4-16-2020

Privacy-aware Security Applications in the Era of Internet of Things

Abbas Acar
Florida International University, aacar001@fiu.edu

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

PRIVACY-AWARE SECURITY APPLICATIONS IN THE ERA OF INTERNET OF THINGS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL AND COMPUTER ENGINEERING

by

Abbas Acar

2020

To: Dean John Volakis
College of Engineering and Computing

This dissertation, written by Abbas Acar, and entitled Privacy-aware Security Applications in the Era of
Internet of Things, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Kemal Akkaya

Alexander Perez-Pons

Bogdan Carbunar

A. Selcuk Uluagac, Major Professor

Date of Defense: April 16, 2020

The dissertation of Abbas Acar is approved.

Dean John Volakis
College of Engineering and Computing

Andres G. Gil
Vice-President for Research and Economic Development
and Dean of University of Graduate School

Florida International University, 2020

DEDICATION

To my parents.

ACKNOWLEDGMENTS

First of all, I would like to express my most sincere gratitude and appreciation to my advisor Dr. A. Selcuk Uluagac, for his guidance and support from the beginning till the end of the completion of this work. His positivity and encouragement always pushed me to achieve higher success throughout my journey.

I would also like to thank Dr. Kemal Akkaya for his comments during our collaboration for several projects. Many thanks to Dr. Bagdan Carbunar, serving on my dissertation committee as well as for his studies on authentication and privacy. They were a great inspiration for my studies in this thesis. I also appreciate Dr. Alexander Perez-Pons for serving on my dissertation committee and giving me valuable comments.

Throughout my doctoral journey, I had a chance to work with great collaborators. I would like to thank Dr. Z. Berkay Celik of Purdue University and Dr. Patrick McDaniel of Pennsylvania State University for their great contributions to our distributed differential privacy work; Dr. Mauro Conti of University of Padua for guiding me during my time as a visitor researcher in his lab and his great comments for homomorphic encryption and smart home user privacy work; Dr. Engin Kirda and Dr. Long Lu of Northeastern University for their guidance in our enterprise malware work during my time at Northeastern University as a visitor researcher; Dr. Ahmad-Reza Sadeghi of Technische Universität Darmstadt and his lab members for their contributions to our smart home user privacy work. I would also like to thank our postdoctoral researcher Dr. Hidayet Aksu and my labmates Amit Kumar Sikder and Leonardo Babun, and many others for their help with everything.

Finally, I would like to thank Florida International University and U.S. National Science Foundation for financially supporting my research as well as the FIU community for providing a great environment for my research for this dissertation.

ABSTRACT OF THE DISSERTATION

PRIVACY-AWARE SECURITY APPLICATIONS IN THE ERA OF INTERNET OF THINGS

by

Abbas Acar

Florida International University, 2020

Miami, Florida

Professor A. Selcuk Uluagac, Major Professor

In this dissertation, we introduce several novel privacy-aware security applications. We split these contributions into three main categories: First, to strengthen the current authentication mechanisms, we designed two novel privacy-aware alternative complementary authentication mechanisms, Continuous Authentication (CA) and Multi-factor Authentication (MFA). Our first system is Wearable-assisted Continuous Authentication (WACA), where we used the sensor data collected from a wrist-worn device to authenticate users continuously. Then, we improved WACA by integrating a noise-tolerant template matching technique called NTT-Sec to make it privacy-aware as the collected data can be sensitive. We also designed a novel, lightweight, Privacy-aware Continuous Authentication (PACA) protocol. PACA is easily applicable to other biometric authentication mechanisms when feature vectors are represented as fixed-length real-valued vectors. In addition to CA, we also introduced a privacy-aware multi-factor authentication method, called PINTA. In PINTA, we used fuzzy hashing and homomorphic encryption mechanisms to protect the users' sensitive profiles while providing privacy-preserving authentication. For the second privacy-aware contribution, we designed a multi-stage privacy attack to smart home users using the wireless network traffic generated during the communication of the devices. The attack works even on the encrypted data as it is only using

the metadata of the network traffic. Moreover, we also designed a novel solution based on the generation of spoofed traffic. Finally, we introduced two privacy-aware secure data exchange mechanisms, which allow sharing the data between multiple parties (e.g., companies, hospitals) while preserving the privacy of the individual in the dataset. These mechanisms were realized with the combination of Secure Multiparty Computation (SMC) and Differential Privacy (DP) techniques. In addition, we designed a policy language, called Curie Policy Language (CPL), to handle the conflicting relationships among parties.

The novel methods, attacks, and countermeasures in this dissertation were verified with theoretical analysis and extensive experiments with real devices and users. We believe that the research in this dissertation has far-reaching implications on privacy-aware alternative complementary authentication methods, smart home user privacy research, as well as the privacy-aware and secure data exchange methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

CHAPTER 1

INTRODUCTION

1.1 Motivation and Threat Models

With the advancements in the technology, our computers and mobile devices have become our identities as they handle and store very sensitive personal information such as identity numbers, bank account credentials, email passwords. Similarly, nowadays, IoT devices (e.g., smart locks, smartwatches) have also become part of our daily lives, and these devices record our daily activities. The amount of sensitive user information recorded, handled, and stored by our computers, mobile devices, and IoT devices is huge. The leakage of this information may result in both serious security and privacy issues. Therefore, it is more important than ever to protect these devices with robust security and privacy mechanisms. In this dissertation, we address three different threats, where the sensitive user information is revealed by the attacker.

In Part I of the dissertation, we investigate solutions against an attacker who wants to access our devices by bypassing the existing authentication mechanisms. Today's authentication systems mostly rely on passwords. However, many practical attacks have been demonstrated that the passwords can be either stolen or bypassed [Dic16, TGC16]. For example, they can be easily stolen via shoulder-surfing or bypassed via phishing attacks. Specifically, in password-based authentication systems, the user is verified one time, which does not guarantee that the identified user is the actual user throughout the login session. In order to strengthen the current authentication systems, there is a need for alternative complementary authentication methods. Continuous Authentication (CA) and Multi-factor Authentication (MFA) are two promising solutions. In CA, the user is periodically

verified throughout the entire session, while in MFA, the user is verified via multiple independent authentication factors. There are several CA and MFA methods proposed in the literature; however, there are two issues with the proposed methods. First, they are either not secure or not practical enough to be deployed in a real-life applications [AAAU16, AAUA20]. Second, these methods are mostly based on the biometrics to provide a more convenient and secure authentication. However, biometric-based systems demand more user information in their operations, yielding privacy issues for users in biometric-based continuous authentication systems [ALB+19].

In Part II of the dissertation, we investigate solutions against a nearby attacker within the range of radio frequency (e.g., WiFi, ZigBee, Bluetooth Low Energy) to our house, who can sniff and record all the network generated by the pairwise communication of the smart home devices. Even though this communication is encrypted, an attacker can perform a fingerprinting attack, and infer the user activities occurring at home [AFA+18]. The mechanisms like VPN or TOR do not protect against such kind of adversary and the solutions like faraday cage is not realistic. This type of attacker has an advantage of not being detected easily as it is a passive attacker.

Finally, in Part III of the dissertation, we investigate solutions against an attacker called honest-but-curios attacker, who wants learn the about the user data shared for genuine purposes. Previous works have shown the benefit of data sharing within distributed, collaborative, and federated learning [DCM+12, SCST17, APP+18]. However, the hospitals as well as the patient may not want to reveal their data to third parties. Here, the methods such as Secure Multi-party Computation (SMC) [BDNP08] (e.g., homomorphic encryption [AAUC18], garbled circuits [H+11]) allow computation of a joint function (e.g., regression function) with-

out revealing individual input of the parties. However, there are two issues SMC does not address during the data sharing. First, SMC does not consider the conflicting relationships (e.g, politics, regulations) among members. Second, SMC does not guarantee that the final result of distributed computation would not leak any information about an individual in a sensitive dataset [EESA+12, NS08, GKS08]. Therefore, privacy of individuals and their data can be easily violated.

1.2 Research Problem

Considering the threats and the issues in the previous section, in this dissertation we address seven unique but related research problems:

1. *A Robust and Usable Continuous Authentication Method:* Continuous Authentication (CA) is a good mechanism to re-verify a user identity periodically throughout a login session. In the literature, a number of studies have been proposed for the use of biometrics in continuous user authentication [PCCB16, FFS17]. However, one of the desired features in CA is *non-intrusiveness* [AAUA20]. Physiological characteristics like iris pattern or fingerprint are not applicable in this manner since they can not be extracted seamlessly. More plausible approaches for CA would be behavioral characteristics [Sea16, ERLM17, WWZJ18] like typing rhythm, gait as they can be collected without interrupting the user. Therefore, they are ideal candidates to increase the security of the current systems as an additional authentication factor rather than a standalone authentication system. Among all behavioral biometrics, the most promising results are proposed using keystroke dynamics [ASL15, WDL+18, CZY+15]. However, in a recent work [TGG13], the reliability of classical keystroke dynamics is analyzed, and an interface

was designed to help an attacker so that the attacker can mimic the typing rhythm of a legitimate user by using the feedback provided by the interface. Therefore, there is a need for a system, which is both reliable and usable.

2. *A Privacy-Aware Biometric-based Continuous Authentication Protocol:* In traditional biometric authentication systems, it is generally assumed that an authentication server and a decision module have access to feature vectors of users in plaintext form. Even though the sensitive biometric data may be communicated through secure channels, and it may be stored in encrypted form, the feature vectors would have to be decrypted during the verification phase. This, in principle, violates the privacy of biometric data, and the adversaries may be able to exploit this in their attacks [FDCA11, GR12, BGK+15, FLE14]. Such vulnerabilities can be prevented in traditional password-based authentication mechanisms by computing the hash of a password and storing this hash value instead of the password itself. A matching decision is made by comparing the two hash values. Here, the use of cryptographic hash functions provides some level of protection because given the hash value of a string, it is computationally infeasible to determine the input string as a preimage of that hash value. However, traditional cryptographic hash functions (e.g., MD5, SHA-2) cannot be adapted in biometrics because even some slight changes in the input would result in a significant change in the hash function's output. The previous studies are either using the cryptographic primitives, which suffer from high computational overhead and also they have been overlooking and missing the details of a full protocol. Therefore, there is a need for a complete privacy-aware biometrics-based continuous authentication protocol.

3. *Privacy-Aware Multi-factor Authentication:* In a typical MFA system, each user is verified via the first authentication factor (usually password) along

with a second or even a third factor such as smartcards [Kum04], fingerprints [BSSB06], or user's mouse movements [ZPW11]. MFA solutions based on physical devices or physiological characteristics depend on the introduction of specialized hardware, such as a token or a fingerprint reader, which hinders usability and deployability by causing additional cost for manufacturing and implementation. Alternatively, the more usable (and therefore more likely to be widely-adopted) MFA solutions are based on users' behavior; however, they do little to protect the privacy of the user data as the data needs to be revealed to the authentication server in plain form. This approach has two kind of risks. First, the owner of the database server may use it for malicious purposes (e.g., selling user's information for economic interest). Second, if an attacker succeeds in obtaining the database storing the user data, he/she can masquerade as a legitimate user by crafting required authentication factors. Therefore, there is a need for a system providing MFA while preserving the privacy of user data.

4. *Smart Home User Privacy:* A myriad of IoT devices such as bulbs, switches, speakers in a smart home environment allows users to easily control the physical world around them and facilitate their living styles. However, an attacker inside or near a smart home environment can potentially exploit the innate wireless medium used by these devices to exfiltrate sensitive information about the users and their activities, invading user privacy. This allows an adversary to efficiently aggregate extensive behavior profiles of targeted users. The smart home devices are usually encrypted using standard protocols like WPA2, in the case of WiFi, the contents of the exchanged messages or commands are hidden. However, the encryption only hides the payload, related meta-data (e.g., packet lengths, traffic rate) of the network traffic still leaks some informa-

tion about the messages exchanged [SSW+02, VČČD15, SFSM13, LXZ+16, CMSV16, CBS+18]. Some earlier works [SSW08, ARS+17, ARF17] have shown that it is relatively easy to make some simple inferences such as device type inference [MMH+17], identifying the user occupancy via detecting the mode transition between the device activities [CLBR16], or simple device mode inference [ARS+17]. However, combining such partial information from different smart home devices to get a more meaningful picture about a user's actions or his/her activity profile is challenging. This is because a successful attacker must aggregate information about actions over a longer period of time from a multitude of smart home devices, which is only feasible if activity detection and identification can be automated to a large degree to keep the required effort manageable. Therefore, there is a need for the investigation of this attack vector and promising countermeasures.

5. *Policy-based Secure Data Exchange:* Inter-organizational data sharing is crucial to the advancement of many domains including security, health care, and finance. Previous works have shown the benefit of data sharing within distributed, collaborative, and federated learning [DCM+12, SCST17, APP+18]. Privacy-preserving machine learning offers data sharing among multiple members while avoiding the risks of disclosing the sensitive data (e.g., health-care records, personally identifiable information) [EESA+12]. For example, Secure Multiparty Computation (SMC) enables multiple members, each with its training dataset, to collaboratively learn a shared predictive model without revealing their datasets [MZ17]. These approaches solve the privacy concerns of members during model computation, yet do not consider the complex relationships such as regulations, competitive advantage, data sovereignty, and jurisdiction among members on private data sharing. Members want to be

able to articulate and enforce their conflicting requirements on data sharing. Therefore, there is a need for a system, where the members can easily specify their requirements on the data exchanged without compromising the security and privacy of the data and users.

6. *Secure and Differentially Private Computations in Multiparty Settings:* Secure and private computation of statistical models is increasingly used in different operational settings from healthcare [KHK+16, CAA+19] to finance [BTW12] and security sensitive applications [FDCB15]. Given the distributed nature of these applications, security and privacy are mostly achieved by utilizing Secure Multiparty Computation (SMC). SMC allows distributed parties to compute a joint function (e.g., regression function) over their private inputs without revealing those inputs to other parties. Each party learns the final result, but no other information. However, SMC has a major privacy concern for a targeted individual as it does not guarantee that the final result of distributed computation would not leak any information about an individual in a sensitive dataset [EESA+12, NS08, GKS08]. As such, privacy of individuals and their data can be easily violated. Therefore, there is a need for a mechanism, where individual parties do not see each others' inputs and further can not infer their data from the final constructed model. Indeed, combining SMC with Differential Privacy (DP) could solve this privacy problem as DP introduces sufficient noise into the final result to prevent any leakage about a single individual. However, combining SMC with DP is not a trivial task. Adding noise in a distributed manner may lead to a significant accuracy loss in the final models, which may cause catastrophic consequences in, for example, the healthcare domain. Therefore, enabling distributed differential privacy on local data with differential privacy guarantees on final results is a challenging

problem and needs a novel mechanism combining SMC and DP to provide both data privacy and individual privacy.

7. *Survey of Homomorphic Encryption Schemes:* Legacy encryption systems depend on sharing a key (public or private) among the peers involved in exchanging an encrypted message. However, this approach poses privacy concerns. The users or service providers with the key have exclusive rights on the data. Especially with popular cloud services, the control over the privacy of the sensitive data is lost. Even when the keys are not shared, the encrypted material is shared with a third party that does not necessarily need to access the content. Moreover, untrusted servers, providers, and cloud operators can keep identifying elements of users long after users end the relationship with the services. Indeed, Homomorphic Encryption (HE), a special kind of encryption scheme, can address these concerns as it allows any third party to operate on the encrypted data without decrypting it in advance. Although this extremely useful feature of the HE scheme has been known for over 30 years, the first plausible and achievable Fully Homomorphic Encryption (FHE) scheme, which allows any computable function to perform on the encrypted data, was introduced by Craig Gentry in 2009. Even though this was a major achievement, different implementations so far demonstrated that FHE still needs to be improved significantly to be practical on every platform. Therefore, many follow-up works are proposed in the literature to improve the FHE schemes and it attracted the interest of people from very different research areas in terms of theoretical, implementation, and application perspectives. Therefore, there is a need for a study providing a structured way to understand the state-of-the-art HE schemes and to understand how HE or FHE would applicable in the provision of privacy in other works in this dissertation.

1.3 Research Objectives

In this dissertation, we introduced several unique solutions to the research problems given in Section 1.2. Our objectives for these solutions are sevenfold:

- **Objective #1**: The proposed novel authentication mechanisms should increase the security of the existing technologies while keeping the usability and deployment cost minimal.
- **Objective #2**: The proposed privacy-aware continuous authentication protocol should protect the biometrics of the users against both malicious attackers and curious advertisers while allowing the continuous authentication.
- **Objective #3**: The proposed privacy-aware multi-factor authentication system should protect the user profiles against both malicious attackers and honest-but-curious advertisers while allowing the authentication of the user from multiple independent sources.
- **Objective #4**: While the proposed novel attack on smart home users mechanisms shows the feasibility of the multi-stage privacy attacks, the proposed countermeasure should protect the privacy of the smart home users against both local and remote adversaries.
- **Objective #5**: The proposed policy-based secure data exchange method should allow the members to express their privacy requirements on the data exchange.
- **Objective #6**: The proposed differentially private and secure data exchange method should protect both the data privacy and individual privacy in the dataset.

- **Objective #7**: Our overview of HE schemes should provide an understanding of the state-of-the-art HE schemes and how HE or FHE would applicable in the provision of privacy in other works in this dissertation.

1.4 Contributions

With the objectives above in mind, the contributions of this dissertation are as follows:

WACA: Wearable-Assisted Continuous Authentication. In this work, we introduced a usable and reliable Wearable-Assisted Continuous Authentication (WACA), which relies on the sensor-based keystroke dynamics and the authentication data is acquired through the built-in sensors of a wearable (e.g., smartwatch) while the user is typing. The acquired data is periodically and transparently compared with the registered profile of the initially logged-in user with one-way classifiers. With this, WACA continuously ensures that the current user is the user who logged-in initially. We also tested WACA against powerful attacks, including imitation, statistical attacks, and insider attackers. For this purpose, we designed a scenario for the imitation attacks with real participants. On the other hand, we developed three generic attacking scenarios for the statistical attacks that can also be utilized by other future continuous authentication studies.

PACA: Privacy-aware Continuous Authentication. In this work, we constructed a lightweight, privacy-aware, and secure continuous authentication protocol, called PACA. Previous works have been overlooking this and missing the details of a full protocol. PACA is initiated through a password-based key exchange protocol, and it continuously authenticates users based on their biometrics. Moreover, it is generic in the sense that one can instantiate it using a large class of secure template

generation and matching algorithms, and biometrics-based authentication systems. Moreover, we also design an actual system (the system, its full implementation, and its detailed evaluation) under the proposed protocol: a hybrid (password and keystroke dynamics), continuous, and privacy-preserving biometric authentication for utilized and optimized a wearable-assisted continuous authentication mechanism, and NTT-Sec to handle the real-valued feature vectors while preserving the accuracy. The use of PAKE and NTT-Sec allows one to avoid TLS, any certification authority, verification of certificates, and long term private keys [GGB13, ŠGGB15]. In addition, we performed a detailed security and privacy analysis of the proposed protocol against eight different well-known attacks [RCB01] for the biometrics-based authentication methods. We first identify several security requirements. Moreover, we particularly described detailed attack strategies, and then analyzed the resistance of our protocol against those attacks. Moreover, we deployed the proposed scheme and provided extensive results with data collected from users wearing an Apple smartwatch to assess the security, accuracy, and resource consumption. Particularly, we provided some concrete estimates for the security of the proposed system, and we report on the timing results, and the false acceptance/rejection rates. Finally, we also measured the resource consumption on a real computing device (e.g., smartwatch).

PINTA: Privacy-Aware Multi-factor Authentication. In this work, we designed a privacy-preserving multi-factor authentication (MFA) system which collects hybrid user behavior profiles to serve as a second authentication factor along with the user password as the first. Instead of just focusing on one specific category of user behavior, like system processes or user's mouse movements, we integrated features from several categories to generate a user's profile. We also adopted fuzzy hashing and fully homomorphic encryption (FHE) techniques to ensure that a user's personal

information is not leaked to servers or a third party. For the experiments, we used a user profile database derived from several public datasets [BSPvdM12, She12, KM12] and a dataset we generated. We evaluated the performance of the proposed system in terms of recall, false positive rate, size of information required for authentication, system overhead, and resource utilization. Our results show that the proposed scheme can well detect imposters from legitimate users while protecting user privacy.

Peek-a-Boo: Smart Home User Privacy. In this work, we discovered a novel multi-stage privacy attack against user privacy in a smart environment. It is realized utilizing state-of-the-art machine-learning approaches for detecting and identifying particular types of IoT devices, their actions, states, and ongoing user activities in a cascading style by only observing the wireless traffic passively from smart home devices. The attack effectively work on both encrypted and unencrypted communications. In contrast to earlier approaches, our multi-stage privacy attack can perform activity detection and identification automatically, without extensive background knowledge or specifications of analyzed protocols. This allows an adversary to efficiently aggregate extensive behavior profiles of targeted users. We evaluated the effectiveness of the novel multi-stage privacy attack with 22 different off-the-shelf IoT devices utilizing the most popular wireless protocols for IoT. Our experimental results show that an attacker can achieve very high accuracy (above 90 %) in identification of the types, actions, states, activities of the devices and sensors. To protect against this privacy leakage, we also proposed a countermeasure based on generating spoofed network traffic to hide the real activities of the devices.

CURIE: Policy-based Secure Data Exchange. In this work, we introduced a policy-based data exchange approach, called Curie, that allows secure data exchange among members that have such complex relationships. Members specify their requirements on data exchange using a policy language (CPL). The require-

ments defined with the use of CPL form the local data exchange policies of members. Local policies are defined separately for data sharing and data acquisition policies. This property allows asymmetric relations on data exchange. For example, a member does not necessarily have to acquire the data that the other members dictate to share. By using these two policies, members specify statements of who to share/acquire and what to share/acquire. The statements are defined using conditional and selection expressions. Selections allow members to filter data and limit the data to be exchanged, whereas conditional expressions allow members to define logical statements. Another advanced property of CPL is predefined data-dependent conditionals for calculating the statistical metrics between member's data. For instance, members can define a conditional to compute the intersection size of data columns without disclosing their data. This allows members to define content-dependent conditional data exchange in their policies. We validated Curie through an example of real healthcare application used to prescribe warfarin dosage. A privacy-preserving joint dose model among medical institutions is compiled with the use of various data exchange policies while protecting the privacy of members' healthcare records. Finally, we showed Curie incurs low overhead and policies are effective at improving the dose accuracy of medical institutions.

Achieving Secure and Differentially Private Computations in Multiparty Settings. In this work, we designed a novel protocol for achieving Secure Multiparty Distributed Differentially Private (SM-DDP) computations on sensitive data. The protocol provides the guarantees of both SMC and DP. SMC is provided through Homomorphic Encryption (HE) [Gen09] while DP is provided via Functional Mechanism (FM) [ZZX+12]. An important characteristic of FM is that it injects noise into the feature matrices (i.e., coefficients of objective function), which can be computed independently by each party in a multiparty computational environment. We

explored this feature of FM and apply it to linear regression using our SM-DDP protocol, but it can be applied to the computation of any statistical model function that allows independent calculation from the local statistics. We show that the accumulated noise in our protocol is still bounded and convergent by using the infinite divisibility property of Laplacian distribution [McN02]. Finally, we evaluated SM-DDP protocol's computational efficacy on linear regression using two real-world datasets. We compared our results with the use of Centralized DP (CDP) in a multiparty setting. The intuition is that the distributed setting of DP (DDP), which is proposed in this work, would cause a greater accuracy loss than the typical client-server setting of SMC systems. However, we showed exactly same trade-off can be achieved using the SM-DDP protocol. The extensive evaluation results indicate that the proposed SM-DDP protocol yields minimal computational overhead—less than a minute for 20 parties with 32 attributes and 10K samples. The individual parties obtain better accuracy than that would be obtained from a single party model. Finally, SM-DDP is scalable while providing security and privacy guarantees.

Investigation of Practical Usage of Privacy-Aware Technologies In an effort to better understand the state-of-the-art privacy-aware technologies, as part of this dissertation, we also investigated the homomorphic encryption technologies. Particularly, we provided a comprehensive survey of all the main FHE schemes. We also covered a survey of important PHE and SWHE schemes as they are the first works in accomplishing the FHE idea and are still popular as FHE schemes are computationally very costly. Furthermore, we included the FHE implementations focusing on the improvements with each scheme. In addition, we mentioned the challenges and future perspectives of HE to motivate the researchers and practitioners to explore and improve the performance of HE schemes and their applications.

1.5 Ethical Considerations

As the collected data in these works may raise some ethical and privacy concerns, we acknowledge that our research study with the human subjects was conducted with the appropriate Institutional Review Board (IRB) approvals (FIU-IRB-16-0296 and FIU-IRB-18-0443).

1.6 Outline

The rest of this dissertation is organized as follows:

- **Chapter 2** describes the related work of the studies in this dissertation.
- **Chapter 3** describes the architecture of our wearable-assisted continuous authentication, called WACA.
- **Chapter 4** describes the details of our privacy-aware continuous authentication protocol, called PACA.
- **Chapter 5** describes the details of our privacy-preserving multi-factor authentication system called PINTA.
- **Chapter 6** describes our multi-stage privacy attack on smart home users as well the details of our countermeasure against that attack.
- **Chapter 7** describes an overview of state-of-the-art homomorphic encryption schemes.
- **Chapter 8** describes our policy-based secure exchange approach.
- **Chapter 9** describes our approach to combine the multiparty computation and distributed differential privacy.

- **Chapter 10** explains our conclusions and the recommended future works that can be built upon the studies in this dissertation.

CHAPTER 2

RELATED WORK

In this chapter, we examine the related work of the studies presented in this dissertation.

2.1 Wearable-assisted Continuous Authentication

Currently, the most common method used to verify the user periodically depends on session time-outs. In session time-outs, if the time window is kept too short, the user's convenience will be reduced due to frequent interruptions of the session for authentication. On the other hand, if the time window is set too long, in the case of a breach, the attacker would have more time on the victim's system.

In the literature, a number of works have been proposed for the use of biometrics in continuous user authentication [Car03, KJ06, AMSS08, PCCB16, FFS17]. However, one of the desired features in the continuous authentication is *transparency*. Hard biometrics like iris pattern or DNA are not applicable since they can not be extracted transparently. In another work [KYSR09], a special mouse with a fingerprint sensor is proposed. In addition to requiring a custom mouse, its reliability is also an issue. The ease of counterfeiting fingerprints was shown, and the fingerprint-based biometrics was easily bypassed [Clu07, Clu13]. Facial recognition methods may seem a good candidate; however, the liveliness detection is still an issue to be addressed, and several attacks are possible under practical conditions [DM09, BCF+13]. In addition, several other biometrics like pulse-response [MRRT17] or eye movements [ERLM15] are also proposed. However, since these approaches require special equipment, deployment costs are increasing significantly.

Recent suspicions on keystroke dynamics. Among all the biometrics, the most promising results are proposed using keystroke dynamics and mouse movements [ASL15, WDL+18, CZY+15]. However, in a recent work [TGG13], the reliability of classical keystroke dynamics are analyzed and an interface, called Mimesis, was designed so that a user can mimic the typing rhythm of another user by using the feedback provided by Mimesis. In another study [SP13], the statistical attacks with bots generating synthetic typing patterns are examined for the conventional keystrokes biometrics. In our work, we test WACA against both these imitation and statistical attacks using similar configurations presented in these studies. We show that WACA is secure against the powerful imitation and statistical attacks. The detailed analysis of these attacks are given in Section 3.5.2.

Inference attacks using smartwatch sensors. Another direction on sensor-based keystroke research is using the motion sensors of wearables as a side channel attack to infer some valuable assets like passwords. The main motivation behind this attack is similar to WACA. Motion sensors will move in the same way with keystrokes while typing and the wrist rotations and displacement will cause to leak the keystrokes. This attack is deployed firstly on smartphones [MVBC12, OHD+12, CC12, ASBS12, XBZ12], and recently on smartwatches [WLRC15, LDW+18]. Restricting access to motion sensors is not a realistic suggestion to defend against this attack. In our work, we propose a pairing and synchronization session before using the smartwatch with its paired computer. In this way, an encryption-supported secure channel can be used to communicate between smartwatch and computer.

Comparative evaluation of WACA. In the literature, there is not a widely accepted standard framework to compare device authenticators. However, Usability-Deployability-Security (UDS) framework proposed in [BHVOS12] is a highly accepted framework for web authentication schemes. To compare our work with its

Table 2.1: Comparative evaluation of WACA using the UDS framework [BHVOS12] with continuous authentication alternatives.

	Memorywise-Effortless	*Nothing-to-Carry*	*Physically-Effortless*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	*No-Constraint-on-Using-the-Device*	*Accessible*	*Negligible-Cost-per-User*	*Resilient-to-Physical-Observation*	*Resilient-to-Targeted-Impersonation*	*Resilient-to-Internal-Observation*	*Resilient-to-Leaks-from-Other-Verifiers*	*Resilient-to-Phishing*	*Resilient-to-Theft*	*Requiring-Explicit-Consent*	*Unlinkable*	*Insider-Identification*	*Resilient-to-Insider-Threat*
	Usability						Dep.		Security									
Password		●		●	●	●	●	●							●	●		
Time-out	●	●	◐	●	●	●	●	●		na	●		●	na		na		
Proximity [CN02]	●	◐	◐	◐	◐	●	●	◐	◐	na	●							
Face [Beu14]	●	●	◐			◐	◐	◐	●	◐		●		●	◐	●	●	●
Fingerprint [KYSR09]	●	●	◐				◐		●	◐		●		●	◐	●	●	●
Eye-movement [ERLM15]	●	●	◐				◐		●	●	●	●	●	●	●	●	●	●
Keystroke [MR97]	●	●	◐	◐		●	●	●	◐		●		●		●			●
ZEBRA [MMC+14]	●	◐	◐	◐	◐	●	●	◐			●	●	●	◐	●	●	◐	●
WACA (this work)	●	◐	◐	◐	◐	●	●	◐	●	●	●	●	●	●	●	●	●	●

● = offers the benefit; ◐ = almost offers the benefit; no circle = does not offer the benefit.

alternatives, we remove some of the irrelevant and non-applicable benefits and use only the relevant ones of the UDS framework. The complete list of benefits can be found in [BHVOS12]. After also adding three new benefits, we end up with 18 benefits in total. Table 2.1 rates WACA using these 18 benefits. For space, we cannot compare WACA to all continuous authentication methods proposed in the literature. Therefore, we choose representatives for each continuous authentication method.

WACA captures the sensor readings through a smartwatch without interrupting the user, i.e., unobtrusively. However, unlike time-out or classical keystroke dynamics, it requires an extra channel to collect data, but obviously a smartwatch is a not a customized hardware, i.e., it is an off-the-shelf device, so we say it partially supports the benefit of *Nothing-to-Carry* and since its error is deficient, it also of-

fers the advantage of *Infrequent-Errors*. On the other hand, WACA outperforms all other methods in terms of security benefits. In addition to WACA, eye-movement based authentication method also seems as secure as WACA. However, WACA's performance for usability and deployability is better. For example, WACA offers much lower error rates, and eye-movement based methods require a specialized eye or gaze-trackers and the user should be in a certain distance and in front of the eye tracker which obstructs the usability of the eye-movement based methods. They are more convenient for challenge-response type authentication methods [SRRM16] even though they have the capability to provide data continuously and transparently. In brief, our conclusion from this comparative evaluation shows that WACA offers better security benefits while keeping the usability at the same level as other notable methods.

2.2 Privacy-aware Continuous Authentication

In the literature of Continuous Authentication (CA), keystroke dynamics and mouse movements [BW12, TTY13] are the ones having the most promising results in terms of usability and deployability as they can work transparently and have almost zero cost. However, with a recent attack [TGG13], the reliability of classical keystroke dynamics have become suspicious. The idea of using motion sensors to extract user behavior is first used in smartphones [FBM+13, ZBHW14, TO13] and later used for computer users by using smartwatch [AAUA20]. The advantage of the sensory-based approach is that sensors provide not only one-dimensional timing information but also some features in other dimensions like the pressure of keystrokes and the rotation of hand during key pressing. This obstructs the imitation attacks [TGG13] and statistical attacks [SSCG16] since it requires to mimic the user's acceleration and rotation behavior simultaneously in three dimensions.

Privacy Attacks on Keystroke Dynamics. Even though keystroke dynamics is considered as a good candidate for continuous authentication and identification systems, the information that can leak from the collected keystroke data raises serious privacy concerns. It has been shown that the gender [FDCA11, GR12], the demographics [BGK+15] or the emotional state [FLE14] of the user can be predicted from keystroke dynamics. What the user is typing (e.g., password inference) [SWT01, ZW09] can also be effectively inferred from the keystroke dynamics.

Secure and Privacy-preserving Biometrics. There are three approaches proposed to address security and privacy issues in biometric schemes: *biometric cryptosystems* (BC), *cancelable biometrics* (CB), and *keyed biometrics* (KB). Some of the key references include [JW99, JS06, DRS04] for BC, [JLG04, TGN06, RCCB07] for CB, and [BCI+07, BBCdS08, Sto10, BBC+10, BG11] for KB. In addition to these three main techniques, there are *hybrid biometrics* (HB), that blend BC, CB, and multi-factor authentication [BCK08, FAD06]. Some of the above methods have been used to secure multi-biometric traits simultaneously for improved performance and security (see [NJ15, NMX+16], and the references therein). These constructions can also be considered under HB.

Several theoretical and practical attacks (record-multiplicity, hill-climbing, masquerade attacks, and brute-force attacks) have been developed on BC and CB, many of which result in a total break of the system with respect to irreversibility and indistinguishability. For attacks on BC and CB, see [SB07, STP09, RU12, WRDI12, Tam14] and [NNJ10, FLY14], respectively. Several countermeasures have been proposed to guard against these attacks, including hardening with secrets [FAD06, NNJ07, BA11], hybrid approaches and multi-biometrics [BCK08, RTWB16], employing encryption or signature schemes [Boy04, BCI+07, BBCdS08, BBC+10, BG11], and new quantization and alignment methods [TMM15]. Recommended safeguards

come at the cost of degrading performance and usability, increasing communication and computational bandwidth to impractical ranges, and introducing secret parameters or trusted third parties. These are also the major common problems shared over HB and KB in general. See [NJ15, NMX+16] for other drawbacks of HB and KB.

Several cryptographic primitives including Secure Multiparty Computation [BA09, EHKM11], Verifiable Computation [BCK+15], and Bloom Filters [RBBB14] have been proposed for the secure biometrics. However, the main drawback of the cryptographic primitives is the computational overhead. Moreover, in addition to cryptographic primitives, the biometric template protection methods such as cancellable biometrics [ASNM05, KPDD09] and biohashing [RU10] have been proposed for the secure biometrics. However, Biohashing has been shown as vulnerable to several attacks [KV10, KCZ+06] and even though cancellable biometrics is more secure, they do not apply to behavioral biometrics, which is more ideal for continuous authentication.

Secure and Privacy-preserving Continuous Authentication. Although there is extensive literature on the privacy-preserving biometrics, most of the work is on physiological biometrics such as fingerprints, iris, etc. However, the physiological biometrics are not feasible for a continuous authentication mechanism [SLM+16]. A potential solution is to use homomorphic systems [AAUC18] to address privacy issues in template matching. A solution using a homomorphic system can be implemented with two main approaches. In the first approach, the user generates a public key-private key pair for HE; the user encrypts his biometric data using his public key, registers it with the server. At the time of verification, the user queries the server with his fresh biometric data encrypted under the same public key. The server uses the public key of the user and computes the encrypted and randomized

distance between the template and the queried biometric, and sends it to the user. The user decrypts using his private key and sends the randomized distance back to the server. The server de-randomizes to recover the actual distance and outputs the result (accept or reject). This approach requires users to maintain long-term and individual secret keys, highly interactive with non-trivial computation and bandwidth requirements. See [SSNS14, SSNS15] for some recent implementations of this approach. In the second approach, the server generates public-private key pair for HE, and users encrypt their biometric data under the server's public key during registration and authentication. Even though the key generation/storage/decryption and computations on the encrypted data are performed on two separated independent components of the server, the server has the ability to decrypt and recover users' biometric data, whence has to be trusted by all users in the system; see [YSK+13] for some recent implementations of this approach. Indeed, it has also been shown that the proposed protocol is vulnerable to biometric template recovery attacks under the presence of even a malicious computational server, which is only one of two servers [AM14].

2.3 Privacy-aware Multi-factor Authentication

A number of researchers have proposed the design and implementation of MFA systems [PMZ+11, Ver12, SP12, JY11, ZPW11], with each presenting its own specific advantages and trade-offs. *Knowledge factor* (i.e., passwords) is the most ubiquitous authentication factor. It is widely known that the sole use of passwords has many weaknesses. Nevertheless, passwords are still in use and are the de-facto standard [BHOS12]. Thus, to reduce the security risk of the sole use of a *knowledge factor*, researchers have added *possession factor* and *identity factor* to authentication sys-

tems. MFA systems using passwords along with a *possession factor* are commonly found in electronic commerce and online banking. Security tokens, also called One-Time-Password (OTP) tokens, is one of the most commonly-used *possession factor*, which generates a pseudo-random number at pre-determined intervals (e.g., RSA SecurID[AUT12] and VeriSign Security Token[Ver12]). Additionally, these cards serve as an additional authentication factor, especially in corporate network environments. For instance, in [Kum04], Kumar proposed a secure remote user authentication scheme with smart cards for corporate networks. Unfortunately, an MFA system with a *possession factor* usually depends on the distribution of some specific device, which is cumbersome and not user-friendly. Besides, the introduction of physical devices may pose further security risks if the devices are lost, stolen or replicated without the knowledge of the legitimate user. Czeskis et al. [CDK+12] first consider the usability of an MFA system with a *possession factor* by proposing authentication through opportunistic cryptographic identity. Nevertheless, their proposed scheme requires the presence of the user's phone, which limits the usability of the system.

Finally, authentication via an identity factor is also a well-studied area of research. Identity factors are further categorized as either physiological biometrics or behavioral characteristics [YG08, SP12]. Physiological biometrics, such as fingerprints, iris, and face, have already drawn considerable attention in academia and have been implemented widely in industry [BSSB06]. Behavioral biometrics, such as mouse movements, keystroke dynamics [SO19, TCCL14, KDPP16], graphical passwords, though not widely utilized, have also gained popularity in the research community [JY11]. Similar to MFA systems with possession factors, MFA systems with physiological biometrics suffer from relatively low usability and deployability due to the implementation cost of biometrics recognition devices. Meanwhile, the

Table 2.2: Comparative evaluation of PINTA .

	Security	Privacy	Usability	Low Deployment Cost
Password-only	◐	na	●	◐
2FA	◐		◐	●
Czeskis et al. [CDK+12]	◐	◐		●
Sujithra et al. [SP12]	●			
Bhargav-Spantzel et al. [BSSB06]	●	●		
PINTA (this work)[1]	●	●	●	●

● = offers the benefit; ◐ = almost offers the benefit; no circle = does not offer the benefit.

downside of using behavioral characteristics is that the system may induce relatively low authentication accuracy and large system overhead [ZPW11]. To the best of our knowledge, no consideration has been given to the privacy issue when authenticating based on user behavior. This is critical, given that the validity of the specific user characteristics shared with a site will likely significantly outlast the period of time for which the site's services are needed. That is, the shared characteristic is not a mere pseudonym, but a characteristic that can identify a user for years to come. Therefore, our goal in this work is to develop a privacy-preserving multi-factor authentication system based on passwords along with hybrid user profiles, that considers usability, privacy, and deployment cost.

Comparative evaluation of PINTA. In Table 2.2, we perform a comparative evaluation of our proposed scheme in this work, PINTA, where we compare PINTA with its alternatives in terms of the benefits offered by the schemes. As can be seen from the Table 2.2, PINTA offers more benefits than its alternatives.

2.4 Smart Home User Privacy

Identification using the encrypted network traffic . The meta-data (e.g. MAC, traffic rate) of encrypted network traffic triggers possible threats including unintentional disclosure of the content or user. There is an extensive literature in

the identification of the content from the encrypted network traffic. For example, web page identification [SSW+02], web user identification [LL06], protocol identification [WMM06] are some of the research on the identification using the encrypted traffic. Not only identification attacks, but also the countermeasures have been studied in several studies [DCRS12, CZJJ12].

Smartphone Fingerprinting. Recently, this research has been extended to smartphone users. For example, Conti et al. [CMSV16] showed a way of identifying user action on Android apps and Taylor et al. [TSCM16] presented their work on the fingerprinting of apps from an encrypted network of the smartphone. In addition, [SFSM13] fingerprints the smartphones using the network traffic captured generated from the popular applications such as Facebook, WhatsApp. Finally, in [AHM+15], Ateniese et al. showed a new adversary model that can infer the location of the user from the encrypted network traffic.

Fingerprinting Methods. In all the aforementioned studies, either statistical techniques [VČČD15] or machine learning methods [CMSV16] were used to infer different sensitive information about the user and the context. Even ML has been used for the task of identification such as user, device, or website identification, in none of these studies, the attacks are timing-based as we have in our work.

IoT Fingerprinting. So far, in all the aforementioned studies the results showed that the used methods are efficient and the threat is real, but the threat was limited to the web and online privacy of the user. Now with the emergence of IoT, it has been extended to every part of our daily lives and, with this, threats and countermeasures have also evolved. The number of studies on the IoT fingerprinting through the network traffic has been increasing every day. Many studies have investigated the device type identification problem, where it has been sometimes proposed for both attacking [MMH+17, SBZD18, BTB17, DLT+19, SECK19] and improving the secu-

rity of smart home platforms [OMM+19, OER19, BEM19]. Moreover, some other works [CLBR16, ARF17, ARS+17, AHR+19, JFF19, TVMD20, RDC+19] worked on the device activity (event) inference problem, where the phrases device activity inference and user activity inference sometimes have been used interchangeably. In our work, we refer to the device activity (event) as the activity inferred from only one device. Even though sometimes the device activity and user activity would be the same thing (e.g., "coffee maker is ON" is the same as "the user is making coffee"), sometimes information from multiple devices is needed to infer one user activity correctly (e.g., see Figure 6.4). We differentiate those two types of activities and provide a more generalized activity types in the fourth stage of our attack when we are modeling the user activities using HMM in Section 6.5.8.

Difference from existing work. Our work differs from the aforementioned studies in several ways: First, we are proposing a comprehensive method of end-to-end attack to infer the on-going user activities in a cascaded manner, where the previous studies have focused on only one stage of the attack. Note that putting all the different attack mechanisms and executing them successfully is a non-trivial task. Second, we are proposing the use of HMM for user activity modeling, where the device activities from multiple devices have been used to infer user activities. Last but not least, for the analysis of our attack, we performed experiments using the devices with WiFi, ZigBee, and BLE, where most of the previous studies have focused only on one of those wireless protocols.

2.5 Survey of Homomorphic Encryption Schemes

Like our work in this dissertation, there are similar useful surveys in the literature. In fact, unfortunately, some of the surveys only cover the theoretical information of

the schemes as in [PPP+14, AS14] and some of them are directly for expert readers and mathematicians as in [Vai11, Sil13, Gen14]. Compared to these surveys, our survey has a broad reader perspective including researchers and practitioners interested in the advances and implementations in the field of HE, especially FHE. Furthermore, while the survey in [AMFF+13] only covers the signal processing applications, other in [HP14] covers a few FHEs on only cloud applications. Since our survey is not limited to specific application areas, we do not articulate these specific application areas in detail but we list the theory and implementation of all existing HE schemes, which can be used in possible futuristic application areas with recent advancements. After [FG07] and [Aki09], many HE schemes were introduced. Compared to these useful surveys, our survey focuses on the most recent HE schemes, since most of the significant improvements are introduced recently (after 2009). Although [MOO+14] is one of the most recent surveys, it focuses on the hardware implementation solutions of FHE schemes. This survey is not limited to hardware solutions, as, in addition to hardware solutions, it covers software solutions of implementations as well in the implementation section. After [Wu15], several new FHE schemes, which improves FHE in a sufficiently great way as to be worthy of attention, were proposed in the literature. Finally, it is worth mentioning that [ABC+15] provides a systematic explanation of the new terminology related to FHE and [AKP13] provides security and a characterization of all existing group homomorphic encryption schemes, where they do not present all the HE schemes and their implementations in detail. Compared to these useful prior works, nonetheless, our survey is intrinsically different from the aforementioned surveys.

2.6 Policy-based Privacy-aware Secure Data Exchange

Policy has been used in several contexts as a vehicle for representing configuration of secure groups [MP06], network management [RJR+16], threat mitigation [FDCB15], and access control [DZC+16]. These approaches define a schema for their target problem and do not consider the challenges in secure data exchange. In contrast, Curie defines a formal policy language to dictate the data exchange requirements of members and enforces the agreement in collaborative ML settings.

On the other hand, secure computation on sensitive proprietary data has recently attracted attention. Federated learning [TBA+19, SCST17], anonymization [EESA+12], multi-site statistical models [Dan15], secure multi-party computation [BCD+09], and secure and differentially-private multi-party computation [ACA+17] have started to shed light on this issue. Such techniques have been used both for training and classification phases in deep learning [SS15], clustering [GLN13], and decision trees [BPTG15]. To allow programmers to develop such applications, secure computation programming frameworks and languages are designed for general purposes [HKoS+10, RHH14, O+16, BKLS18, EESA+12]. However, these approaches do not consider complex relationships among members and assume members share their all data or nothing. We view our efforts in this work to be complementary to much of these works. CPL can be integrated into these frameworks to establish partnerships and manage data exchange policies before a computation starts.

2.7 Secure and Differentially Private Computations in Multiparty Settings

There have been many works on the secure computation of linear regression over distributed databases [KLSR05, DHC04, KLSR09, SKLR04, HFN11]. In these, the threat model is considered as a third party that does not have access to data, but curious about it. However, one of the parties may want to release the model function after computing function securely, which still poses threats to the individuals [NS08, GKS08, EESA+12]. DP copes with this problem as it injects a certain amount of noise to the results of the queries to mask the individuals in the database. Indeed, there have been different works about the DP [DKM+06, DMNS06, Dwo08, MT07] and particularly about differentially private linear regression [CMS11, BST14, DJW13, F+14, JT13, ZZX+12, STU17]. However, these works consider DP without SMC. Although they are useful, they only provide privacy guarantees that the output of queries does not carry information about the individuals.

Approaches combining SMC and DP to provide both individual-level privacy and secure computation would be more secure. However, combining DP and SMC is not trivial; indeed, it is a rather challenging task since the application of centralized DP just after SMC in client-server settings would leak the model to an untrusted data collector, which results in a privacy violation of individuals in the database. Applying distributed DP directly on the local data held by the parties is more secure, but if each user independently injects noise randomly, it may lead to an excessive or uncontrollable amount of accumulated noise at the data collector end. Recent works focused on combining SMC and DP [GXS13, CA13, SCR+11], but none of them focused on linear regression. As pointed in [ZZX+12], the main reason behind this is that the regression analysis involves an optimization problem, which makes

it harder to control the required amount of noise, and if the data is also distributed among parties, that makes it much more difficult to control the privacy-accuracy trade-off introduced by DP. In another relevant work [PRR10], a combination of SMC and DP is proposed for aggregate classifiers. However, this approach injects the noise to the optimum model parameter. This resulted in excessive noise in the global model and significant loss in the accuracy. Particularly, the experimental evaluation shows that when the classifier is locally trained, the error rate obtained from locally trained classifiers is higher than the optimum error rates that could be obtained from a centralized approach. However, in our work, we take a different approach from this work. We deploy FM [ZZX+12], which adds noise to local statistics, which provides the same model as the centralized approach. Lastly, even though a similar idea is proposed in [AHPW15], it is not analyzed in detail.

PART I - PRIVACY-AWARE ALTERNATIVE AUTHENTICATION METHODS

CHAPTER 3

WACA: WEARABLE-ASSISTED CONTINUOUS AUTHENTICATION

3.1 Introduction

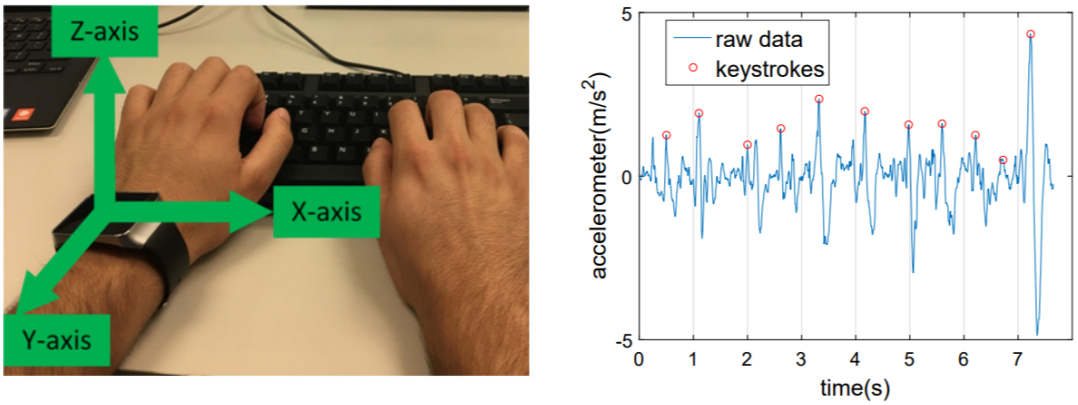
The majority of the current user authentication methods rely on password authentication. However, password authentication methods are subject to many security drawbacks [BHvOS15, GU13]. Many practical attacks have been demonstrated that the passwords can be either stolen or bypassed [Dic16, TGC16]. To mitigate these threats, Multi-Factor Authentication (MFA) methods were proposed [Dis17, SKH+19, ALB+19]. In MFA, the user credentials are checked from two or more independent sources, and even if the attacker steals one factor, it would still have to overcome the burden of other factors. Though, whether it is one-factor or MFA [ALB+19], a one-time login process does not guarantee that the identified user is the real user throughout the login session. Even if it is a legitimate insider who has been authorized once, a forever access is provided in most cases not to interrupt the current user.

An authentication mechanism, which re-verifies the user periodically without breaking the continuity of the session, is vital [Goo16]. For example, users may share their passwords with family members, friends, colleagues, or an already-authenticated user may walk away without locking his/her computing platform (e.g., laptop) for a short time or may intentionally hand it to a non-authenticated co-worker trusting that s/he will not perpetrate anything nonsensical or malicious or a malicious former employee or disgruntled worker may want to use his/her former privileges. In all these cases, as long as the original login session is actively used,

there is no mechanism to verify that the initial authenticated user is still the user in control of the computing environment.

In this chapter, we introduce a novel *Wearable-Assisted Continuous Authentication* framework called WACA, where a wearable device (e.g., smartwatch) is used to authenticate a computer user continuously utilizing the motion sensors of the smartwatch. Specifically, WACA uses *sensor-based keystroke dynamics*, where the typing rhythm of the user is captured by the motion sensors of the smartwatch worn by the user. In essence, keystroke dynamics is one of the behavioral biometrics that characterizes the users according to their typing pattern. Note that most conventional keystroke-based authentication schemes in the literature [TTY13] have used *dwell-time* and *flight-time* as unique features of the users. These features are directly obtained by logging the timing between successive keystrokes. However, in WACA, the feature set is richer and more flexible since 6-axis motion sensor data can provide not only timing information, but also the key-pressing pressure, hand rotation, and hand displacement, etc. Our feature set consists of 14 different sensory features from both time and frequency domains. These features are applied to 6-axis motion sensor data, obtaining 84 features in total, jointly considering the 6-axis data. Finally, different distance measures are used to compare the registered and the unknown profile of the user as it was shown that they performed well in similar contexts [KM09a, SPW13]. Also, in another work [MMC+14], users are classified according to the sequence of interactions (e.g., typing, scrolling), where the user wears a bracelet with motion sensors and radio. However, that work [MMC+14] has been shown as insecure in another work [HSU+16]. As explained, our work differs from other works in several ways to tackle those flaws and strengthen our design.

We tested the performance, efficiency, and security of WACA with more than thirty real users and data collected from them. We specifically evaluated WACA in



(a) (b)

Figure 3.1: (a) The reference coordinate system for accelerometer and gyroscope sensors. (b) A sample raw data collected from the accelerometer of the smartwatch and keystrokes detected by using peak detection methods while typing the word "smartwatch".

terms of three metrics: (i) *How accurately can it authenticate the genuine users and lock out the and impostor users?* (ii) *How fast can it detect an impostor?* (iii) *How accurately can it identify an impostor from its typing pattern?* Moreover, we also evaluated the robustness of our proposed method against powerful attacks, including, *imitation* [TGG13, HSU+16], *statistical* [SP13, SSCG16], and insider attacks.

3.2 Design Rationale: Why Should it Work?

In this section, we study how motion sensors of a smartwatch are impacted when typing on a keyboard and see if the data can be used to identify users. Particularly, we analyze a case that a user wears a smartwatch and types on a qwerty-type built-in keyboard of a computer. Our goal is to collect keystroke information from the built-in motion sensors (i.e., accelerometer and gyroscope) of the smartwatch during the typing activity. To collect smartwatch sensor data, we developed an Android Wear app that records the raw sensor readings from the motion sensors.

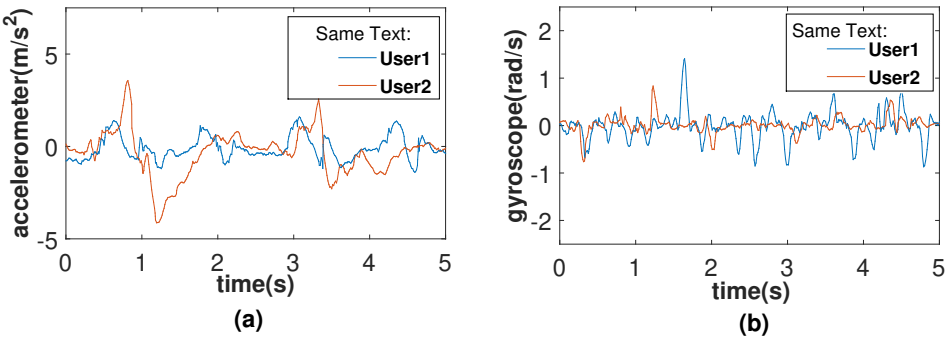


Figure 3.2: Comparison of two different users' (a) accelerometer (b) gyroscope readings while typing the same text.

In our experiments, we used linear acceleration composite sensor data, which combines the data of accelerometer and gyroscope to exclude the effect of gravity[1]. Note that the accelerometer and gyroscope sensors provide three-dimensional sensor data, where the reference coordinate system associated with the sensors are illustrated in Figure 3.1a. As z-axis of the accelerometer sensor is directly affected by the key up-down movements of a user while typing, the most significant changes are observed in the z-axis. Therefore, the z-axis of the data provides the best information for keystroke features such as holding time, pressing pressure, etc. Moreover, another observation is that even if the device is placed flat on a desk, the sensors generate a certain level of noise, which needs to be removed by filtering, as explained later.

Sample data in Figure 3.1b was acquired from the z-axis of the accelerometer while typing the word "smartwatch". It can be seen how the value of the accelerometer makes peak points. As the acceleration through the gravity corresponds to the going down of the accelerometer, the peak points in the figure correspond to the keystrokes in the typing activity. While the amplitude of the peak is related to how strong the key press is, the width of the peaks is associated with how long the key

[1]For brevity, we use acceleration to refer to the linear acceleration.

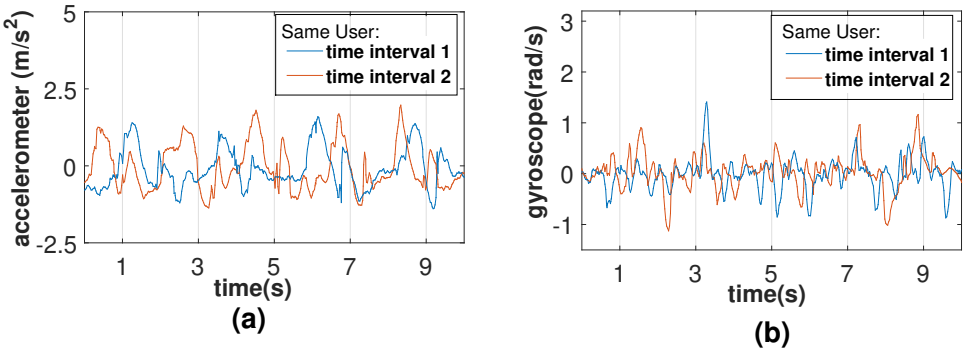


Figure 3.3: Comparison of the same user's sensor data over two different time intervals with (a) accelerometer, (b) gyroscope.

is pressed. These are simple statistics that can be used to identify users. These and other features will be further analyzed in detail in Section 3.4.

Moreover, we conducted two more simple experiments using the accelerometer and gyroscope data on the smartwatch, and we made the following two observations:

- *Observation 1: Different users exhibit different patterns even if they type the same text.*

In this experiment, we compared the data collected from two different users while typing the same text. Figure 3.2 presents the sensor data of the two users' accelerometer and gyroscope data for a given time interval. The distribution of the accelerometer data in Figure 3.2a shows clear differences such as the magnitude of peaks, inter-arrival time of peak points, the width of peaks, etc. On the other hand, the gyroscope sensor measures the rotation of the watch. As seen in Figure 3.2b, the number of peaks or the magnitude of the peaks are different for different users; so these features are viable candidates to recognize different users.

- *Observation 2: Same user follows similar patterns over different time intervals even while typing different texts.*

In the second experiment, the data was collected from the same user over two different time intervals corresponding to the different texts, and the plots are given in Figure 3.3. As seen in Figure 3.3a, the amplitudes and widths of the peaks are similar in magnitude, but with a phase shift, meaning leading or lagging. On the other hand, the same leading or lagging of similar shapes can also be seen in the gyroscope data in Figure 3.3b.

These two observations justify the rationale that keystroke dynamics obtained from smartwatch accelerometer and gyroscope sensors can differentiate different users as classical keystroke dynamics and the same users can be detected over different times even while typing different texts. Although these are just preliminary observations, our framework will be further tested and evaluated with extensive experiments using real user data in Section 3.5.

3.3 System Model

In this section, we explain design goals, our assumptions, and the adversary model.
Design Goals: In WACA, our design goals is similar to the ones given in [PPJ03]: Our system should be *universal* (i.e., the biometric features exist for everyone), *unique* (the features are specific for everyone), *permanent* (the biometric features always exist), *transparent* (the system works without interrupting the user), *continuous* (the system should provide continuous user data), and *accurate* (the system works with low error rate). WACA achieves the first five goals by its design and the accuracy is tested in Section 3.5.
Assumptions: For WACA, the following assumptions are considered:

- We assume that the user wears a smartwatch, which is equipped with motion sensors and either Bluetooth or WiFi. We also assume that an app to collect

the motion data is already installed on the smartwatch, and it is paired with the computer that will be authenticated. For our work, we built a custom Android Wear app to collect and process the sensor data.

- We assume that by pairing devices, a secure communication channel is already established between the computer and smartwatch as well as between the computer and the remote or local authentication server. This secure communication channel should keep the sensor data secure in both transitions and at rest.
- The WACA framework acts as a complementary second-factor, and it has the flexibility to work any first-factor authentication system, and it is assumed that the system has already a first authentication factor. The first factor could be one of the password-, token-, or biometric-based systems. Note that the first factor of authentication is beyond the scope of this work.

Adversary Model: The primarily considered adversary model is an attacker who somehow bypassed the first factor (e.g., password, token) of the authentication system and it has physical access to the computing terminal. The attacker is likely to be an insider or co-worker, but it can also be an outsider, just passing by the victim's computer. Attacker's goals can include, but not limited to, trying to get some important information from the victim's computer, taking action on behalf of the victim, or trying to get access to the assets that s/he does not have permission (i.e., privilege abuse). More specifically, we consider the following attack scenarios by considering WACA is deployed in a real-world system:

- *Attack Scenario 1:* The victim is one of the employers and forgets to lock his computer and *an outsider* (e.g., a mail courier) who is just passing through the office tries to get access to the victim's computer. In this scenario, if

the attacker is not aware of WACA, s/he will attempt to use the victim's computer. If the attacker is aware of WACA, s/he will first look for the victim's smartwatch and then try to keep the system logged in.

- *Attack Scenario 2:* We consider the attacker can also be a malicious insider and thereby the attacker also has a registered smartwatch, but its typing profile is registered together with its username. This type of attacker tries to get access to the system's assets that s/he does not have permission (i.e., privilege abuse). In this scenario, the attacker watches its victim (e.g., supervisor) for a proper timing that its victim leaves the computer unlocked for some time to go to lunch or to get coffee, etc. (aka *lunchtime attack* [ERLM15]). The attacker can either try to bypass the system via providing data from his smartwatch or can try to use the victim's smartwatch somehow obtained (e.g., can steal it or victim can leave it behind).

- *More Powerful Adversaries:* Furthermore, a powerful adversary can be aware of WACA and try to defeat it using special tools and skills by *imitating* legitimate users [TGG13, HSU+16] or launching *statistical attacks* [SP13, SSCG16]. This powerful adversary (insider or outsider) can be a human or a trained bot. In imitation attacks, the attacker wears the victim's smartwatch either via after stealing it, or the victim can leave it behind for a while and the attacker can try to impersonate the victim. On the other hand, the statistical attack is more complex and requires special tools and skills. Hence, WACA also considers these powerful attack scenarios in its adversary model.

The security evaluation of these attack scenarios and how WACA is robust against insiders, imitators, and statistical attackers are explained more in Sections 3.5.1 and 3.5.2.

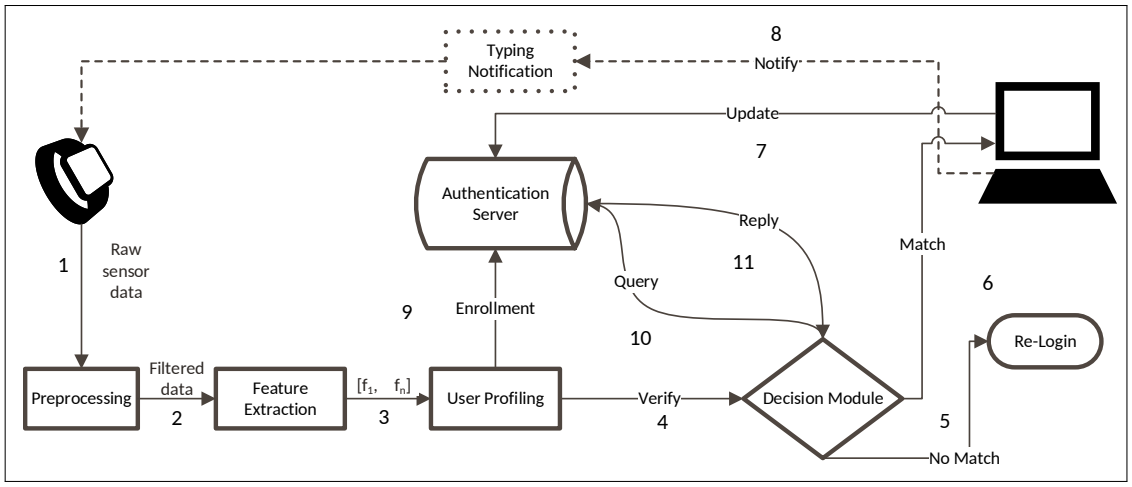


Figure 3.4: WACA framework architecture and key components.

3.4 WACA Architecture

In this section, we present the details of the WACA. WACA is a typing-based continuous authentication system using the accelerometer and gyroscope sensors of a smartwatch. WACA framework is complementary to the first-factor authentication mechanisms, and it is flexible to work with any first factor.

3.4.1 Overview

WACA consists of four main stages: *Preprocessing, Feature Extraction, User Profiling, and Decision Module*. These stages, which are shown in Figure 3.4, work as follows:

- First, the raw sensor data is acquired from a smartwatch (1) through an app installed on the watch. Then, the raw data is transmitted to the computer through a secure wireless channel, and the rest of the stages are performed on the computer except that Authentication Server (AS) is located in a trusted place.

- As the collected data includes a certain level of noise, in the preprocessing stage, the raw data is cleaned up by filtering (2) and transformed into a proper format for the next stages.
- Then, incoming data is used to extract a set of features (3). This set of features, namely *feature vector*, represents the characteristics of the current user profile.
- In the enrollment phase (9), the created feature vector is stored in the AS.
- In the verification phase (4), the queried user profile is dispatched from the AS to the decision module (10, 11).
- The decision module computes a similarity score between the returned profile and the provided profile for the current user to make a binary authentication decision (match/no match). If the decision is a no match (5), then the user's access to computing terminal will be suspended, and the user will be required to re-authenticate using the primary authentication method (e.g., password).
- However, when the decision is a match (6) then the user's access will be maintained. The profile of the current user in the AS will be updated after the correct match of the user profile (7). In WACA, this update frequency is a system parameter and can be set by the admin in the security policy. An optimum value of this parameter can be set after experimenting with different values in a real-world implementation. In this way, the user profile will be kept up-to-date over time.
- Whenever a typing activity is initiated on the keyboard of the computer, the smartwatch will be notified (8) again by the terminal to start over the authentication process continuously.

In the following subsections, we explain the details of WACA and its key stages.

3.4.2 Data Collection

In WACA, *data collection* refers to capturing sensor readings from the user's smartwatch through a secure wireless communication channel (i.e., via WiFi or Bluetooth). An app is installed on the smartwatch to listen to the physical sensors. Then, the raw sensor data is transmitted to the computer through a secure communication channel.

Each row of the collected raw data of accelerometer is represented in the format of $\vec{acc} =< t_a, x_a, y_a, z_a >$ and gyroscope is represented as $\vec{gyro} =< t_g, x_g, y_g, z_g >$, where t stands for timestamps and x, y, z represent the different axis values of the accelerometer and gyroscope sensors. Each of t, x, y, and z is stored as a different vector. The length of the vectors directly depends on the sampling rate of the sensors and the time interval of the data collection. In WACA, the parameter *sample size* refers to the length of these vectors, and it is set as a configurable parameter while the parameter *sample rate* is a constant system parameter that is characterized by the wearable device and app.

3.4.3 Preprocessing

In WACA, *preprocessing* stage refers to the preparation of raw sensor readings for the next stages. It consists of cleaning and transformation of the raw data. In the cleaning part, the noise is removed. To remove the effect of the noise from data, we apply M-point Moving Average Filter (MAF), which is a simple low-pass filter and it operates by taking the average of M neighbor points and generates a single output. M-point filtering in equation form can be expressed as follows:

$$\dot{y}[i] = \frac{1}{M} \sum_{j=0}^{M-1} \dot{x}[i+j], \tag{3.1}$$

where $\dot{x}$ is the raw sensor data, $\dot{y}$ is the new filtered data, and i indicates the current sample that is averaged. The filtered data becomes smoother than the raw data without altering the value at that point.

After filtering the noise, the data is transformed into appropriate forms for the next stage. Particularly, different types of sensor data are separated according to an assigned ID number during the sensor registration and then x, y, and z axes of the sensor values are recorded as different vectors e.g., $\vec{x_a} =< {x_a}^1, ..., {x_a}^n >$ and $\vec{x_g} =< {x_g}^1, ..., {x_g}^n >$ for a profile of n samples.

3.4.4 Feature Extraction & User Profiling

In WACA, *Feature Extraction* (FE) refers to the transformation of the time series raw data into a number of features. In order to create the feature vector, each feature is computed using the data vectors. As an example, the first feature is calculated from a function f, i.e., $f_1 = f(x_a, y_a, z_a, x_g, y_g, z_g)$ and the second feature is calculated from another function g, i.e., $f_2 = g(x_a, y_a, z_a, x_g, y_g, z_g)$ etc. Then, the final feature vector $\vec{f} =< f_1, f_2, ..., f_n >$ is generated using all the calculated features.

As each element of the feature vector has different ranges, some of the features can be dominant in the distance measurement. To prevent this and create a scale-invariant feature vector, we apply normalization to the feature vector to map the interval $[x_{min}, x_{max}]$ into the unit scale [0,1]. We formulate this linear normalization process in WACA as follows:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{3.2}$$

where x_{min} and x_{max} are the minimum, and maximum value of the features of the user's enrolled templates.

Table 3.1: Feature set extracted from sensor data in WACA.

Domain	**Feature**	**Length**
Time	Mean, Median, Variance, Average Absolute Difference of Peaks, Range, Mode, Covariance, Mewan Absolute Deviation (MAD), Inter-quartile Range (IQR), Correlation between axes (xy, yz, xz), Skewness, Kurtosis	$12 * 6 = 72$
Frequency	Entropy, Spectral energy	2*6=12
Total #	-	**84**

After generating the final feature vector $\vec{f}$, in the user profiling stage, a user profile $\vec{p}$ is generated by adding the user ID and start and end timestamps of the data sample, i.e., $\vec{p} =< userID, t_{start}, t_{end}, \vec{f} >$. If the user is in the enrollment phase, this profile is transmitted to the AS to be stored in a database. Finally, if the user is unknown, and a typing activity notification comes from the computer, the profile is passed to the Decision Module.

The feature set used in our framework is presented in Table 3.1. These features were chosen as they performed well in similar contexts [KM09a, SPW13].

3.4.5 Decision Module

The last stage in WACA is the *decision module*. The task of this stage is classifying the user as authorized or unauthorized for given credentials entered during the initial login. For authentication, we use distance measures. The distance measure methods simply calculate the distance between two vectors or data points in a coordinate

plane. It is directly related to the similarity of compared time-series data sets. The most widely used distance measure is *Euclidean Distance*. It is actually just the distance between two points in vector space and is the particular case of *Minkowski Distance*, which is expressed as follows:

$$distance(\vec{x}, \vec{y}) = (\sum_{i=1}^{n}(x_i - y_i)^p)^{\frac{1}{p}}, \tag{3.3}$$

where $\vec{x} = (x_1, x_2, ..., x_n)$ and $\vec{y} = (y_1, y_2, ..., y_n)$ are the set of sensor observations to be compared. If $p = 2$, it is Euclidean distance and has been extensively used in the keystroke-based authentication methods. WACA calculates the distance and returns the result by comparing it with a configurable predetermined threshold value (i.e., genuine if $distance < threshold$, impostor if $distance \geq threshold$), the impact of which is analyzed in Section 3.5.1. Indeed, this threshold measures the confidence of the decision for a given user.

In addition to Euclidean and Minkowski Distances, there are several distance measurement methods utilized in biometric authentication systems which may perform differently depending on the context. Therefore, we also tested different distance metrics in our experiments to see, which shows the best for WACA. Other distance metrics that we tested in our experiments are *Cosine Distance, Correlation Distance, Manhattan (Cityblock) Distance*, and *Minkowski with p=5*. The performance of each one is given in Section 3.5.1.

3.5 Performance Evaluation

We tested the performance, efficiency, and security of WACA with more than thirty real users and data collected from them. We specifically evaluated WACA in terms of three metrics: (i) *How accurately can it differentiate between genuine and impostor users?* (ii) *How fast can it detect an impostor?* (iii) *How accurately can it identify*

an impostor? In for these purposes, we first conduct authentication experiments. In these, we measure how WACA performs when users type a different or the same text. We also analyze how the sample size and the detection technique impact WACA's performance. The effect of the sample size allowed to evaluate the quickness of WACA. Finally, we also conducted an experiment to show how successful WACA would be in identifying insider threats.

Data and Collection Methodology. In our experiments, we collected data from 34[2] human subjects. [3] During the collection of data, an Android Wear smartwatch with an installed data collection app was distributed to the participants, and the participants were asked to type a text. The participants were free to choose the hand (left/right) on which they wore the smartwatch. The choice of the hand that the participants wore the smartwatch was left to the participants. Moreover, they were also given the freedom to adjust the sitting position and the keyboard and screen position according to their comfort levels. It is also worth noting that sitting and wrist position (i.e., if it is resting on the table or maintained in the air) may affect the performance. Therefore, a real-world implementation may require further calibration before enrolling the users to the system.

Throughout these experiments, we utilized a standalone qwerty keyboard to have generic results. Before typing each text, the participants were also given enough time to read the texts to make them familiar with the text as typing a familiar text is a more common activity.

[2]Not all of them participated in all experiments.

[3]Our research study with the human subjects was conducted with the appropriate Institutional Review Board (IRB) approvals.

The participants were involved in two typing tasks conducted in two different sessions. They were asked to type with their normal typing style without noticing that their data was recorded. The two data sets were compiled as follows:

- *Typing Task-1:* 20 participants are involved in this task, and the participants were asked to type a story from a set of short and simple stories from the American Literature[4] for four minutes. The story was chosen randomly by the participants. On average, four minutes of data corresponds to 25000 samples for each participant (Total: 850000 samples).
- *Typing Task-2:* 20 participants are involved in this task and for this data set, all the participants were asked to type the same text[5] for four minutes. For each participant, almost the same amount of data is collected as Typing Task-1. This dataset is essential to be able to measure the quality of the features.
- *Typing Task-3:* 34 participants are involved in this task, and the participants were instructed to imitate someone else' typing pattern by watching the prerecorded video of the other person. For these experiments, one of the participants was recorded on video while typing a short and simple sentence for 15 seconds from a perspective that the hand motions, smartwatch, keyboard, and the screen could be seen. Although it was not required, the perspective allowed to infer what the victim was typing by watching. This dataset was primarily used to analyze the attacking scenarios.

Note that in all the experiments, the dataset obtained from all these tasks were always used by dividing them into equal size chunks. Therefore, even if all the

[4] `https://americanliterature.com/100-great-short-stories`

[5] `https://en.wikipedia.org/wiki/The_Adventures_of_Tom_Sawyer`

participants in Typing Task-2 typed the same text, the compared samples always corresponded to different texts for a participant.

Moreover, in our experiments, we split the collected data sets into equal size chunks, called *sample size*. It is the number of samples (i.e., row) in a chunk. Each chunk consists of 8 columns of data, two of which are timestamp, and the others are 6-dimensional sensor data. The sample size is the main system design parameter in our experiments as it has a direct impact on the time required to collect data. Particularly, the time t required to collect data with the sample size can be represented as $t = sample\ size/100$ in seconds as the sampling rate in our experiments was $100Hz$.

Performance Metrics. In the authentication experiments, we used *Equal Error Rate* (EER) as it is a commonly accepted metric to assess the accuracy of WACA. EER is calculated using two metrics: False Acceptance Rate (FAR) and False Reject Rate (FRR). FAR is the rate of incorrectly accepted unauthorized users among all the unauthorized attempts: The increase in FAR is a direct threat to the system's security level (i.e., confidence level on the decision). For more valuable assets, increasing the threshold will decrease FAR. On the other hand, FRR is the rate of incorrectly rejected authorized users among all the legitimate authentication attempts. Contrary to FAR, FRR can be decreased by decreasing the value of the threshold. Indeed, the threshold value effectively measures the confidence of the decision for a given user. Finally, EER is the point that gives the closest FAR and FRR point for a given threshold (ideal EER is the intersection point of FAR and FRR) and the lower the EER, the better is an authentication system.

3.5.1 Results

In this section, we present and discuss the evaluation results.

Impact of the text dependency. In this experiment, our goal is to analyze how EER changes among the participants. We try to answer: *How does WACA perform with the typed text?* This is also a more advanced analysis of the framework and the fundamental idea than that of in Section 2.

Specifically, for this experiment, we used Typing Tasks 1 (any text) and Typing Task 2 (the same text) dataset and we fixed the sample size to 1000 and used Manhattan (Cityblock) as a representative distance measure to compare the samples. Note that as later shown and analyzed in Figures 3.7-3.8, this distance metric was chosen as it performed the best among the different distance measurement techniques. This is because Manhattan is rectilinear distance, considering the absolute differences and is more suitable for natural settings [KCB03, PGR07, BG04]. For each sample of a particular user, we computed the differences from other users' samples. For this purpose, we computed the $N \times N$ dissimilarity matrix, where N is the total number of samples for all the participants. The dissimilarity matrix was calculated by measuring the similarity of each sample to all the other samples using leave-one-out cross-validation[6] method [JL10].

Then, for a given threshold and participant, the ratio of the rejected and accepted samples was computed to obtain FRR and FAR, respectively. This process was repeated by incrementing the threshold by 0.01 in each step for all the samples of all the participants. This gave us a set of EER for each participant. Note that in a real system, FAR/FRR rate can be tuned according to the system preferences,

[6]Even though to show the feasibility of our method, we tested our method with leave-one-out cross-validation, collecting and storing more than one samples from each user at the enrollment phase may impact the accuracy in real-life implementations.

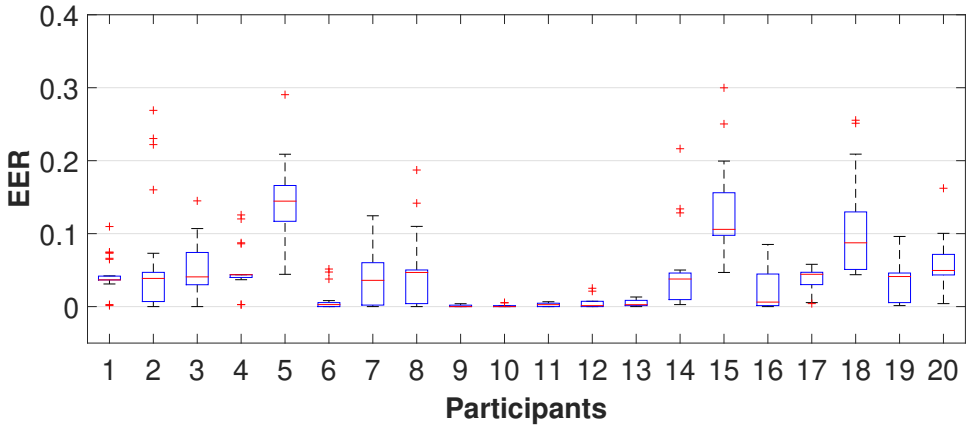


Figure 3.5: EER for each participant with a sample size of 1000 using Manhattan (Cityblock) distance metric during Typing Task-1. Average EER is 0.0513.

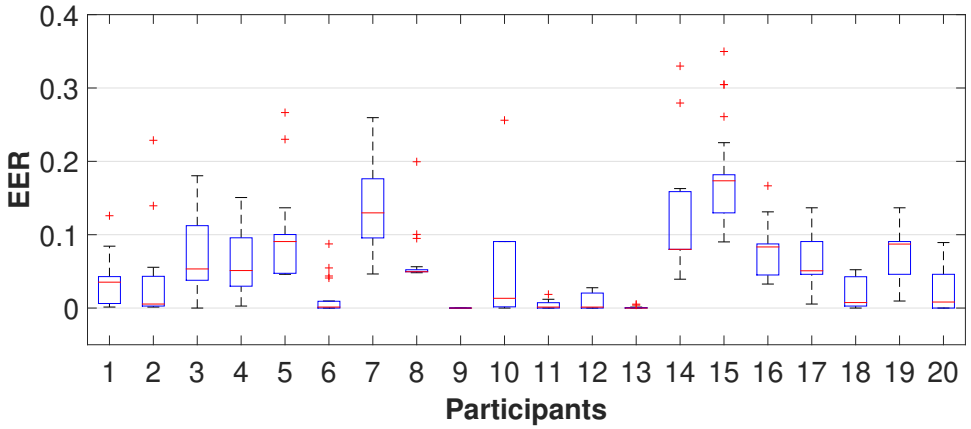


Figure 3.6: EER for each participant with a sample size=1000 using Manhattan (Cityblock) distance metric during Typing Task-2. Average EER is 0.0647.

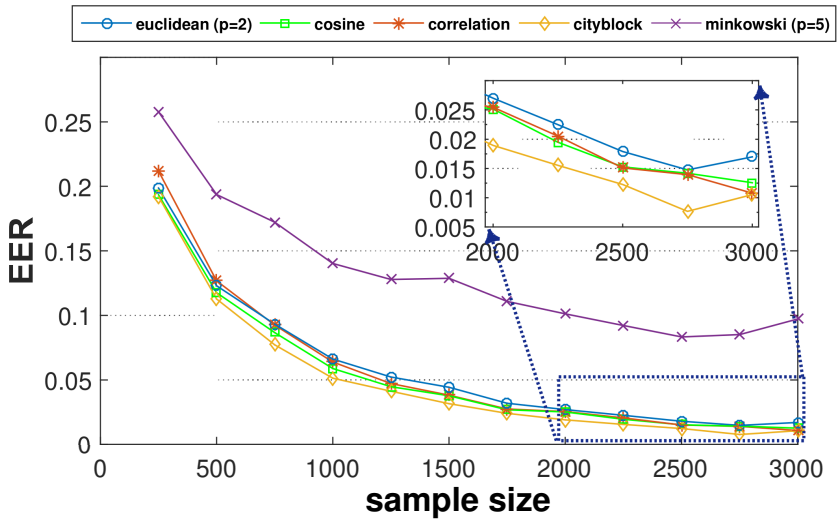


Figure 3.7: Average EER according to different sample sizes using different distance metrics while users are performing Typing Task-1.

but here our purpose is to find an acceptable performance metric for WACA. The results are plotted in Figure 3.5 for Typing Task-1 and Figure 3.6 for Typing Task-2. Average EER for the Typing Task-1 experiment was 0.0513. Figure 3.6 compares the EER of participants for the Typing Task-2 experiment. Average EER for this experiment was 0.0647.

If we compare the ERR of each participant in both the experiments, we see that they are also close to each other, where a few of the participants perform very distinctive behaviors (e.g., participant 15). However, the overall distribution of EER over the participants is similar in both the experiments. Recall that in Typing Task-1, all the participants typed different texts, while they typed the same text in Typing Task-2.

Overall, in this analysis we report the average EERs of both the experiments are close (around %1), which supports the usability of WACA regardless of the typed text for the continuous authentication session.

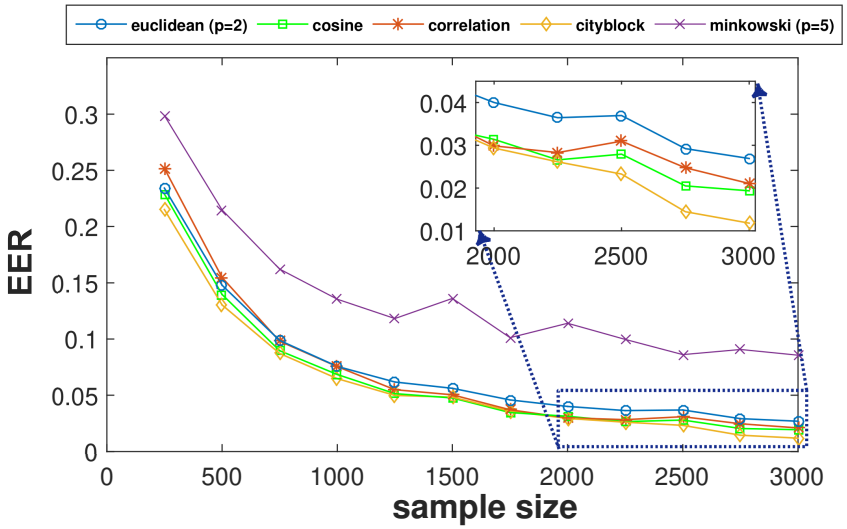


Figure 3.8: Average EER according to different sample sizes using different distance metrics while users are performing Typing Task-2.

Impact of the sample size and the distance measuring technique. In these experiments, *our goal was to assess how different sample sizes and the distance measuring techniques used in WACA impact the performance.* For this, we varied the sample size from 300 to 3000 and utilized five different distance measuring techniques, Euclidean (p=2), Cosine, Correlation, Cityblock, and Minkowski (p=5). Again, two types of participant dataset, Typing Task-1 (any text) and Typing Task-2 (the same text), were used. Figure 3.7 (Typing Task-1) and Figure 3.8 (Typing Task-2) present the main results when the sample size increases.

As can be seen in Figure 3.7 when the participants typed different texts, the EERs are generally decreasing with the increase of sample sizes as expected. The EERs go under 0.05 after the sample size of 1500 for all the distance metrics utilized except for Minkowski (p=5). Then, the EER is converging to the value of 0.01-0.02 through the sample size of 3000. In the best case, EER 0.007 is achieved with the sample size of 2750 for the Manhattan (i.e., Cityblock) distance measurement technique.

Figure 3.8 presents the results of the same-text experiment (Typing Task-2). As in Figure 3.7, the general behavior is that the EERs are decreasing with the increase of the samples. The lowest EER of 0.01 is achieved using the Cityblock distance measuring technique at 3000. We also see the convergence of EER in Figure 3.8 as Figure 3.7. Plots are starting to converge around sample sizes 1500-2000 and converging to 0.01 for Cityblock and Correlation distance measuring techniques. We also see that at 3000, 0.02 EER is obtained for Cosine and Correlation techniques. However, if shorter data collection time is of interest, a sample size of 2000, which needs 20 seconds for data collection, gives 0.03-0.04 EER. However, if we increase the sample size, both the accuracy and the data collection time are increasing. This means the time needed to catch an adversary or more generally, the re-verification period would also increase. Therefore, an optimal sample size should be adjusted according to the preferences in a real application based on the usage needs or security policies.

To conclude, the features in WACA can successfully differentiate the users from their typing rhythm with a minimal error rate (1%) independent of the typed text. There is an inherent trade-off between the EER and data collection time, which should be configured according to the security needs of an organization.

The accuracy of insider threat identification. As noted earlier, the insider threat detection is important in continuous authentication systems as a potential attacker is likely to be an insider. To effectively locate such an insider attacker within an organization where WACA is employed, an identification mechanism is needed. Depending on the security policy of the organization, the management may want to do an investigation to find the insider attacker. In this case, we will have many unknown samples of the attacker to find the owner of the samples, and we will need a one-to-many classification task to exactly detect an insider attacker. For

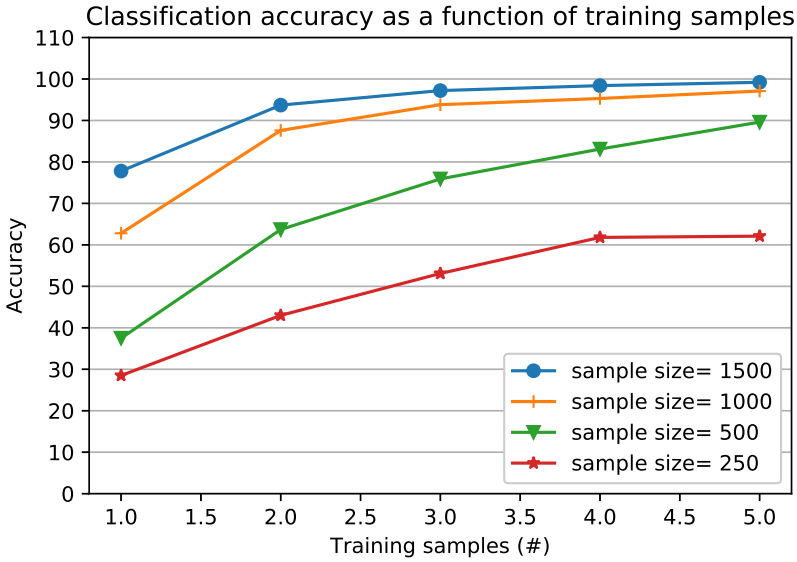
Table 3.2: Evaluation of the insider threat identification results with seven different machine learning algorithms. MLP yields the best result and the training/validation graphs of the MLP algorithm are given in Table 3.9.

Classifier	**Typing Task-1**	**Typing Task-2**
SVM	98.7	98.1
Random Forest	98.9	97.8
Naive Bayes	93.6	87.3
Decision Tree	62.1	62.1
MLP	**99.0**	**99.2**
kNN	96.4	96.8
Logistic Regression	90.5	93.7

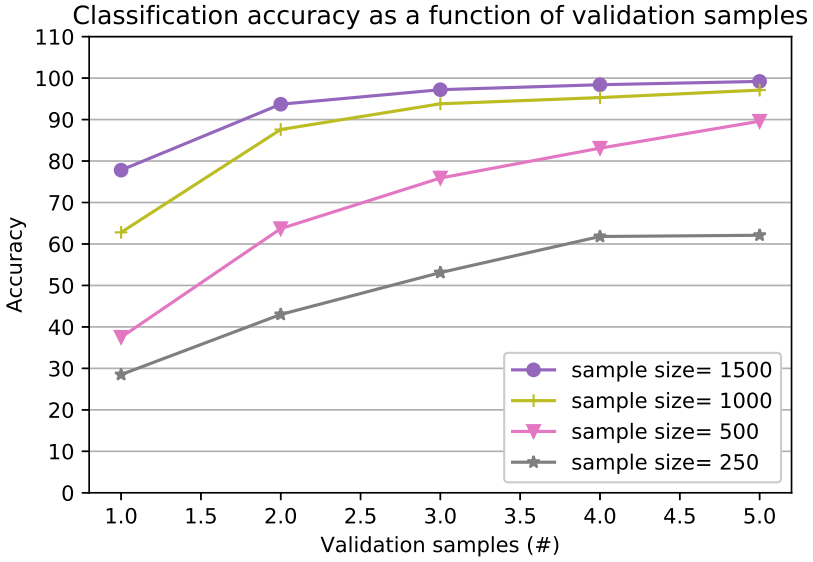
this purpose, we fix the sample size to 1500 and the number of training sample to five. With those parameters, we tested different machine learning algorithms and results are presented in Table 3.2. Here, we assume that the insider's data is also stored in the authentication server's database (training set) as a legitimate user.

According to the results given in Table 3.2, the most accurate results are obtained with the Multilayer Perceptron (MLP) algorithm. This happens because of two reasons. First, MLP is a neural network model, which maps a set of input data into a set of outputs through the interconnected processing elements (neurons). The main advantage of MLP is that it approximates highly nonlinear functions between input and output [GD98]. Second, when we look at literature [ASL15, SIG+19, MT16, BVACM18, BCE+18], MLP is giving very high accuracy with the features obtained from noisy sensor data collected from devices like smartphone or smartwatch. Moreover, Table 3.3 shows the parameters used in the machine learning algorithms given in Table 3.2.

We also analyzed the impact of the sample size and the size of the training data on the accuracy. For this, we focused on two test scenarios that would be relevant in real investigations and tested the efficacy of 7 different machine learning algorithms. As seen in Table II, MLP performed the best and accordingly we picked MLP as a



(a)



(b)

Figure 3.9: a) Training and b) Validation curve of MLP algorithm. Please note that since the validation set size is 0 as provided in Table 3.3, the two curves are same.

Table 3.3: Parameters used in the Machine learning algorithms in Table 3.2.

Classifier	**Parameters**
SVM	The complexity parameter $c = 1$, $\gamma = 0.01$ Polynomial kernel with exponent 1
Random Forest	# of iterations (trees) = 100, # of features = unlimited
Naive Bayes	Kernel estimator = False Supervised Discretization = False
Decision Tree	tree= J48, confidence factor = 0.25
MLP	# of hidden layers = (attribs + classes) / 2 # of neurons per hidden layer = 1 Learning rate = 0.3 Momentum = 0.2 Validation set size = 0% Validation threshold = 20
kNN	$k = 1$ Search algorithm : linearNNSearch Distance function : Euclidean Distance
Logistic Regression	$Ridge = 10^{-8}$

representative algorithm to be used in these scenarios: *Scenario 1:*[7] In order to show that MLP does not show any over-fitting, we plot the training and validation curves in Figure 3.9. In the first scenario, we built our test model using the same text and tested again using the same text with the 5-fold cross-validation technique. For this scenario, we utilized Typing Task-2 Dataset for both the training and testing. This type of scenario can be useful as all the users are asked to type a provided text, and during the investigation, all users are requested again to type the same text. The results are presented in Table 3.4. *Scenario 2:* In the second scenario, the test model was trained with the same text dataset, which is the same for all the participants and tested using random-text experiments, where each user typed a randomly chosen text. For this scenario, we utilized Typing Task-2, Typing Task-1

[7]Please note that this is different the Typing Task-1 in Figure 3.2.

Table 3.4: The accuracy results insider threat identification experiments for different sample sizes in Scenario 1 and 2.

Scenario 1: Accuracy (%)

	Training Set				
Sample size	1	2	3	4	5
1500	77.8	**93.7**	**97.2**	**98.4**	**99.2**
1000	62.8	87.6	**93.8**	**95.3**	**97.1**
500	37.5	63.7	75.9	83.1	89.6
250	28.5	43	53.1	61.8	62.1

Scenario 2: Accuracy (%)

	Training Set				
Sample size	1	2	3	4	5
1500	55.8	80.1	88.7	89.8	**91.8**
1000	51.7	82.7	83.2	86.1	86.8
500	29.9	51.3	66.7	73.8	76.5
250	22.1	33.6	41.9	49.8	54.1

Table 3.5: Time taken to build the MLP model used in Scenario 1 in Section 3.5.

Time taken to build the model (seconds)

	Training Set				
Sample size	1	2	3	4	5
1500	0.99	2.02	3	3.99	4.94
1000	1	1.99	2.98	3.95	4.94
500	0.98	2.02	3.01	3.98	4.95
250	1	1.95	2.93	3.92	4.95

Datasets for training and testing, respectively. This scenario is suitable for cases where all the users are enrolled using the same text, but a user is verified while typing a random text. The results for this test scenario are presented in Table 3.4.

As can be seen in Table 3.4, in the best case, 99.2% identification rate of an insider threat can be achieved with the sample size of 1500 while the model is trained with five samples. Even with two samples of the insider, 93.7% accuracy rate can be achieved with the sample size of 1500. Finally, we also present the model training time for the insider threat detection in Table 3.5 and 3.6.

Table 3.6: Time taken to build the MLP model used in Scenario 2 in Section 3.5.

Time taken to build the model (seconds)					
	Training Set				
Sample size	1	2	3	4	5
1500	1	1.97	2.97	4.01	4.99
1000	1	1.99	2.95	4.04	4.93
500	0.99	2	2.98	3.98	4.98
250	0.99	1.96	3.02	3.96	4.96

Scenario 2 aims to answer the question of "*Can an insider be identified while typing a random text even if s/he is enrolled while typing a given text ?*" Table 3.4 presents the result of this question for Scenario 2. As can be seen from Table 3.4, similar to Scenario 1, the accuracy rates increase as the sample sizes and training set increase, and the time to build model and time required to catch the attacker is also increasing. Three training samples and the sample size is 1500 or four training samples with the sample size of 1000 may be the two most optimal choices for real cases.

Overall, WACA can achieve 0.01 error rate with almost 30 seconds of the data collection (see Figure 3.7 and 3.8) in the best case. If a shorter time is of interest, 0.02 error rate is achieved with 20 seconds of the data collection. Moreover, if five training samples with 1500 sample sizes are obtained from a potential insider threat, WACA could identify the insider with 99.2% accuracy rate while typing the provided text (see Table 3.4) or with 91.8% accuracy rate while typing a random text (see Table 3.4).

3.5.2 Advanced Attacks on WACA with More Powerful Adversaries

In this subsection, we evaluate the performance of WACA against two powerful attacks: imitation [TGG13, HSU+16] and statistical [SP13, SSCG16] attacks. In these attacks, the attacker is aware of WACA and can try to defeat WACA using special tools and skills.

Numerous attacks against classical keystroke dynamics that exist in the literature can also be used to attack WACA. The attacker can be a human or a trained bot. A human-type attacker can perform *zero-effort attacks*[8] [RP13] or *imitation attacks* [TGG13] to defeat the WACA's authentication system. In *zero-effort attacks*, the attacker tries to defeat the authentication system without any effort or prior knowledge. Zero-effort attacks will not be successful due to the low EER values in WACA as analyzed in the previous sub-sections. However, the effectiveness of the imitation attacks performed by a human should be investigated as noted in some recent studies [TGG13, HSU+16].

In addition to these attacks, another recent attack against the behavioral biometrics [SP13, SSCG16] has emerged, which is called *statistical attacks*. In this attack, a bot is first trained using typical user data from a large population. Then, the bot generates random permutations of the features to mimic a legitimate user. In addition to human and robot attacks, a *replay attack* using a key-logger [GOC12] is noted in the literature, which can also be performed against the keystroke dynamics. However, a key-logger installed on the computer can obtain only the timing of the keystrokes, which is solely not enough to use it in a replay attack against WACA as

[8]Also called *zero-information attack*.

there is not a way that a key-logger can obtain the three-dimensional sensor data collected by the smartwatch.

In the next sub-sections, we consider these two powerful attacks (imitation and statistical) and investigate the effectiveness of WACA against them. In these cases, the attacker would have somehow obtained the victim's smartwatch or manipulates his smartwatch. We use the zero-effort attacks as a baseline to evaluate the success of the imitation and statistical attacks. In imitation attacks, the attacker either can steal the victim's smartwatch or the victim can leave it behind for a while, then the attacker wears the victim's smartwatch and can try to impersonate him while attacking. On the other hand, the statistical attack is more complex and requires special tools and skills. In this type of attack, we assume the attacker can provide its input data to the system. It manipulates its username and profile data to get access to the computer that he does not have permission.

Imitation Attacks

In this subsection, we evaluate the performance of an imitating adversary, who knows that WACA is already installed on the current system. The adversary is assumed to be watching his victim by standing nearby or trying to imitate the victim's typing style by looking at the previously recorded video of the victim while typing. S/he is also assumed to be opportunistically waiting for the right time to mimic the victim.

To replicate this imitation attack scenario, we recorded a 15 seconds video of a legitimate user and presented this video to an attacker (i.e., another participant in our experiments). The video showed the user as s/he was typing and thus the hand, fingers, watch and keyboard were all visible. By watching the video (multiple times allowed in experiments), the attacker tried to imitate the legitimate user.

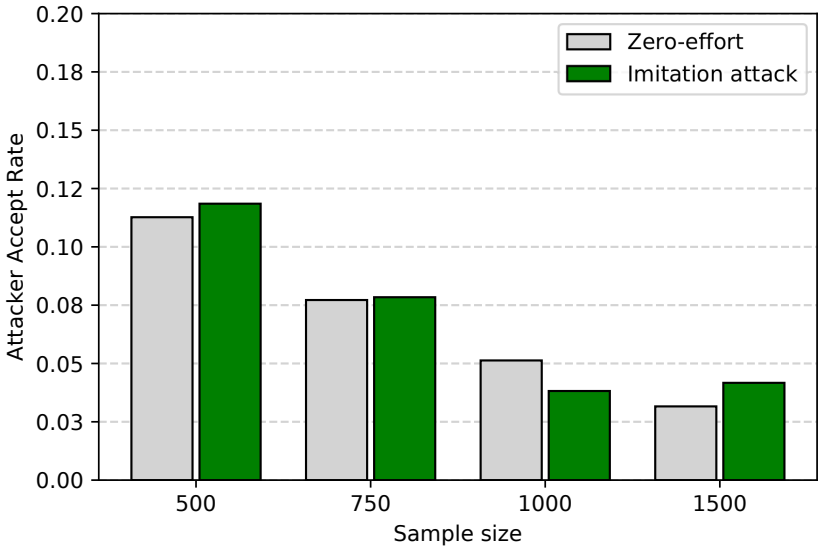


Figure 3.10: Attacker accept rates for different sample sizes. The results show that an imitation attacker has no more advantage than a zero-effort attacker.

Note that this scenario would increase the chances of a successful attack when compared to a real-life case where the attacker would possibly only have limited opportunity to watch a victim. We also collected the victim's typing data to evaluate the performance of the attackers. We computed EER for this attack scenario and compared it with the case when there was a zero-effort for the attack. In the zero-effort attack, we used the data set obtained in Typing Task-1 Dataset. We applied the leave-one-out method [JL10] by leaving the victim's data out as in the other authentication experiments While calculating EER (i.e., the intersection of FAR and FRR) of the victim. In the imitation attack, since we only had the impostor attempts, EER would be equal to the attacker's acceptance rate. We also note that WACA was directly tested without any change. The results are presented in Figure 3.10.

As presented in Figure 3.10, the attackers have different success rates (attacker accept rate) for different sample sizes. The highest success rate was achieved when

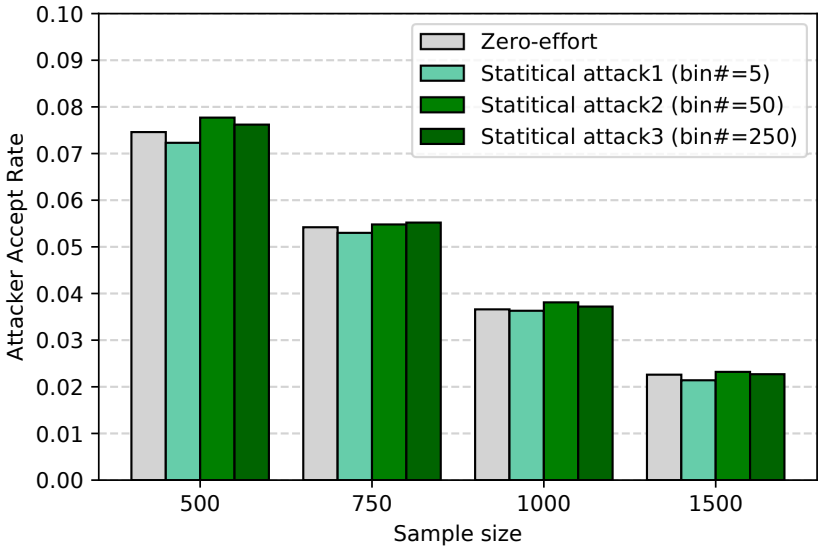


Figure 3.11: 3 different statistical attacks against WACA with different sample sizes.

the sample size is equal to 500, but the success rates are decreasing to the much lower rates as the sample sizes increase. A sample size of 500 corresponds to almost 2-3 keystrokes for the sampling rate used, which is not enough to measure and settle down for some of the features. So, this is not practical from the attacker's perspective. Beyond 1500, which corresponds to 15 seconds of sensor readings, the probability of an attacker to imitate a user is significantly decreasing (i.e., 0.04). *These results indicate that even though an attacker is aware of WACA in a targeted system, s/he still has a meager chance to be successful.*

Statistical Attacks

In this subsection, we evaluate WACA against statistical attacks. In this attack scenario, it is assumed that the attacker has a database obtained from AS consisting of the user profiles. Similar to the imitation attack, it is also assumed that the attacker can provide its input to the system. As mentioned earlier, this can occur either by obtaining the victim's smartwatch or if the attacker is an insider, it can

manipulate its input data to deceive WACA. It is also worth mentioning that we assume the attacker has only a limited amount of time to attack; therefore, it only tries the data that has the highest chance to get in, which we refer to as *topBins* in the attack algorithm that will be utilized and noted below.

Note that statistical attacks are very powerful attacks and it is successfully implemented to bypass the conventional keystroke-based systems [SP13]. It is based on the generation of fake (synthetic) inputs using common features of a given population. The idea behind this attack is using the random combination of the most common features of the population to defeat the authentication system. We designed the following attack scenario to test WACA against the statistical attacks.

In our attack, we used both Typing Task-1 and Typing Task-2 dataset as input. Each participant was chosen as a victim iteratively, and the other participants' samples were used to generate forged data samples. Then, the forged samples were used to attack the victim. For this, a histogram was created for each feature of all the participants in the dataset except the victim. The forged samples were generated as in Algorithm 1.Overall, we created three different statistical attackers with three different capabilities (bin sizes in the histogram). Statistical Attacker-1, Statistical Attacker-2, and Statistical Attacker-3. Before running the algorithm for attacking WACA, we first calculated the EER for each user without adding any forged data. Similar to the imitation attacks, the attacker acceptance rate in zero-effort attack corresponds to the average EER. We conducted experiments without attack under varying sample and bin sizes. The results are shown in Figure 3.11.

In Figure 3.11, we can see that bin number 50 has the most successful result on attacking victims. This is because if we increase the bin number and keep the bins with the highest number of occurrences constant, the width of the bins will narrow; so, the range of the forged data will be confined to a very small range. On the other

Algorithm 1 Calculation of EER for a statistical attacker.

Require: $Samples_{MXN}[]$: M is # of samples and N is # of features
Require: $outNumber$: # of generated forged samples
Require: $binNumber$: # of bins
Require: $topBins$: # of top bins used to generate forge samples
Ensure: new_eer: # new error rate against the attack

for each $user$ **do**
  $victim \leftarrow user$;
  $victimSamples \leftarrow$ getSamples($victim$);
  $attackSamples \leftarrow$ getSamples($\sim victim$);
  $combin[] \leftarrow$ ComGen($N, outNumber, topBins$);
  for each forgeid $s_i \in attackSamples$ **do**
    for each feature $f_j \in attackSamples$ **do**
      $[freq, edges] \leftarrow$ histGen();
      [$\sim$,$index$] $\leftarrow$ sortBins($freq$);
      $index(topBins + 1 : end) \leftarrow []$;
      $m \leftarrow edges(index(combin[f_j, forgeid]))$;
      $forgedSamples \leftarrow$ random($[m, m + 1]$);
    end for
  end for
  $victimSamples \leftarrow$ addSamples($forgedSamples$);
  $D \leftarrow$ calculateDissMatrix($TestingSamples$);
  $eer\_for\_victim \leftarrow$ calculateEER(D);
end for
$new\_eer \leftarrow$ mean(eer_for_victim);
return new_eer

hand, if we decrease the bin number significantly, the less frequently occurred bins will also be included in the sample generation range, which will reduce the success rate of the attacks. Finally, we note that in the attack scenario, we choose each user in our dataset as a victim in an iterative way. *These results show that despite the small increase compared to zero-effort, the attacker does not have a chance to defeat WACA using the systematically generated fake data due to its high dimensional feature vector in WACA's design.*

As a summary, neither the imitation nor statistical attacks put WACA in danger as their success rates are very close to zero-effort attacks.

Table 3.7: Resource consumption of the smartwatches used in the experiments: *LG Watch R* and *Samsung Gear Live*.

		LG G Watch R	**Samsung Gear Live**
CPU (no WACA)		4.5%	4.5%
CPU (w/WACA)		7.5%	16.8%
Memory (no WACA)		4.5 MB	4.5 MB
Memory (w/WACA)		15.2 MB	13.8 MB
Battery	10Hz	1.1%	1.2%
	30Hz	1.6%	0.3%
	100Hz	2.1%	2.4%
Data Size	10Hz	0.3 MB	0.3 MB
	30Hz	0.6 MB	0.9 MB
	100Hz	4.1 MB	6.5 MB

3.5.3 Resource Consumption

In WACA, a smartwatch, a computer, and an authentication server work together. In this subsection, we only analyze the resource consumption of relatively constrained smartwatches. It is worth noting that we monitored the consumption of our application while it was running continuously; however, in WACA, the data collection app does not have to be running continuously. It can happen periodically or on-demand because the data collection runs only when the smartwatch is notified by the computer that the user is typing on. We analyze the performance of both LG G Watch R and Samsung Gear Live smartwatches used in the experiments. Both smartwatches have Cortex-A7 at 1.2GHz and 512MB RAM, but Samsung uses 300mAh battery, while LG is using 410mAh battery. The results are presented in Table 3.7.

In all the experiments, we both monitored the memory and CPU resource utilization of the smartwatches in the default mode (i.e., not actively running any app - no WACA) and while the app was running (w/WACA). In the default mode, both smartwatches used almost 4.5MB memory and 4.5% CPU their consumption while

the app was running, as shown in Table 3.7. As compared to the default memory usage (no WACA), the memory consumption in the smartwatch in WACA is increasing, but it is still at an acceptable rate.

In addition to memory and CPU consumption, we also analyzed the power consumption and data size while running our app for 10 minutes. We excluded the power consumption of the screen as the screen can be turned off or the smartwatch can be in the ambient mode during the data collection of WACA. We see that the power consumption of the app scales by the sampling frequency. However, when we decrease the sampling rate, the time needed to collect a certain amount of data will also increase. Hence, the optimum sampling rate should be tuned according to the desired security policy.

3.6 Discussion

Security Policy Implementation Considerations. WACA works by checking if the current user's profile matched the profile of the logged-in user. When an unauthorized access attempt is detected, the reaction depends on the previously decided security policy. Depending on the security policy, when an attacker is detected, the screen can be locked, and the user can be challenged to re-login; the management and security teams can be alerted in real-time, or a notification e-mail can be sent to the registered e-mail of the logged-in user, and so on. Moreover, we showed that WACA could differentiate an insider from an outsider accurately. In suspicious cases, the administrator can do further investigation to detect the insider, and as we noted earlier, the insider detection is possible in WACA. We also note that even if WACA catches an insider attacker, WACA can not know if the attacker has the full key, which is out of scope this work. Therefore, even if the system is

logged-out, an insider can log-in again if it has the full key. Therefore, resetting the initial authentication factor should be considered in the security policy in this case. Finally, the server can also log the failed attempts to prevent attacks aiming to drain the smartwatch's battery.

Moreover, if WACA is deployed in an environment where typing is not required much, the actions that will be performed when the user is not typing should be defined in the security policy. A straightforward solution to this problem can be reducing the system to default security, i.e., locking out the user if there is inactivity for a certain duration.

WACA captures the typing patterns of the user only from one wrist. If the wrist wearable is on the left hand, for example, the typing pattern for the words "and" and "aod" would be the same. This can be perhaps exploited by the attacker by using the letters on the right. However, this would be a remote possibility. In WACA, we wanted to test our proposed method in a more realistic scenario assuming a user will wear a wearable on both hands might be an unrealistic assumption. However, in highly extreme cases, i.e., highly critical environments, two smartwatches can be utilized to collect data from two hands of the users. This will prevent against this type of attack. This should be considered while deploying WACA in a real-world application.

Privacy. In WACA, the computer and the wearable are the devices that belong to the user or belong to the same authentication realm and thus are trusted. The only device that may threaten privacy is the AS. As for the security of the data at rest at the server, the existing industry standards such as AES, RSA, ECC, RC4, can be employed to establish the security of the data in these cases. In WACA, after collecting the raw sensor data from the smartwatch, either the raw sensor data can be transmitted to the AS, or the features can be computed on the smartwatch and the

feature vector can be transmitted.No data is stored on the watch and as noted in the Assumptions Section (Section 3.3), this channel is secured with existing methods. If the raw sensor data is sent to the AS, the AS may try to infer the user's typed characters from the raw sensor data. The more secure way would be to compute the features on the smartwatch and to keep the feature vectors of the profiles of the users in the AS. In that case, the transmitted feature vector has only the mean of the values of the multi-dimensional sensor data and thus inferring the typed characters would not be possible at the AS.

3.7 Conclusion

In this chapter, we introduced a novel Wearable-Assisted Continuous Authentication (WACA) utilizing the sensory data from the built-in motion sensors available on smartwatches. WACA is a practical and usable wearable-assisted continuous authentication system that combines the functionality of wearables and usability of continuous authentication. Particularly, WACA decreases the vulnerable time window of a continuous authentication system to as low as 20 seconds, prevents the privilege abuse and insider attacks and also allows the insider threat identification. We evaluated the efficacy and robustness of WACA with real data from real experiments. The results showed that WACA could achieve 1% EER for 30 seconds or $2-3\%$ EER for 20 seconds of data collection time and error rates are as low as 1% with almost a perfect (99.2%) insider threat identification rate.

CHAPTER 4

PACA: A LIGHTWEIGHT PRIVACY-AWARE CONTINUOUS AUTHENTICATION PROTOCOL

4.1 Introduction

Efforts to improve the security of the authentication services have historically progressed from what-you-know (i.e., passwords) to what-you-have (i.e., tokens), then to what-you-are (i.e., biometrics) as attacks have increased in sophistication and become widespread [TGG13, SSY12]. While the deployment of biometric authentication systems increases the usability of the authentication systems, the plethora of cyber-attacks demands more user information from biometrics, which introduces additional security and privacy challenges in the authentication systems. In this landscape, another challenge is due to the nature of one-time authentication, which verifies users only at the initial login session regardless of being single- or multi-factor. This is a serious security risk as once the attacker bypasses the initial authentication, it will have a forever access or if the user leaves the system intentionally/unintentionally unlocked, anyone such as an insider or a strong outsider adversary [ALUK19], who has physical access to the system will have access to the system without the actual user notification. Therefore, the user should be continuously monitored and re-authenticated. In the literature, several solutions such as time-out or token (or even RFID) based solutions are proposed to address these issues in the authentication systems [KSC10]. Indeed, biometric-based systems are considered to be ideal and usable for such cases as they can not be easily misplaced unlike tokens, or forgotten unlike passwords, or easily forged by an imposter.

In this chapter, we tackled these challenges and constructed a novel lightweight privacy-aware continuous authentication protocol, called PACA. In our protocol, we

utilize the password-authenticated key exchange (PAKE) primitive, which we adapt for the biometric continuous authentication. This provides basic security requirements of our protocol, such as a secure channel between the user and server, mutual authentication, forward secrecy, as well as the resistance against pre-computation attacks. In our design for an actual privacy-aware continuous authentication method, we utilize a secure and noise-tolerant template generation and matching technique called NTT-Sec-$\mathbb{R}$, and combine it with a wearable-assisted continuous authentication method called WACA. NTT-Sec-$\mathbb{R}$ irreversibly transforms the feature vectors, but still allows us to distinguish genuine pairs from imposter pairs. The novel security enhancements proposed in this chapter are applicable to a wide range of biometric authentication mechanisms when feature vectors are represented as fixed-length real-valued vectors. One of the important applications of such systems is sensor-based keystroke dynamics, which could be used in the authentication of computer [AAUA18, AAUA20], smartphone [LL17], and wearable [FBM+13] users.

4.2 System and Security Model

In this section, in order to understand the threat model, we present the basic security requirements of the protocol. Then, we also present the system components and parameters of the protocol, and the assumptions made.

4.2.1 Security Requirements of the Protocol

The security requirements of our continuous authentication protocol are as follows:
Secure channel. The communication between the user and the authentication server should be secure against any eavesdropping or interception.

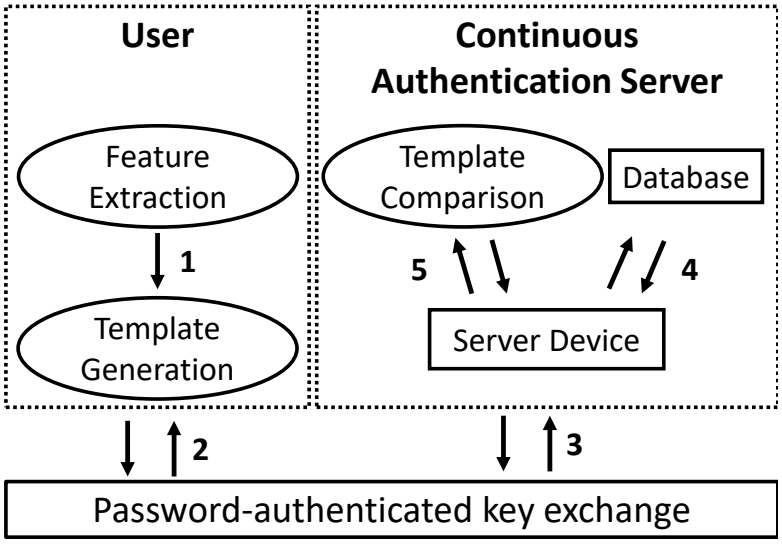


Figure 4.1: System components of our proposed continuous authentication protocol, PACA. The detailed definitions are given in Section 4.2.2.

Mutual authentication. The proposed authentication protocol should support the mutual authentication between the user and the authentication server.

Forward secrecy. This protects past sessions and session keys even after the long term secret keys of the parties, future sessions, and future sessions keys are compromised.

Resistance against known attacks. In addition to the security requirements above, our proposed protocol should be resistant to the main threats known against privacy-aware biometrics-based authentication protocols [RCB01, PM17]. We split these threats into four categories: 1) Password recovery attack, 2) Impersonation attacks, 3) Session intervene attacks, and 4) Biometric recovery attacks. The more details about the attacks and their analysis for PACA are explained in Section 4.4.

Table 4.1: The symbols used throughout the chapter.

Symbol	**Description**
User_i	i'th user
Id_i	the identity of user i
Pwd_i	i'th user's password
$\mathsf{Bio}_{i,t}$	Biometrics of the user i at time t
UsrDev_i	i'th user's device
$\mathsf{UsrDev}_{i,k}$	the k'th device of the i'th user
Ext	Feature Extraction
$\mathsf{f}_{i,t}$	feature vector of user i at time t
$\mathsf{t}_{i,t}$	biometric template of the user i at time t
$\mathsf{TempGen}$	Template Generation algorithm
$\mathsf{TempComp}$	Template Comparison algorithm
s	similarity score
DB_i	Information of the user i at database
SerDev_k	k'th Server Device
CAS	Continuous Authentication Server
T_i	threshold value of the user i
Time	constant time
$\mathsf{NumQuer}$	number of query
$\mathsf{NumMatch}$	number of match query
$\mathsf{MinQuer}$	constant number of minimum query
$\mathsf{MinMatch}$	constant number of match query
K_i	shared session key between user i and server

4.2.2 System Components and Parameters

Our continuous authentication protocol consists of three main components: 1) Password-authenticated Key Exchange (PAKE), 2) User, and 3) Continuous Authentication Server (CAS). Figure 4.1 illustrates the interactions between the main components of the protocol. (1) User extracts the elements of the feature vector via a feature extraction algorithm and transforms the feature vector to its corresponding template through template generation function. (2,3) After that, the template is transmitted to (CAS) through the secure channel provided by the PAKE protocol, which also provides the mutual authentication between the user and the server and forward secrecy properties. (4,5) After receiving the user's template, the AS also extracts the user's information from the database and compares it with the incoming the template of the user via the template comparison function, which return a similarity score. The server device decides the authentication result by comparing this similarity score with a predetermined threshold value. In the end, the final authentication result is returned to the user side via the underlying PAKE method. Before explaining the details of our protocol and its components, we also give the description of the symbols used in the protocol in Table 4.1. In the following sub-sections, we explain the details of the components.

1. Password-authenticated Key Exchange (PAKE). Our authentication protocol utilizes a (strong) password-authenticated key exchange (PAKE) method with some strong security properties. OPAQUE [Kra18] and SRP-6 [Wu07] are examples of such a protocol that satisfy the following features:

1. Public key infrastructure (PKI) is not needed because PAKE protocols reduce the security of the system to only the user's password without relying on an outside keying material such as public keys [Kra18]. This is a big advantage

from the efficiency point of view because TLS, any certification authority, verification of certificates, long term private keys, etc. are not required. Another advantage from a security point of view is that any potential failure in PKI is not an issue anymore (such as invalid certificates [Osb], stolen private keys [Kas], etc.).

2. A user and server can mutually authenticate each other.
3. The server stores only a cryptographic transformation of the user's password. The password is never sent in clear, and the server does not learn the password of the user.
4. Pre-computation attacks [Kra18] are not applicable. Such attacks do apply to some password-based protocols if salts are not used, or they are sent in clear from a server to a user, but they do not apply to the specific PAKE instantiations as specified above.
5. To recover the password of a selected user, the adversary can only mount an exhaustive offline dictionary attack after compromising user data on the server. This attack can not be avoided but it is computationally not feasible as only the cryptographic transformation of the password is stored on the server.

2. User. Throughout this chapter, $\mathsf{User}_i = (\mathsf{Pwd}_i, \mathsf{Bio}_{i,t}, \mathsf{UsrDev}_i)$ denotes a user indexed with i, her password, her biometric data indexed with t, indicating different measurements of the biometric data of a user, and her device used for collecting the biometric data. A user has access to feature extraction and template generation algorithms:

- $\mathsf{f}_{i,t} \leftarrow \mathsf{Ext}(\mathsf{Bio}_{i,t})$: denotes a feature extraction algorithm. The parameter t considers that different measurements of the same biometric data may result in different feature vectors, i.e., in general, $\mathsf{f}_{i,t_1} \neq \mathsf{f}_{i,t_2}$ for $t_1 \neq t_2$. We assume that

the feature extraction always runs on a user device such as user's computer. Biometric data and extracted features are stored only temporarily on this device, and they are deleted after communicated to another device or entity in our protocol.

- $\mathsf{t}_{i,t} \leftarrow \mathsf{TempGen}(\mathsf{f}_{i,t})$ refers to the one-way transformation of the feature vector into a more secure template, while allowing comparison on the transformed version as well as providing irreversibility and indistinguishability [JNN08] and it corresponds to the traditional hash in password-based systems. Similar to the feature extraction, the operation can be performed on the user device.

3. Continuous Authentication Server (CAS). CAS denotes an authentication server that validates or invalidates an enrollment or an authentication query initiated by a user (or by an adversary who is trying to impersonate a user). CAS indicates the validity or invalidity of a query by an output of 1 or 0, respectively. CAS has access to a template comparison algorithm and manages a database and server devices that the users interact with:

- $s \leftarrow \mathsf{TempComp}(\mathsf{t}_{i,t_1}, \mathsf{t}_{i,t_2})$: denotes a template comparison algorithm that takes two templates t_{i,t_1} and t_{j,t_2} captured at two different times as input, and outputs a similarity score, $s \in \mathbb{R}$ quantifying the similarity of the underlying biometric data pair $(\mathsf{Bio}_{i,t_1}, \mathsf{Bio}_{j,t_2})$.
- DB denotes a database that stores information about the users who are enrolled with CAS. For convenience, DB_i denotes the information about User_i. This information is comprised of the user's identity, a cryptographic transformation of her password, and her biometric template along with her matching thresholds. The full definition is given in Section 4.3.

- SerDev denotes a server device which the user is trying to log in, such as desktop or laptop computers of the user. CAS may manage more than one server device. In this case, the k'th device of CAS is denoted SerDev_k.

4.2.3 Assumptions

We list our assumptions regarding the components of the system as follows:

- In general, $\mathsf{Bio}_{i,t_1} \neq \mathsf{Bio}_{i,t_2}$. We assume that each user has at least one device that can extract biometric information of that user. In one of the applications described in this chapter, we equip users with a smartwatch that extracts the typing behavior of its user. As they are commodity devices, they are easily accessible to many users.
- The user-specific values and devices are distinct and not shared among other users in a regular run of our protocol. More formally, we assume $\mathsf{User}_i \neq \mathsf{User}_j$, $\mathsf{Bio}_{i,t_1} \neq \mathsf{Bio}_{j,t_2}$, and $\mathsf{UsrDev}_i \neq \mathsf{UsrDev}_j$ for $i \neq j$. It also worth noting that a malicious user may control a user device, or a user password may be stolen, but we treat these scenarios as attack scenarios and analyze them in detail to show how our work is robust against these attacks in Section 4.4.
- A user may have more than one device. In this case, the k'th device of the i'th user is denoted $\mathsf{UsrDev}_{i,k}$.
- We assume the adversary is a computationally bounded, active adversary who tries to achieve some *adversarial goals* in Section 4.4.2 to break the security and/or privacy of the users or the system.

4.3 Continuous Authentication Protocol

In this section, we describe our novel continuous authentication protocol, which includes both the password authentication phase and continuous authentication using the biometrics of the user. Particularly, in our continuous authentication protocol, a user, User_i, is involved in two phases. The enrollment phase is implemented only once and can be implemented at any time before the authentication phase. The authentication phase consists of two parts. The initialization part is implemented only one time, but it has to be implemented every time the user wants to log in. It is required to establish a secure and authentic channel between the user and CAS. Finally, the authentication phase is performed periodically, in which the period depends on the underlying biometrics-based authentication mechanism. The quicker and more accurate systems are better for security. We explain the details of enrollment and authentication phases below and illustrate them in Figure 4.2, 4.3, and 4.4, respectively.

Enrollment Phase

In the enrollment phase, a secure template is generated from a biometric trait and stored in CAS. The following are the steps of the enrollment phase:

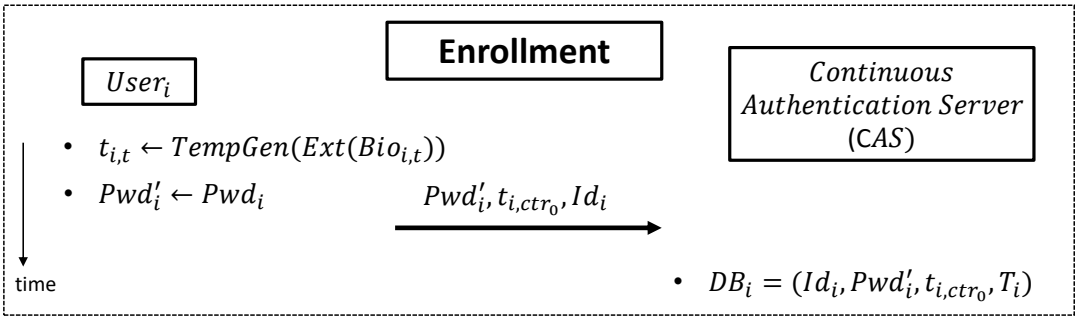


Figure 4.2: The enrollment phase of our proposed continuous authentication protocol.

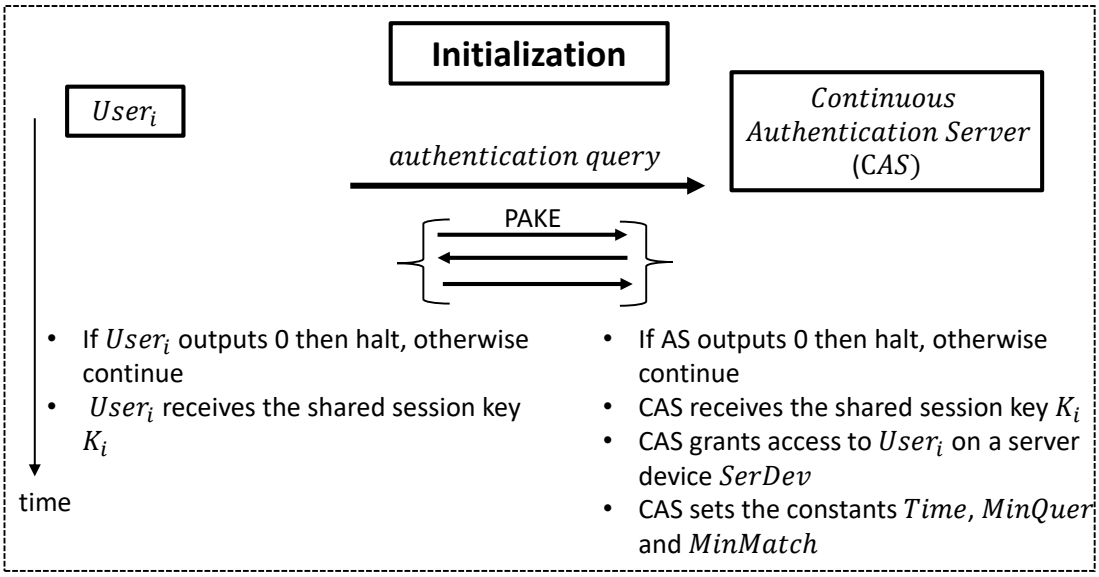


Figure 4.3: The initialization phase of our continuous authentication protocol.

1. User_i computes $\mathsf{t}_{i,0} = \mathsf{TempGen}(\mathsf{Ext}(\mathsf{Bio}_{i,0}))$.

2. User_i registers a cryptographic transformation of her password Pwd_i, her template $\mathsf{t}_{i,0}$ along with her identity, Id_i by following the underlying PAKE protocol. Note that during enrollment, CAS may want to authenticate User_i and her information through her physical presence. Moreover, CAS may store additional information about the user in $\mathsf{DB}i$ such as her matching threshold value T_i. The final stored information for each user is shown as $DB_i = (Id_i, Pwd_i^{'}, t_{i,ctr_0}, T_i)$.

Authentication phase

In the authentication phase, a user's biometric template is periodically verified by the authentication server after a secure and authentic channel is initialized based on PAKE.

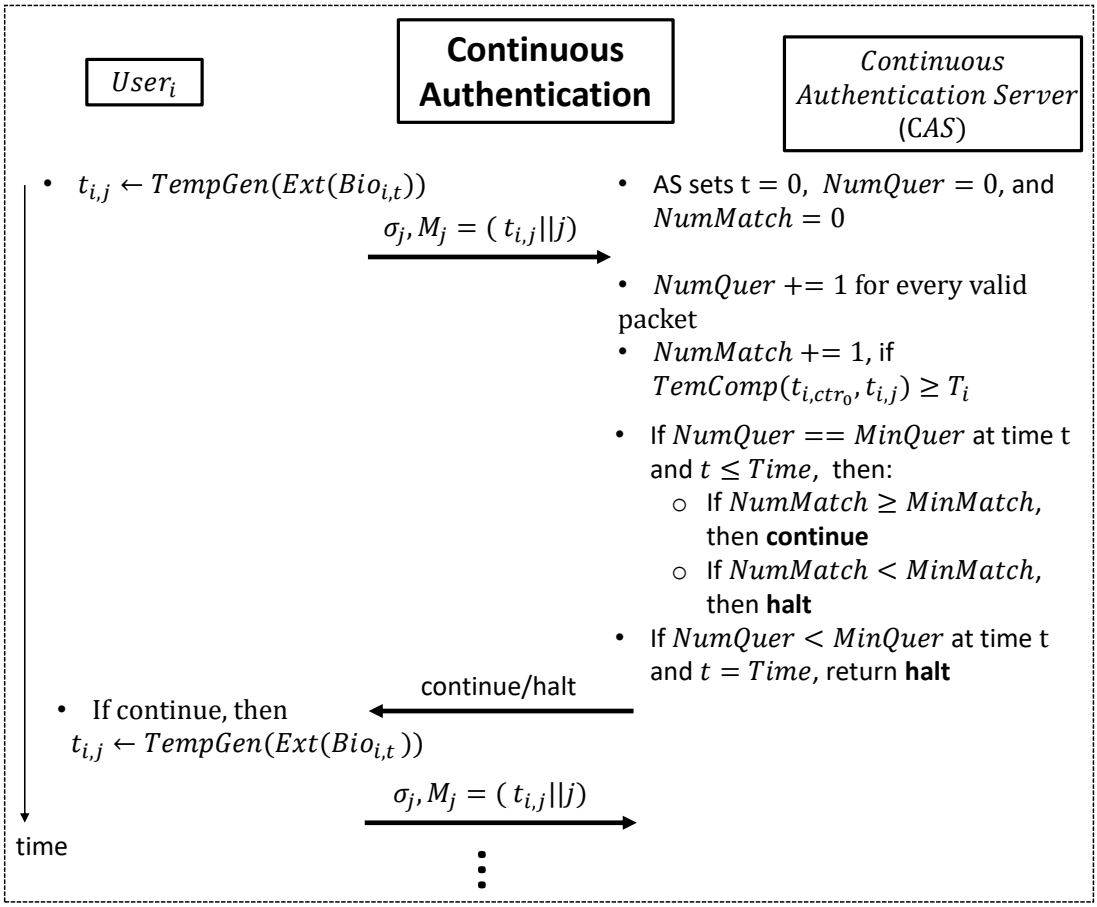


Figure 4.4: The continuous authentication phase of our continuous authentication protocol.

Initialization: User_i and CAS execute the underlying PAKE protocol to authenticate each other mutually and generate a session key K_i, which then establishes a secure and authentic channel. If the mutual authentication is successful, then

1. CAS sets the constants Time, MinQuer, and MinMatch.
2. CAS initiates a continuous session for User_i (granting access to User_i on the server device, SerDev such as her computer.

Continuous Authentication: In this phase, a user generates her templates and sends them to CAS through the secure and authentic channel established in the

initialization phase. Assuming the mutual authentication in the initialization phase is successful, the following steps are executed and are shown in Figure 4.4:

1. CAS sets its system time $t = 0$, NumQuer $= 0$, NumMatch $= 0$, and keeps track of the system time t.
2. User_i continuously computes her secure biometric templates $\mathsf{t}_{i,j}$ for $j = 1, 2, ...,$ and sends packages of the form $(\mathsf{t}_{i,j}||j)$[1] to CAS using the confidential and authentic channel established through the shared session key K_i. For each valid package that CAS receives, CAS increments NumQuer by one, and for each $\mathsf{t}_{i,j}$ with $\mathsf{TempComp}(\mathsf{t}_{i,0}, \mathsf{t}_{i,j}) \geq T_i$, CAS increments NumMatch by one.
3. If NumQuer == MinQuer at time t and $t \leq$ Time, then
   - If NumMatch $\geq$ MinMatch, then return (1), indicating the continuity of the session.
   - If NumMatch $<$ MinMatch, then CAS terminates the session and the protocol halts.
4. If NumQuer $<$ MinQuer in time t and $t =$ Time, then CAS terminates the session and the protocol halts.
5. If AS returns 1, the protocol continues from step (2), where the user computes her new biometric template periodically.

4.4 Security & Privacy Analysis

In this section, we analyze the security requirements of the protocol and we show how PACA is secure and robust against eight well-known malicious in against the privacy-aware biometrics-based authentication protocols.

[1]Here, appending j plays the role of a counter to prevent some obvious replay attacks.

4.4.1 Analysis of Security Requirements of the Protocol

In our protocol, the basic requirements are provided through the PAKE protocol executed during the initialization phase. A particular example of this PAKE protocol could be OPAQUE [Kra18], which is a strong asymmetric PAKE protocol providing security against pre-computation attacks. In [Kra18], the different versions of the PAKE protocol with different security features are proposed. We specifically consider the version called "the generic OPRF+AKE construction". This version is based on Oblivious Pseudo-Random Functions (OPRF) and Authenticated Key Exchange (AKE). The full description of the OPAQUE protocol providing the full forward secrecy and mutual authentication and generating the shared session key between the parties with only three messages exchanged between the user and server can be found in Figure 12 of [Kra18]. In the following sub-sections, we will show how this specific version of OPAQUE provides the secure channel and satisfies the perfect forward secrecy and mutual authentication in PACA.

Secure channel. The primary use case of PAKE protocols [Kra18, Wu07] is that the user does not need to rely on any outside key other than his password. The shared session key K_i is generated from the user's low-entropy password during the execution of the OPAQUE protocol. Then, throughout the entire session, the secure channel is provided through this shared session key. This provides a confidential and authentic communication channel and protects the current session against eavesdropping and man-in-the-middle attacks. Particularly, in PACA, this shared session key K_i is generated during the initialization phase of our protocol after executing the PAKE protocol between the User_i and CAS.

Mutual authentication. TLS provides only server-side authentication through the certificates, and the password provides the authentication for the user side. However, PAKE protocols achieve mutual authentication without the need for TLS or

any PKI infrastructure. For example, the OPAQUE [Kra18] protocol uses HMQV [Kra05] as a base AKE protocol to provide mutual authentication. HMQV extends the computational Diffie-Hellman (DH) key exchange with Exponential Challenge-Response (XCR) signatures. These signatures are proven to be unforgeable in [Kra05], and they are computed directly on the identity of parties. The ability of parties to provide the signature shows the proof that exchange is carried by the claimed parties and since the messages on which the signatures are computed are directly the identity of the user and server, it proves that the key they computed is uniquely associated with the correct identities (i.e., mutually authenticated).

Forward secrecy. A protocol is said to have the forward secrecy [Kra05], if the session keys of previous runs can not be recovered by the attacker after the keys are established, used, and deleted from the memory even after the compromise of long-term keys. Similar to the mutual authentication, the forward secrecy in OPAQUE is provided by the HMQV protocol. The perfect forward secrecy can be achieved if one of the user messages depends on the user's private key. This is achieved by letting DH values by both parties for the session.

4.4.2 Attack Resistance.

In this section, we first present several known attacks against privacy-aware biometrics-based authentication systems [RCB01, PM17] and we also analyze if our protocol is robust against these attacks. More specifically, we present the adversarial goals (AGs); hence, the adversarial model, attack strategies, their analysis, and the countermeasures our protocol provides against the attacks.

In our proposed continuous authentication protocol, the underlying PAKE method provides the basic security features given in 4.4.1; however, it does not guarantee

that the user data will not be revealed to the CAS. This violates the privacy of the user against the server, and to protect the user privacy, in the next section, we propose the use of NTT-Sec-$\mathbb{R}$ for realizing the function $TempGen()$ which is used as a black-box function. The proposed template generation function and its benefits are separately evaluated in Section 4.5.3 and 4.5.4.

AG-1: Password recovery attacks. In this attack, the adversary's goal is to recover the password of a user. An adversary who is capable of actively controlling sessions or compromising a server can achieve this goal only if he succeeds in an exhaustive offline dictionary attack.

Analysis of the attack: Such dictionary attacks cannot be prevented perpetually, but strong passwords would increase the run time of the attack. More precisely, assuming that users choose their passwords uniformly at random from a password space $PassSpace$, then the attack would require $|PassSpace|$ trials. In OPAQUE [Kra18], it has been shown that the cost can be increased by increasing the number of iterations in the hashing operation (i.e., replacing H with H^n in the full protocol). Moreover, we quantify the cost of this attack in Section 4.5.3 particularly for our implementation.

AG-2: Impersonating the user (or the server) at the initial login phase. In this attack, the adversary's goal is to initiate a session and generate a valid session key on behalf of a user, or impersonate a server to a user. These may be achieved by the following two attacks:

1. *Replay attack:* Adversary may try to replay messages from the previous runs of the protocol between the user and the server.

   Analysis of the attack: Such an attempt would fail thanks to the fresh and randomized session keys generated per session with PAKE.

2. *Password recovery attack through the user impersonation:* If an adversary achieves AG-1 above, she can clearly achieve AG-2.

   Analysis of the attack: As in the analysis of AG-1, an offline dictionary attack cannot be prevented perpetually, but strong passwords would increase the run time of the attack.

AG-3: Session intervene. In this attack, the adversary's goal is to intervene an active session of a user, and to stay undetected as long as possible while behaving maliciously (e.g. interacting with SerDev and impersonating User_i). In the following, we assume that User_i initiates a session with CAS and they both computes the shared session key K_i. We also assume that UsrDev_i stores a copy of K_i. The adversary can achieve AG-3 as follows:

1. *Package delay attack:* In this first attack scenario, an adversary eavesdrops the communication between User_i and CAS, and interrupts a sequence of legitimate packages (including templates and their counters $(\mathsf{t}_{i,j}||j)$, $j = 1, 2, ..., k$, encrypted under K_i) going from User_i to CAS. Now, suppose that User_i is out for lunch after sending her last package and leaves the server device SerDev unlocked [2]. Then, the adversary forwards the packages she already collected to CAS while behaving maliciously on SerDev. Receiving sufficiently many legitimate packages (e.g., at least $\mathsf{MinQuer}$ in time Time), CAS cannot distinguish the adversary from User_i, and therefore, the adversary stays undetected and achieves her goal.

   Analysis of the attack: This attack can be detected easily if the server device SerDev (i.e., the user's computer) acknowledges CAS immediately after an ac-

[2]This is a reasonable user behavior in a continuous biometric authentication scheme as such systems assure that adversaries can successfully be detected when they try to impersonate legitimate users.

tion is received on SerDev because CAS can detect whether or not the adversary is interrupting and delaying legitimate packages while User_i is legitimately interacting with SerDev.

2. *Zero-effort and mimicking attacks:* In this second attack scenario, we assume that User_i is out for lunch after initiating a session with CAS and establishing K_i. We also assume that User_i leaves her device UsrDev_i behind, and leaves the server device SerDev unlocked. Now, the adversary captures UsrDev_i, and presents her own biometric measurements to CAS (zero-effort attack), or tries to reproduce the physiological or behavioral characteristics of User_i (i.e., imitation attack).

   Analysis of the attack: The success rate of this attack would be strongly correlated to the FAR of the system, and the uniqueness of the underlying biometric trait. See Section 4.6.1 for the FAR rates of our protocol and see [AAUA18] for the robustness of WACA against the mimicking attacks.

3. *Session key reveal attacks:* The third attack scenario is similar to the second one, but we consider a more powerful adversary. We assume that User_i is out for lunch after initiating a session with CAS, establishing K_i, and sending some packages to CAS (including templates and their counters $(\mathsf{t}_{i,j}|j)$, $j = 1, 2, ..., k$, encrypted under K_i). We also assume that User_i leaves her device UsrDev_i behind and the server device SerDev unlocked. In addition, we assume that the adversary recovers the session key K_i from UsrDev_i. Having captured some of the previously exchanged packages, the adversary can now recover $\mathsf{t}_{i,j}$ using the knowledge of the key K_i. Next, the adversary can form legitimate packages with the appropriate counters and impersonate User_i during that current session.

   Analysis of the attack: We do not consider this cascaded third attack to be a

practical attack because it requires an attacker to extract the session key from the user device in a relatively short amount of time (i.e., before a new session starts, and a new session key is generated).

4. *Input device replacement attack:* The fourth attack that we consider is rather a physical attack. After a session is initiated between User_i and CAS, the attacker replaces the legitimate input device of SerDev (e.g., a keyboard or a smartwatch) by her own malicious input device. User_i may still think that she is interacting with SerDev through the legitimate input device, and she may keep sending valid packages to CAS. In the meantime, SerDev receives the adversary's malicious input through the legitimate input device, and CAS keeps the session live based on the legitimate packages it receives from User_i.
   Analysis of the attack: This attack may work in theory, but it may be challenging to deceive User_i that she is interacting with SerDev through the legitimate input device while indeed she is providing her input through a malicious input device. Therefore, we do not consider this fourth attack to be practical.

AG-4: Recovering biometrics. In this attack, the adversary tries to recover the biometric information of a user.

1. *Server compromise attack:* The adversary may be able to capture some of the biometric templates $\mathsf{t}_{i,j}$ of User_i by compromising the server database, or by capturing some of the packages from a previous session and the session key K_i of that specific session. Then, the adversary can try to reverse the templates back to the biometric information.
   Analysis of the attack: The success rate of the adversary would depend on the difficulty of reversing templates for the given template generation algorithm. Therefore, this attack does not seem to be feasible if an irreversible and in-

distinguishable template generation algorithm TempGen is deployed. We show the proof of this in Section 4.5.3 more formally for our implementation.

4.4.3 Further notes on the attacks, their limitations, and justification for multi-factors

1. If an adversary captures the secure template of a user $\mathsf{t}_{i,t}$ and a particular session key K_i, but not the password (Pwd_i), then the adversary can impersonate User_i only for that session, because in the next session a fresh session key is generated by the underlying PAKE method and without the knowledge of the new session key, the adversary cannot produce secure templates and legitimate packages to send to CAS.

2. If an adversary captures the password of a user (Pwd_i), but not the template of a user ($\mathsf{t}_{i,t}$), then she can initiate a session, but her chances for avoiding detection are limited by the success probability of the mimicking attack or the zero-effort attack (or FAR attack).

3. Another interesting scenario is when an adversary steals the template ($\mathsf{t}_{i,t}$) and the password of a user (Pwd_i). In this case, the adversary can impersonate the user forever unless the user becomes aware and resets the password and re-enrolls in the system. Therefore, one may consider equipping the user device UsrDev with a public key private key pair and involve UsrDev in the session key generation at the beginning of the protocol. For example, a UsrDev signature together with a timestamp can independently be used to confirm that User_i is initiating a session with CAS. In this scenario, the adversary would need the template, password, and also the user device to impersonate User_i. If the user cannot locate her device at any time, she may acknowledge CAS, reset her

password, and re-enroll. Moreover, even if the two-way TLS is affordable in the system, a password would still provide an extra barrier for the adversary in case she captures the user's biometric information and device.

We note that even if UsrDev has a long-term private key and it becomes a part of the protocol in the session key generation, one would still need a biometric factor because otherwise, an adversary would successfully impersonate User_i in lunchtime type attacks by temporarily accessing UsrDev.

The use of a password is also important in our case because PAKE eliminates the need for TLS for mutually authenticating User_i and CAS. Moreover, passwords provide extra protection against an adversary who already captured the user's biometric information and device. In summary, combining all three factors what you know (password), what you have (device), and who you are (biometrics) would provide the most comprehensive secure and privacy-aware setup.

4.5 Full Implementation

In this section, we describe our hybrid (password and keystroke dynamics), continuous, and privacy-preserving biometric authentication system, which is illustrated in Figure 4.5. Both for performance evaluation purposes and as a walk-through proof-of-concept case study, we fully deployed Wearable-Assisted Continuous Authentication framework called WACA [AAUA18, AAUA20] as an example continuous biometric authentication system in this study. On the other hand, we utilized NTT-Sec-$\mathbb{R}$ by improving NTT-Sec [KC16] as a template generation and comparison algorithm, to address the aforementioned privacy issues in continuous biometric authentication settings. We applied a feature selection algorithm and selected the top 15 features from WACA. These features included the time and frequency domain

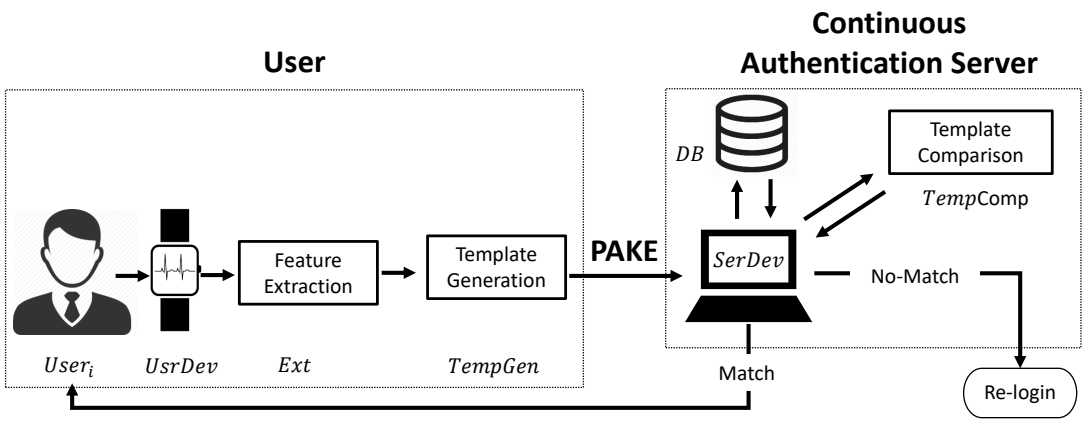


Figure 4.5: The architecture of our concrete continuous authentication system used as a case study.

statistics (e.g., mean, median, entropy) of the raw sensor values for a certain period. Moreover, we also improved NTT-Sec to handle the real-valued feature vectors while moderately preserving accuracy. As both of these example methodologies are generic, they can be applied to any biometric-based authentication algorithm, which has a real-valued feature vector. For example, this could be sensor-based keystroke dynamics, which can be used for the authentication of computer [AAUA18], smartphone [LL17], or wearable [FBM+13] users.

In the following sections, we first explain WACA's design and our modifications. Then, we explain NTT-Sec-$\mathbb{R}$ and show how to use it in the settings of continuous authentication. Finally, we also show how NTT-Sec-$\mathbb{R}$ provides security properties such as irreversibility and indistinguishability while protecting the user templates from the third parties.

4.5.1 Testing with a sample continuous authentication system

WACA is based on the idea of sensor-based keystroke dynamics, where the authentication data is collected and extracted from the accelerometer and gyroscope sensors of a wearable device (e.g., smartwatch). In our protocol, the wearable device corresponds to the parameter UsrDev, while the user's computer corresponds to SerDev parameter. Since the continuous authentication is based on the sensor data traveling from the smartwatch to the computer, the security and privacy of the data become very important. $\mathsf{NTT\text{-}Sec\text{-}}\mathbb{R}$ ensures the security and privacy of the data and also makes authentication possible over the noisy nature of the context. In this subsection, before presenting the evaluation study, we first provide an overview of WACA.

In WACA, the raw motion sensor data of the smartwatch is acquired through an app installed on the watch, and the sensor data is transmitted to the computer. In our concrete system, we encrypt this data with the shared session key K_i generated by the underlying PAKE protocol.

In the enrollment phase of WACA, the created feature vector and the user's id are stored together as a profile in the AS, which is located in a trusted place. Then, in the authentication phase of WACA, the decision of the authentication is made by the decision module by computing a similarity score between the feature vector dispatched from the AS and the incoming feature vector of the current user. In the end, the decision module makes a binary authentication decision (match/no-match) by comparing the similarity score with a predetermined threshold value. If the decision is a no-match, then the user's access to computing terminal is suspended, and the user is required to re-login using the initial authentication method (e.g., pass-

word, or 2FA [LUB14b, ALB+19]). If the decision is a match, then the user's access is maintained without interrupting the user. This process is repeated periodically with a predetermined period.

Moreover, in WACA, the raw accelerometer data is represented in the format of $\vec{acc} =< t_a, x_a, y_a, z_a >$ and gyroscope data is represented as $\vec{gyro} =< t_g, x_g, y_g, z_g >$, where t is timestamp information and x, y, z represent three axis values of the accelerometer and gyroscope sensors. In the preprocessing, to remove the effect of the noise from data, M-point Moving Average Filter (MAF) is applied. After filtering, to obtain a scale-invariant feature vector, the feature vector is normalized through the linear normalization. In this chapter, we start with the original size of 84 statistical features, but to increase both the computational efficiency and accuracy, we applied a feature selection algorithm, which is explained in the next subsection. In the *decision module* of WACA, the user is classified as authorized or unauthorized for the claimed credentials entered during the initial login. The final authentication decision is given by comparing the samples of through the distance measures such as Euclidean or Manhattan distance.

New feature extraction and optimization

As noted above, in WACA, the length of the feature vector is 84. However, we observed that this both affects the security and performance of the system negatively. To prevent this, we improved WACA by applying a feature selection algorithm. Specifically, we applied different univariate feature selection algorithms. The reason we chose univariate algorithms is that we did not want the final feature vector to be dependent on the algorithms used in the decision module. Particularly, we tested three different univariate feature selection algorithms: Chi2 [PVG+11a], Mutual Information [KL87], and F-score [PVG+11b]. The results are plotted according to

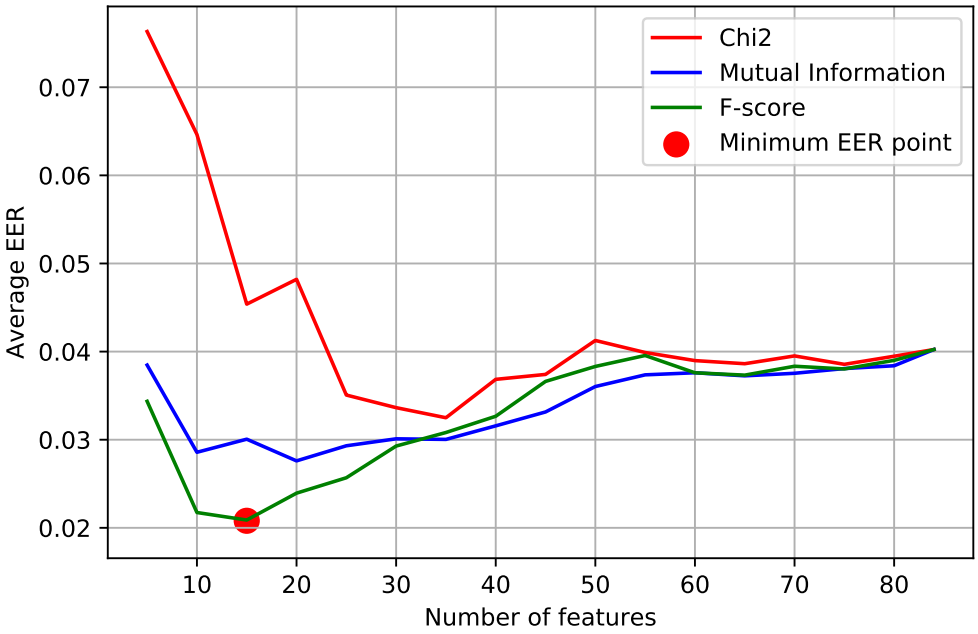


Figure 4.6: The result of feature selection algorithms.

a varying number of features in Figure 4.6. As can be seen in Figure 4.6, the result of F-score algorithms with the feature vector length of 15 gives the least (average) EER, 0.0208. The top 15 features selected by F-score are specified in Table 4.2. For the rest of the chapter, we use these 15 features of WACA, instead of originally proposed 84 features in WACA.

4.5.2 Testing with a secure template generation and comparison method: NTT-Sec-$\mathbb{R}$

As mentioned earlier (Section 4.2.2) that our protocol description requires secure template generation (TempGen) and comparison (TempComp) functions. Our implementation is based on the cryptographic primitive NTT-Sec [KC16]. We chose NTT-Sec because (1) the NTT-Sec is solely based on publicly computable functions (generalizing cryptographic hash functions in a setting with noisy measurements

Table 4.2: 15 features chosen by F-score algorithm and used in our experiments.

Feature	**Formula**	**F-scores**
mean of accelerometer' x-axis	$mean(acc_x)$	1289.51
cross-correlation between accelerometer' x- and z-axis	$sum(abs(xcorr(acc_x, acc_z)))$	989.89
median of accelerometer' x-axis	$median(acc_x)$	626.99
median of accelerometer' y-axis	$median(acc_y)$	497.72
mean of accelerometer' y-axis	$mean(acc_y)$	466.377
entropy of accelerometer' y-axis	$entropy(acc_y)$	377.96
entropy of accelerometer' x-axis	$entropy(acc_x)$	285.51
mean absolute deviation of gyroscope' y-axis	$mad(gyro_y)$	205.32
cross-correlation between accelerometer' y- and z-axis	$sum(abs(xcorr(acc_y, acc_z)))$	175.16
range of gyroscope' y-axis	$range(gyro_y)$	171.11
covariance of gyroscope' y-axis	$cov(gyro_y)$	151.62
spectral energy of gyroscope's y-axis	$sum(fft(gyro_y). * conj(fft(gyro_y)))$	144.86
spectral energy of accelerometer's z-axis	$sum(fft(acc_z). * conj(fft(acc_z)))$	136.50
mean absolute deviation of gyroscope's z-axis	$mad(gyro_z)$	131.15
mean of accelerometer's z-axis	$mean(acc_z)$	122.00

of data), and (2) NTT-Sec offers formal security analysis with no known attacks to date. Overall, NTT-Sec offers certain advantages over its alternatives. More specifically, (1) homomorphic encryption-based methods [AAUC18] are not suitable for the CA protocol due to the requirement of public key and private key on the user side; (2) many of the previously known biometric cryptosystems (e.g., fuzzy extractors [JW99]) are known to have security issues for their reusability [BA11] and they are limited in their noise tolerance capability; (3) Cancelable biometrics constructions [RU11], in general, lack formal security analysis, and many constructions have been shown to be vulnerable under false acceptance and stolen key attacks [AGKL19].

NTT-Sec consists of two algorithms called Proj (project) and Decomp (decompose). The Proj algorithm maps a length-n binary vector (considered as the feature vector) to a finite field element (considered as its secure template) using a priori-fixed set of public parameters and a factor basis. Given a pair of secure templates, the Decomp algorithm can detect whether the templates originate from a pair of binary feature vectors that differ in at most t indices for a priori-fixed error threshold value

t. In Decomp, the detection is achieved by checking whether a particular finite field element can be written (decomposed) as a product of the factor basis elements in a certain form. Computations in NTT-Sec are performed in a cyclotomic subgroup $\mathbb{G}$ of the multiplicative group of a finite field. We adapt the same group structure in our modification. More specifically, let $\mathbb{F}_q$ be a finite field with q elements where $q = p^m$. Let $c \in \mathbb{F}_q$ be a non-quadratic residue with minimal polynomial of degree m over $\mathbb{F}_p$. Let $\mathbb{F}_{q^2} = \mathbb{F}_q(\sigma)$ be a degree two extension of $\mathbb{F}_q$ where σ is a root of $x^2 - c$. $\mathbb{F}_{q^2}$ has a cyclotomic subgroup $\mathbb{G}$ of order q and every non-identity element in $\mathbb{G}$ can be represented as $\frac{a+\sigma}{a-\sigma}$ for some $a \in \mathbb{F}_q$. Moreover, we say an element $a \in \mathbb{G}$ is *k-decomposable* over $\mathbb{F}_p$ if it can be written as a product $a = \prod_{i=1}^{k} \left(\frac{a_i+\sigma}{a_i-\sigma}\right)$ for some $\mathbb{F}_p$-elements $a_1, a_2, \ldots a_k$.

The original NTT-Sec is only limited working with binary feature vectors by its design. On the other hand, biometric data [AAUA18] we deal with in this work and in most cases such as physiological biometrics [RKBT07] or behavioral biometrics [FBM+13] is represented through real-valued feature vectors. Therefore, we extend NTT-Sec to a new construction NTT-Sec-$\mathbb{R}$, which comprises two algorithms called NTT-Hash-$\mathbb{R}$ and NTT-Match-$\mathbb{R}$. We use the *scale-then-round* transformation in [AKK19] to transform the real-valued feature vectors to integer-valued vectors. Moreover, we describe NTT-Param-$\mathbb{R}$ for the new parameters required in NTT-Sec-$\mathbb{R}$.

NTT-Param-$\mathbb{R}$

We assume that n and t are some fixed values that represent the length of feature vectors and (original) system threshold, respectively. More specifically, a (non-cryptographic) biometric authentication system would declare match for an input pair of biometric data if and only if $d(x, y) \leq t$, where d is the Manhattan distance

function (ℓ_1), and x, y are the length-n feature vectors of the biometric data. The parameters of NTT-Sec-$\mathbb{R}$ are defined as follows:

- a scaling factor s,
- a prime number p such that $p > 2n$,
- an integer m such that $m \geq \lfloor st \rceil$,
- a set $\mathsf{B} = \{g_1, g_2, \ldots, g_n\}$ such that $1 \leq g_i \leq \frac{p-1}{2}$ for each i.

We pack all of these parameters under the set $\mathsf{SP} = \{n, t, s, p, m, \mathsf{B}\}$, and call this as the system parameter set. Note that SP can be made public, and commonly used in the NTT-Hash-$\mathbb{R}$ and NTT-Match-$\mathbb{R}$ algorithms. SP should be determined in accordance with the desired security parameter λ in order to make NTT-Sec-$\mathbb{R}$ resistant to adversarial attacks.

NTT-Hash-$\mathbb{R}$

This algorithm maps a real-valued feature vector to a $\mathbb{G}$-element called *hash*[3] *value* as follows: Assume a fixed-length real-valued feature vector $x = (x_1, x_2, \ldots, x_n)$ in $[0, 1]^n$ is given. Using the *scaling factor* s and the *basis* $\mathsf{B} = \{g_1, g_2, \ldots, g_n\}$, it is mapped to a $\mathbb{G}$-element, defined as

$$\text{NTT-Hash-}\mathbb{R}(x) = \prod_{i=1}^{n} \left(\frac{g_i + \sigma}{g_i - \sigma} \right)^{\lfloor s x_i \rceil}$$

where $\lfloor \cdot \rceil$ is the nearest integer function.

NTT-Match-$\mathbb{R}$

For a given hash value $h = \text{NTT-Hash-}\mathbb{R}(x)$ for some $x = (x_1, \ldots, x_n)$ in $[0, 1]^n$, a real-valued vector $y = (y_1, \ldots, y_n)$ in $[0, 1]^n$ and a positive real number t, the goal

[3]We have chosen the name "*hash" because the algorithm eventually satisfies randomness and irreversibility similar to the hash functions.*

of NTT-Match-$\mathbb{R}$ is to decide whether $\sum_{i=1} |x_i - y_i| \leq t$ or not by using their hash values. To achieve this goal, the following process is performed.

NTT-Hash-$\mathbb{R}$ computes $h_y =$ NTT-Hash-$\mathbb{R}(y)$, and then it decides whether the $\mathbb{G}$-element h/h_y is $\lfloor st \rceil$-decomposable. Furthermore, if the retrieved $\mathbb{F}_p$-elements belong to the basis B, NTT-Match-$\mathbb{R}$ returns Match, otherwise No $-$ Match.

4.5.3 A Security Analysis of NTT-Sec-$\mathbb{R}$

The best strategy for an adversary to attack the new NTT-Sec-$\mathbb{R}$ method (with respect to both irreversibility and indistinguisahbility notions) is to solve the discrete logarithm problem in the underlying cyclotomic group, which belongs to the finite field $\mathbb{F}_{p^{2m}}$. Discrete logarithms in $\mathbb{F}_{p^{2m}}$ can be computed in time bounded by $(\max(p, m))^{O(\log_2 m)}$ [BGJT14]. As analyzed in [KC16], an attacker needs to solve $(n + 1)$ discrete logarithms, and so we calculate the cost of this discrete logarithm attack to be $(n + 1)(\max(p, m))^{\log_2 m}$.

Security Levels

In Section 4.6.1, we analyze the security level of our NTT-Sec-$\mathbb{R}$ implementations using the scalars $s = 100$ and $s = 400$, and denoted by NTTSec_{100} and NTTSec_{400}, respectively. The prime number $p = 31$ is chosen for both implementations. Note that the vector length is fixed as $n = 15$. Using these parameters, the security levels λ, which correspond to the minimum cost of the DLP attack [BGJT14] and considered as 2^λ, are provided in Table 4.3.

Remark. We note that we are rather conservative in our security analysis, and our estimated bit security levels can be increased in practice at almost no-cost. For example, since a user is already equipped with a password in the protocol,

Table 4.3: Security Levels of NTT-Sec-$\mathbb{R}$ for each user tested against the Discrete Logarithm Problem (DLP) attack [BGJT14].

User	DLP			
	$\mathbf{NTTSec}_{100}$		$\mathbf{NTTSec}_{400}$	
	m	λ	m	λ
1	71	42	281	70
2	101	48	389	78
3	73	42	307	72
4	59	39	227	65
5	101	48	397	79
6	67	41	241	67
7	97	48	367	77
8	131	53	509	85
9	59	39	239	66
10	229	65	919	101

User	DLP			
	$\mathbf{NTTSec}_{100}$		$\mathbf{NTTSec}_{400}$	
	m	λ	m	λ
11	89	46	347	75
12	67	41	241	67
13	89	46	337	75
14	167	59	673	92
15	191	61	739	95
16	149	56	587	89
17	43	33	173	59
18	137	54	541	86
19	97	48	359	76
20	113	51	443	81

that password can be taken as part of the input in the feature extraction process, while making attacker's task harder in the template reversing attack. This would also allow a legitimate user to revoke his template, and reissue a new template by changing his password and re-enrolling to the system, and also to reuse his biometric data over different systems by choosing different passwords.

4.5.4 Security benefits of NTT-Sec-$\mathbb{R}$

In this section, we show the extra security benefits of NTT-Sec-$\mathbb{R}$, in addition to the security properties provided by the PAKE protocol.

User Data privacy. The user data is very sensitive as it contains the biometrics information so it should be protected from any third party including the authentication server and as well as any kind of eavesdropping. In our protocol, the data is transformed in an indistinguishable and irreversible way before transmitted to any party from the users. Therefore, no party sees the sensitive user data in cleartext.

No key required. The security of NTT-Sec-$\mathbb{R}$ is based on a discrete logarithm problem, where it does not require to store any keys. Therefore, the security of NTT-Sec-$\mathbb{R}$ is not based on a key.

4.6 Performance Evaluation

In this section, we evaluate our proposed system in terms of accuracy and resource consumption.

Implementation Details. To evaluate the performance of our proposed concrete privacy-aware continuous authentication system, we implemented it on a real system. Specifically, for the timing results of NTT-Sec-$\mathbb{R}$ algorithm's implementation, the codes were written in the C programming language using the GCC 5.4.0 compiler. The Core i7-7700 CPU @ 3.60GHz desktop computer was used with Ubuntu 16.04 LTS running. The CPU time of the match operation was measured using the function clock() from the time.h library. All the timings were provided in milliseconds. For the linear algebra and finite field computations, we used the popular FLINT C-library by William Hart et al. [HJP13]. In our implementation, for accuracy analysis, we used the scalars $40, 100, 400$, and 1000 and denoted by NTTSec_{40}, NTTSec_{100}, NTTSec_{400}, and NTTSec_{1000}, respectively.

To measure the resource consumption of our proposed continuous authentication system, we used an Apple smartwatch. The results of the resource consumption experiments are given in Figure 4.9. The feature calculation was strictly done on the device. The measurements were taken on a 38mm Apple Watch. The feature calculation code was implemented in C. We used KissFFT library [Bor] for FFT calculations in spectral entropy and cross-correlation features. It is worth noting that we only implemented the most time-consuming parts and the computations

on the relatively constrained devices (e.g., smartwatches). The implementation of PAKE was already implemented and analyzed in many works and languages [Wu07, Coc20] before.

4.6.1 Accuracy Analysis

A User-based evaluation model with training

We pick the first 10 feature vectors of the i'th user for training. Denote this set by $\text{Train}_i = \{[i,j] : \ j = 1, ..., 10\}$, and the remaining 10 feature vectors by $\text{Test}_i = \{[i,j] : \ j = 11, ..., 20\}$. We picked a subset of 5 feature vectors from Train_i, and computed the mean of these 5 feature vectors combinations. This is also called as the *gallery* feature vector of a user. As a result, we generated $\binom{10}{5} = 252$ gallery feature vectors per user (simulating 252 different enrollments of a user), and denoted this set by Gallery_i. In our EER_i calculations, we paired each vector from Gallery_i and Test_i. This yielded $252 \cdot 10 = 2520$ genuine comparisons for the i'th user. For the i'th user, we also paired the first 10 vectors from Gallery_i with all the vectors from Test_j for all $j \neq i$. This yielded $10 \cdot 10 \cdot 19 = 1900$ imposter comparisons for the i'th user.

Implementation Results

In this section, we discuss our implementation results. Using the same dataset, we implemented two different techniques: Manhattan Distance (MD) (i.e., no secure template generation) and NTT-Sec-$\mathbb{R}$ algorithm. Unlike the MD, the NTT-Sec-$\mathbb{R}$ algorithm requires the feature vector elements to be an integer; therefore, using the accuracy preserving transformation idea in [AKK19], we transformed real-valued to the integer-valued feature vector. The selection of scalar for scaling purposes is

important, and for that reason, we analyzed the experimental results of different (suitable) scalars. Note that we are reporting the FRR and FAR values at the first threshold point where FRR becomes less than or equal to FAR, implying EER.

For the FRR and FAR values, the MD was implemented using the Python Programming language, where the threshold value was incremented by 0.001 in each step to obtain accurate results. Furthermore, the referenced vectors and query vectors were both taken as floating-point values. Hence, the error rates of the MD can be used as a point of reference — in terms of the result accuracy — for the NTT-Sec-$\mathbb{R}$ implementations. Using the EER threshold points from the MD results, we determined the parameters for the NTT-Sec-$\mathbb{R}$ algorithm. Among the parameters, the threshold values were computed as $T = \lfloor s \cdot t \rceil$ where "s" is the (chosen) scalar and "t" is the threshold value obtained from MD results. Considering accuracy and computational efficiency, we tested different scalars. Smaller scalar means faster computation but more loss of error rate accuracy while larger scalar implies slower computation and lesser loss of accuracy.

To show the change of error rate accuracy, we computed $|\text{MD} - \text{NTTSec}_i|$ using the FRR and FAR values for each user where $i \in \{40, 100, 400, 1000\}$. The absolute FRR and FAR differences of the NTTSec implementations w.r.t. the MD are presented in Figure 4.7 and Figure 4.8, respectively.

For the ease of our readers and to save space, the Figures 4.7 and 4.8 are the summary of our extensive results. It is evident that scalar 40 results in more loss of error rate accuracy than other selected scalars in both figures. One the other hand, the accuracy is well preserved using the scalar 1000. 1000 is 25 times of 40, which implies that the efficiency of the match function of NTT-Match-$\mathbb{R}$ algorithm will be affected even by a higher factor due to the very large underlying algebraic

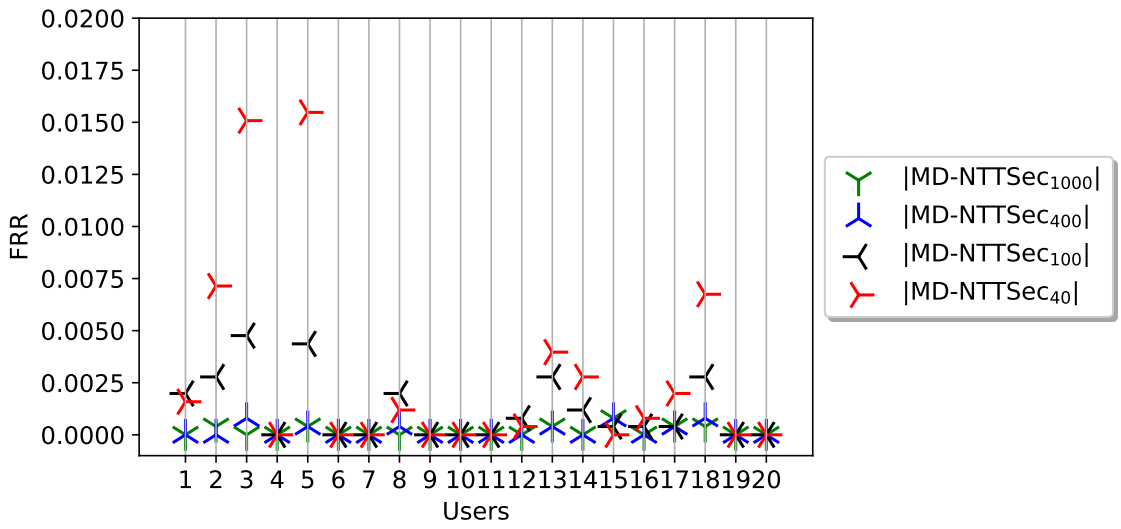


Figure 4.7: The absolute FRR difference between the Manhattan distance (MD) and NTT-Sec-$\mathbb{R}$ implementations using the scalars $40, 100, 400$ and 1000 for all the 20 users.

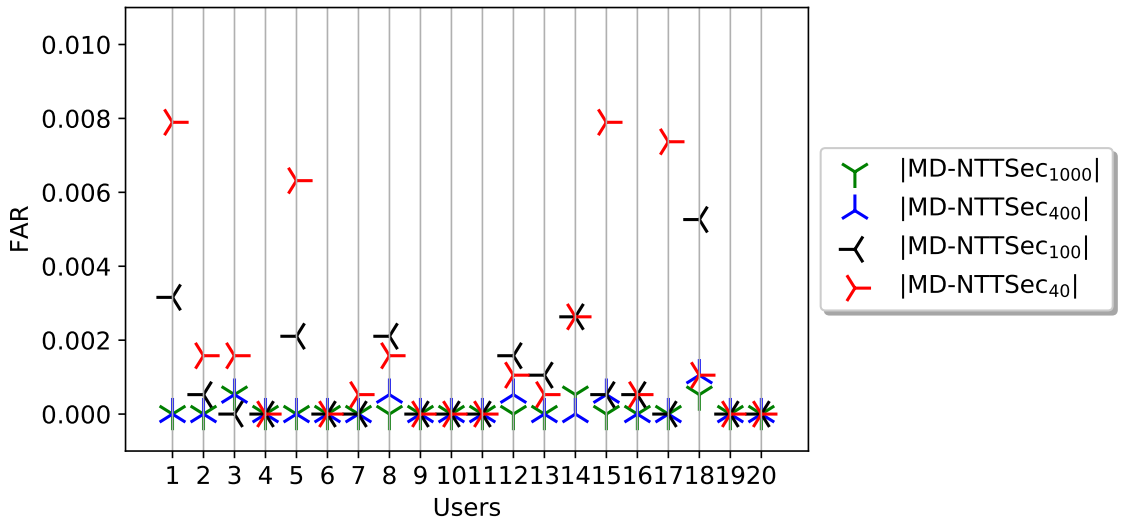


Figure 4.8: The absolute FAR difference between the Manhattan distance (MD) and NTT-Sec-$\mathbb{R}$ implementations using the scalars $40, 100, 400$ and 1000 for all the 20 users.

structure. Hence, the scalars 100 and 400 seem to be good candidates for deciding on efficiency and keeping security into consideration.

Timing Results

In this section, we report only the timing results of the matching function because it is the bottleneck in the running time of NTT-Sec-$\mathbb{R}$ algorithm. The Match operation corresponds to TempComp function in Figure 4.5 and in our protocol. As explained in Section 4.5.2, the NTT-Sec-$\mathbb{R}$ requires finite field and linear algebra operations over integers; therefore, the scalar is directly proportional to the efficiency of the NTT-Sec-$\mathbb{R}$. But one must also balance the loss of error rate accuracy and computation efficiency. For example, for User-10, we find the average CPU timings of 4.494, 39.652 and 1408.454 milliseconds in the NTTSec_{40}, NTTSec_{100}, and NTTSec_{400}, respectively. The loss of computational efficiency from scalar 100 to 400 is more significant than the loss of error rate accuracy. But the security of the protected template should also be considered. Therefore, we are focusing on and reporting the timing results of the NTTSec_{100} and NTTSec_{400} for all users.

The NTT-Match-$\mathbb{R}$ algorithm is comprised of Hash and Match functions, as also explained in Section 4.5.2. The Hash function takes the feature vector and finite field parameters as input. Then, the function computes a new field element by using the elements of the feature vector as an exponent of the field element. Therefore, the hash function corresponds to the secure template generation algorithm in PACA (i.e., TempGen). Similarly, the Match function takes the same finite field parameters and two field elements to perform the comparison. The function outputs Match or No-Match according to the (fixed) threshold. Similar to the hash function, the match function corresponds to the secure template comparison algorithm in PACA (i.e., TempComp) Note that the Match function requires the hashed values of both

Table 4.4: The average timing results of the Match functions of NTTSec_{100} and NTTSec_{400} algorithms in milliseconds.

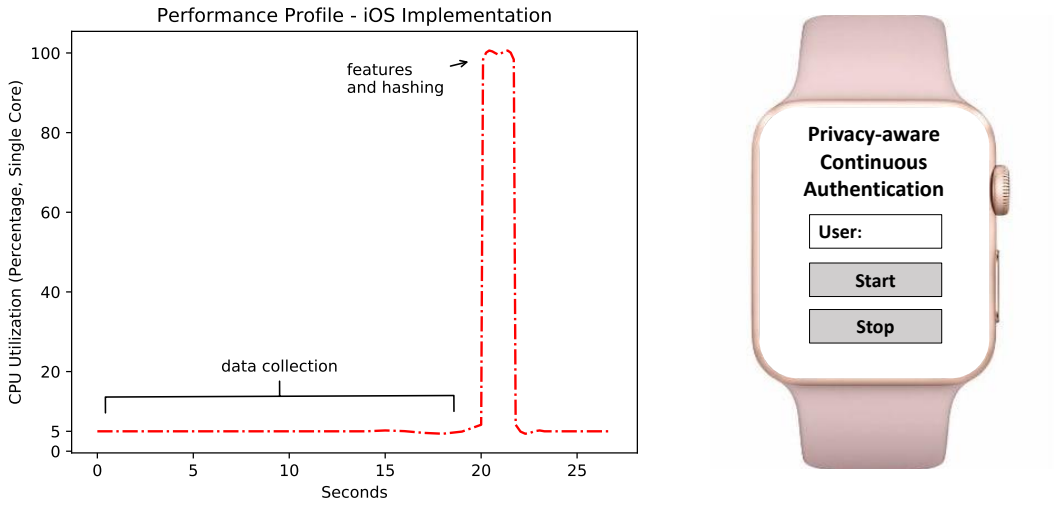
User	1	2	3	4	5	6	7	8	9	10
$\textbf{NTTSec}_{100}$	2.667	5.877	2.689	1.302	5.893	2.290	4.530	9.681	1.286	39.652
$\textbf{NTTSec}_{400}$	68.773	154.441	83.268	39.198	143.128	35.313	113.392	290.592	43.850	1408.454
User	11	12	13	14	15	16	17	18	19	20
$\textbf{NTTSec}_{100}$	4.305	2.264	4.285	15.924	27.268	14.821	0.769	12.034	4.527	6.740
$\textbf{NTTSec}_{400}$	116.286	35.172	97.893	507.976	762.882	378.951	21.117	371.429	109.914	225.091

the query and reference vectors. The reference vectors are stored as hashed values while the query vector is required to be hashed first and then pass to the Match function for comparison. We report the average CPU time of the Match function only for each user in Table 4.4.

In Table 4.4, for NTTSec_{100}, the minimum and maximum average CPU time of 0.769 and 39.652 milliseconds, respectively. For NTTSec_{400}, the minimum and maximum average CPU time of 21.117 and 1408.454 milliseconds, respectively. These minimum and maximum timing results are observed for the users 17 and 10, respectively.

4.6.2 Resource Consumption Analysis

To measure the resource consumption of our proposed continuous authentication system, we implemented our proposed system on an Apple iWatch device. Figure 4.9 shows how the CPU was utilized throughout the feature calculation period and a screenshot of the application used for the experiments. For 20 seconds, the watch collects data from both the gyroscope and the accelerometer. The device yields 50 measurements per second, giving us 1000 data points in one dimension. For both gyroscope and accelerometer, we have three dimensions; hence the total count of measurements amount to 6000. One CPU core is utilized in its complete capacity for a brief amount of time when enough time is elapsed. The total time



(a) (b)

Figure 4.9: a) CPU profile of iOS implementation on an Apple iWatch. b) A screenshot of the application used for the experiments on the Apple iWatch.

required for calculations, which is noted in Figure 4.9 by the peak, is 1.8 seconds; the majority of this time (approximately 90%) is spent to calculate the hash. After, the device begins to collect sensor measurements again, followed by a repeated feature calculation. This profile repeats as long as the framework is in operation. The application cruises at 5% on the sensor measurement collection phase. After this, the app calculates the features from sensor data and hashes it, which is illustrated by the peak. The peak's width is approximately 1.6 seconds, which corresponds to the feature extraction (i.e., Ext) and template generation (i.e., TempGen) in total. The measurement collection phase follows afterward. This profile repeats until the application is stopped.

The memory footprint of the implementation is minuscule and constant, coasting around 3.5 MB. Such a memory profile is expected because upon the completion of the feature calculation period, the previous data is discarded, and since the period and sampling rate is fixed, approximately the same amount of data is recorded anew.

We measured the battery consumption by initiating the algorithm and sampling the overall battery percentage every minute for more than 4 hours. We observed that the application consumed 1% of the battery approximately every 4 minutes, which yields us an operational time of 400 minutes, or 6.5 hours for PACA. Note that this number is an absolute lower bound: during the measurements, a debugger was attached to the device, the device's screen was lit, and the maximum possible values for s and t were selected, greatly increasing computational requirements and the battery consumption. Hence, PACA is very promising in terms of resource consumption.

4.7 Conclusion

Unlike the one-time authentication systems, the continuous authentication systems are more suitable and better suited to the contemporary threats in cyberspace. Due to its sensitivity and uniqueness, the biometric data requires proper security and privacy mechanisms in place. Existing solutions like the password-hash-matching systems do not work in noise-tolerant biometric authentication systems, while the privacy-preserving homomorphic encryption constructs are not feasible for continuous authentication due to its performance limitations. In this chapter, we constructed a lightweight, privacy-aware, and secure continuous authentication protocol and a comprehensive system under the protocol. Formally proving its security and privacy guarantees against eight different attacks, we further deployed our system with $\mathsf{NTT\text{-}Sec}\text{-}\mathbb{R}$ and a continuous biometric authentication system using an Apple smartwatch. We evaluated our protocol's efficiency with data collected from real users and validated that it incurs a minimal overhead. The proposed novel scheme and results can be easily generalized to other biometric authentication mechanisms

for both continuous and traditional noncontinuous settings with real-valued feature vectors. Hence, the proposed protocol enables privacy-aware continuous biometric authentication, which can fundamentally improve the security in cyberspace.

CHAPTER 5

PINTA: A PRIVACY-PRESERVING MULTI-FACTOR AUTHENTICATION SYSTEM

5.1 Introduction

Both for highly sensitive systems such as online retail and e-banking and relatively less critical systems such as desktop machines in a corporate Intranet and social networks, it is crucial to protect users' accounts and assets from malicious third parties. Only password-based authentication systems suffer from many weaknesses, like password cracking, susceptibility to phishing and cross-site password reuse. Once the password is compromised, an adversary can easily misuse the victim's account. Thus, there is a great demand to establish a multi-factor authentication (MFA) system, which requires two or more authentication factors (i.e., knowledge, possession, identity) to validate users during their login. Popular web services such as Amazon Web Services [Que12], Google Accounts [Goo12], Microsoft Outlook [Mic19] have already deployed MFA. However, in all of these techniques, an out-of-band channel (e.g., an App, text message) and an additional action from the user is required. This reduces usability significantly. Similarly, in the literature, there have been a number of academic works proposing MFA systems [PMZ+11, Ver12, SP12, JY11, ZPW11].

The more practical (and therefore more likely to be widely-adopted) MFA solutions are based on users' behavior; however, they do little to protect their privacy. In particular, in an MFA system, users face the risk of exposing their personal information to database servers or a malicious adversary. First, the owner of the database server may use it for malicious purposes (e.g., selling user's information for economic interest) or if an adversary breaks into the database server and succeeds

in obtaining the profile belonging to the targeted user, he/she can masquerade as a legitimate user by crafting required authentication factors.

Nonetheless, there are several challenges in achieving a privacy-preserving MFA system based on user profiles. First, a widely acknowledged challenge in the area of user profiling is how to accurately model a user's behavior while it constantly changes [SKD+10]. Note that even for the same user, there may be a difference between two profiles collected at different times. The second challenge is to identify a unique user from others based on their own varying profiles. The third challenge is to enable a server to verify the aforementioned profile, given that: (1) it cannot be read by the server (to preserve user privacy), and (2) it will vary over time (so standard cryptography cannot be used).

In this chapter, we designed a privacy-preserving method for multi-factor authentication systems, called PINTA, in which the privacy of the collected hybrid user behavior profiles serving as a second authentication factor is protected from the authentication server. Moreover, we also adopt fuzzy hashing [Kor06] and fully homomorphic encryption (FHE) [Gen09] techniques to ensure that a user's personal information is not leaked to servers or a third party. Furthermore, PINTA uses a large combination of host-based characteristics and network-based features to profile users. The combination of multiple features (26 configurable features in total) enables a much simpler, distance metric-based user classification instead of expensive machine learning. Finally, while fuzzy hashing is used to match the strings portion of the user profile, FHE is used to match integer numbers without knowing the actual value. For the experiments, we used a user profile database derived from several public datasets [BSPvdM12, She12, KM12] and a dataset we generate. Our results show that the proposed scheme can well detect imposters from legitimate users while protecting user privacy.

5.2 Problem Formulation & Preliminaries

In this section, we first introduce our assumptions and the adversary model. Then, we outline the design goals of our work. We also provide a brief introduction of *fuzzy hashing* and *homomorphic encryption*.

5.2.1 Assumptions and Adversary Model

In our system, we make the following two assumptions:

1) *Perfect knowledge assumption*: We assume that the adversary has perfect knowledge of the multi-factor authentication system including the strategy of user profile acquisition, the mechanism of profile encryption/hashing, and the details of the authentication protocol.

2) *First-Factor knowledge assumption*: We assume that the adversary knows the victim's first authentication factor, which is the user password.

In our adversary model, a malicious entity can attack the proposed multi-factor authentication system via impersonating a legitimate user (victim) in order to gain access to the victim's account. We note that we do not consider any low-level communication adversary model such as man-in-the-middle and replay attacks because those attacks can be prevented by the appropriate implementation of one or several security protocols [IET08] and, hence, such attacks are not within the scope of our work. We specifically consider the following two types of adversaries: *brute-force attacker* and *honest-but-curious server attacker*.

1) *Brute-force attacker*: A computationally bounded third-party adversary may attempt to authenticate with a spoofed second authentication factor by exhaustively searching for the correct user profile. Such an attack consists of enumerating all

possible user profiles until the correct one is found, which, in the worst case, would involve traversing the entire message space.

2) *Honest-but-curious server attacker*: In our system, we assume the server that processes the authentication requests is *honest-but-curious* that (1) stores incoming cryptographic data without tampering with it; (2) honestly processes each authentication request and returns the corresponding outcome; (3) but tries to derive the underlying sensitive information from the user's cryptographic profile.

5.2.2 Design Goals

The design goals of our work are outlined below.

1) *Privacy Preservation Assurance*: The system should guarantee that the privacy of each user is well preserved. To be specific, never should anyone including the *honest-but-curious server* and the malicious third-party obtain the user profiles in plaintext. We analyze the security of our proposed system in Section 5.5. We adopt fuzzy hashing and fully homomorphic encryption techniques to provide this assurance.

2) *Authentication Usability Assurance*: The system should ensure a pleasant user experience by satisfying the following three conditions. First, a login should not need extra effort other than typing the username-password pair. Second, the system overhead (i.e., authentication delay) of the entire authentication process should be tolerable. Third, the program that runs on the client's machine should not consume a lot of computing resources. We evaluate system overhead and resource utilization in Section 5.4.

3) *Authentication Accuracy Assurance*: The system should ensure that the authentication is accurate with acceptable recall and false positive rate (FPR), which

poses two challenges. First, the system should be adaptive and flexible enough to tolerate slight changes in the users' profiles. Second, the system should be sensitive enough to recognize a login request initiated by an adversary.

5.2.3 Fuzzy Hashing and Homomorphic Encryption

Fuzzy Hashing: The system proposed in this work uses fuzzy hashing, also called Context Triggered Piecewise Hashing [Kor06], which is a hashing function that can match data with similarities. Jesse Kornblum first proposed a generic fuzzy hashing scheme in [Kor06] and implemented his algorithm as `ssdeep` [Kor18]. In Rousseu's approach, each similarity digest SD for a byte stream is generated by employing a sequence of Bloom Filters, which are bit vectors used for space-efficient set representation. Given two similarity digests SD_1 and SD_2, the similarity digest score is generated by the function SD_{score} $(SD_1$ $, SD_2)$, which yields a score of zero for a mismatch or a matching score ranging from 1 to 100. In industry, fuzzy hashing has been applied in the realm of security forensics, especially in identifying morphing malware and spam [BCH+09]. In our system, we adopt fuzzy hashing to evaluate the similarity of two fuzzy hash values of user behavior features in the forms of strings, without revealing the user's sensitive information to the authentication server.

We conducted a simple experiment to show the efficacy of fuzzy hashing and how fuzzy hashed strings can be compared to gauge their level of similarity. The procedure of the experiment is as follows: (1) We used a newly-installed (clean) Windows 7 OS and ten different software installation packages, labeled as A, B, ... J. (2) We installed Software A on the clean OS and used the **tree** command to output the file hierarchy in the folder path 'C:"Program Files' as a string, denoted

Table 5.1: Similarity score between fuzzy hashes. The element in the ith row and jth column represents the value of $SD_{Score}(SD_i, SD_j)$

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	100	89	89	74	63	62	62	40	40	40
2	89	100	86	76	76	55	55	45	45	45
3	89	86	100	92	82	77	58	58	48	38
4	74	76	92	100	90	90	70	60	50	50
5	63	76	82	90	100	81	81	72	62	62
6	62	55	77	90	81	100	91	81	72	72
7	40	55	58	70	81	91	100	91	81	81
8	40	45	58	60	71	81	91	100	92	92
9	40	45	48	50	62	72	81	92	100	100
10	40	45	38	50	62	72	81	92	100	100

by Str_1. (3) We installed ten software packages one by one and obtained 10 strings, denoted $Str_1, Str_2...Str_10$. Then, we generated the fuzzy hash for each string using *ssdeep*, denoted by $SD_1, SD_2...SD_10$. We computed the *similarity score* (SD_{Score}) between every hash generated by *ssdeep*. Since we had ten fuzzy hashes, there are 100 combinations in total. The SD_{Score} for each combination is presented in Table 5.1. From Table 5.1, we make several observations. First, all the similarity scores are diagonally symmetric because SD_{Score} (SD_1, SD_2) and SD_{Score}(SD_2, SD_1) are the same. Second, the similarity score for the fuzzy hashes of the same string is always 100. Third, with the installation of more software (i.e., more of a change to the file hierarchy and corresponding string), the similarity score reduces. Therefore, we can say that the *similarity score* of fuzzy hashes can roughly represent how similar two fuzzy hashed strings are.

Homomorphic Encryption: The second technique, in this work, to achieve the privacy of a user's profile involves the application of fully homomorphic encryption (FHE). After being an open problem for a long time, the first plausible FHE scheme was introduced by Craig Gentry in his Ph.D. thesis [Gen09] in 2009. Homomorphic encryption has an additional algorithm compared to the traditional encryption al-

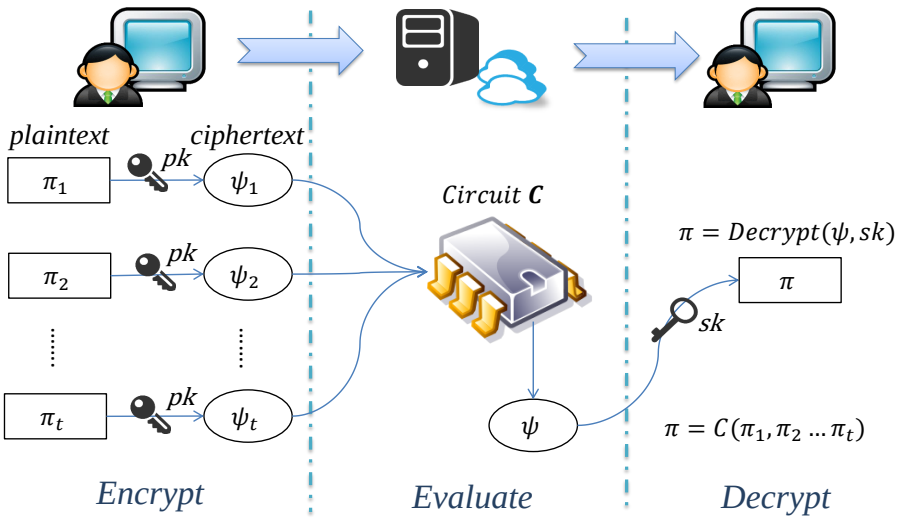


Figure 5.1: Fully Homomorphic Encryption

gorithms, which is called homomorphic evaluation algorithms and allows to be able to perform operations on the encrypted data without decrypting it. Particularly, an homomorphic encryption scheme ε has an algorithm $Evaluate_\varepsilon$ that, given plaintext $\pi_1, \pi_2, ...\pi_t$, for any valid ε, private, public key pair (sk, pk), any circuit C, and any ciphertext $\psi_i \longleftarrow Encrypt_\varepsilon(pk, \pi_i)$, yields

$$\begin{gathered} \psi \longleftarrow Evaluate_\varepsilon(pk, C, \psi_1 ... \psi_t) \\ \text{such that } Decrypt_\varepsilon(sk, \psi) = C(\pi_1, \pi_2 ... \pi_t) \end{gathered} \tag{5.1}$$

A typical scenario of FHE is illustrated in Figure 5.1. In this scenario, the user encrypts the data with public key pk and the function $Encrypt$ and sends the encrypted data to the server. The server in the cloud performs operations on the encrypted data by using the function $Evaluate$ with public key pk and outputs ψ. The server sends ψ back to the user. The user then decrypts ψ by function $Decrypt$ with his private key sk and obtains the result of $C(\pi_1, \pi_2 ... \pi_t)$. In this way, the server conducts the desired operation for the user without acquiring any plaintext. Van Dijk

et al. later used Gentry's technique to establish a fully homomorphic encryption system over integers [VDGHV10]. Coron and his colleagues further improved van Dijk's work by reducing the size of public keys and time complexity [CMNT11b]. Since then many improvements have been proposed [BV11, Bra12, BGV14, BV14a] and the research on FHE still an active research area [AAUC18]. In our experiments, we used Microsoft's open-source homomorphic encryption library called `SEAL` [MR18] in our authentication system, which is a C++ implementation of homomorphic encryption. The current version of the SEAL implements two different encryption schemes: BFV and CKKS. We used the BFV version, which is originally proposed in [FV12a] and we chose the parameters to provide 128 bits security.

5.3 Proposed System

In this section, we introduce the design of PINTA. Our proposed system collects the user behavior to serve as a second authentication factor along with the user's password. However, unlike conventional user behavior profiling, the user information acquisition, transmission, and storage all occur in a privacy-preserving fashion. Furthermore, to prove that none of these stages in the system violates user privacy, the proposed system is assumed to be open to the public, which corresponds to our *Perfect Knowledge Assumption* as stated in Section 3.1.

5.3.1 System Overview

The architecture of the developed system has four primary components and is shown in Figure 5.2: (1) an open-source *profile acquisition program* (PAP) that runs on the user's local host; (2) a *user profile database* (UPDB) that stores the user's information in a privacy-preserving fashion; (3) an *authentication server* (AS) that

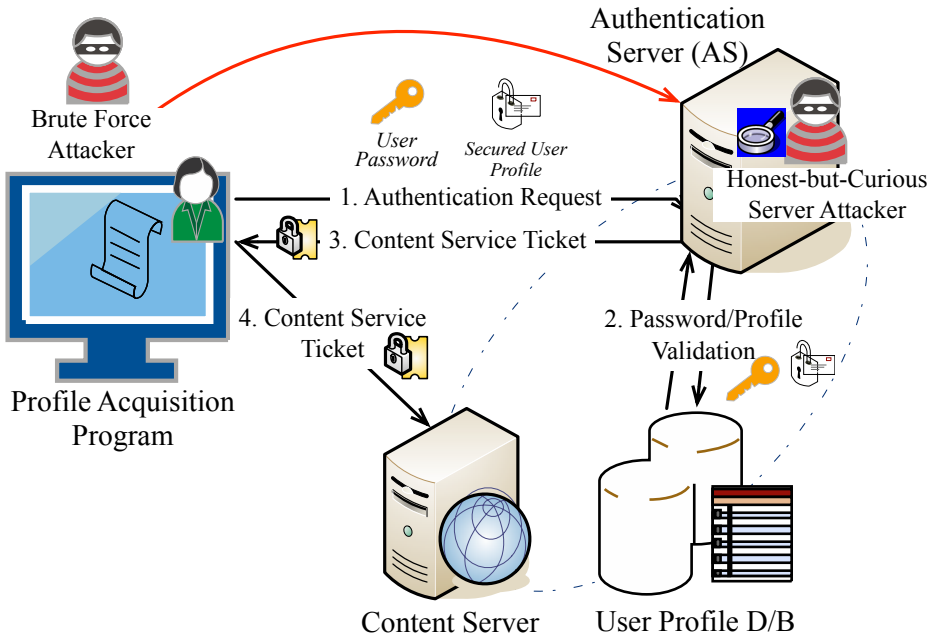


Figure 5.2: Overview of the Privacy-Preserving MFA System

processes and validates user's login request; and (4) a content server. We assume UPDB, AS and the server are owned by the same organization, but this is not a strict requirement.

The first time a user uses PINTA with a specific site, he/she must go through the process of *enrollment*. During the enrollment, the user information acquisition program collects a user profile, denoted by P and then hashes or encrypts P with

Table 5.2: Operations in PINTA.

Function	**Operation**
$FuzzyHash(a)$	Obtain the fuzzy hash of bytestream a, denoted by $\dot{a_F}$
$FuzzyCmp(\dot{a_F}\, \dot{b_F})$	Compare the distance between two fuzzy hash $\dot{a_F}$ and $\dot{b_F}$, return a value ranging from 0 to 100.
$FHE\_KeyGen()$	Generate a key pair (pk, sk) for fully homomorphic encryption.
$FHE\_Encrypt(a, pk)$	Encrypt a with public key pk via FHE, outputting $\dot{a_H}$
$FHE\_Decrypt(\dot{a_H}, sk)$	Decrypt $\dot{a_H}$ with public key sk via FHE, output a
$FHE\_Sub(\dot{a_H}\, \dot{b_H}, pk)$	Subtract $\dot{a_H}$ with $\dot{b_H}$ via FHE under public key pk
$FHE\_Div(\dot{a_H}\, \dot{b_H}, pk)$	Divide $\dot{a_H}$ by $\dot{b_H}$ via FHE under public key pk

the FHE public key pk. Then, the user ID and user-assigned password, denoted by uid and Psw, along with the cryptographic user profile, denoted by $\dot{P}$, are passed to the AS. The AS will interact with the UPDB and thus insert $\dot{P}$ into the user profile database. For each login attempt afterward, the individual will pass his uid, typed password, denoted by Psw' as well as the newly captured user profile in ciphertext, denoted by $\dot{P}'$. The AS is responsible for evaluating how much $\dot{P}'$ is different from $\dot{P}$ (hereinafter referred to as *distance*) and returning the corresponding authentication result denoted as $AuthResult$, a boolean value indicating authentication as *success* or *failure*. If $AuthResult$ is a success, the AS will send a content service ticket along with $AuthResult$ to the user, which contains a session ID and a timestamp. Any user holding a valid service ticket may initiate a service request to the content server.

The major challenge of the privacy-preserving multi-factor authentication system is how to preserve the privacy of the user profile from servers and any third party while enabling the server to determine the distance between user profiles. To achieve this, we use fuzzy hashing and fully homomorphic encryption techniques. The seven operation primitives used in PINTA are summarized in Table 5.2.

From Equation 5.1, we can derive operations (Circuit C in Figure 5.1) that are used by the server on ciphertext values ($\psi_1..\psi_k$) sent by the user when generating the ψ. Specifically, we implemented two simple arithmetic operations, *substraction* and *division* using the underlying And/Xor gates that were proven to be secure [CMNT11b]. Assuming that we have FHE key pair (pk, sk), $\dot{a_H} \leftarrow FHE\_Encrypt\ (a, pk)$, $\dot{b_H} \leftarrow FHE\_Encrypt(b, pk)$, $\psi_1 \leftarrow FHE\_Sub(\dot{a_H}, \dot{b_H}, pk)$, and $\psi_2 \leftarrow FHE\_Div\ (\dot{a_H}, \dot{b_H}, pk)$, we can show the existence of Equations 5.2 and 5.3 as follows:

$$FHE\_Decrypt(\psi_1, sk) = a - b \tag{5.2}$$

$$FHE\_Decrypt(\psi_2, sk) = a/b \tag{5.3}$$

5.3.2 User Profile Acquisition

To collect user profiles, we developed two user profile acquisition programs in C# and Python for Windows and Linux OSs, respectively.

Hybrid User Profiling Model

The program has three main steps as illustrated by the cascading blocks in Figure 5.3: *data summation, feature derivation*, and *hashing-encryption*.

1) *Data Summation*: The Data Summation block is responsible for collecting the user information in a *sliding window* - collection for some user information occurs continuously, and at the end of each sliding window period, the collected information is handed over to the Feature Derivation block, and Data Summation starts again. To minimize the development effort, we use several third-party tools (i.e., TSTAT [dT12]) to assist with data collection.

2) *Feature Derivation*: The Feature Derivation block receives the raw data from the previous block and extracts the required features. The derivation of some features might demand further calculation, like the packet interval and the keystroke press interval. After each feature is ready for processing, the Feature Derivation block passes the data to the next block.

3) *Hashing-Encryption*: The Hashing-Encryption block is responsible for generating a cryptographic profile based on all of the available features via fuzzy hashing and fully homomorphic encryption. The fuzzy-hashed user profile is denoted by $\dot{P}_F$

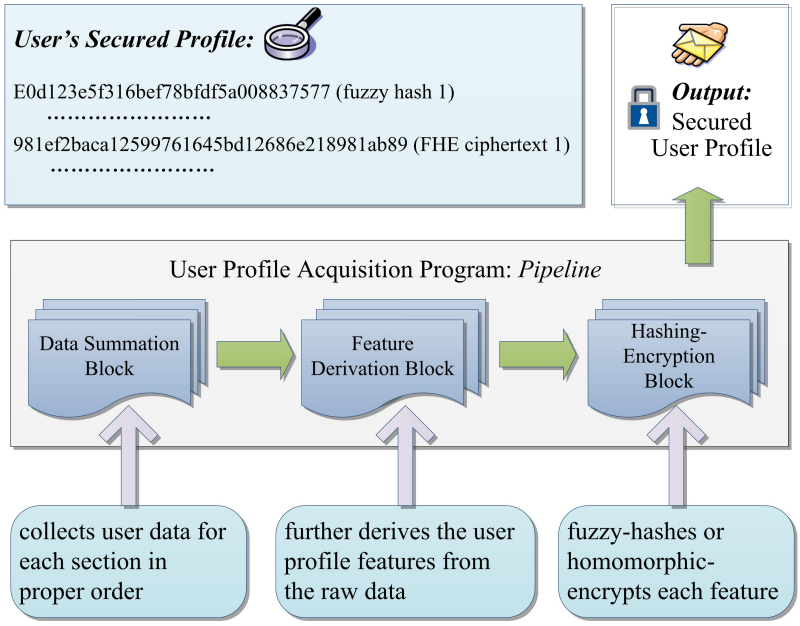


Figure 5.3: The Pipeline of User Profile Acquisition Program

Table 5.3: Features Used for User Profile Modeling

Category	Section	Number[1]	Type
Host-Based	1 File System and Registry	2	string
	2 Mouse Dynamics	3	number
	3 Keystroke Activity	2	number
	4 System Process	4	string
Network-Based	5 Browser Information	3	string
	6 Flow-Based Features	19	number

[1] Number of features in the section.

while the homomorphically encrypted user profile is denoted by $\dot{P}_H$. Thus, the block outputs a hybrid cryptographic user profile $\dot{P} \longleftarrow \{\dot{P}_F, \dot{P}_H\}$. In addition, the Feature Derivation Block continues to produce feature information at the end of every window period; therefore, the Hashing-Encryption block generates a new cryptographic profile based on the latest incoming feature information and overwrites the former one.

Furthermore, as illustrated in Table 5.3, the features are divided into two categories: host-based and network-based, which are further classified into several subcategories as follows.

1) *File System and Registry*: This sub-category consists of fuzzy hashes of the file hierarchy at a critical path and a portion of the registry contents (for Windows systems). In particular, we target the information pertaining to the installed software since, for most users, the installed software is relatively stable and is a good representation of user habits compared to other attributes. For the Windows OS, the folder path for installed software is typically 'C:\Program Files' and 'D:\Program Files' while for the Linux OS, it is '/usr/bin/'. Besides, the Windows Registry is a hierarchical database which stores configuration settings and options for both low-level OS components and high-level running applications, which reflects users' utilization of software well. In our case, we only retrieve the part of the registry which contains the information of the installed software.

2) *Mouse Dynamics*: User's mouse movements can be characterized via three fine-grained metrics: *direction, curvature distance*, and *curvature angle*. Nan Zheng et al. [ZPW11] proved that these three angle-based features are relatively unique from person to person and independent of the computing platforms and can, therefore, be used to distinguish legitimate users from intruders. To obtain a stable and representative sample, we use the average values of these three metrics in a time window as several features in the user's profile. According to [ZPW11], a reasonable choice of the window size is 20 mouse clicks, which takes up to approximately 3.03 minutes.

3) *Keystroke Activity*: User's keystroke activity is modeled via two features: the average key press-down time and the average time interval between key presses. Kevin Killourhy and Roy Maxion described in [KM09b] a method by which timing

Table 5.4: Notations Used

$\dot{P}$	Cryptographic user profile stored	$P_F \leftarrow \{p_{F_1}, ...p_{F_i}...p_{F_n}\}$	User's string features
$\dot{P}'$	Newly captured cryptographic user profile	$P_H \leftarrow \{p_{H_1}, ...p_{H_i}...p_{H_m}\}$	User's number features
n	Number of string characteristics	$W_F \leftarrow \{w_{F_1}, ...w_{F_i}...w_{F_n}\}$	Weight for each string feature
m	Number of number characteristics	$W_H \leftarrow \{w_{H_1}, ...w_{H_i}...w_{H_m}\}$	Weight for each number feature
pk	User's public key	$k_j \leftarrow \{0, 1...k_j\}$	Threshold for the jth number feature
sk	User's private key	$V_j \leftarrow \{0, 1...v_j\}$	ADV for jth number feature ($j \in [1, m]$)

data for keystroke activity of typing a password is collected and used to classify impostors from legitimate users.Their experimental results show that these two-time metrics are sufficient to represent different users. In our proposed system, the timing metrics of keystroke activity are captured and derived while a user is typing his password in order to log into the operating system.

4) *System Processes*: This sub-category is composed of fuzzy hashes of four clusters of system process names. The clustering strategy is alphabetically and then divide the names into four blocks.

5) *Browser Information*: In this sub-category, we utilize the auto-fill information in browsers. We derive the fuzzy hash of personal information with attributes of "Email", "Username" and "Address". The significance of auto-fill information is that one tends to have the same auto-fill value for those frequently-used attributes, in different browsers or hosts.

6) *Flow-Based Features*: We model users' general network behavior via 19 flow-based features. Note that it is commonly known that flow-based features indicate the category of network traffic (e.g., stream video, online chat) and thus serve a good reflection of user's network usage patterns. In our system, we use the average values of each feature, given a specific window of time.

We profiled users' network behavior via the network flow-based features listed in Table 5.5. Note that except 'flowDuration', each feature is measured on both client-to-server direction and server-to-client direction. Also, the value of each feature is

Table 5.5: Network flow-based features

No	Identifier	Definition	Weight
1,2	noPacket	Number of packets	0.012, 0.007
3,4	maxSegment	Maximum TCP segment	0, 0.001
5,6	maxWindow	Maximum TCP Window	0.001, 0
7,8	minWindow	Minimum TCP Window	0, 0
9,10	avgRTT	Average Round-Trip Time	0.045, 0.032
11,12	stddevRTT	Stddev of Round-Trip Time	0.003, 0
13,14	minTTL	Minimum Time-To-Live	0, 0.005
15,16	maxTTL	Maximum Time-To-Live	0, 0.005
17,18	avgInterPkt	Average Packet Arrival	0.134, 0.119
19	flowDuration	Flow Completion Time	0.087

based on one flow. A flow is defined as a sequence of packets sent from a particular source to a particular destination. Taking feature 'stddevRTT' as an example, it represents the standard deviation of all the round-trip times in that flow.

Algorithm for User Profile Generation

From Table 5.3, we observe that the user characteristics can be classified as two types: string (Sections 1, 4, 5) and numerical (Sections 2, 3, 6) values. Due to the respective working mechanisms of fuzzy hashing and homomorphic encryption, we apply the former on string characteristics and the latter on numerical ones. Suppose each user has n string characteristics as well as m number characteristics where each string feature is denoted by $p_{F_i}(i \in [1, n])$ and each numerical feature is denoted by $p_{H_j}(j \in [1, m])$. Also suppose that the user has already generated one pair of homomorphic encryption keys denoted by (pk, sk). The process of generating a cryptographic user profile is given in Algorithm 1. The notations used in Algorithm 1 are presented in Table 3. Note that the user profiling approach we propose is an extensible and configurable framework, which means that one can always edit the user profile by deleting or inserting new user behavioral features.

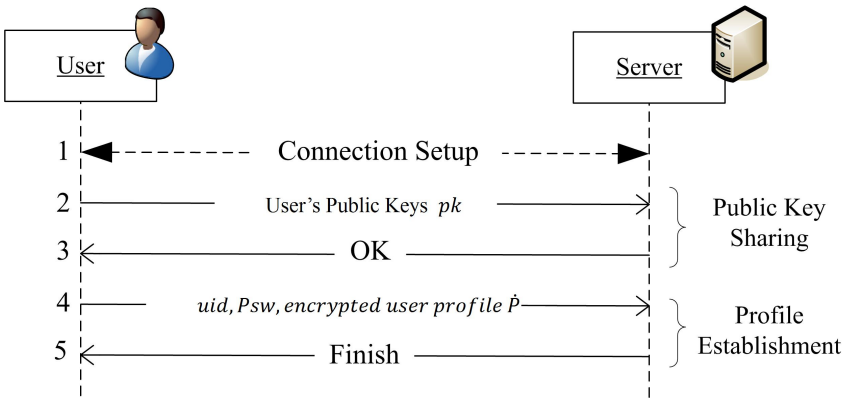


Figure 5.4: Sequence Diagram for User Enrollment

5.3.3 Enrollment of First-Time User and Profile Update

The sequence diagram of a first-time user's *enrollment* is shown in Figure 5.4. To initiate the *enrollment*, a user shares his public key pk with the authentication server (message 1). After receiving the server's acknowledgement (message 2), the user passes his uid, Psw along with his cryptographic user profile $\dot{P}$. Now, the server has the user's cryptographic profile stored in the database. As mentioned earlier, the user profiles change constantly. Thus, the cumulative change of user profiles makes it harder to recognize the distance between the newly captured profile and the original one stored in the database. To address this issue, the UPDB should update the cryptographic user profile for each user once every time period T. When updating a user's profile, the AS first authenticates the user as discussed in Section 4.4. After the user is validated, the profile update proceeds similar to the process of enrollment, which is shown in Figure 5.4. The system administrator is responsible for setting a reasonable T when deploying PINTA.

$$distance_j = \frac{|\dot{P}_{H_j} - P_{H_j}|}{P_{H_i}} \times 100 \tag{5.4}$$

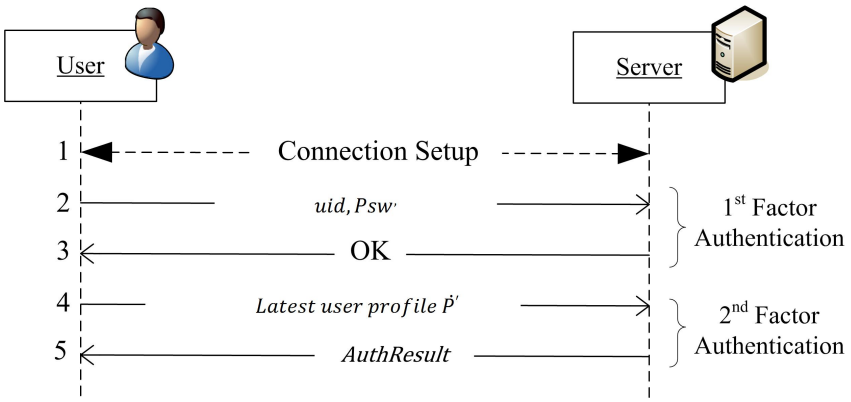


Figure 5.5: Sequence Diagram for Authentication

5.3.4 Server Authentication

The sequence diagram of the server handling each authentication request is presented in Figure 5.5. To initiate an authentication attempt, the user passes the uid, Psw' for the first-factor authentication (message 1). The failure of first-factor authentication terminates the conversation. If the user passes the first step of the authentication, he then passes his newly generated user profile, denoted by $\dot{P}'$ (message 3), to the server. After evaluating the distance between $\dot{P}'$ and $\dot{P}$, the server returns a boolean value $AuthResult$ (message 4), indicating the success or failure of the second-factor authentication. In order to yield $AuthResult$, we introduce a new concept, the *Accepted Distance Value (ADV)*. To define ADV, we first define *distance value*. Assuming P_{H_j} and $\dot{P}_{H_j}$ denote the values of the j^{th} feature in profile P and profile $\dot{P}$, the distance value between P_{H_j} and $\dot{P}_{H_j}$ is defined in Equation 5.4:

ADV is used as the array of all accepted distance values, which is assigned by the server for every numerical characteristic. In our system, for the j^{th} numerical feature, we propose a threshold k_j of the distance value. We define ADV V_j as the array of all integers ranging from 0 to k_j (i.e., $V_j \longleftarrow \{0, 1...k_j\}$). In addition, we place different weights on each string feature and each numerical feature, denoted by

Algorithm 1: *GenerateCryptographicUserProfile()*

Require: P_F, P_H, pk
Ensure: $\dot{P} \leftarrow \{\dot{P}_F, \dot{P}_H\}$
  for each $i \in [1, n]$ **do**
    $\dot{p_{F_i}} \leftarrow FuzzyHash(p_{F_i})$;
  end for
  for each $j \in [1, m]$ **do**
    $\dot{p_{H_j}} \leftarrow FHE\_Encrypt(p_{H_j}, pk)$;
  end for
  $\dot{P_F} \leftarrow \{\dot{p_{F_1}} ... \dot{p_{F_i}} ... \dot{p_{F_n}}\}$;
  $\dot{P_H} \leftarrow \{\dot{p_{H_1}} ... \dot{p_{F_j}} ... \dot{p_{F_m}}\}$;
  $\dot{P} \leftarrow \{\dot{P_F}, \dot{P_H}\}$;
  return $\dot{P}$;

Algorithm 2: *CalculateAuthenticationResult()*

Require: $\dot{P}, \dot{P'}, n, m, pk, W_F, W_H, V_j$
Ensure: $AuthResult$: truelfalse
  initial $Dis \leftarrow 0$
  for each $i \in [1, n]$ **do**
    $dis_{F_i} \leftarrow FuzzyCmp(\dot{p_{F_i}}, \dot{p_{F_i}}')$;
    $Dis \leftarrow Dis + dis_{F_i} * w_{F_i}$;
  end for
  for each $j \in [1, m]$ **do**
    $dis_{H_j} \leftarrow$
    $FHE\_Div(FHE\_SUB(\dot{p_{H_j}}', \dot{p_{H_j}}, pk), \dot{p_{H_j}}, pk)$;
    for each $v \in V_j$ **do**
      if $dis_{H_j} == FHE\_Encrypted(v, pk)$ **then**
        $dis_{H_j} = 1$; break;
      end if
      $dis_{H_j} = -1$;
    end for
    $Dis \leftarrow Dis + dis_{H_j} * w_{H_j}$;
  end for
  $AuthResult \leftarrow (Dis > 0)?true; false$;
  return $AuthResult$;

W_F and W_H, respectively. We will address the problem of how to reasonably set the threshold and weight for each feature in Section 5.3. The process of estimating profile distance and calculating $AuthResult$ is given in Algorithm 2 and the notations used are presented in Table 3.

5.4 Experimentation and Evaluation

In this section, we evaluate the feasibility of our proposed system through a series of experiments. We specifically conducted experiments with a combination of several public datasets and a dataset which we generated. Particularly, we evaluate PINTA in terms of authentication performance, the overhead caused on the system, and computational performance.

5.4.1 Datasets and User Profile Generation

The names of the public datasets used in our experiments are shown in Table 5.6.

Table 5.6: Data Source for Experiments

Section	Data Set	Subjects[1]
Mouse Dynamic	NSKEYLAB Dataset [She12]	10
Keystroke Activity	CMU Dataset [KM12]	51
General Network	DACS Dataset [BSPvdM12]	132

[1] number of subjects involved in the data set

A description of the datasets, along with a description of the necessary preprocessing approach for each dataset are as follows. 1) *NSKEYLAB Dataset* [She12] is a dataset containing mouse dynamics information from 10 subjects, each of who accomplishes at least 30 data sessions. Each session consists of about 30 minutes of a user's mouse activity in a free environment [SCG12, She12]. We derived three angle-based metrics using the approach proposed in [ZPW11], thus, generating three data points that represent the user's mouse movement profile. 2) *CMU Dataset* [KM12] is a dataset consisting of keystroke-timing information from 51 subjects (typists), each typing a password 400 times [KM09b, KM12]. 3) *DACS Dataset* [BSPvdM12] is a dataset consisting of the network trace for an educational organization. A 100 Mbit/s Ethernet link connecting the organization to the Internet was monitored to generate the trace. Each user at this site is assigned a fixed IP address [BSPvdM12]. We used SplitPcap [AB12] to obtain separate pcap files for each IP address inside the organization's network and further partition each pcap file into smaller pcap files with a period of 30 minutes. For each small pcap file, we ran TSTAT [dT12] to derive the flow-based features, thus, generating the user's network behavior profile. This data set spans two months.

In addition to these three datasets, we generated the dataset that contains *file system* and *registry* information as follows. First, we generated two *Software Pools*, A and B, which are two lists of software names with 30 and 100 types of software. Software Pool A represents the OS pre-installed software and common software,

while Pool B represents personalized software. For 15 users, we randomly chose 20 software names from Pool A and five from Pool B for each user. We fuzzy-hashed the list of the software names, thereby obtaining the first piece of the profile for each individual. Then, for each user, we randomly chose two uninstalled software packages from Pool B and inserted them into the user's original software list, thus generating a new piece of the user profile. We conducted this procedure iteratively for 30 times in order to generate a data set representing the "file system" with 15 individuals, each of which, has 30 profiles. We generated two software pools for two reasons: First, in the initial stage, users have similar software environments, in which most installed software is pre-installed with the OS by default and some are commonly used software. Second, the software environment for users tends to diverge afterward according to users' habits, interests, and occupations. In our approach, *Software Pool* A represents OS pre-installed and common software while *Software Pool* B represents personalized software.

Based on these four datasets, we generated hybrid user profiles for each user. As shown in Table 5.6, the NSKEYLAB Dataset only contains data from 10 participants. Therefore, we can at most generate user profiles for 10 distinctive subjects. To generate one piece of the hybrid user profile for a certain user, we first randomly chose four profiles, each from one dataset. Then, we combined 30-minute chunks from each of the four datasets to create 30 hybrid profiles belonging to one user. Hence, in our experiments, thirty 30-minute hybrid profile samples are created per user. The users are labeled as User A, User B, and so on. Finally, we eliminated the previously used profiles from the four datasets. By repeating this procedure, we generated 30 hybrid user profile samples for each of the ten users, yielding 300 different hybrid profile samples, as illustrated in Figure 5.6.

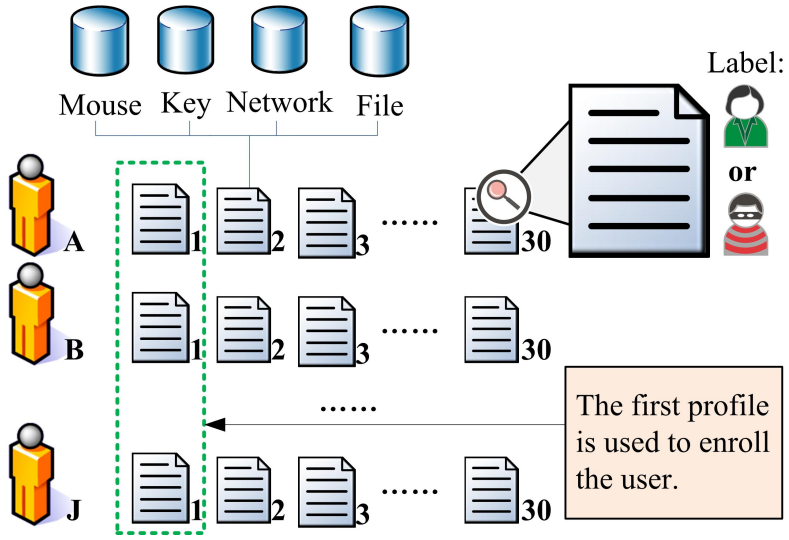


Figure 5.6: The Generation of Hybrid User Profile Samples

5.4.2 Experimental Setup

For each user, we registered his/her first piece of the cryptographic hybrid user profile in the database (the enrollment of that user). Then, we appended a label to each profile, indicating the owner of that profile. We randomly chose half of all the hybrid user profile samples and intentionally mislabeled them making the actual owner of that profile appears as an intruder. For the other half of the hybrid user profile samples, we correctly labeled them, treating the owner as a legitimate user. Finally, we randomly split all the labeled user profiles into two equal sized parts: the training set and the testing set. We encrypted each profile in the testing set via Algorithm 1 and left the profiles in the training set as plaintext (the data from this training set can result from a bank's focus group). The training set serves as a priori knowledge for threshold setting and weight adjustment while the testing set was the input of the system during experiments for performance evaluation. We first performed a threshold setting and weight adjustment based on the training set, as explained in Section 5.3. Then, we conducted a series of experiments by iteratively

comparing the result generated by Algorithm 2 with the actual situation for each piece of the user profiles. We conducted all the experiments on a standard laptop computer with Intel CPU i5-M430 (2.27GHz) and RAM of 4 GB.

5.4.3 Decision Process

To decide whether a hybrid user profile is legitimate, we must first determine appropriate thresholds for individual characteristics that comprise the profiles. Next, we used a majority vote to make the final decision on the legitimacy of a profile sample. A training set containing 150 labeled hybrid user profile samples served as a priori knowledge in this section.

Threshold Setting

We used the following approach to find an optimal threshold for each feature based on the a priori knowledge we had. Assume there are p features in each hybrid user profile and altogether q hybrid profile samples as prior knowledge. d_{ij} denotes the change percentage of the j^{th} feature in the i^{th} profile compared to the enrollment profile stored in the database. l_i denotes the identity of the owner of the i^{th} profile. $l_i = 1$ if the owner is legitimate and $l_j = 0$ if the owner is an intruder. th_j denotes the threshold for the jth feature. Thus, for the j^{th} feature, we design a function F of th_j:

$$\begin{aligned} F(th_j) &= l_1 * Sgn(th_j - d_{1j}) + ...l_i * Sgn(th_j - d_{ij}) \\ &+ ....l_n * Sgn(th_j - d_{nj}) \\ &= \sum_{i=1}^{q} l_i * Sgn(th_j - d_{ij}) \end{aligned} \tag{5.5}$$

where $F(th_j)$ represents the total number of correctly classified cases for the j^{th} feature and Sgn denotes the sign operation. Thus, the optimal th_j is the one that makes $F(th_j)$ reach a relatively maximum value. In this way, the threshold setting problem is transformed to a simple linear function maxima problem, which can be easily solved by Matlab.

Weight Adjustment

After having a vector of optimal thresholds for all the features, each feature is used as a voter, either voting for or against the user. The weight placed on each vote is adjusted via linear programming as described in Equation 5.6, where $v_j \in \{1, 0\}$ denotes the vote of the jth feature and $rst \in \{1, 0\}$ denotes the actual result.

$$min \sum_{j=1}^{p} w_j * v_j - rst, s.t. \sum_{j=1}^{p} w_j = 1 \tag{5.6}$$

After the implementation of linear programming with Matlab, we identified the features that were assigned far more weight than others, which indicates that those features better represent user behavior. See Appendix C for the weight adjustment results. Note that in the deployment of our system, the system administrator can always modify the weight adjustment based on his knowledge/experience. Additionally, he can eliminate the feature with little or zero weight and add new features to the system, which makes PINTA highly configurable and robust.

The result for weight adjustment is as follows: (1) For the mouse movement sub-category, the weights for *direction*, *curvature distance*, and *curvature angle* were 0.106, 0.035 and 0.004, respectively. (2) For the keystroke activity sub-category, the weights for *key press-down time* and *average key-press interval* were 0.048 and 0.023. (3) For the general network section, the weight for each feature is shown in Table 5.5. (4) For the file system and registry, the weights for the *file system* feature

and *registry* feature are 0.132 and 0.129. We have 26 features in total and the sum of the weights equals to 1.

5.4.4 Results

In this section, we evaluate PINTA in terms of authentication performance, system overhead, computational performance, and present our results.

Authentication Performance

In order to evaluate the PINTA's authentication performance, we used recall, false positive rate as a metric. We define recall and false positive rate as follows:

- *Recall*: Recall is the proportion of positive cases that are correctly identified and was calculated using $\frac{TP}{TP+FN}$. TP denotes True Positive, and FN denotes False Negative while in our authentication system, an intruder is labeled as *Positive* and a legitimate user as *Negative*. Thus, recall indicates the authentication system's ability to identify intruders.
- *False Positive Rate (FPR)*: FPR is the proportion of negatives cases that are incorrectly classified as positive and was calculated using $\frac{TN}{FP+TN}$. TN denotes True Negative while FP denotes False Positive. FPR refers to the probability of the system's falsely rejecting a legitimate user.

We conducted a series of experiments on the testing set containing 150 labeled hybrid user profile samples within different time window sizes for data collection and with or without weight adjustment. The experimental results in terms of recall and false positive rate (FPR) are presented in Table 5.7.

From the table, it can be seen that the longer the time window for the data collection, the better the system performance is in terms of recall and FPR. When

Table 5.7: Experiment Results: Recall and FPR

Time For Data Collection	Weight Adjustment	Recall (%)	FPR (%)
5 min	Without	74.2	26.9
	With	75.1	24.3
10 min	Without	75.4	19.7
	With	77.1	19.3
20 min	Without	78.8	16.2
	With	79.9	15.1
30 min	Without	80.8	14.7
	With	82.7	13.2

Table 5.8: Average timing results

	GenerateCryptographicUserProfile		*CalculateAuthenticationResult*			
FHE_KeyGen	*FuzzyHash*	*FHE_Encrcypt*	*FuzzyCmp*	*FHE_Sub*	*FHE_Div*	*FHE_Decrypt*
1449 microsecs	677 microsecs	47784 microsecs	57 microsecs	1392 microsecs	370728 microsecs	29040 microsecs
	Subtotal = 0.05 seconds			**Subtotal = 0.37 seconds**		

the time window is as long as 30 minutes (initial bootstrap latency - this drops to 0 for every attempt after 30 minutes), we achieve an optimal result with recall of 82.7% and FPR of 13.2%. Nevertheless, a longer data collection time tends to reduce the usability of the system. Therefore, there is a trade off between system accuracy and efficiency. Also, it can be observed that weight adjustment makes a positive impact on the system performance via a relatively higher recall and a relatively lower FPR.

System Overhead & Utilization

In order to evaluate the overhead caused by PINTA on the system and the utilization of PINTA, we computed packet size, system overhead, and resource utilization like CPU and RAM utilization. We define and calculate them as follows:

- Size of Authentication Packets (hereafter referred to as *packet size*): Packet size is the size of all the authentication information transmitted from client to the server.
- *System Overhead*: System Overhead is the initial latency during the entire multi-factor authentication process. Measuring total system overhead is a complex task. In Section 5.4, we evaluated the overhead introduced by user profile acquisition, cryptography, data transmission, and server processing.
- *Resource Utilization*: Resource Utilization is defined as how much system resources it takes for a user information acquisition program to retrieve information and derive features continuously. We evaluated it in terms of CPU and RAM utilization.

In terms of packet size, for the first-time user enrollment, the packet size is the sum of the size of the user's public key pk, uid, Password Psw, and cryptographic user profile $\dot{P}$. Since the size of uid and Psw are far less than the size of the rest, we can neglect uid and Psw in the calculations. Packet size, $Size$, was calculated using Equation 5.7, in which $pkSize$ denotes the size of public key pk, n denotes the number of string features, α denotes the size of each fuzzy hash, m denotes the number of number features, and β denotes the size of each FHE ciphertext.

$$Size = pkSize + n * \alpha + m * \beta \tag{5.7}$$

In PINTA, with $pkSize = 128KB, n = 9, \alpha = 0.125KB, m = 24$, and $\beta = 128KB$, the packet size for the enrollment is approximately 3201 KB (message 1 and message 3 in Figure 5.4). In the authentication procedure, the user is not expected to transmit their pk again; so, the packet size is around 3201 KB after enrollment (message 3 in Figure 5.5).

System overhead, T, is comprised of the time spent on authentication data transmission, T_t and the server process, T_s. We can derive T using Equation 5.8. T_t largely depends on the network environment. In a high-speed Internet environment, T_t is normally under 30 seconds. T_s was less than 5 seconds in our experiments. In total, the system latency after users initiate a login request is around 35 seconds. It is worth mentioning that system overhead may vary because it is dependent on the computing ability of both the client and the server sides and also to the specific network conditions.

$$T_{after} = T_t + T_s \tag{5.8}$$

In terms of resource utilization, in our experiments, the running of the user profile acquisition program takes less than 3% of CPU resources and about 15MB of RAM. The most significant proportion of computing resource consumption stems from the capturing of network packets. Because we only capture packet headers, the resource demanded for network monitoring is still in an acceptable range. Note that the statistic for resource utilization is based on a laptop computer with Intel CPU i5-M430 (2.27GHz) and a RAM of 4 GB. The CPU utilization percentage will decrease with a more powerful machine.

Computational Performance

Finally, we measure PINTA's computational performance. For this purpose, we used the timing results of two main algorithms: $GenerateCryptographicUserProfile()$ and $Calcu\ lateAuthenticationResult()$. Moreover, we also computed the timing results of operations in Table 5.2.

$CalculateAuthenticationResult$ includes subtraction and division operations. While performing the subtraction, while we used the built-in subtraction function

in the homomorphic library, the division is implemented by multiplying the reverse of the second multiplier. Therefore each division is indeed the combination of encryption and multiplication.

The results are given in Table 5.8. According to the results, while the generation of cryptographic profiles takes around 0.05 seconds, the calculation of authentication result takes about 0.37 seconds. When we look at the operations performed in these algorithms, we see that the division operation dominates the authentication result calculation algorithm. That's because the division operation is a multiplication operation and the homomorphic multiplication operations are computationally much more expensive than the addition. However, still, the results show that PINTA is feasible in terms of computational feasibility.

5.4.5 Security Analysis

In this subsection, we demonstrate how the authentication system thwarts the adversaries discussed in Section 5.2.1.

1) *Security Against Brute-Force Attacker*: In this case, an adversary attempts to log in by guessing a user's profile, which, in the worst case, would involve traversing the entire message space. For each fuzzy hash, the message space is 2^{128}. For each FHE, the message space is $2^{1.5 \cdot 10^5}$. Since the message space for the fuzzy hash is arbitrarily small compared to that of FHE ciphertext, we only consider the computational cost to brute force the FHE ciphertext. First, denoting the length of ciphertext by l, the probability of a malicious code correctly forging one correct feature is:

$$P_{forge} = \frac{1}{2^l} \tag{5.9}$$

In our system, we realize $l = 1.5 \cdot 10^5$, so P_{forge} is $\frac{1}{2^{1.5 \cdot 10^5}}$. Second, an adversary that is brute-forcing the FHE ciphertext for each feature will have to put an overall effort of

$$\Psi = \frac{\gamma}{\kappa} \tag{5.10}$$

where γ is the average number of possible ciphertext values (e.g., $2^{1.5 \cdot 10^5}$) and κ is the frequency of the attacker's computer at 1 *attack/sec*. Assuming an attacker with computational resources such as Intel Core i7-3960 at 3.9GHz, approximately $2^{1.499 \cdot 10^5}$ years of Ψ would be needed to generate the correct ciphertext. Therefore, it is safe to conclude that it is impossible for a computationally bounded adversary to match a user's profile by brute-force.

2) *Security Against Honest-But-Curious Server Attacker*: In this scenario, the *honest-but-curious server* tries to reverse engineer a user's cryptographic profile $\dot{P}'$. The capability of resisting reverse engineering is also referred to as *semantic security* in cryptology [GM84]. Informally, a traditional definition of *semantic security* is that a system is semantically secure if any computationally bounded adversary is not able to compute the plaintext even with the knowledge of both ciphertext and the corresponding public key. Note that in our work, we also treat fuzzy hashing as a special form of encryption. Because each $\dot{P}'$ comprises a piece of fuzzy-hashed user profile $\dot{P_H}'$ and a piece of homomorphically encrypted user profile $\dot{P_F}'$, we demonstrate the semantic security of fuzzy hashing and FHE, respectively. For fuzzy hashing, the only available information available to the *honest-but-curious* [AMN18] server or any third-party adversary are the two fuzzy hashed values (after the initial authentication attempt) and the only answer the server can obtain is the similarity between two fuzzy hashes, which is just as intended. No further information can be derived from the hash values due to the one-way property of fuzzy hash-

ing [Kor06]. For FHE, Gentry and Coron showed that an FHE system is semantically secure [Gen09, CMNT11b], which means it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public key.

5.5 Conclusion

In this chapter, we designed a privacy-preserving multi-factor authentication system, called PINTA. In PINTA, while the first authentication factor is a password and the second one is a hybrid behavioral user profile. PINTA focuses on the privacy preservation of the second factor, which has two advantages over previously proposed systems. First, user privacy is not leaked to the authentication server. We have proven that the system is secure against both a brute-force matching attacker and an honest-but-curious attacker. Second, the hybrid user profiling model is highly usable and configurable. One can always modify the feature list for user profiling in PINTA according to the actual circumstances. We evaluated the system performance via a series of experiments, resulting in an optimal recall of 82.7% and FPR of 13.2%. In addition, we also show that PINTA's system overhead is within the acceptable range.

PART II - SMART HOME USER PRIVACY

CHAPTER 6

PEEK-A-BOO: REVEALING THE USER ACTIVITIES VIA MULTI-STAGE PRIVACY ATTACKS

6.1 Introduction

Previously, the Internet was mainly used for accessing and displaying content of web pages (i.e., web browsing). However, with the emergence of IoT devices in smart homes, users have now the ability to control their home's electronic systems (e.g., smart bulbs, smart locks, sensors, etc.) using appropriate smartphone apps and also from remote locations [SAA+18, BSAU18, BSAU19, SBC+19]. To realize smart home automation, the devices are mostly equipped with embedded sensors. These sensors collect data from the environment and help users to control them. Moreover, smart home devices are also continuously communicating with associated back-end system servers or other devices (e.g., smart hubs) to transmit the sensor data in a real-time manner. On the other hand, as IoT devices usually are single-purpose devices, the capabilities of individual smart home devices are relatively limited, comprising only a few states or actions. For example, a motion sensor allows a user to detect any movement in a physical space, but the sensor has only two states: motion and no-motion. If an attacker can reveal the current state of the sensor, the attacker will also reveal the presence of the user at home.

In this chapter, we demonstrate how machine learning methods based on traffic profiling of smart home IoT device communications can be used by an adversary to automatically identify actions and activities of the IoT devices and its users in a victim's smart home with very high accuracy, even if only encrypted data are available. Indeed, device types, daily mundane activities of the users (e.g., left home, walking from kitchen to bedroom), or states of the devices (e.g., door locked, unlocked) can

all be easily identified even if the traffic is encrypted, thus posing a threat to user privacy. We refer to this novel attack to user privacy as *multi-stage privacy attack*, which is achieved in a cascading style by only observing passively the wireless traffic from smart home devices. In this, a passive attacker can easily realize the multi-stage privacy attack to extract meaningful data from any smart environment equipped with smart devices including personal homes, residences, hotel rooms, offices of corporations or government agencies. Here, unlike earlier approaches, the presented attack is device-type and protocol-agnostic, making it easily applicable to a wide variety of different IoT device types without the need for tedious harvesting of device-type or protocol-specific knowledge about specifications for supporting the activity identification task.

We evaluate the effectiveness of the novel multi-stage privacy attack with 22 different off-the-shelf IoT devices utilizing the most popular wireless protocols for IoT. Our experimental results show that an attacker can achieve very high accuracy (above 90 %) in identification of the types, actions, states, activities of the devices and sensors. Moreover, to counter the identified privacy threats posed by the multi-stage privacy attack, we also designed a new effective countermeasure solution based on generating spoofed traffic to hide the real states of targeted IoT devices and thereby the real activities of the users. Our solution does not require modifications in targeted IoT devices and is, therefore, easier to deploy than previously proposed solutions for IoT devices, for which it is very difficult to implement client-based countermeasures due to the vast heterogeneity of smart devices and limited resources available on the IoT devices. Also, even if the user is not at home, a fake traffic-based solution for the user's presence will mask the user's absence, further improving privacy.

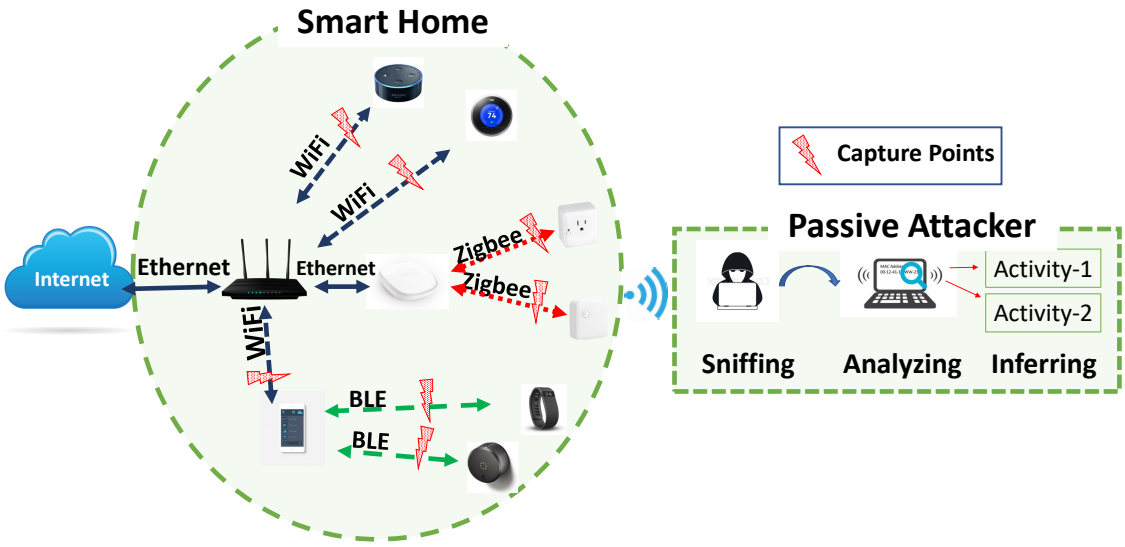


Figure 6.1: Local adversary model considered in this work.

6.2 Adversary Model

One of the unique challenges in the domain of IoT, and particularly smart home, is that the attack surface is naturally extended and comprises a diverse set of devices and sensors deployed at the user's home. Figure 6.1 shows some of the data capturing points that an attacker can take advantage of when inferring user activities. In this work, we consider a local adversary located physically within the wireless range of the targeted user's smart home devices similar to [Fea16a, Fea16b, HCS+18]. For this, the attacker can install the sniffers only once and even manage them remotely. Or, it could compromise a device inside the smart home, remotely, and turn it into a sniffer. In this way, the attacker may never need to be present. In all these cases, the adversary can eavesdrop on various wireless IoT network communications transmitted by the user's smart home devices. For example, as presented in Figure 6.1, the attacker can sniff all the network traffic transmitted over WiFi, BLE, and ZigBee protocols. The attacker only needs to passively sniff the network traffic and does not need to interrupt. Therefore, the attacker may stay

active long enough without being detected by the victim. An alternative adversary would be an adversary who can launch the attack remotely, i.e., intercepting the network traffic over the Internet such as a malicious ISP. We further discuss the advantages and limitations of such an adversary in Section 6.7.

Assumptions. We further make the following assumptions:

- The attacker has access to the same kind of smart home devices and sensors as the targeted user, s/he can analyze the devices by collecting the traffic of these devices, and use the collected data to train its algorithms.
- The attacker has access to protocol headers data on all layers that are not protected by encryption. The attacker does not need to know the specifications of analyzed protocols, instead it only needs to know how to run the already publicly available scripts, which does not require an extensive knowledge about the specifications of the protocol itself. Moreover , it can also use Layer 2 information like MAC addresses, or BLE advertisement packets, to automatically identify additional information, the brand of individual devices, thereby reducing the search space of devices to guess the set of smart home devices that the targeted user is using. Moreover, it is also worth noting that the attacker does not need exactly same devices to train its algorithm, but it needs exact brand and device type to get the results presented in this work as we use the $< brand, device - type >$ pair to uniquely identify devices.

Attacker's goals. We model the attacker's goals under four different categories:

- Goal-1: The attacker aims to infer the devices used in a smart home. (Section 6.5.5)
- Goal-2: The attacker aims to infer the daily routine of the user. (Section 6.5.6)

- Goal-3: The attacker aims to infer the state of a specific smart home device. (Section 6.5.7)
- Goal-4: The attacker aims to infer specific user activities from the states of multiple devices. (Section 6.5.8)

6.3 Smart Home Devices

In this section, we describe the typical characteristics of smart home devices relevant to this chapter. First, we classify the smart devices according to their capabilities. This capability-based classification can also be used to classify the device actions. Second, we present required background information about the communication protocols used by these devices.

6.3.1 Capabilities of Smart Devices

We categorize smart devices in our study into three categories in terms of their capabilities. The first category is the *Hub-like devices*. They are central communication hubs that connect other devices to both each other and to the Internet. They mostly do not provide a functionality of their own to users as their main purpose is to act as gateways connecting devices using other protocols than WiFi to the smart home network. In some cases, like the Samsung ST Hub, they serve as a centralized platform to install and run smart home apps for different smart devices. The second category of devices is *User-controlled devices*. These devices can be controlled by their users either manually or via a controller device like a smartphone or tablet. Examples of such devices include Smart Lights, Smart Switches or Smart Locks. These devices can be controlled both remotely and locally by the user. The third category is *Sensor-like devices*. These devices are the most primitive ones and have

Table 6.1: The communication protocols and capabilities of the smart home devices used.

		Communication			Capabilities		
ID	**Device**	**WiFi**	**ZigBee**	**BLE**	**Type-I**	**Type-II**	**Type-III**
1	ApexisCam	●	○	○	○	○	●
2	AirRouter	●	○	○	●	○	○
3	AugustSmartlock	○	○	●	○	●	●
4	BelkinWemoLink	○	○	○	○	●	○
5	DLinkCam	●	○	○	○	○	●
6	DLinkDoorSensor	●	○	○	○	○	●
7	DLinkMotionSensor	●	○	○	○	○	●
8	DLinkSiren	●	○	○	○	○	●
9	EdimaxCam	●	○	○	○	○	●
10	EdimaxSPlug1101	●	○	○	○	●	○
11	EdinetCam1	●	○	○	○	○	●
12	EdinetGateway	●	○	○	●	○	○
13	FitbitAria	●	○	○	○	●	○
14	Lightify2	●	○	○	○	●	○
15	PhilipsHueBridge	●	○	○	●	○	○
16	SMCRouter	●	○	○	●	○	○
17	STMotionSensor	○	●	○	○	○	●
18	STOutlet	○	●	○	●	●	○
19	STMultiSensor	○	●	○	○	○	●
20	TPLinkHS110	●	○	○	○	●	○
21	WansviewCam	●	○	○	○	○	●
22	WemoInsightSwitch	●	○	○	○	●	○

Type-I: Hub-like devices, Type-II: User-controlled devices, Type-III: Sensor-like devices

only the capability of sensing the environment via their built-in sensors. An example of this type of device is the Samsung ST Motion Sensor, which can detect persons moving in its proximity. These devices send notification messages to their associated services either when an event takes place, or periodically. All the devices studied in this chapter are shown in Table 6.1.

Apart from these devices, a typical smart home environment uses a smartphone or tablet as a controller device to control smart home devices. The smartphone or tablet can also be used as an interface to connect smart devices and smart home hubs and install different apps on the devices. We consider the smartphone or tablet as the controller device in the user activity inference.

6.3.2 Communication Features

Both the smart home vendors and users mostly prefer wireless communication over wired communication as it is more convenient. However, compared to wired communication, the wireless network traffic from smart home devices is open to the eavesdropping attacks.

In this work, we target three wireless protocols: WiFi, ZigBee, and Bluetooth Low Energy (BLE). Among these, WiFi is used in the wired or plugged-in devices, while other protocols, ZigBee and BLE, are implemented for short range communication tasks of battery-powered devices as they consume less power than WiFi.

WiFi-enabled devices

WiFi-enabled devices are connected to the Internet either through a Hub-like device or directly connected to an access point. In both cases, the adversary can track and capture the traffic through a specific device via MAC address. Even though

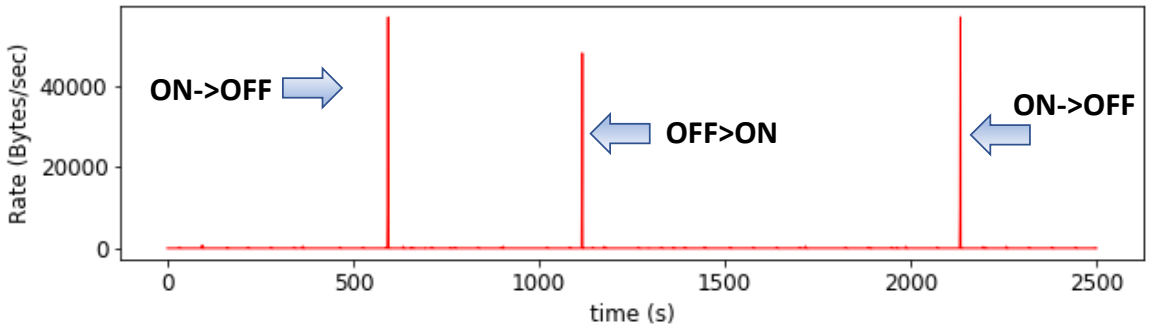
MAC addresses may help the attacker to narrow down the device type, it can not precisely decide the device type from MAC address. It may want to use IP addresses of servers. However, the adversary can only see the traffic that is encrypted by both the network protocols (SSL/TLS) and WiFi encryption (WPA). Therefore, it cannot see the IP or transport layer headers encrypted by the WPA protocol. This prevents the attacker from using header-based features for the device identification. However, the traffic rates of the devices still cannot be hidden from the attacker.

ZigBee-enabled devices

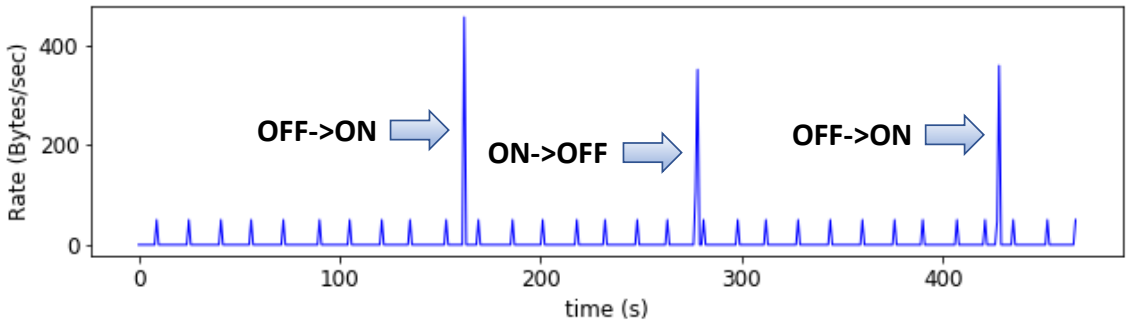
ZigBee devices have two addresses: MAC address and Network Address (NwkAddr). The MAC address is exactly the same as the MAC used in WiFi-enabled devices, which is unique for every device in the world and never changes. On the other hand, NwkAddr is created and assigned when the device joins a network and changes when it leaves and re-joins another network. It is similar to IP, however, it is not encrypted and source and destination NwkAddr of the packets can be seen by the attacker. In addition, the network coordinator (i.e., hub) has the $0x0000$ address and each network has a unique identifier, called the Personal Area Network Identifier (PAN ID). This information may additionally help the attacker.

BLE-enabled devices

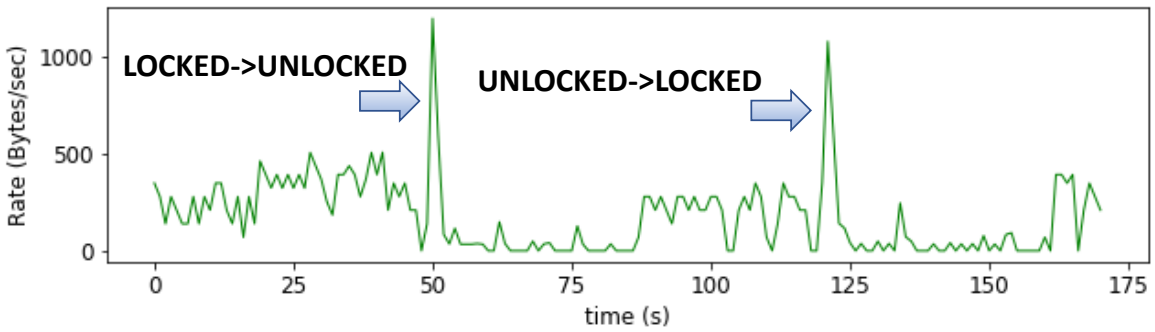
In a BLE network, a device can be either a master or a slave. A slave can connect to only one master node while a master can connect to multiple slave nodes. In all the smart home devices that we used, while the smartphone acts as a master, targeted smart device acted as a slave. Before establishing the connection, a slave device broadcasts advertising packets (ADV_IND) randomly on channel 37, 38, and 39. Once a connection starts, they agree on a channel map, where they follow in the



(a) Wemo Insight Switch (WiFi)



(b) Samsung SmartThings Outlet (ZigBee)



(c) August Smart Lock (BLE)

Figure 6.2: The traffic rates of (a) Wemo Insight Switch, (b) Samsung ST outlet, and (c) August Smart Lock. Here, a number of actions are illustrated, with many signals easily discerned by the naked eye. For instance, when the lock is turned on, the significant amount of packets are transmitted and received, which creates a peak in the traffic rate for a certain duration.

rest of the communication. If an attacker wants to follow the BLE traffic through a smart device, it needs to capture the first packet so that it can learn the channel mapping. Once the attacker captures the access address, it can follow the rest of the communication.

6.4 Case Studies

In this section, we show the feasibility and possibility of privacy leaks from encrypted network traffic of smart home devices. We show that an attacker who can sniff the network traffic of the devices can easily infer some simple information without using any advanced techniques. We consider one device for each protocol: Wemo Insight Switch (WiFi), Samsung ST Outlet (ZigBee), and August Smart Lock (BLE). We analyze the raw network traffic of each device and see if it is really possible to extract information from the network traffic, specifically from data rate.

Wemo Insight Switch (WiFi)

Wemo Insight Switch is a Wifi-enabled device and used to monitor and control other appliances (e.g., smart light) from a smartphone. It has only two capabilities: ON and OFF. Figure 6.2a shows the data rate of the sample traffic collected from Wemo Insight Switch, where we illustrated a number of actions of the user to change the state of the device. As can be seen from the figure, the data rate shows a significant increase when the device state is changing. Therefore, the data rate clearly reveals the device state changes. In the first peak, the device's state is changed by the user, i.e., the device is turned on and in the second peak, the user turned off the device and so on.

Samsung ST Outlet (ZigBee)

Samsung SmartThings (ST) Outlet uses ZigBee protocol to communicate with Samsung ST Hub. It can also act as a repeater and repeats the broadcast packet of Hub for the smart devices, which is not in the range of Hub. This increase the range of Hub. Other than repeating Hub's broadcasting packets, it has only two capabilities: ON and OFF. The traffic rate of a sample network capture of Samsung ST Outlet is plotted in Figure 6.2b. In the given sample network traffic, the device's activity has been changed by the user three times, which clearly corresponds to the three large peaks. On the other hand, small peaks correspond to the repeating of the broadcast packets of the hub, which is periodic with 15 seconds.

August Smart Lock (BLE)

The August Smart Lock communicates with the user's smartphone via BLE. In addition to locking and unlocking from the app on the smartphone, the owner (main user) can also give access to guest users through the web servers. The user can also enable the auto-unlock, where the lock is unlocked when the user is in range. However, the lock itself does not have the remote control capability. For remote access, it needs other accessories (e.g., WiFi bridge). Here, we only consider the BLE communication between the lock and smartphone. Figure 6.2c shows the plot of the sample packet capture of August Smart Lock. As in the previous case studies, the transition between the device's actions can be clearly identified by the attacker. The small increase in the traffic rate in the first part of the capture is because of the advertising packets.

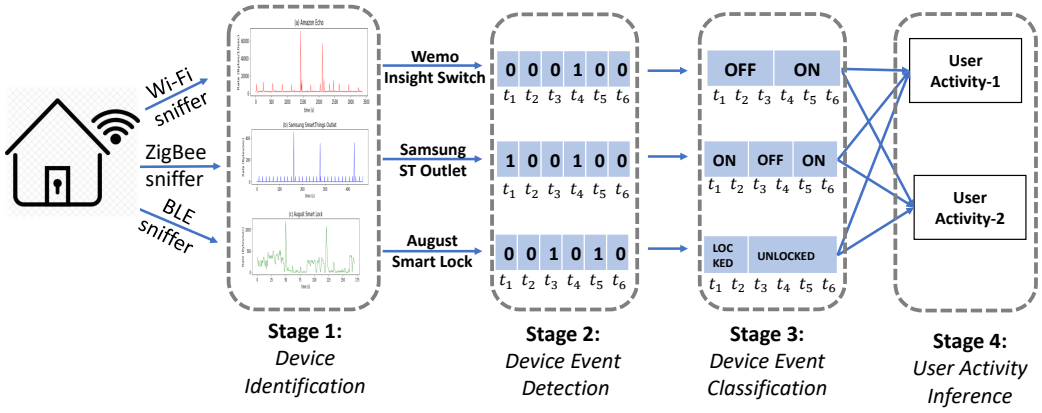


Figure 6.3: Overview of our multi-stage privacy attack.

6.5 Multi-stage Privacy Attack

As shown in Figure 6.3, our novel multi-stage privacy attack consists of four stages connected in a cascaded manner. While the goal of the attack is to infer user activities at the final stage, every stage also leaks partial information about devices and their actions and can be independently used by the attacker for various purposes. In the following, we first outline the high-level overview of the attack and then present details of individual stages and related results.

6.5.1 Attack Stages

Stage-1: In the first stage, the attacker's goal is to identify the type of each smart home device. Even though used protocols use unique identifiers for each device (e.g., MAC address, NwkAddr), the attacker does not know the device type a specific address corresponds to. By sniffing packets of individual protocols, the attacker will obtain network traffic profiles of all devices using that protocol. Identifying individual devices' types becomes then a multi-class classification task based on the traffic profiles of individual devices.

Stage-2: After discovering the types of individual devices, the attacker's goal is to infer the state of individual devices. As shown in Figure 6.2, a state change typically results in a significant increase in network traffic related to the device, causing an increase in the data rate and decrease in the inter-arrival time of the packets. Therefore, the attacker can in most cases detect state changes of devices by observing changes in these metrics. At the end of this stage, as shown in Figure 6.3, the attacker converts the network packets into 1s and 0s, where the 1s show where the transition occurred.

Stage-3: After detecting transitions between device states, the attacker splits the network trace of a device into segments corresponding to different device states (e.g., ON, OFF). Identifying these states is then reduced to a multi-class classification problem, where classes represent possible device states.

Stage-4: In this stage, by using the results of the state classification in Stage-3, the attacker knows the inferred states of all devices. For example, at a particular moment, the attacker may know the smart lock is in the LOCKED state, no motion is detected in the motion sensor placed in the kitchen and so on. Using the state information of the devices, the attacker can guess that the user is sleeping. Any user activity in a smart home can be inferred by observing the inferred states of devices and sensors and using a Hidden Markov Model to infer the corresponding user activity.

In the next sections, we evaluate the efficiency of our multi-stage privacy attack on network traffic data collected from 22 different off-the-shelf IoT devices used in smart homes.

Table 6.2: Characteristics of network traces used in experiments.

Device	**Period (mins)**	**Size (MB)**	**Packets**
ApexisCam	133	80	152220
AirRouter	85	49	115192
AugustSmartLock	25.8	0.66	8129
BelkinWemoLink	71	0.66	2039
DLinkCam	225	1.15	5389
DLinkDoorSensor	74	0.48	3519
DLinkMotionSensor	74	0.47	2849
DLinkSiren	71	0.41	3073
EdimaxCam	225	0.27	1798
EdimaxSPlug1101	74	0.5	2823
EdinetCam1	117	0.3	2779
EdinetGateway	225	0.34	3240
FitbitAria	213	0.043	257
Lightify2	74	0.25	1022
PhilipsHueBridge	53	0.8	2680
SMCRouter	124	47	150768
STOutlet	6	0.04	1061
STMotionSensor	11	0.05	1291
STMultiPurpose	12	0.22	5255
TPLinkHS110	71	0.14	473
WansviewCam	193	11	73759
WemoInsightSwitch	117	0.8	1675

6.5.2 Dataset and Evaluation Metrics

In order to evaluate the attacks in the stages above, we collected the network data from 22 different smart home devices. Data collection was performed in two stages: In the first stage, controlled experiments were performed in which detailed instructions were followed to initiate specific actions on the tested device. These instructions were compiled based on the on-line or hardcopy manual of each tested device (specs and data sheets). The controlled experiments were performed in order to ensure that all relevant actions for each device were represented in the usage dataset sufficiently many times. Each experiment was therefore repeated $n = 20$ times for each device. In addition to the controlled experiments, also uncontrolled testing was performed in order to capture background traffic of relevant devices. In this set-up, several devices were configured to be used simultaneously and device actions were occasionally triggered during a test period of ca. 1-2 hours.

The duration and the total size of the captures and the number of the packets are given in Table 6.2. The devices used include a representative cross-section of IoT device types, typically available in the European and North American markets during the study. The devices were also selected based on the market share of different device categories. The most popular device categories are smart security systems such as smart cameras and smart locks (22.2%), lighting (3.03%), outlets and switches (1%), gateways including hubs and routers (24.5%), and smart speakers (22.39%) [Ana17]. In addition to these categories, we also included several smart sensors as these devices hold significant smart home market share (approximately 23.9%) [Int18]. We installed all the devices in a laboratory network and emulated user inputs triggering device state changes. We captured all the network traffic from a device and performed the analysis offline.

For evaluating the efficiency of our attacks, we use different metrics. First, we use accuracy, which is the ratio of correctly inferred observations to total observations. In some cases, as in real deployments, the collected network data may have imbalanced data, where the duration of the active state is much less than the inactive one. In those cases, we use additional metrics such as Precision, Recall, F1 score, and Support. In the cases that the dataset includes a lot more label 0 (no activity) rows than label 1 (activity) rows, we observed that F1 score is a better performance measurement than accuracy although accuracy is a more intuitive performance measurement, in general.

6.5.3 Performance Metrics

To evaluate our proposed novel attack, we used seven different performance metrics: True Positive Rate (TPR), False Negative Rate (FNR), True Negative Rate (TNR), False Positive Rate (FPR), Precision, Accuracy, and F1-score. These can be calculated using following equations:

$$TPR\,(Recall) = \frac{TP}{TP + FN} \tag{6.1}$$

$$FNR = \frac{FN}{TP + FN} \tag{6.2}$$

$$TNR = \frac{TN}{TN + FP} \tag{6.3}$$

$$FPR = \frac{FP}{TN + FP} \tag{6.4}$$

$$Precision = TP/(TP + FP) \tag{6.5}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.6}$$

$$F1 - score = \frac{2 * TP * TN}{TP + TN} \tag{6.7}$$

where $TP = True\ Positive$, $FP = False\ Positive$, $TN = True\ Negative$ and $FN = False\ Negative$.

6.5.4 Calculating Features from Network traffic

In this sub-section, we explain how we use the traffic flow for the classification task. Particularly, we take advantage of the fact that while the encryption layer in the protocol protects the payload of a packet, it fails to hide other information revealed by network traffic patterns, for instance, sequence of packet lengths (SPL) and direction (incoming/outgoing). We consider each network traffic flow as a time ordered sequence of packets exchanged between two peers during a session. Before processing the network traffic for classification, we converted packet in traffic flow into a Sequence of Packet Lengths and Times (SPLT) as in following format:

$$pkt = [timestamp,\ direction,\ packet\ length], \tag{6.8}$$

where the direction is 1(0) if it is an incoming (outgoing) packet. This transformation is done for each packet in the captured trace, where each result is written to a new row. In the end, we obtained a matrix with three columns. Then, in the feature extraction of each attack, we calculated the features from this matrix.

6.5.5 Stage-1: Device Identification

Several different identification approaches for IoT devices have been proposed in literature. Numerous works have shown that IoT devices can be identified with high accuracy for both WiFi-enabled [MMH+17, DJ17, BBP+18, MBS+17, NMM+19] and BLE-enabled [DPCM16] devices. Therefore, in this section (e.g., Stage-1), we implemented already existing device identification algorithm for ZigBee-enabled smart home devices using our features to see whether we can identify the ZigBee-enabled smart home devices from their network traffic.

In our dataset, each device can be uniquely identified by the $< brand, device - type >$ pair. We did not consider the different models of devices as different devices.

On the other hand, a hub in ZigBee always uses the network address $0x0000$, so it can be easily recognized by the attacker. Therefore, we did not include the hub in the identification of ZigBee devices. After collecting ZigBee network traffic, the second step involves extracting the features to identify the devices. In this step, the features we used include *mean packet length*, *mean inter-arrival time*, and *standard deviation in packet lengths*. We split each individual network traffic trace of a device into equal time intervals (e.g., 5 sec, 10 sec). Then, we calculated these features for each interval.

For the classification, we used the kNN classification algorithm. The classifier could correctly identify devices with an overall accuracy of 93% for ZigBee devices. This shows that as for WiFi and BLE, also devices using ZigBee can be identified with high accuracy.

6.5.6 Stage-2: Device State Detection

When an interaction between the device and the user occurs, a significant amount of data is transmitted, which leads to a significant increase in the traffic rate. After this data exchange, the data transmission drops to the minimum until a new interaction starts. When there is no activity, only the minimum amount of continuation packets like heartbeat messages are sent to minimize the device's power and bandwidth consumption. We also observed that almost the same amount of data transfer occurs for the same activities. All this information allows us to detect transitions between the activities or states of the device. For further validation, we do the following experiments.

Table 6.3: Evaluation results of device activity detection stage.

Device	**Random Forest**		**kNN**	
	F1 Score	Accuracy	F1 Score	Accuracy
ApexisCam	93	97	94	98
AirRouter	98	97	98	97
AugustSmartLock	100	100	100	100
BelkinWemoLink	80	79	85	83
DLinkCam	85	80	85	80
DLinkDoorSensor	94	98	92	97
DlinkMotionSensor	74	96	69	95
DlinkSiren	89	99	91	99
EdimaxCam	84	82	82	81
EdimaxSPlug1101	91	97	92	97
EdinetCam1	76	96	76	96
EdinetGateway	80	99	66	99
FitbitAria	100	100	100	100
Lightify2	86	99	81	98
PhilipsHueBridge	74	98	76	98
SMCRouter	94	91	100	100
STOutlet	83	99	92	99
STMotionSensor	91	97	92	97
STMultiSensor	86	99	92	99
TPLinkPlug1101	98	99	92	99
WansviewCam	91	87	91	86
WemoInsightSwitch	86	98	88	98
Avg	**88**	**99**	**91**	**95**

Feature Extraction

Our goal is to transform a sequence of packets into a supervised learning dataset. To achieve this, we divided the sequence of packets into windows of size W. For a given time interval length W, we extracted a feature vector comprised of three variables: *mean packet length*, *mean inter-arrival time* and *median absolute deviation of packet size*. Based on timestamped labels telling whether an activity was ongoing or not, we labeled the given vector with 1 for an ongoing activity or 0 for no activity. We found that the window size has significant influence on the performance of our model. The window size for the best performance depends on adjusting the size according to the duration of the activity. In general, selecting a smaller window size improves the performance until some level, but any further reduction results in decline of the performance. From our observation, better performance was observed when the window size is about a quarter of the duration of an activity.

Results

After obtaining feature vectors with labels from the sequence of packets, any supervised learning algorithm can be applied on the dataset. We have evaluated two supervised learning algorithms, namely Random Forest classifier (RF) and k-Nearest Neighbors classifier (kNN). As shown in Table 6.3 both RF and kNN have similar performance with RF averaging 88% and kNN with 91% average of correctly detecting activities. F1 Score of each device in Table 6.3 differs slightly. DlinkMotionSensor has the worst F1 score 74% using RF and 69% using kNN and the best F1 score is 100% for the Aria Fitbit and AugustSmartLock.

6.5.7 Stage-3: Device State Classification

In the device state classification experiments, the attacker's goal is to decide the state of the device (e.g., deciding if it is ON or OFF). When looking at the device's exchanged network packets, unlike previous steps, this is more difficult to determine. However, each state has a unique pattern which helps us to differentiate them from each other. In order to see if it is possible to differentiate the states, we did the following experiments:

Feature extraction

To conduct device state classification, informative and distinctive features must be extracted from time-series generated in the preprocessing steps. We used the *tsfresh* [CBN18] tool that automatically calculates a large number of time series characteristics and features and then constructed our feature vector. Examples of the features extracted from time-series are as follows: Absolute Energy of time-series, Length of time-series, Mean of time-series, Median of time-series, Skewness of time-series, Entropy of time-series, Standard deviation of time-series, Variance of time-series, Continuous wavelet transform coefficients, Fast Fourier Transform Coefficients, Coefficients of polynomial fitted to time-series.

Feature selection

The output of the feature extraction phase is a set of feature vectors including 795 binary features. A large number of features, some of which redundant or irrelevant might present several problems such as misleading the learning algorithm, and increasing model complexity. A feature selection technique was therefore used to mitigate these problems and also to reduce over-fitting, training time and improve accuracy. We used a technique leveraging ensembles of randomized decision

trees (i.e., Extra Trees-Classifier) for determining the importance of individual features. We exploited Extra-Trees Classifier to compute the relative importance of each attribute to inform feature selection. The features considered unimportant were discarded. The feature selection phase effectively reduced the feature vector size from 795 binary features to 197 features.

Results

Our objective was to build a performant model to correctly classify IoT devices' states even if their traffic is encrypted. To this end, we employed several machine learning algorithms for the classification such as *XGBoost*, *Adaboost*, *Random Forest*, *SVM with RBF kernel*, *kNN*, *Logistic Regression*, *Naïve Bayes*, and *Decision Tree*. In order to ensure that our machine learning model got the most of the patterns from the training data correctly, and it was not picking up too much noise, we shuffled and split the data-points to conduct the following experiments: *(i)* we performed 5-fold Cross Validation (CV) on a training set of 377 samples (75% of data) for assessing the effectiveness of the machine learning model and *(ii)* we carried out Hold-out Validation on 126 samples (25% of data) to test the machine learning model performance against unseen data.

5-fold Cross Validation: To avoid the risk of missing important patterns or trends in the dataset, we applied cross validation, as it provides ample data for training the model and also leaves ample data for validation. Thus, we conducted a 5-fold cross validation experiment. In 5-fold CV, the data are randomly partitioned into 5 equal-sized sub-samples. Of the 5 sub-samples, a single sub-sample is retained as the validation data for testing the model, and the remaining 4 sub-samples are used as training data. The process is then repeated 5 times with each of the 5 sub-samples used exactly once as the validation data. The 5 results from the folds

Table 6.4: Cross-validation and hold-out validation results for device state classification.

Classifier	**5-fold CV** (75% of data)	**Held-out data** (25% of data)		
		Precision	Recall	F1 Score
SVC RBF Kernel	86	89	87	87
Logistic Reg.	87	90	89	88
Random Forest	**92**	96	94	**94**
Naive Bayes	87	92	87	88
Decision Tree	66	62	63	61
K-NN	84	91	87	87
Adaboost	86	89	87	87
XGBoost	85	91	87	87

can then be averaged to produce a single estimation. We obtained 92% accuracy in terms of F1 Score in the detection of devices' states using Random Forest classifier, as shown in Table 6.4.

Hold-out Validation: To make sure that our classifier can generalize well and is not over-fitted, we tested the classifiers' performance in terms of Precision, Recall, and F1 Score against unseen data (the data was removed from the training set and is only used for this purpose). Table 6.5 shows the detailed results obtained by Random Forest classification algorithm when conducting the device state classification over 126 unseen samples. As can be seen, the F1 Score of each device used in the experiment differs slightly. We obtained an average performance measurement of 0.94 (94%) of correctly classifying activities. This shows that an attacker can easily differentiate the devices' states.

6.5.8 Stage-4: User Activity Inference

Modern smart home environments comprise several sensors and devices that are connected with each other and share information. These devices and sensors are configured as independent entities, but work co-dependently to provide an autonomous

Table 6.5: Hold-out validation results of RF classifier for all IoT devices.

Device name	**Action**	**Pre.**	**Recall**	**F1**	**Supp.**
ApexisCamera	live view	100	100	100	4
AirRouter	surfing on amazon	80	100	89	4
AugustSmartLock	off	100	67	80	3
AugustSmartLock	on	67	100	80	2
BelkinWemoLink	off	80	100	89	8
BelkinWemoLink	on	100	50	67	4
DLinkCamera	live view	100	100	100	3
DLinkDoorSensor	open	100	100	100	5
DLinkSensor	motion detection	100	100	100	6
DLinkSiren	turn on	100	100	100	1
EdimaxCam	live view	100	100	100	1
EdimaxSPlug1101	on	100	100	100	5
EdinetCam1	live view	100	100	100	2
EdinetGateway	on	100	100	100	3
FitbitAria	measure weight	100	100	100	4
Lightify2	change light type	100	100	100	6
PhilipsHueBridge	turn scene off	100	100	100	3
PhilipsHueBridge	turn scene on	100	100	100	5
SMCRouter	surfing on amazon	100	80	89	5
STOutlet	on	100	89	94	9
STMotion	active	88	100	93	7
STMotion	inactive	100	71	83	7
STMultiSensor	acceleration active	100	100	100	8
STMultiSensor	acceleration inactive	71	100	83	5
TPLinkPlugHS110	turn off	100	100	100	5
WansviewCam	reboot	100	100	100	9
WemoInsightSwitch	on	100	100	100	2
Avg./Total	———	**96**	**94**	**94**	**126**

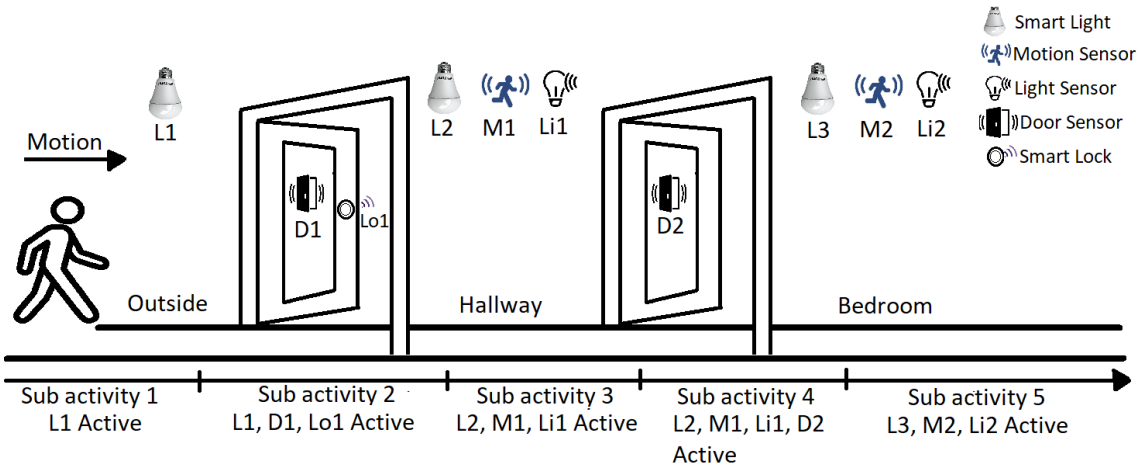


Figure 6.4: User walking scenario in a smart home environment.

system. Any user activity in a smart home can be inferred by observing the states of the devices and sensors.

Modelling User Activities via Hidden Markov Model

In Figure 6.4, we demonstrate a simple walking scenario of a user. Here, a user is entering the smart home from outside to the bedroom through the hallway. The scenario consists of five different devices with lights both inside and outside the home controlled by the motion sensor (M) and light sensor (L). This simple activity can be illustrated as a sequential pattern: Sub-activity 1- moving towards the door from outside (L1 is active), Sub-activity 2- user opens the front door (L1, D1, Lo1 are active), Sub-activity 3- user enters the hallway (L2, M1, Li1 are active), Sub-activity 4- user enters the room (Li2, L2, M2, D1, Lo1 are active), Sub-activity 5- user inside the home (L2, M2, Li2 are active). To complete the activity, a user must follow the same sequence of sub-activities and complete each step. As discussed earlier, the devices' states (active/inactive) for a specific time can be determined from the network traffic captured from the devices. These device states can be used to infer an on-going activity in a smart home setting.

Feature Extraction

To infer user activities, different device features must be extracted from network traffic data. Network traffic data contain several features including timing information, sensor information, device states, location, etc. Based on the data-type, the extracted features from the network traffic for user activity inference can be represented as follows:

$$Data\ array, E_T = \{S, D, M, L\}, \tag{6.9}$$

where T is the set of timing features extracted from the network traffic, S is the set of sensors' features, D is the set of device features, M is the features extracted from the controlling device (smartphone/tablet), and L is the set of location features extracted from the network traffic. We describe the characteristics of these features below.

- *Timing features (T):* Smart home devices change their state according to user activities and commands. Some devices perform time-independent tasks (e.g., switching lights with motion), while some devices perform a task in a certain pattern with different user activities (e.g., walking from one point to another) based on smart home settings. We extract the time of an event from the network traffic captured from different devices to build the overall state of the smart home at the time of the user activity.
- *Sensor State features (S):* Smart home environment consists of different sensors (e.g., motion sensor, light sensor, door sensor, etc.) which act as a bridge between devices and the peripheral. Sensors in a smart home can sense different environment parameters which can trigger different pre-defined tasks in multiple devices. Moreover, sensors can sense any change occurred because of

a user interaction and forward this information as an input to the associated devices. These sensor data can be both logical (motion sensor) and numerical (temperature sensor) depending on the nature of the sensor. We observe the changes in both logical and numerical value of a sensor from the captured network traffic and use as a feature to infer user activities. We represent the changes in sensor data as binary output: 1 for active state and 0 for inactive state.

- *Device State features (D):* In a smart home environment, multiple devices such as smart light, smart thermostat, etc. can be connected with each other and with a central hub to perform different tasks. These devices can be configured to change their states (active/inactive) to perform a pre-defined task or to perform a task based on user activities. We consider the state information of all the connected devices as features and extract this information from captured network traffic to infer the on-going user activity. The active and inactive states of the devices are illustrated as 1 and 0 respectively in the data array.

- *Controller State features (M):* Smart home devices can be controlled in an autonomous way and also by using a controller device (smartphone/tablet). To understand the changes in states of the sensors and devices, one should consider the control commands generated by the controller devices. We consider the state of controller device as active (represented as 1 in data array) when a user interacts with smart home devices via controller device and inactive otherwise (represented as 0 in data array). This state information of the controller devices can be extracted from the captured network traffic to build the data array.

- *Controller Location features (L):* The devices connected in a smart environment can be controlled from a different location and this location information can be collected from the captured network traffic. We consider the location of the controller device as a feature to understand any activities on smart home. We consider the home location of the controller device as 1 and the away location of the controller as 0 to represent the location feature as a binary number in the data array.

For Stage 4, we captured the network traffic from a smart home environment and create the feature array explained in Equation 6.9. We captured the network traffic for a specific time to correctly portray user activities from the network data. Each element of the data array represents the operating conditions of different smart devices, sensors, and controller devices. These data were then used to train a Hidden Markov Model (HMM) to detect user activities in a smart home environment.

Hidden Markov Model (HMM) is a statistical Markov model, where each state of the model contains unobserved states. In traditional Markov model, all the states of an ongoing process are observable while in Hidden Markov model the states are not directly visible. Here, only the output depending on the states is visible. The main assumptions of HMM are similar to the Markov Chain model which are as follows: (1) The probability of occurring a particular state depends only on the previous state. (2) The transition between two consecutive states is independent of time. (3) Hidden states are not visible, but each hidden state randomly generates one of the defined observations or visible states. We use these properties of HMM to detect different user activities from the captured network traffic in a smart home environment. The probabilistic condition of HMM is shown in Equation 6.10, where

X_t denotes the state at time t for a user activity in a smart home [SAU17].

$$P(X_{t+1} = x|X_1 = x_1, X_2 = x_2..., X_t = x_t) = P(X_{t+1} = x|X_t = x_t), \quad (6.10)$$
$$when,\ P(X_1 = x_1, X_2 = x_2..., X_t = x_t) > 0$$

For each activity in the smart home environment, multiple feature arrays were created and these arrays maintain different, but specific sequences for different user activities. For a specific time, t, the state of the smart home can be represented by the data array E_T where each element of this data array illustrates the conditions of smart home devices' and sensors' as binary output (1 for active status of an entity and 0 for inactive status). Thus, each state can be represented as an n-bit binary number, where n is the total number of devices in the smart home. Let assume the smart home environment is in state i at time t and changing to state j at time $t + 1$. The transition probability from state i to state j can be noted as P_{ij}. If the smart home environment comprises of n number of devices and $m = 2^n$ states in the system, the transition matrix of HMM is given as follows:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & \dots & P_{1m} \\ P_{21} & P_{22} & P_{23} & \dots & \dots & P_{2m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{m1} & P_{m2} & P_{m3} & \dots & \dots & P_{mm} \end{bmatrix} \quad (6.11)$$

If the smart home environment has X_t number of states where $t = 0, 1, \dots, T$, the elements of the transition matrix can be shown as follows [NPVB05]:

$$P_{ij} = \frac{N_{ij}}{N_i}, \quad (6.12)$$

where N_{ij} denotes the number of transition from X_t to X_{t+1}, where X_t is the state at time t and X_{t+1} is the state at time $t + 1$.

To build the observation probability matrix, we consider different user activities as hidden states of the smart home environment and correlates with the system's states build from the data arrays. Let assume the smart home environment has k number hidden states (H) in the system. The observation probability matrix of HMM is given as follows:

$$B = \begin{bmatrix} X_1(H_1) & X_2(H_1) & X_3(H_1) & \dots & \dots & X_m(H_1) \\ X_1(H_2) & X_2(X_2) & P_3(X_2) & \dots & \dots & X_m(H_2) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ X_1(H_k) & X_2(H_k) & X_3(H_k) & \dots & \dots & X_m(H_k) \end{bmatrix} \tag{6.13}$$

where $X_m(H_k)$ is the probability of observing H_k from state X_m. $X_m(H_k)$ can be represented by Equation :

$$X_m(H_k) = P(H_k|X_m), \tag{6.14}$$

For our work, we want to detect the hidden state (user activity) from a given state sequence. To calculate the probability of user activity, we use the Forward-Backward (FB) algorithm to decode HMM. The FB algorithm can be expressed by the following equations.

$$Forward\ recursion, P_m(t+1) = B_{mH_{t+1}} \sum_{a=0}^{m} P_a(t) P_{am} \tag{6.15}$$

$$Backward\ recursion, B_i(t) = \sum_{b=1}^{k} P_{ij} B_{jH_{t+1}} B_{j(t+1)}, \tag{6.16}$$

where, t= 0,1, ..., T-1. The probability of occurring a hidden state (user activity) from the sequence of observable states (device states) can be calculated from the following equation.

$$P(H_1, H_2, \dots, H_k) = \sum_{l=1}^{K} P_k(t) B_k(t). \tag{6.17}$$

Table 6.6: Typical activities of users in a smart home environment.

Task Category	**Task Name**
Time-independent	1. Controlling device within smart home.
	2. Controlling device from outside of the home.
	3. Presence in a specific point at home.
Time-dependent	4. Walking in the smart home.
	5. Opening/ closing doors/windows.
	6. Entering/ exiting from smart home

To train this HMM, we collected data from a smart home environment with real smart devices. We consider common smart home devices to build our training environment [dL17]. Our test smart home environment included Samsung SmartThings hub, Samsung multipurpose sensor, Samsung motion sensor, Netgear Arlo security camera, Philips Hue smart light, Ecobee Smart Thermostat, and August Smart Lock. We collected network traffic data from 10 different users for different user activities.

Activity Types

User activities in a smart home environment can be instantaneous (e.g., switching on a device) or sequential over time (e.g., walking from one place to another). We categorized user activities in a smart home environment in two categories - time-independent and time-dependent user activities.

- *Time-independent Activities:* These user activities are instantaneous, non-sequential activities which do not depend on time. For example, a user can switch on/off a device in the smart home environment at a specific time instance. This activity will show changes in different features for only one time.
- *Time-dependent Activities:* These user activities are time-dependent, sequential activities. For example, a user can move from one point to another point.

This activity will show changes in different features over time in a specific sequence.

We tested our HMM model with data collected from six different user activities. Our user activity model is explained below.

- *User Activity- 1.* A user is controlling a device from inside of the smart home environment.
- *User Activity- 2.* A user is controlling a device from outside of the smart home environment.
- *User Activity- 3.* A user is performing tasks from a specific point of a smart home environment.
- *User Activity- 4.* A user is walking from one point to another inside the smart home environment.
- *User Activity- 5.* A user is entering/ exiting from the smart home environment.
- *User Activity- 6.* A user is opening/ closing a window/ door in smart home environment.

Results

To train our proposed HMM for user activity inference, we collected user activity data for a week from 15 different people (total 30 datasets) in an emulated smart home environment. We asked the users to perform their daily activities in a timely manner (from morning to night) and performed the same activities in defined sequences in a real-life smart home setting. We considered single authorized smart home user interacting with smart devices at a time for data collection. We trained our HMM model with these data. We also collected data for this activity model to

Table 6.7: User activity inference from network traffic data in a smart home environment.

Smart Home User Activity	**TPR**	**FNR**	**TNR**	**FPR**	**Accuracy**	**F-score**
Activity-1	1	0	1	0	1	1
Activity-2	1	0	1	0	1	1
Activity-3	1	0	1	0	1	1
Activity-4	0.96	0.03	0.94	0.05	0.95	0.95
Activity-5	0.95	0.04	0.87	0.12	0.93	0.91
Activity-6	0.97	0.02	0.91	0.08	0.94	0.94

test our proposed method. We collected two datasets for each activity (12 in total) to test the efficacy of the activity inference model.

In Table 6.7, the evaluation results of our activity inference model are shown. For time-independent activities (Activity-1, Activity-2, and Activity-3), one can infer with 100% accuracy and F-score from the captured network traffic data in a smart home environment. On the contrary, accuracy and F-score decreases slightly for time-dependent activities as these activities introduce FP and FN instances in the activity inference model. For Activity-4, our proposed stage 4 activity inference HMM can achieve both accuracy and F-score over 95%. The false positive rate (FPR) and false negative rate (FNR) are over 5% and 3% respectively for Activity-4. For Activity-4 and Activity-5, the accuracy of user activity inference decreases (93% and 94% respectively) while FPR and FNR increases. The reason for the increment of FPR and FNR is that different time-dependent user activities can have similar patterns over time with small changes in specific time instances. This affects the probability of occurring an activity calculated from HMM. In summary, an attacker can infer time-independent activities more accurately (with 100% accuracy and F-score) than the time-dependent activities (with over 95% accuracy and F-score).

Finally, note that an accurate user activity inference means that all the stages in the multi-stage attack have to be correctly guessed, which may lower the end-to-end successful inference rate of the attacker. For example, if the stage 1, 2, 3, and 4 are X, Y, Z, and T, respectively, for an attacker, the probability of correctly guessing the Activity-4 of the user is $X \times Y \times Z \times T$. However, we also note that independently inferred information in every stage is also valuable as it may also include sensitive information (e.g., inferring the device type of a connected medical device may reveal the health status of the subject [Sø17]).

6.6 Mitigating the Privacy Leaks

Despite the security vulnerabilities exploited before, as these privacy concerns are *inherent* and *insidious*, it is too hard to detect and avoid these types of threats associated with smart home devices. An attacker can passively listen to the wireless medium and record all the network traffic from a smart home environment without interrupting the normal activities of devices and their users.

6.6.1 Straightforward Solutions

Using VPN or Tor-like Tools

The use of VPN will prevent an attacker from recording the victim's outgoing traffic after the gateway as it is going to be encrypted by the VPN provider. An ISP cannot record the network traffic of the user anymore. On the other hand, Tor will make the source and destination IPs impossible to determine for the ISP. Both methods will protect the communication between the home AP and the server of the hub.

However, they provide no protection against an attacker within range (e.g., outside, near home) and sniffing the internal traffic.

Signal Attenuation

A signal attenuator can be used in theory to protect from an attacker sniffing the internal network traffic of the smart home. This can be realized via a wired connection or using Faraday cages [SSW08]. Nonetheless, forcing all the devices to such a modification in the hardware level and a Faraday cage could be too unrealistic and very expensive to set up for the smart home users.

Traffic Shaping

The traffic shaping solutions have been widely studied in the literature of website classifiers. Padding to proper MTU, exponential padding, or random padding are some of the countermeasures with the traffic shaping methods. Indeed, not only padding, but also constant or random delays can be applied to the packets transmitted to protect from inference attacks. In all these solutions, the underlying protocol, which needs to provide a real-time accurate values from the devices, is modified in a way that unfortunately lowers the efficiency and accuracy of the devices.

6.6.2 Proposed Approach

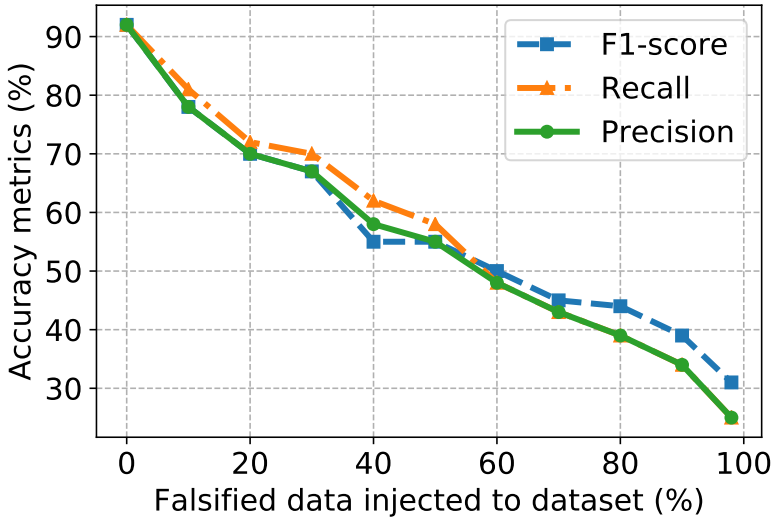
In this sub-section, we introduce a solution based on generating spoofed traffic. In this way, even if the user is not at home, generating false activity for the user's presence traffic will mask the user's absence.

In order to measure the efficacy of our proposed spoofed traffic, we investigated the injection of false packets by modifying the feature vectors and evaluated how

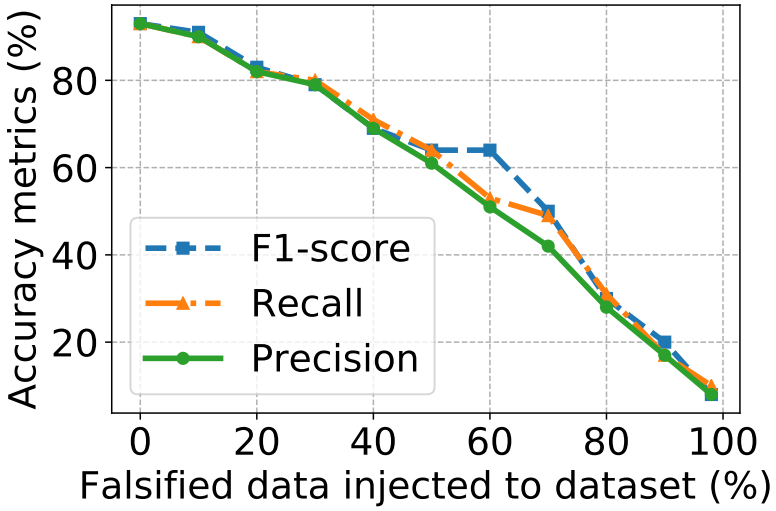
the performance measurements would change. Then, we applied it to the device state detection and device activity classification attacks. Since the user activity inference is based on the results of the device state detection and device activity classification attacks, if we can falsify their results, the attacker will not able to infer the activities correctly. Particularly, we conducted a set of experiments where we injected falsified data into the training set to observe how the previously shown detection and classification algorithms would behave in such a situation. The results are shown in Figure 6.5.

Impact of False Data Injection on Device State Detection. Figure 6.5a shows the average of the accuracy measures for the kNN algorithm after increasingly injecting false packets. When there is no injected false packet, all of the devices have 91% F1 score, then it linearly decreases with the increase of false packets. For example, injecting false data equivalent to 10% of packets exchanged during the observation time resulted in a decrease by 13%. For 90% false traffic addition, the accuracy of device state detection declined by about 57%. This shows that traffic injection can be efficiently used for hiding the state of devices from the adversary.

Impact of False Data Injection on Device State Classification. We injected the falsified data into the training data and computed the accuracy metrics in terms of F1 Score, Precision, and Recall. We injected 10% falsified data and continued injecting until 90% of the dataset contained false data. As can be seen in Figure 6.5b, the F1 Score plunges dramatically when injecting 90% false data and reaches 15%. This is due to the fact that randomly falsified features deteriorate traffic patterns used for classifying the devices' states. Also here, we can see that by injecting increasing amounts of fabricated traffic, the adversary can effectively be prevented from making inferences about the types of device events occurring.



(a)



(b)

Figure 6.5: Impact of false data injection experiments on the attack accuracy. Its impact on device state detection and device state classification attacks are shown in a) and b), respectively.

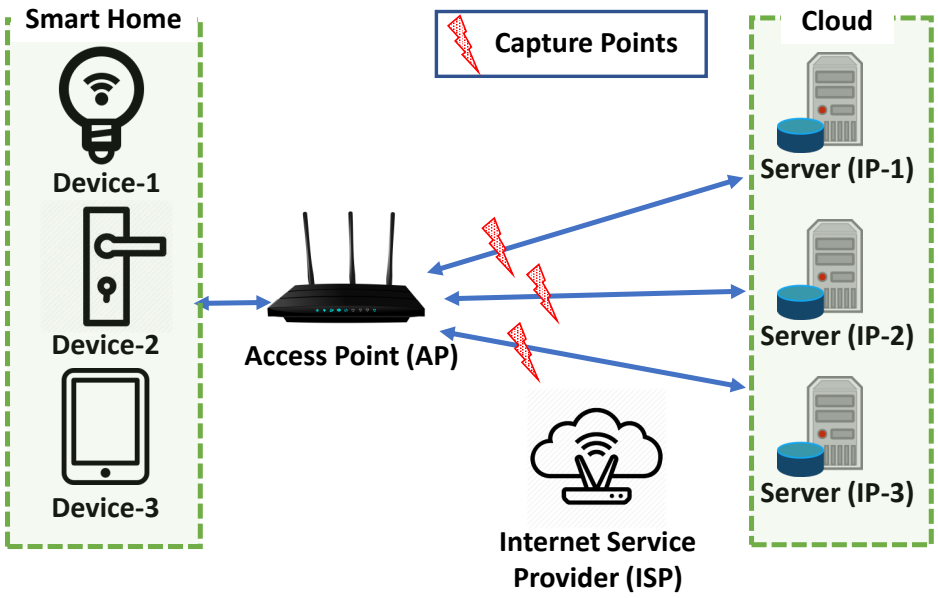


Figure 6.6: Remote adversary model (e.g., a malicious ISP).

6.7 Discussion

ISP as an adversary: Note that so far, our adversary model included only local adversary, where the adversary is within the range of radio frequency. An extension to this adversary model can be a remote adversary that can monitor outgoing network traffic of the smart home. A concrete example of such an adversary is an ISP. Compared to the local adversary model considered in this work, an ISP-like adversary has both advantages and disadvantages. It does not have to be within a range and it can see the source and destination IPs of the packets, which a local adversary can not see if the WPA encryption is enabled. However, it can only collect the outgoing network traffic, not the internal two-way (upstream and downstream) network traffic as all the traffic is merged by the gateway (i.e., access point). Figure 6.6 shows the complete topology of the network from device to cloud.

As can be seen in Figure 6.6, an ISP will only see the router's (i.e., gateway/access point) MAC address. Therefore, it can not use the MAC addresses of the smart home

devices for the device identification. However, it can still try to use IPs in order to identify the devices and infer activities. Though there are number of challenges that attacker needs to solve in order to able use IP as a device identifier. First of all, if Network Address Translation (NAT) is deployed by the AP[1], the ISP can not find out the topology of the smart home and the number of devices. Even though NAT is not enabled, ZigBee and BLE devices have never been assigned an IP as they communicate with the AP through a hub, where only the hub they are connected to gets an IP. Moreover, devices do not communicate with only one server. Instead, sometimes multiple devices use one server (i.e., destination IP) as in the Samsung ST Hub, or sometimes one device can use multiple servers [CLBR16, ARS+17]. Therefore, even though the ISP-like attacker has some advantages (i.e., seeing IPs) over the local adversary, there are several additional challenges that it needs to solve to get the same attack working. We leave this kind of adversary out of scope for now and will be studied in a future work.

Multi-user vs. single user: Smart home devices support multiple authorized users. In a multi-user smart home scenario, more than one user can control and change the settings of smart devices. Additionally, different users can perform different activities within the smart environment at a time. This can create some false positive and false negative cases in user activity inference using our proposed method. Nonetheless, an attacker can still infer the device type and devices states from the network traffic. Additionally, the attacker can also infer the presence of multiple users and the specific point of ongoing activities in multi-user smart home environment using the network traffic. Compared to a multi-user scenario, a single user smart home environment is more vulnerable to our proposed threat as it is easier to infer a single on-going user activity in the smart home.

[1]Assuming IPv4 is still in use.

Local vs. remote control: To improve the user control over smart devices and increase convenience, smart homes offer remote access control in addition to traditional local access. Our proposed threat model can guess both local and remote access from location feature of the captured network traffic. This is a serious threat to user privacy as attackers can detect when a user is changing the state of a specific device remotely and perform malicious activities. For example, an attacker can infer when a user is accessing the smart lock remotely, which may result in physical access to the home environment.

Smart device diversity: Smart devices have no common network protocols. Indeed, some of them such as WiFi, ZigBee, and BLE are more popular than others. This makes it harder to sniff all the devices that the smart home user is using. In addition to the diversity of network protocols, smart home devices come with different computational resources, hardware types, capabilities, exchanged data format etc. All of these differences in smart devices make it very challenging to build a generic solution as well as an attack. However, with our automated multi-stage privacy-attack, we showed the feasibility of the attack with the most popular network protocols, which covers the most of the commercial devices.

Limitations of defense: As our results show, injecting false data to the communication clearly decreases the accuracy of the attacks. However, even though it is an effective method and it has the advantage of not affecting the efficiency of real traffic on the devices, but it requires an extension to the protocols to put a flag on the fake activities, which will be known by both devices and server. Here, we propose two different ways to implement this solution with trade-offs on the power consumption and security.

1. *Only on the Hub*: This countermeasure can be implemented only on the smart hub devices and does not require relatively-constrained smart home devices to

be part of the countermeasure. Even though this type of solution is effective and better for battery-powered devices, it can be discovered by the attacker; after a while, an attacker can use the network traffic from the device(s) to hub only, but the attacker's success and accuracy will decrease in that case. '

2. *On the Hub and Device*: This countermeasure requires to modify the communication protocol both on the smart home device and the hub. This will generate a more realistic interaction between the device and the hub, but this may cause slightly more power consumption (depending on the size of the extra field for the flag) in the device and requires a modification on the devices to send the false data.

Depending on the devices, where the solution to be implemented on, the required modification in the current system and the implementer can change. For example, if our solution is preferred to be implemented only on the hub, then it can be implemented by the manufacturer of the hub only. However, if it is going to be implemented on the devices and the hub together, it requires either the collaboration of both sides or a protocol-level modification.

Generalizability of the attack: As we noted in the assumptions, the attacker we considered in this work does not need exactly the same devices to train its attack model, but it needs exact brand and device type to get the results presented in this work as we use the $< brand, device - type >$ pair to uniquely identify devices. In other words, we assume at the end of the device identification stage of our attack, the attacker knows $< brand, device - type >$ pair. However, this assumption weakens the attack model. An attacker who can infer the device type and does not need the same device with the same brand would be more realistic. In order to remove this assumption, the same device type with different brands should be used to train the models and to attack (i.e., testing). It would be interesting to train and test the

attack models on the same device type with different brands, or the same brand with different device types. Moreover, it would be also interesting to test the affects of model numbers, device configurations, or firmware updates etc.

6.8 Conclusion

In this chapter, we explored how encrypted traffic from a smart home environment can be used to infer sensitive information about smart devices and sensors. Specifically, we introduced a novel multi-stage privacy attack, which an attacker can exploit to automatically detect and identify particular types of devices, their actions, states, and related user activities by passively monitoring the traffic of smart home devices and sensors. Our evaluation on an extensive list of off-the-shelf smart home devices, sensors, and real users showed that an attacker can achieve very high accuracy (above %90) in all the attack types. As opposed to to earlier straightforward activity identification approaches, the novel multi-stage privacy attack can perform detection and identification automatically, is device-type and protocol-agnostic, and does not require extensive background knowledge or specifications of analyzed protocols. Finally, we proposed a new yet effective mitigation mechanism to hide the real activities of the users. The effectiveness of the multi-stage privacy attack raises serious privacy concerns for any smart environment equipped with smart devices and sensors including personal homes, residences, hotel rooms, offices of corporations or government agencies.

PART III - PRIVACY-AWARE SECURE DATA EXCHANGE METHODS

CHAPTER 7

HOMOMORPHIC ENCRYPTION

In the next two chapters, in order to provide user privacy, we use homomorphic encryption (HE). In this section, we first give an overview of HE schemes in the literature.

7.1 Introduction

In ancient Greeks, the term "ὁμός" (homos) was used in the meaning of "same" while "μορφή" (morphe) was used for "shape" [LS96]. Then, the term *homomorphism* is coined and used in different areas. In abstract algebra, homomorphism is defined as a map preserving all the algebraic structures between the domain and range of an algebraic set. The map is simply a function, i.e., an operation, which takes the inputs from the set of domain and outputs an element in the range, (e.g., addition, multiplication). In the cryptography field, the homomorphism is used as an encryption type. The *Homomorphic Encryption* (HE) is a kind of encryption scheme which allows a third party (e.g., cloud, service provider) to perform certain computable functions on the encrypted data while preserving the features of the function and format of the encrypted data. Indeed, this homomorphic encryption corresponds to a mapping in the abstract algebra. As an example for an additively HE scheme, for sample messages m_1 and m_2, one can obtain $E(m_1 + m_2)$ by using $E(m_1)$ and $E(m_2)$ without knowing m_1 and m_2 explicitly, where E denotes the encryption function.

Normally, encryption is a crucial mechanism to preserve the privacy of any sensitive information. However, the conventional encryption schemes can not work on the encrypted data without decrypting it first. In other words, the users have to

sacrifice their privacy to make use of cloud services such as file storing, sharing and collaboration. Moreover, untrusted servers, providers, popular cloud operators can keep physically identifying elements of users long after users end the relationship with the services [McM13]. This is a major privacy concern for users. In fact, it would be perfect if there existed a scheme which would not restrict the operations to be computed on the encrypted data while it would be still encrypted. From a historical perspective in cryptology, the term *homomorphism* is used for the first time by Rivest, Adleman, and Dertouzous [RAD78] in 1978 as a possible solution to the computing without decrypting problem. This given basis in [RAD78] has led to numerous attempts by researchers around the world to design such a homomorphic scheme with a large set of operations. In this work, the primary motivation is to survey the HE schemes focusing on the most recent improvements in this field, including *partially*, *somewhat*, and *fully* HE schemes.

A simple motivational HE example for a sample cloud application is illustrated in Figure 7.1. In this scenario, the client, C, first encrypts her private data (Step 1), then sends the encrypted data to the cloud servers, S, (Step 2). When the client wants to perform a function (i.e., query), $f()$, over her own data, she sends the function to the server (Step 3). The server performs a homomorphic operation over the encrypted data using the $Eval$ function, i.e., computes $f()$ blindfolded (Step 4) and returns the encrypted result to the client (Step 5). Finally, the client recovers the data with her own secret key and obtains $f(m)$ (Step 6). As seen in this simple example, the homomorphic operation, $Eval()$, at the server side does not require the private key of the client and allows various operations such as addition and multiplication on the encrypted client data.

An early attempt to compute functions/operations on encrypted data is Yao's *garbled circuit*[1] study [Yao82]. Yao proposed two party communication protocol as a solution to the Millionaires' problem, which compares the wealth of two rich people without revealing the exact amount to each other. However, in Yao's *garbled circuit* solution, ciphertext size grows at least linearly with the computation of every gate in the circuit. This yields a very poor efficiency in terms of computational overhead and too much complexity in its communication protocol. Until Gentry's breakthrough in [Gen09], all the attempts [RSA78, GM82, ElG85, Ben94, NS98, OU98, Pai99a, DJ01, KTX07, Yao82, BGN05, SYY99, IP07] have allowed either one type of operation or limited number of operations on the encrypted data. Moreover, some of the attempts are even limited over a specific type of set (e.g., branching programs). In fact, all these different HE attempts can neatly be categorized under three types of schemes with respect to the number of allowed operations on the encrypted data as follows: (1) *Partially Homomorphic Encryption* (PHE) allows only one type of operation with an unlimited number of times (i.e., no bound on the number of usages). (2) *Somewhat Homomorphic Encryption* (SWHE) allows some types of operations with a limited number of times. (3) *Fully Homomorphic Encryption* (FHE) allows an unlimited number of operations with unlimited number of times.

PHE schemes are deployed in some applications like e-voting [Ben87] or Private Information Retrieval (PIR) [KO97]. However, these applications were restricted in terms of the types of homomorphic evaluation operations. In other words, PHE schemes can only be used for particular applications, whose algorithms include only addition or multiplication operation. On the other hand, the SWHE schemes sup-

[1]A circuit is the set of connected gates (e.g., AND and XOR gates in boolean circuits), where the evaluation is completed by calculating the output of each gate in turn.

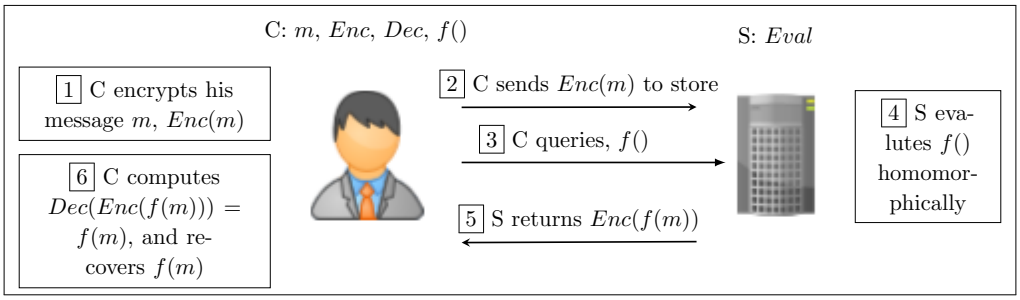


Figure 7.1: A simple client-server HE scenario, where C is Client and S is Server

port both addition and multiplication. Nonetheless, in SWHE schemes that are proposed before the first FHE scheme, the size of the ciphertexts grows with each homomorphic operation and hence the maximum number of allowed homomorphic operations is limited. These issues put a limit on the use of PHE and SWHE schemes in real-life applications. Eventually, the increasing popularity of cloud-based services accelerated the design of HE schemes which can support an arbitrary number of homomorphic operations with random functions, i.e. FHE. Gentry's FHE scheme is the first plausible and achievable FHE scheme [Gen09]. It is based on ideal-lattices in math and it is not only a description of the scheme, but also a powerful framework for achieving FHE. However, it is conceptually and practically not a realistic scheme. Especially, the *bootstrapping* part, which is the intermediate refreshing procedure of a processed ciphertext, is too costly in terms of computation. Therefore, a lot of follow-up improvements and new schemes were proposed in the following years.

7.2 Homomorphic Encryption Schemes

In this section, we explain the basics of HE theory. Then, we present notable PHE, SWHE and FHE schemes. For each scheme, we also give a brief description of the scheme.

Definition 1. *An encryption scheme is called homomorphic over an operation '$\star$' if it supports the following equation:*

$$E(m_1) \star E(m_2) = E(m_1 \star m_2), \quad \forall m_1, m_2 \in M, \tag{7.1}$$

where E is the encryption algorithm and M is the set of all possible messages.

In order to create an encryption scheme allowing the homomorphic evaluation of arbitrary function, it is sufficient to allow only addition and multiplication operations because addition and multiplication are functionally complete sets over finite sets. Particularly, any boolean circuit can be represented using only XOR (addition) and AND (multiplication) gates. While an HE scheme can use the same key for both encryption and decryption (symmetric), it can also be designed to use the different keys to encrypt and decrypt (asymmetric). A generic method to transform symmetric and asymmetric HE schemes to each other is demonstrated in [Rot11].

An HE scheme is primarily characterized by four operations: $KeyGen$, Enc, Dec, and $Eval$. $KeyGen$ is the operation, which generates a secret and public key pair for the asymmetric version of HE or a single key for the symmetric version. Actually, $KeyGen$, Enc and Dec are not different from their classical tasks in conventional encryption schemes. However, $Eval$ is an HE-specific operation, which takes ciphertexts as input and outputs a ciphertext corresponding to a functioned plaintext. $Eval$ performs the function $f()$ over the ciphertexts (c_1, c_2) without seeing the messages (m_1, m_2). $Eval$ takes ciphertexts as input and outputs evaluated ciphertexts. The most crucial point in this homomorphic encryption is that the format of the ciphertexts after an evaluation process must be preserved in order to be decrypted correctly. In addition, the size of the ciphertext should also be constant to support unlimited number of operations. Otherwise, the increase in the ciphertext size will require more resources and this will limit the number of operations.

Of all HE schemes in the literature, PHE schemes support *Eval* function for only either addition or multiplication, SWHE schemes support for only limited number of operations or some limited circuits (e.g., branching programs) while FHE schemes supports the evaluation of arbitrary functions (e.g., searching, sorting, max, min, etc.) with unlimited number of times over ciphertexts. The well-known PHE, SWHE, and FHE schemes are summarized in the timeline in Figure 7.2 and are explained in the following sections with a greater detail. The interest in the area of HE significantly increased after the work of Gentry [Gen09] in 2009. Therefore, we articulate the HE schemes, FHE anymore, after Gentry's work in a greater detail and we also discuss their implementations and recent techniques to make it faster in Section 7.3. Here, we start with the PHE schemes, which are the first stepping stones for FHE schemes.

7.2.1 Partially Homomorphic Encryption Schemes

There are several useful PHE examples [RSA78, GM82, ElG85, Ben94, NS98, OU98, Pai99a, DJ01, KTX07] in the literature. Each has improved the PHE in some way. However, in this section, we primarily focus on major PHE schemes that are the basis for many other PHE schemes.

RSA

RSA is an early example of PHE and introduced by Rivest, Shamir, and Adleman [RSA78] shortly after the invention of public key cryptography by Diffie Helman [DH76]. RSA is the first feasible achievement of the public key cryptosystem. Moreover, the homomorphic property of RSA was shown by Rivest, Adleman, and Dertouzous [RAD78] just after the seminal work of RSA. Indeed, the first attested

use of the term "privacy homomorphism" is introduced in [RAD78]. The security of the RSA cryptosystem is based on the hardness of *factoring problem* of the product of two large prime numbers [Mon94][2] RSA is defined as follows:

- *KeyGen Algorithm:* First, for large primes p and q, $n = pq$ and $\phi = (p-1)(q-1)$ are computed. Then, e is chosen such that $gcd(e, \phi)$ and d is calculated by computing the multiplicative inverse of e (i.e, $ed \equiv 1 \pmod{\phi}$). Finally, (e, n) is released as the public key pair while (d, n) is kept as the secret key pair.
- *Encryption Algorithm:* First, the message is converted into a plaintext m such that $0 \leq m < n$, then the RSA encryption algorithm is as follows:

$$c = E(m) = m^e \pmod{n}, \quad \forall m \in M, \tag{7.2}$$

  where c is the ciphertext.

- *Decryption Algorithm:* The message m can be recovered from the ciphertext c using the secret key pair (d, n) as follows:

$$m = D(c) = c^d \pmod{n} \tag{7.3}$$

- *Homomorphic Property:* For $m_1, m_2 \in M$,

$$E(m_1)*E(m_2) = (m_1^e \pmod{n})*(m_2^e \pmod{n}) = (m_1*m_2)^e \pmod{n} = E(m_1*m_2). \tag{7.4}$$

The homomorphic property of RSA shows that $E(m_1 * m_2)$ can be directly evaluated by using $E(m_1)$ and $E(m_2)$ without decrypting them. In other words, RSA

[2]Here, we do not mean that RSA is secure. We mean the most basic attack on RSA (e.g., key recovering attack) has to solve the problem of factoring of two large primes. For example, plain RSA is not secure against *Chosen Plaintext Attacks* (CPA) as its encryption algorithm is deterministic. We use the same idea for the rest of the paper as well. Because of the limited space, we do not discuss the details of the security of each encryption scheme.

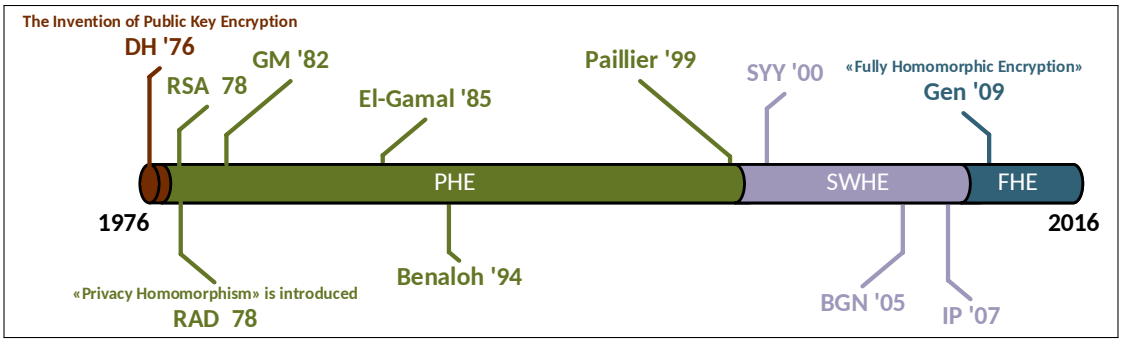


Figure 7.2: Timeline of HE schemes until Gentry's first FHE scheme

is only homomorphic over multiplication. Hence, it does not allow the homomorphic addition of ciphertexts.

Goldwasser-Micali

GM proposed the first probabilistic public key encryption scheme proposed in [GM82]. The GM cryptosystem is based on the hardness of *quadratic residuosity problem* [Pai99b]. Number a is called *quadratic residue modulo* n if there exists an integer x such that $x^2 \equiv a \pmod{n}$. Quadratic residuosity problem decides whether a given number q is quadratic modulo n or not. GM cryptosystem is described as follows:

- *KeyGen Algorithm:* Similar to RSA, $n = pq$ is computed where p and q are distinct large primes and then, x is chosen as one of the quadratic nonresidue modulo n values with $(\frac{x}{n}) = 1$. Finally, (x, n) is published as the public key while (p, q) is kept as the secret key.
- *Encryption Algorithm:* Firstly, the message (m) is converted into a string of bits. Then, for every bit of the message m_i, a quadratic nonresidue value y_i is produced such that $gcd(y_i, n) = 1$. Then, each bit is encrypted to c_i as follows:

$$c_i = E(m_i) = y_i^2 x^{m_i} \pmod{n}, \quad \forall m_i = \{0, 1\}, \tag{7.5}$$

where $m = m_0 m_1 ... m_r$, $c = c_0 c_1 ... c_r$ and r is the block size used for the message space and x is picked from ${Z_n}^*$ at random for every encryption, where ${\mathbb{Z}_n}^*$ is the multiplicative subgroup of integers modulo n which includes all the numbers smaller than r and relatively prime to r.

- *Decryption Algorithm:* Since x is picked from the set ${Z_n}^*$ $(1 < x \leq n-1)$, x is quadratic residue modulo n for only $m_i = 0$. Hence, to decrypt the ciphertext c_i, one decides whether c_i is a quadratic residue modulo n or not; if so, m_i returns 0, else m_i returns 1.

- *Homomorphic Property:* For each bit $m_i \in \{0, 1\}$,

$$\begin{aligned} E(m_1) * E(m_2) &= (y_1^2 x^{m_1} \pmod n) * (y_2^2 x^{m_2} \pmod n) \\ &= (y_1 * y_2)^2 x^{m_1+m_2} \pmod n = E(m_1 + m_2). \end{aligned} \tag{7.6}$$

The homomorphic property of the GM cryptosystem shows that encryption of the sum $E(m_1 \oplus m_2)$ can be directly calculated from the separately encrypted bits, $E(m_1)$ and $E(m_2)$. Since the message and ciphertext are the elements of the set $\{0, 1\}$, the operation is the same with exclusive-OR (XOR)[3] Hence, GM is homomorphic over only addition for binary numbers.

El-Gamal

In 1985, Taher Elgamal proposed a new public key encryption scheme [ElG85] which is the improved version of the original Diffie-Hellman Key Exchange [DH76] algorithm, which is based on the hardness of certain problems in discrete logarithm [Kev90]. It is mostly used in hybrid encryption systems to encrypt the secret key of a symmetric encryption system. The El-Gamal cryptosystem is defined as follows:

[3]XOR can be thought as binary addition.

- *KeyGen Algorithm:* A cyclic group G with order n using generator g is produced. In a cyclic group, it is possible to generate all the elements of the group using the powers of one of its own element. Then, $h = g^y$ computed for randomly chosen $y \in \mathbb{Z}_n{}^*$. Finally, the public key is (G, n, g, h) and x is the secret key of the scheme.

- *Encryption Algorithm:* The message m is encrypted using g and x, where x is randomly chosen from the set $\{1, 2, ..., n-1\}$ and the output of the encryption algorithm is a ciphertext pair $(c = (c_1, c_2))$:

$$c = E(m) = (g^x, mh^x) = (g^x, mg^{xy}) = (c_1, c_2), \tag{7.7}$$

- *Decryption Algorithm:* To decrypt the ciphertext c, first, $s = {c_1}^y$ is computed where y is the secret key. Then, decryption algorithm works as follows:

$$c_2 \cdot s^{-1} = mg^{xy} \cdot g^{-xy} = m. \tag{7.8}$$

- *Homomorphic Property:*

$$E(m_1) * E(m_2) = (g^{x_1}, m_1 h^{x_1}) * (g^{x_2}, m_2 h^{x_2}) = (g^{x_1+x_2}, m_1 * m_2 h^{x_1+x_2}) = E(m_1 * m_2). \tag{7.9}$$

As seen from this derivation, the El-Gamal cryptosystem is multiplicatively homomorphic. It does not support addition operation over ciphertexts.

Benaloh

Benaloh proposed an extension of the GM Cryptosystem by improving it to encrypt the message as a block instead of bit by bit [Ben94]. Benaloh's proposal was based on the higher residuosity problem. Higher residuosity problem (x^n) [Pai99b] is the generalization of quadratic residuosity problems (x^2) that is used for the GM cryptosystem.

- *KeyGen Algorithm:* Block size r and large primes p and q are chosen such that r divides $p-1$ and r is relatively prime to $(p-1)/r$ and $q-1$ (i.e., $gcd(r,(p-1)/r) = 1$ and $gcd(r,(q-1)) = 1$). Then, $n = pq$ and $\phi = (p-1)(q-1)$ are computed. Lastly, $y \in \mathbb{Z}_n{}^*$ is chosen such that $y^\phi \not\equiv 1 \mod n$, where $\mathbb{Z}_n{}^*$ is the multiplicative subgroup of integers modulo n which includes all the numbers smaller than r and relatively prime to r. Finally, (y,n) is published as the public key, and (p,q) is kept as the secret key.

- *Encryption Algorithm:* For the message $m \in Z_r$, where $Z_r = \{0,1,...,r-1\}$, choose a random u such that $u \in Z_n{}^*$. Then, to encrypt the message m:

$$c = E(m) = y^m u^r \pmod{n}, \tag{7.10}$$

  where the public key is the modulus n and base y with the block size of r.

- *Decryption Algorithm:* The message m is recovered by an exhaustive search for $i \in \mathbb{Z}_r$ such that

$$(y^{-i}c)^{\phi/r} \equiv 1, \tag{7.11}$$

  where the message m is returned as the value of i, i.e., $m = i$.

- *Homomorphic Property:*

$$\begin{aligned} E(m_1) * E(m_2) &= (y^{m_1}{u_1}^r \pmod{n}) * (y^{m_2}{u_2}^r \pmod{n}) \\ &= y^{m_1+m_2}(u_1 * u_2)^r \pmod{n} = E(m_1 + m_2 \pmod{n}). \end{aligned} \tag{7.12}$$

Homomorphic property of Benaloh shows that any multiplication operation on encrypted data corresponds to the addition on plaintext. As the encryption of the addition of the messages can directly be calculated from encrypted messages $E(m_1)$ and $E(m_2)$, the Benaloh cryptosystem is additively homomorphic.

Paillier

In 1999, Paillier [Pai99a] introduced another novel probabilistic encryption scheme based on *composite residuosity problem* [Jag12]. *Composite residuosity problem* is very similar to quadratic and higher residuosity problems that are used in GM and Benaloh cryptosystems. It questions whether there exists an integer x such that $x^n \equiv a \pmod{n^2}$ for a given integer a.

- *KeyGen Algorithm:* For large primes p and q such that $gcd(pq, (p-1)(q-1)) = 1$, compute $n = pq$ and $\lambda = lcm(p-1, q-1)$. Then, select a random integer $g \in \mathbb{Z}^*_{n^2}$ by checking whether $gcd(n, L(g^{\lambda \bmod n^2})) = 1$, where the function L is defined as $L(u) = (u-1)/n$ for every u from the subgroup $Z^*_{n^2}$ which is a multiplicative subgroup of integers modulo n^2 instead of n like in the Benaloh cryptosystem. Finally, the public key is (n, g) and the secret key is (p, q) pair.

- *Encryption Algorithm:*

  For each message m, the number r is randomly chosen and the encryption works as follows:

$$c = E(m) = g^m r^n \pmod{n^2}, \tag{7.13}$$

- *Decryption Algorithm:* For a proper ciphertext $c < n^2$, the decryption is done by:

$$D(c) = \frac{L(c^\lambda \pmod{n^2})}{L(g^\lambda \pmod{n^2})} \bmod n = m, \tag{7.14}$$

  where private key pair is (p, q).

- *Homomorphic Property:*

$$\begin{aligned} E(m_1) * E(m_2) &= (g^{m_1} {r_1}^n \pmod{n^2}) * (g^{m_2} {r_2}^n \pmod{n^2}) \\ &= g^{m_1+m_2}(r_1 * r_2)^n \pmod{n^2} = E(m_1 + m_2). \end{aligned} \tag{7.15}$$

This derivation shows that Paillier's encryption scheme is homomorphic over addition. In addition to homomorphism over the addition operation, Paillier's encryption scheme has some additional homomorphic properties, which allow extra basic operations on plaintexts $m_1, m_2 \in Z_{n^2}^*$ by using the encrypted plaintexts $E(m_1)$, $E(m_2)$ and public key pair (n, g):

$$E(m_1) * E(m_2) \pmod{n^2} = E(m_1 + m_2 \pmod{n}), \tag{7.16}$$

$$E(m_1) * g^{m_2} \pmod{n^2} = E(m_1 + m_2 \pmod{n}), \tag{7.17}$$

$$E(m_1)^{m_2} \pmod{n^2} = E(m_1 m_2 \pmod{n}). \tag{7.18}$$

These additional homomorphic properties describe different cross-relation between various operations on the encrypted data and the plaintexts. In other words, Equations (7.16), (7.17), and (7.18) show how the operations computed on encrypted data affects the plaintexts.

Others

Moreover, Okamoto-Uchiyama (OU) [OU98] proposed a new PHE scheme to improve the computational performance by changing the set, where the encryptions of previous HE schemes work. The domain of the scheme is the same as the previous public key encryption schemes, Z_n^*, however, Okamoto-Uchiyama sets $n = p^2 q$ for large primes p and q. Furthermore, Naccache-Stern (NS) [NS98] presented another PHE scheme as a generalization of Benaloh cryptosystem to increase its computational efficiency. The proposed work changed only the decryption algorithm of the scheme. Likewise, Damgard-Jurik (DJ) [DJ01] introduced another PHE scheme as a generalization of Paillier. These three cryptosystems preserve the homomorphic property while improving the original homomorphic schemes.

Table 7.1: Homomorphic properties of well-known PHE schemes

	Homomorphic Operation	
Scheme	**Add**	**Mult**
RSA [RSA78]		✓
GM [GM82]	✓	
El-Gamal [ElG85][4]		✓
Benaloh [Ben94]	✓	
NS [NS98]	✓	
OU [OU98]	✓	
Paillier [Pai99a]	✓	
DJ [DJ01]	✓	
KTX [KTX07]	✓	
Galbraith [Gal02]	✓	

Similarly, Kawachi (KTX) et al. [KTX07] suggested an additively homomorphic encryption scheme over a large cyclic group, which is based on the hardness of underlying lattice problems. They named the homomorphic property of their proposed scheme as *pseudohomomorphic*. Pseudohomomorphism is an algebraic property and still allows homomorphic operations on ciphertext, however, the decryption of the homomorphically operated ciphertext works with a small decryption error. Finally, Galbraith [Gal02] introduced a more natural generalization of Paillier's cryptosystem applying it on elliptic curves while still preserving the homomorphic property of the Paillier's cryptosystem. Homomorphic properties of well-known PHE schemes are briefly summarized in Table 7.1.

[4]The method to convert El-Gamal into an additively homomorphic encryption scheme is shown in [CGS97]. However, it is still PHE as it still supports only addition operation, not both at the same time.

7.2.2 Somewhat Homomorphic Encryption Schemes

There are useful SWHE examples [Yao82, SYY99, BGN05, IP07] in the literature before 2009. After the first plausible FHE published in 2009 [Gen09], some SWHE versions of FHE schemes were also proposed because of the performance issues associated with FHE schemes. We cover these SWHE schemes under the FHE section. In this section, we primarily focus on major SWHE schemes, which were used as a stepping stone to the first plausible FHE scheme.

BGN

Before 2005, all proposed cryptosystems' homomorphism properties were restricted to only either addition or multiplication operation i.e., SWHE schemes. One of the most significant steps toward an FHE scheme was introduced by Boneh-Goh-Nissim (BGN) in [BGN05]. BGN evaluates 2-DNF[5] formulas on ciphertext and it supports an arbitrary number of additions and one multiplication by keeping the ciphertext size constant. The hardness of the scheme is based on the *subgroup decision problem* [Gjø04]. Subgroup decision problem simply decides whether an element is a member of a subgroup G_p of group G of composite order $n = pq$, where p and q are distinct primes.

- *KeyGen Algorithm:* The public key is released as (n, G, G_1, e, g, h). In the public key, e is a bilinear map such that $e : G \times G \rightarrow G_1$, where G, G_1 are groups of order $n = q_1 q_2$. g and u are the generators of G and set $h = u^{q_2}$ and h is the generator of G with order q_1, which is kept hidden as the secret key.
- *Encryption Algorithm:* To encrypt a message m, a random number r from the set $\{0, 1, ..., n-1\}$ is picked and encrypted using the precomputed g and h as

[5]Disjunctive Normal Form with at most 2 literals in each clause.

follows:

$$c = E(m) = g^m h^r \mod n \tag{7.19}$$

- *Decryption Algorithm:* To decrypt the ciphertext c, one firstly computes $c' = c^{q_1} = (g^m h^r)^{q_1} = (g^{q_1})^m$ (Note that $h^{q_1} \equiv 1 \mod n$) and $g' = g^{q_1}$ using the secret key q_1 and decryption is completed as follows:

$$m = D(c) = \log_{g'} c' \tag{7.20}$$

  In order to decrypt efficiently, the message space should be kept small because of the fact that discrete logarithm can not be computed quickly.

- *Homomorphism over Addition:* Homomorphic addition of plaintexts m_1 and m_2 using ciphertexts $E(m_1) = c_1$ and $E(m_2) = c_2$ are performed as follows:

$$c = c_1 c_2 h^r = (g^{m_1} h^{r_1})(g^{m_2} h^{r_2}) h^r = g^{m_1+m_2} h^{r'}, \tag{7.21}$$

  where $r = r_1 + r_2 + r$ and it can be seen that $m_1 + m_2$ can be easily recovered from the resulting ciphertext c.

- *Homomorphism over Multiplication:* To perform homomorphic multiplication, use g_1 with order n and h_1 with order q_1 and set $g_1 = e(g, g)$, $h_1 = e(g, h)$, and $h = g^{\alpha q_2}$. Then, the homomorphic multiplication of messages m_1 and m_2 using the ciphertexts $c_1 = E(m_1)$ and $c_2 = E(m_2)$ are computed as follows:

$$\begin{aligned} c &= e(c_1, c_2) {h_1}^r = e(g^{m_1} h^{r_1}, g^{m_2} h^{r_2}) {h_1}^r \\ &= {g_1}^{m_1 m_2} {h_1}^{m_1 r_2 + r_2 m_1 + \alpha q_2 r_1 r_2 + r} = {g_1}^{m_1 m_2} {h_1}^{r'} \end{aligned} \tag{7.22}$$

It is seen that r' is uniformly distributed like r and so $m_1 m_2$ can be correctly recovered from resulting ciphertext c. However, c is now in the group G_1 instead of G. Therefore, another homomorphic multiplication operation is not allowed in

Table 7.2: Comparison of some well-known SWHE schemes before Gentry's work

	Evaluation Size	**Evaluation Circuit**	**Ciphertext Size**
Yao [Yao82]	arbitrary	garbled circuit	grows at least linearly
SYY [SYY99]	polly-many AND & one OR/NOT	NC^1 circuit	grows exponentially
BGN [BGN05]	unlimited add & 1 mult	2-DNF formulas	constant
IP [IP07]	arbitrary	branching programs	doesn't depend on the size of function

G_1 because there is no pairing from the set G_1. However, resulting ciphertext in G_1 still allows an unlimited number of homomorphic additions. Moreover, Boneh et al. also showed the evaluation of 2-DNF formulas using the basic 2-DNF protocol. Their protocol gives a quadratic improvement in terms of the protocol complexity over Yao's well-known garbled circuit protocol in [Yao82].

Others

In the literature of HE schemes, one of the first SWHE schemes is Polly Cracker scheme [FK94]. It allows both multiplication and addition operation over the ciphertexts. However, the size of the ciphertext grows exponentially with the homomorphic operation, especially multiplication operation is extremely expensive. Later more efficient variants [LdVP04, VL06] are proposed, but almost all of them are later shown vulnerable to attacks [Ste10, LdVMPT09]. Therefore, they are either insecure or impractical [Le03]. Recently, [AFFP11] introduced a Polly Cracker with Noise cryptosystem, where the homomorphic addition operations do not increase the ciphertext size while the multiplications square it.

Another idea of evaluating operations on encrypted data is realized over different sets. Sander, Young, and Yung (SYY) described first SWHE scheme over a semi-group, NC^1,[6] [SYY99], which requires less properties than a group. NC^1 is a complexity class which includes the circuits with poly-logarithmic depth and polynomial size. The proposed scheme supported polynomially many ANDing of

[6]NC stands for "Nick's Class" for the honor of Nick Pippenger

ciphertexts with one OR/NOT gate. However, the ciphertext size increased by a constant multiplication with each OR/NOT gate evaluation. This increase limits the evaluation of circuit depth. Yuval Ishai and Anat Paskin (IP) expanded the set to branching programs (aka Binary Decision Diagrams), which are the directed acyclic graphs where every node have two outgoing edges with labeled binary 0 and 1 [IP07]. In other words, they proposed a public key encryption scheme by evaluating the branching programs on the encrypted data. Moreover, Melchor et al. [MGH10] proposed a generic construction method to obtain a chained encryption scheme allowing the homomorphic evaluation of constant depth circuit over ciphertext. The chained encryption scheme is obtained from well-known encryption schemes with some homomorphic properties. For example, they showed how to obtain a combination of BGN [BGN05] and Kawachi et al. [KTX07]. As mentioned before, BGN allows an arbitrary number of additions and one multiplication while Kawachi's scheme is only additively homomorphic. Hence, the resulting combined scheme allows arbitrary additions and two multiplications. They also showed how this procedure is applied to the scheme in [MCG08] allowing a predefined number of homomorphic additions, to obtain a scheme which allows an arbitrary number of multiplications as well. However, in multiplication, ciphertext size grows exponentially while it is constant in a homomorphic addition. The summary of some well-known SWHE schemes is given in Table 7.2. As shown in Table 7.2, while in Yao, SYY, and IP cryptosystems, the size of the ciphertext grows with each homomorphic operation, in BGN it stays constant. This property of BGN is a significant improvement to obtain an FHE scheme. Accordingly, Gentry, Halevi, and Vaikuntanathan later simplified the BGN cryptosystem [GHV10]. In their version, the underlying security assumption is changed to hardness of the LWE problem. The BGN cryptosystem chooses input from a small set to decrypt correctly. In contrast,

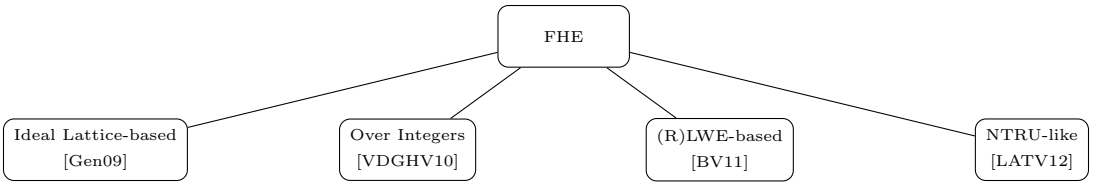


Figure 7.3: Main FHE families after Gentry's breakthrough

a recent scheme introduced in [GHV10] have much larger message space. Moreover, some of the attempts to obtain an FHE scheme based on SWHE schemes are reported as broken. For instance, vulnerabilities for [iF96, GP06, DF02] were reported in [CBW07, Wag03, CKN06], respectively.

7.2.3 Fully Homomorphic Encryption Schemes

An encryption scheme is called Fully Homomorphic Encryption (FHE) scheme if it allows an unlimited number of evaluation operations on the encrypted data and resulting output is within the ciphertext space. After almost 30 years from the introduction of privacy homomorphism concept [RAD78], Gentry presented the first feasible proposal in his seminal PhD thesis to a long term open problem, which is obtaining an FHE scheme [Gen09]. Gentry's proposed scheme gives not only an FHE scheme, but also a general framework to obtain an FHE scheme. Hence, a lot of researchers have attempted to design a secure and practical FHE scheme after Gentry's work.

Although Gentry's proposed ideal lattice-based FHE scheme [Gen09] is very promising, it also had a lot of bottlenecks such as its computational cost in terms of applicability in real life and some of its advanced mathematical concepts make it complex and hard to implement. Therefore, many new schemes and optimization have followed his work in order to address aforementioned bottlenecks. The security

of new approaches to obtain a new FHE scheme is mostly based on the hard problems on lattices.

A lattice is the linear combinations of independent vectors (basis vectors), $b_1, b_2, ..., b_n$. A lattice L is formulated as follows:

$$L = \sum_{i=1}^{n} \vec{b_i} * v_i \quad , v_i \in \mathbb{Z}, \tag{7.23}$$

where each vectors $b_1, b_2, ..., b_i$ is called a basis of the lattice L. The basis of a lattice is not unique. There are infinitely many bases for a given lattice. A basis is called "good" if the basis vectors are almost orthogonal and, otherwise it is called "bad" basis of the lattice [MR09]. Roughly, while good bases are typically long, bad bases are relatively shorter. Indeed, the lattice theory is firstly presented by Minkowski [Min68]. Then as a seminal work, Ajtai mentioned a class of random worst-case lattice problem in [Ajt96]. Two well-known modern problems suggested in [Ajt96] for lattice-based cryptosystems are Closest Vector Problem (CVP) and Shortest Vector Problem (SVP) [Pei16]. A year after, Goldreich, Goldwasser, and Halevi (GGH) [GGH97] proposed an important type of PKE scheme, whose hardness is based on *the lattice reduction problems* [Pei16]. Lattice reduction tries to find a good basis, which is relatively short and orthogonal, for a given lattice. In GGH cryptosystem, the public key and the secret key is chosen from "bad" and "good" basis of the lattice, respectively. The idea behind this choice is that CVP and SVP problems can easily be solved in polynomial time for the lattices with the known good bases. However, best known algorithms (for example LLL in [LLL82]) solve these problems in exponential time without knowing the good bases of the lattice. Hence, recovering the message from a given ciphertext is equal to solving the CVP and SVP problems. In GGH cryptosystem, the message is embedded to the noise to

obtain the ciphertext. In order to recover the message from ciphertext, the secret key (good basis) is used to find the closest lattice point.

Before Gentry's work, in [Reg06], cryptographers' attention is drawn to lattice-based cryptology and especially its great promising properties for post-quantum cryptology. Its promising properties are listed as its security proofs, efficient implementations, and simplicity. Moreover, another lattice-related problem, which gains popularity in last few years, especially after being used as a base to built an FHE scheme is LWE [Zha14]. One of the most significant works for lattice-based cryptosystems was studied in [HPS98], which presented a new PKE scheme and whose security is based on SVP on the lattice. In the SVP problem, given a basis of a lattice, the goal is to find the shortest nonzero vector in the lattice.

After Gentry's work, the lattices have become more popular among cryptography researchers. First, some works like [SV10] focused on just improving Gentry's ideal lattice-based FHE scheme in [Gen09]. Then, an FHE scheme over integers based on the Approximate-GCD problems is introduced [VDGHV10]. The main motivation behind the scheme is the conceptual simplicity. Afterwards, another FHE scheme whose hardness based on Ring Learning with Error (RLWE) problems is suggested [BV11]. The proposed scheme promises some efficiency features. Lastly, an NTRU-like FHE is presented for its promising efficiency and standardization properties [LATV12]. NTRUEncrypt is an old and strongly standardized lattice-based encryption scheme whose homomorphic properties are realized recently. So, these and similar attempts can be categorized into under four main FHE families as shown in Figure 7.3: (1) Ideal lattice-based [Gen09], (2) Over integers [VDGHV10], (3) (R)LWE-based [BV11], and (4) NTRU-like [LATV12]. In the following sections, we will articulate these four main FHE families in greater detail. And, we will also explore other follow-up works after these.

Ideal Lattice-based FHE schemes

Gentry's first FHE scheme in his PhD thesis [Gen09] is a GGH-type of encryption scheme, where GGH is proposed originally by Goldreich et al. [GGH97]. However, Gentry encrypted the message by embedding noise using double layer instead of one layer idea in GGH cryptosystem. Indeed, Gentry started his breakthrough work from SWHE scheme based on ideal lattices.

As mentioned earlier, an SWHE scheme can evaluate the ciphertext homomorphically for only a limited number of operations. After a certain threshold, the decryption function fails to recover the message from the ciphertext correctly. The amount of noise in the ciphertext must be decreased to transform the noisy ciphertext into a proper ciphertext. Gentry used genius blueprint methods called *squashing* and *bootstrapping* to obtain a ciphertext which allows a number of homomorphic operations to be performed on it. This processes can be repeated again and again. In other words, one can evaluate unlimited operations on the ciphertexts which make the scheme fully homomorphic.

As an initial construction, Gentry used ideals and rings without lattices to design the homomorphic encryption scheme, where an ideal is a property preserving subset of the rings such as even numbers. Then, each ideal used in his scheme was represented by the lattices. For example, an ideal I in $\mathbb{Z}[x]/(f(x))$ with $f(x)$ of degree n in an ideal lattice can easily be represented by a column of lattice with basis B_I of length n. Since the bases B_I will produce an $n \times n$ matrix. Gentry's SWHE scheme using ideals and rings is described below:

- *KeyGen Algorithm:* For the given ring R and the basis B_I of ideal I, $IdealGen(R, B_I)$ algorithm generates the pair of (B_J^{sk}, B_J^{pk}), where $IdealGen()$ is an algorithm outputting the relatively prime public and the secret key bases of the ideal

lattice with basis B_I such that $I + J = R$. A $Samp()$ algorithm is also used in key generation to sample from the given coset of the ideal, where a coset is obtained by shifting an ideal by a certain amount. Finally, the public key consists of $(R, B_I, B_J^{pk}, Samp())$ and the secret key only includes B_J^{sk}.

- *Encryption Algorithm:*

  For randomly chosen vectors $\vec{r}$ and $\vec{g}$, using the public key (basis) B_{pk} chosen from one of the "bad" bases of the ideal lattice L, the message $\vec{m} \in \{0,1\}^n$ is encrypted by:

$$\vec{c} = E(\vec{m}) = \vec{m} + \vec{r} \cdot B_I + \vec{g} \cdot B_J^{pk}, \tag{7.24}$$

  where B_I is basis of the ideal lattice L. Here, $\vec{m} + \vec{r} \cdot B_I$ is called "noise" parameter.

- *Decryption Algorithm:*

  By using the secret key (basis) B_J^{sk}, the ciphertext is decrypted as follows:

$$\vec{m} = \vec{c} - B_J^{sk} \cdot \lfloor (B_J^{sk})^{-1} \cdot \vec{c} \rceil \mod B_I, \tag{7.25}$$

  where $\lfloor \cdot \rceil$ is the *nearest integer function* which returns the nearest integers for the coefficients of the vector.

- *Homomorphism over Addition:* For the plaintext vectors $\vec{m_1}, \vec{m_2} \in \{0,1\}^n$, additive and multiplicative homomorphisms can be verified easily as follows:

$$\vec{c_1} + \vec{c_2} = E(\vec{m_1}) + E(\vec{m_2}) = \vec{m_1} + \vec{m_2} + (\vec{r_1} + \vec{r_2}) \cdot B_I + (\vec{g_1} + \vec{g_2}) \cdot B_J^{pk} \tag{7.26}$$

  It is clear that $\vec{c_1} + \vec{c_2}$ still preserves the format and is within the ciphertext space. And, to decrypt the sum of the ciphertext, one computes $(\vec{c_1} + \vec{c_2})$

mod B_J^{pk} which is equal to $\vec{m_1} + \vec{m_2} + (\vec{r_1} + \vec{r_2}) \cdot B_I$ for the ciphertexts whose noise amount is smaller than $B_J^{pk}/2$. Then the decryption algorithm works properly and recovers the sum of the message $m_1 + m_2$ correctly by taking the modulo B_I of the noise.

- *Homomorphism over Multiplication:* Similarly for the multiplication, after setting $\vec{e} = \vec{m} + \vec{r} \cdot B_I$, the homomorphic property can be expressed as follows:

$$\vec{c_1} \times \vec{c_2} = E(\vec{m_1}) \times E(\vec{m_2}) = \vec{e_1} \times \vec{e_2} + (\vec{e_1} \times \vec{g_2} + \vec{e_2} \times \vec{g_1} + \vec{g_1} \times \vec{g_2}) \cdot B_J^{pk} \quad (7.27)$$

where $\vec{e_1} \times \vec{e_2} = \vec{m_1} \times \vec{m_2} + (\vec{m_1} \times \vec{r_2} + \vec{m_2} \times \vec{r_1} + \vec{r_1} \times \vec{r_2}) \cdot B_I$. It can be easily verified that the multiplication operation on ciphertexts yields the output still within the ciphertext space. It is said that if the noise $|\vec{e_1} \times \vec{e_1}|$ is enough small enough the multiplication of plaintexts $\vec{m_1} \times \vec{m_2}$ can be correctly recovered from the multiplication of ciphertexts $\vec{c_1} \times \vec{c_2}$.

To have a better understanding of the "noise" concept, let us consider the encryption scheme over integers[7]. The encryption of the bit b is the ciphertext $c = b + 2r + kp$, where the key $p > 2N$ is an odd integer and r is a random number from the range $(-n/2, n/2)$ and k is an integer. The decryption works as follows: $b \leftarrow (c \mod p) \mod 2$, where $(c \mod p)$ is called as *noise parameter*. If the noise parameter exceeds $|p/2|$, the decryption fails since $(c \mod p)$ is not equal to $b + 2r$ anymore. And, the noise parameter grows linearly with each addition and exponentially with each multiplication operation. If the noise parameter is very close to a lattice point (i.e., $(c \mod p) << |p/2|$), further addition and multiplication operations are still allowed. This is why Gentry's ideal lattice based scheme is called Somewhat Homomorphic "for now" allowing only limited number of operations. Since the noise grows much faster with the multiplication operations, the number of

[7]Further details about FHE over integers will be explained in Section 7.2.3.

multiplication operations before exceeding the threshold is more limited. In order to make the scheme fully homomorphic, the bootstrapping technique was introduced by Gentry. However, the bootstrapping process can be applied to the bootstrappable ciphertexts, which are noisy and have small circuit depth. The depth of the circuit is related to the maximum number of operations. Hence, first the circuit depth is reduced with *squashing* to the degree that the decryption can handle properly.

Squashing: Gentry's bootstrapping technique is allowed only for the decryption algorithms with small depth. Therefore, he used some "tweaks" to reduce the decryption algorithm's complexity. This method is called *squashing* and works as follows:

First, choose a set of vectors, whose sum equals to the multiplicative inverse of the secret key ($(B_J^{sk})^{-1}$). If the ciphertext is multiplied by the elements of this set, the polynomial degree of the circuit is reduced to the level that the scheme can handle. The ciphertext is now "bootstrappable". Nonetheless, the hardness of the recovering the secret key is now based on the assumption of Sparse Subset Sum Problem (SSSP) [HPSS08]. This basically adds another assumption to the provable security of the scheme.

Bootstrapping: Bootstrapping is basically "recrypting" procedure to get a "fresh" ciphertext from the noisy ciphertext corresponding to the same plaintext. A scheme is called *bootstrappable* if it can evaluate its own decryption algorithm circuit [Gen09]. First, the ciphertext is transformed into a bootstrappable ciphertext using squashing. Then, by applying bootstrapping procedure, one gets a "fresh" ciphertext. The bootstrapping works as follows: First, it is assumed that two different public and secret key pairs are generated, $(pk1, sk1)$ and $(pk2, sk2)$ and while the

secret keys are kept by the client, the public keys are shared with the server. Then, the encryption of the secret key, $Enc_{pk1}(sk1)$, is also transmitted to the server, which already has $c = Enc_{pk1}(m)$. Since the above obtained SWHE scheme can evaluate its own decryption algorithm homomorphically, the noisy ciphertext is decrypted homomorphically using $Enc_{pk1}(sk1)$. Then, the result is encrypted using a different public key $pk2$, i.e., $Enc_{pk2}(Dec_{sk1}(c)) = Enc_{pk2}(m)$. Since the scheme is assumed semantically secure, an adversary can not distinguish the encryption of the secret key from the encryption of 0. The last ciphertext can be decrypted using $sk2$, which is kept secret by the client, i.e., $Dec_{sk2}(Enc_{pk2}(m)) = m$. In brief, first the homomorphic decryption of the noisy ciphertext removes the noise, and then the new homomorphic encryption introduces new small noise to the ciphertext. Now, the ciphertext is like just encrypted. Further homomorphic operations can be computed on this "fresh" ciphertext until reaching again to a threshold point. Note that Gentry's bootstrapping method increases the computational cost noticeably and becomes a major drawback for the practicality of FHE. In a nutshell, starting from constructing a SWHE scheme and then squashing method to reduce the circuit depth of decryption algorithm and the bootstrapping to obtain fresh ciphertext completes the creation of an FHE scheme. Hence, one can apply bootstrapping repetitively to compute an unlimited number of operations on the ciphertexts to successfully have an FHE scheme.

After Gentry's original scheme, some of the follow-up works tried to generally improve Gentry's original work. In [Gen09], Gentry's key generation algorithm is used for a particular purpose only and the generation of an ideal lattice with a "good" basis is left without a solution. Gentry introduced a new $KeyGen$ algorithm in [Gen10] and improved the security of the hardness assumption of SSSP by presenting a quantum worst case/average case reduction. However, a more aggres-

sive analysis of the security of SSSP was completed by Stehle and Steinfeld [SS10]. They also suggested a new probabilistic decryption algorithm with lower multiplicative degree, which is square root of previous decryption circuit degree. Moreover, a new FHE scheme, which was a variant of Gentry's scheme was introduced in [SV10]. The scheme uses smaller ciphertext and key sizes than Gentry's scheme without sacrificing the security. Some later works [GH11, SS11a, OYKU10] focused on the optimizations in the key generation algorithm in order to implement the FHE efficiently. Moreover, Mikuš proposed a new SWHE scheme with bigger plaintext space to improve the number of homomorphic operations with a slight increase in complexity of the key generation algorithm [Mik12].

FHE schemes Over Integers

In 2010, one year after Gentry's original scheme, another SWHE scheme is presented in [VDGHV10] which suggests Gentry's ingenious bootstrapping method in order to obtain an FHE scheme. The proposed scheme is over integers and the hardness of the scheme is based on the *Approximate-Greatest Common Divisor* (AGCD) problems [GGM16]. AGCD problems try to recover p from the given set of $x_i = pq_i + r_i$. The primary motivation behind the scheme is its conceptual simplicity. A symmetric version of the scheme is probably one of the simplest schemes. The proposed symmetric SWHE scheme is described as follows:

- *KeyGen Algorithm:* For the given security parameter λ, a random odd integer p of bit length η is generated.

- *Encryption Algorithm:* For a random large prime numbers p and q, choose a small number $r << p$. Then, the message $m \in \{0, 1\}$ is encrypted by:

$$c = E(m) = m + 2r + pq, \tag{7.28}$$

where p is kept hidden as private key and c is the ciphertext.

- *Decryption Algorithm:* The ciphertext can be decrypted as follows:

$$m = D(c) = (c \mod p) \mod 2. \tag{7.29}$$

Decryption works properly only if $m + 2r < p/2$. This actually restricts the depth of the homomorphic operations performed on the ciphertext. Then, Dijk et al. used Gentry's squashing and bootstrapping techniques to make the scheme fully homomorphic. The homomorphic properties of the scheme can be shown easily as follows:

- *Homomorphism over addition:*

$$E(m_1)+E(m_2) = m_1+2r_1+pq_1+m_2+2r_2+pq_2 = (m_1+m_2)+2(r_1+r_2)+(q_1+q_2)q. \tag{7.30}$$

The output clearly falls within the ciphertext space and can be decrypted if the noise $|m_1 + 2r_1 + m_2 + 2r_2| < p/2$, where p is the private key. Since $r_1, r_2 << p$, various number of additions can still be performed on ciphertext before noise exceeds $p/2$.

- *Homomorphism over Multiplication:*

$$E(m_1)E(m_2) = (m_1+2r_1+pq_1)(m_2+2r_2+pq_2 = m_1m_2+2(m_1r_2+m_2r_1+2r_1r_2)+kp. \tag{7.31}$$

The output preserves the format of original ciphertexts and holds the homomorphic property. The encrypted data can be decrypted if the noise is smaller

than half of the private key, i.e., $|m_1 m_2 + 2(m_1 r_2 + m_2 r_1 + 2 r_1 r_2)| < p/2$. The noise grows exponentially with the multiplication operation. This puts more restriction over homomorphic multiplication operation than addition.

In fact, the scheme presented so far [VDGHV10] was the symmetric version of the homomorphic encryption. Transforming the underlying symmetric HE scheme into an asymmetric HE scheme is also presented in [VDGHV10]. It is enough to compute many "encryptions of zero" $x_i = pq_i + 2r_i$, where p is private key. Then, many x_is are shared as the public key. To encrypt the message with the public key, it is enough to add the message to a subset sum of $x_i s$. Same decryption is used to decrypt the ciphertext. As there is no efficient algorithm to recover p from the given x_is in polynomial time, the scheme is considered as secure. The scheme is now basically a public key encryption scheme, since it uses different keys to encrypt and decrypt.

The FHE scheme proposed in [VDGHV10] is conceptually very simple. However, this simplicity comes at a cost in computations. So, the scheme is not very efficient. Hence, some early attempts directly tried to improve the efficiency. For example, some follow-up optimizations focused on reducing the size of public keys [CMNT11a] ($O(\lambda^{10}) \rightarrow O(\lambda^7)$), [CNT12] ($O(\lambda^7) \rightarrow O(\lambda^5)$, [YXWT12] ($O(\lambda^5) \rightarrow O(\lambda^3)$. A more efficient public key generation [RK12b] and re-encryption [CBH14] are other suggested works without reducing the security of the scheme. Later, an important variant, which is batch FHE over integers, was proposed [CCK$^+$13] (merged version of [CLT13] and [KLYC13]). Batch FHE has the ability to pack multiple ciphertexts into a single ciphertext. Moreover, the proposed scheme provides two options for the hardness of the base problem: Decisional AGCD and Error-free AGCD. In [CCK$^+$13], it is also shown how to achieve recryption operation in parallel l-slots.

Some further approaches for FHE schemes over integers are also proposed: a new scale invariant FHE over integers [CLT14], a new scheme with integer plaintexts [RK12a], a new SWHE scheme for computing arithmetic operations on large integer numbers without converting them into bits [PAD12], a new symmetric FHE without bootstrapping [AGS14], and a new FHE for non-binary message spaces [NK15]. All these schemes improved FHEs over integers in the way that their names imply.

LWE-based FHE schemes

Learning with Error (LWE) is considered as one of the hardest problems to solve in practical time for even post-quantum algorithms. First, it was introduced by Oded Regev as an extension of "learning from parity with error" problem [Reg09]. Regev reduced the hardness of worst-case lattice problems like SVP to LWE problems, which means that if one can find an algorithm that can solve LWE problem in an efficient time, the same algorithm will also solve the SVP problem in an efficient time. Since then, it is one of the most attractive and promising topics for post-quantum cryptology with its relatively small ciphertext size. Lyubashevsky et al. suggested another significant improvement on the LWE problem which may lead to a new applications by introducing ring-LWE (RLWE) problem [LPR13]. The RLWE problem is an algebraic variant of LWE, which is more efficient for practical applications with strong security proofs. They proved that the RLWE problems are reducible to worst-case problems on ideal lattices, which is hard for polynomial-time quantum algorithms.

In the LWE-based FHE schemes, an important step towards to a practical FHE scheme is made in [BV11]. Brakerski and Vaikuntanathan established a new SWHE scheme based on Ring-Learning with Error (RLWE) to take advantage of the efficiency feature of RLWE [BV11]. In other words, although both LWE and

RLWE problems can be used as the hardness assumption of an FHE scheme, RLWE shows better performance. Then, the scheme uses Gentry's blueprint squashing and bootstrapping techniques to obtain an FHE scheme. They used polynomial-LWE (PLWE), which is simplified version of RLWE. PLWE is also reducible to worst-case problems such as SVP on ideal lattices. The schemes proposed after [BV11] is also called second generation FHE schemes.

Below, for the sake of simplicity, as we did in the previous part, we first show symmetric version.

Notation: A very common notation is that $\langle a, b\rangle$ is used to denote the inner product of vectors a and b. Moreover, $d \xleftarrow{\$} \mathcal{D}$ denotes that d is randomly assigned by an element from the distribution $\mathcal{D}$ and $\mathbb{Z}[x]/(f(x))$ denotes the ring of all polynomials modulo $f(x)$. The ring of polynomials modulo $f(x)$ with coefficients in $\mathbb{Z}_q$ is denoted with $R_q \equiv \mathbb{Z}_q[x]/(f(x))$. Finally, χ denotes an error distribution over the ring R_q.

The symmetric version of the underlying scheme is given as follows:

- *KeyGen Algorithm:* An element of the ring is chosen as a secret key from the error distribution, i.e., $s \xleftarrow{\$} \chi$. Then, the secret key vector is described as $\vec{s} = (1, s, s^2, ..., s^D)$ for an integer D.
- *Encryption Algorithm:* After choosing a random vector $a \xleftarrow{\$} {R_q}^n$ and the noise $e \xleftarrow{\$} \chi$, the message m is encrypted by:

$$\vec{c} = (c_0, c_1) = (as + te + m, -a) \tag{7.32}$$

where $\vec{c} \in R_q^2$.

- *Decryption Algorithm:* In order to decrypt the ciphertext to recover the message, it can be easily computed that:

$$m = \langle \vec{c}, \vec{s} \rangle \pmod{t}. \tag{7.33}$$

  Decryption works properly if $\langle \vec{c}, \vec{s} \rangle$ is smaller than $q/2$. Furthermore, in order to make the scheme asymmetric, it is sufficient to generate a random set of pairs $(a, as+te)$. Also, the homomorphic property of the scheme is very similar to those in [Gen09] and [VDGHV10].

- *Homomorphism over Addition:*

$$E(m)+E(m') = (c_0+c_0', c_1+c_1') = ((a+a')s+t(e+e')+(m+m'), -(a+a')), \tag{7.34}$$

  Similar to previous schemes, decryption works if the noise is small. And, it is clear that homomorpically added ciphertexts keep the format of the original ciphertexts and stay within the ciphertext space.

- *Homomorphism over Multiplication:*

$$E(m)+E(m') = (c_0c_0', c_1c_1') = (-a's^2+(c_0'a+c_0a')s+t(2ee'+em'+e'm)+mm'). \tag{7.35}$$

  The output seems almost like a ciphertext, but it still can be decrypted correctly with the expense of a new cost by adding a new term to ciphertext.

Brakerski and Vaikuntanathan made their scheme fully homomorphic using Gentry's blueprint squashing and bootstrapping. They also showed their SWHE scheme is circular secure (aka Key-Dependent message (KDM) security) with respect to linear functions of the secret key, i.e., the encryption can successfully keep secure linear functions of its own secret key.

After the proposed BGN-type cryptosystem based on LWE, which is additively homomorphic and allowing only one multiplication operation in [GHV10], Brakerski and Vaikuntanathan proposed another SWHE scheme based on standard LWE problems using *re-linearization* technique [BV14a]. Re-linearization makes the long ciphertexts, which are the output of the homomorphic evaluation, regular size. Another important contribution in this work is the dimension-modulus reduction, which does not require an SSSP assumption and squashing method used in Gentry's original framework.

As discussed earlier, Gentry's bootstrapping method is a creative method to obtain an FHE scheme, however, it comes with a huge cost. A *leveled*-FHE scheme without using the bootstrapping technique was introduced by [BGV14]. Levelled FHE can evaluate homomorphic operations for only a predetermined circuit depth level. Brakerski et al. [BGV14] also showed that their scheme with bootstrapping still provides better performance than the one without bootstrapping and also suggested the batching as an optimization. To achieve batching, "modulus switching" technique is used iteratively to keep the noise size constant. Then, Brakerski removed the necessity of modulus switching in [Bra12]. In Brakerski's new scale invariant FHE scheme [Bra12], contrary to the existing FHE schemes, the noise grows linearly with the evaluation of homomorphic operations instead of exponentially and the scheme is based on the hardness of *GapSVP problem* [Pei16]. GapSVP problem is roughly deciding the existence of a shorter vector than the vector with length d for a given lattice basis B. The result returns simply yes or no. Then, Fan and Vercauteren optimized the Brakerski's scheme by changing the based assumption to RLWE problem [FV12a]. Some other modifications to [Bra12] focused on reducing the overhead of key switching and faster evaluation of homomorphic operations [WWL12] and using re-linearization to improve efficiency [ZXJ$^+$14].

Recently, by [GSW13] a significant FHE scheme was introduced claiming three important properties: simpler, faster, and attribute-based FHE. The scheme is simpler and faster due to the "approximate eigenvector" method replacing the relinearization technique. In this method, by keeping only some parameters small, the format of the ciphertext can be preserved under the evaluation of homomorphic operations. In the previous schemes which use the bootstrapping technique, the secret key (evaluation key) of the user is sent to the cloud to evaluate the ciphertext homomorphically for the bootstrapping. In contrast, [GSW13] eliminates that need and leads to propose the first identity-based FHE scheme, which allows homomorphic evaluation by only a target identity having the public parameters. Then, Brakerski and Vaikuntanathan followed [GSW13] to construct an FHE scheme secure under a polynomial LWE assumption [BV14b]. It is shown that the proposed scheme is as secure as any other lattice-based PKE scheme. Recently, Paindavoine and Vialla showed a way of minimizing the number of required bootstrapping based on the linear programming techniques that can be applied to [GSW13] as well.

In addition to more recently proposed LWE-based FHE schemes in [ZXJ+14, CWZX14, ZYZW16, WWL15b], some optimizations focused on better (faster) bootstrapping algorithms [ASP13, ASP14], speeding homomorphic operations [GHPS12], and a new extension to FHE for multi-identity and multi-key usage [CM15]. More recently, a new efficient SWHE scheme based on the polynomial approximate common divisor problem is presented in [CHLR16]. The presented scheme in [CHLR16] can handle efficiently large message spaces.

NTRU-like FHE schemes

To obtain a practical and applicable FHE scheme, one of the crucial steps is taken by showing the construction of an FHE scheme from NTRUEncrypt, which is an

old encryption scheme proposed by Hoffstein, Pipher, and Silvermanin in [HPS98]. Specifically, how to obtain a multi-key FHE from the NTRUEncrypt (called NTRU) was shown by [LATV12]. NTRU encryption scheme is one of earliest attempts based on lattice problems. Compared with RSA and GGH cryptosystems, NTRU improves the efficiency significantly in both hardware and software implementations. However, there were security concerns for 15 years until the study done by [SS11b]. They reduced the security of the scheme to standard worst-case problems over ideal lattices by modifying the key generation algorithm. Since the security of the scheme is improved, efficiency, easy implementation, and standardization issues attract researchers' interest again. López-Alt et al. used the NTRU encryption scheme to obtain a practical FHE [LATV12] with three differences. First, the set from which the noise is sampled is changed from a deterministic set to a distribution. Second, the modification introduced in [SS11b], which makes the scheme more secure, is used and third, the parameters are chosen to allow fully homomorphism. Their proposed NTRU-like encryption scheme in [LATV12] is as follows:

- *KeyGen Algorithm:* For chosen sampled polynomials f' and g from a distribution χ (specifically, a discrete Gaussian distribution), it is set $f = 2f' + 1$ to get $f \equiv 1 \pmod 2$ and f is invertible. Then, the secret key $sk = f \in R$ and public key $pk := h = 2gf^{-1} \in R_q$.
- *Encryption Algorithm:* For chosen samples s and e from the same distribution χ, the message m is encrypted by:

$$c = E(m) = hs + 2e + m, \tag{7.36}$$

  where the ciphertext $c \in R_q$.

- *Decryption Algorithm:* The ciphertext can easily be decrypted as follows:

$$m = D(c) = fc \pmod 2, \tag{7.37}$$

where $fc \in R_q$. The correctness of the scheme can be verified using $h = 2gf^{-1}$ and $f \equiv 1 \pmod 2$. Moreover, the scheme proposed by López-Alt et al. is a new type of FHE scheme, which is called multi-key FHE. Multi-key FHE has the ability to evaluate on ciphertexts which are encrypted with independent keys, i.e., each user can encrypt data with her own public key and a third party can still perform a homomorphic evaluation on these ciphertexts. The only interaction required between the users is to obtain a "joint secret key". The homomorphically evaluated ciphertext is decrypted by using the joint secret key, which is obtained by using all involved secret keys. The message m_i is encrypted by using public key $h_i = 2g_i{f_i}^{-1}$ with the formula, $c_i = h_i s_i + 2e_i + m_i$. The multikey homomorphism properties for two party computation is shown using joint secret key $f_1 f_2$.

- *Multi-key Homomorphism over Addition:*

$$\begin{aligned} f_1 f_2(c_1 + c_2) =& 2(f_1 f_2 e_1 + f_1 f_2 e_2 + f_2 g_1 s_1 + f_1 g_2 s_2) + f_1 f_2(m_1 + m_2) \\ =& \, 2e_{add} + f_1 f_2(m_1 + m_2) \end{aligned} \tag{7.38}$$

- *Multi-key Homomorphism over Multiplication:*

$$\begin{aligned} f_1 f_2(c_1 c_2) =& 2(2g_1 g_2 s_1 s_2 + g_1 s_1 f_2(2e_2 + m_2) + g_2 s_2 f_1(2e_1 + m_1) \\ & + f_1 f_2(e_1 m_2 + e_2 m_1 + 2e_1 e_2)) + f_1 f_2(m_1 m_2) \\ =& 2e_{mult} + f_1 f_2(m_1 m_2) \end{aligned} \tag{7.39}$$

Here, it is seen that multi-key homomorphic operation increases noise more than a single key homomorphic evaluation. However, $m_1 + m_2$ and $m_1 m_2$ can still be recovered correctly using the jointly obtained secret key since f, g, s, e all are sampled from the bounded distribution χ. In other words, the decryption still works if the each of the noise parameters e_{add} and e_{mult} are smaller than $|p/2|$.

As observed in all of the FHE schemes presented in detail in our work, since in [LATV12] noise grows with homomorphic operations on encrypted data, the proposed scheme is actually an SWHE scheme. To make it fully homomorphic, López-Alt et al. also (like all others above) used Gentry's bootstrapping technique. However, to apply bootstrapping, one first needs to make the underlying SWHE scheme bootstrappable. For this reason, first modulus reduction technique described in [Bra12, BV14a] was used. Then, the final scheme was named a leveled-FHE because it had the ability to deal only a limited number of public keys. Although the number of parties that can be used in homomorphic operations is limited, the complexity of circuit that can be used in homomorphic operations is still independent of the number of parties that can join the communication.

Another issue to be taken account in [LATV12] is the assumptions. Specifically, two assumptions are used in the scheme proposed by Lopez-Alt et al. First is RLWE problems and second is Decisional Small Polynomial Ratio (DSPR). Though RLWE is well-studied and about being a standard problem, DSPR assumption is a non-standard one. Hence, in [BLLN13], Bos et al. showed how to modify [LATV12] to remove DSRP assumption. While removing DSRP assumption, the *tensoring* technique introduced in [Bra12] is used to restrict the noise increase during homomorphic operations. However, the tensoring technique used to avoid DSRP assumption results in a large evaluation key and a complicated key switching procedure, which makes the scheme impractical. A practical variant of their scheme, which reintroduces the DSRP assumption is also presented in the same work. However, it is later shown that the optimizations and parameter selection that yield a significant increase in the performance makes it vulnerable to sub-field lattice attacks [ABD16]. The attack shown by Albrecht et al. affected not only [BLLN13], but every other NTRU-like scheme, which relies on DSRP problem and whose parameters (e.g., se-

cret key, modulus) are chosen poorly. Finally, in [DS16], a modified NTRU-like FHE scheme, which does not require the DSRP assumption, thereby secure against subfield lattice attacks, is proposed. Another attractive feature of the new FHE scheme is that it also does not require the use of evaluation key during the homomorphic operations. The new scheme is based on [SS11b] and it uses a *Flattening* noise management technique adopted from the flattening technique of [GSW13].

Two follow-up interesting works also improved the NTRU-like FHE using different techniques. While one of them focuses on a customized and a generic bit-sliced implementation of NTRU-like FHE schemes [DHS16] and the other suggests the use of GPU [DDS14]. Furthermore, in [DHS16], the AES circuit is chosen to evaluate the homomorphic operations, which is faster than the proposed one in [GHS12]. Other improvements on hardware implementations of NTRU-like FHE schemes are more recently published in [LW15, DÖSS15]. Another NTRU-like FHE scheme was suggested in [RC14]. They used the bootstrapping proposed in [ASP13] and "double-CRT" proposed in [GHS12] to modify the representation of the ciphertexts in more efficient way.

7.3 Implementations of SWHE and FHE schemes

The ultimate goal with different HE schemes is to obtain an unbounded and practical FHE scheme. PHE schemes and SWHE schemes proposed before Gentry's breakthrough work in 2009 were stepping stone towards that goal. Nonetheless, they are restricted in terms of the areas that can be applied. However, the SWHE schemes proposed after Gentry's work are mostly the part of the FHE schemes rather than a different scheme. Moreover, a bounded (level) FHE can also be called as SWHE scheme. Hence, it is not possible to separate SWHE and FHE schemes for the works

Table 7.3: "Fully" implemented FHE schemes

Scheme Information		Platform	Parameters		Running Times				
Implemented Scheme	**Base Scheme**	**Software**	**Security parameter, λ**	**dimension, n**	**PK size**	**KeyGen**	**Enc**	**Dec**	**Recrypt**
GH11 [GH11]	Gen09 [Gen09]	C/C++	72	33768	2.25 GB	2.2 h	3 min (SWHE)	0.66 s (SWHE)	31 min
CMNT11 [CMNT11a]	DGHV10 [VDGHV10]	Sage 4.5.3	72	7897	802 MB	43 min	2 min 57 s	0.05 s	14 min 33 s
CNT12 (with compressed PK) [CNT12]	DGHV10 [VDGHV10]	Sage 4.7.2	72	7897	10.3 MB	10 min	7 min 15 s	0.05 s	11 min 34 s
CNT12 (leveled) [CNT12]	DGHV10 [VDGHV10]	Sage 4.7.2	72	5700	18 MB	6 min 18 s	3.4 s	0.00 s	2 h 27 min

proposed after Gentry's work. In this section, we summarize the implementations of the SWHE and FHE schemes, which can lead to the new works and speed up the follow-up works, proposed after Gentry's work.

Implementation of a cryptographic scheme is the middle step between designing the scheme and applying it to a real life service and it provides a realistic performance assessment of the designed scheme. Although some new proposed FHE schemes have increased the efficiency and performance of the implementations significantly, the overhead and cost of the FHE implementations are still too high to be applied transparently in a real life service without disturbing the user.

"Fully" implemented FHE schemes

After solving the long term open problem of designing a fully homomorphic scheme [Gen09], many new fully homomorphic scheme proposals were tested with implementation. In a very first attempt, Smart and Vercauteren implemented their scheme in [SV10], which is a variant of Gentry's original scheme. However, their key generation takes hours up to $N = 2^{11}$, where N is the lattice dimension and does not generate the key pairs after $N = 2^{11}$. More importantly, their implementation did not include the bootstrapping procedure. Hence, it is actually a SWHE scheme as it was implemented. Then, Craig Gentry and Shai Halevi [GH11] succeeded to implement the FHE scheme first time by continuing the way that Smart and Vercauteren had started. The running times for the implementation in [23] and other proposed FHE implementations which are evaluated over random depth circuits are given in

Table 7.3. Moreover, Gentry and Halevi in [GH11] introduced some optimizations and simplifications on the squashing process to obtain a bootstrappable scheme. In their implementation, they showed four security levels: toy, small, medium, and large. They suggested that the large parameter settings are practically secure, which have a lattice dimension of 2^{15}. However, the performance of the implementation is very inefficient in practical terms. For the large parameter setting, a key pair was generated at 2.2 hours and public key size was 2.25 GB. Recrypting the ciphertexts (bootstrapping) took 31 minutes. After that, in [CMNT11a], an integer variant of the FHE scheme introduced originally in [VDGHV10] was implemented. In this implementation, the key generation takes 43 min, and the public key size is 802 MB. The implementation showed that the same security level can be achieved with a much simpler scheme. (The difference comes from the different definitions of security levels). Later, Coron et al. in a different work [CNT12] improved public key size to 10 MB, key generation to 10 minutes, and recryption procedure to 11 min 34 seconds using the similar parameter settings in [CMNT11a]. This performance is obtained using a compression technique on the public key. In [CNT12], a leveled DGHV scheme is also implemented with slightly worse performance. Yuanmi Chen and Phong Q. Nguyen [CN12] proposed an algorithm to break the scheme in [CNT12], which is faster than exhaustive search. This work showed that the security level of the scheme proposed in [CNT12] is much lower than the scheme proposed in [GH11].

FHE implementation for "Low-depth" circuits

The second type of FHE implementations tried to implement leveled-FHE schemes for small depth circuits with given run time for isolated and composed addition and multiplication [NLV11, BLLN13, LN14, RC14]. The comparisons for these small-

Table 7.4: FHE implementations for "Low-depth" circuits

Scheme Information		Platform	Parameters		Running times			
Implemented Scheme	**Base Scheme**	**Software**			**Enc**	**Dec**	**Mult**	**Add**
NLV11 [NLV11]	BV11 [BV11]	Magma	$w = 2^{32}$	q=127	756 ms	57 ms	1590 ms	4 ms
YASHE (by BLLN13 [BLLN13])	LTV12 [LATV12]	C/C++	$t = 2^{10}$	q=130	27 ms	5 ms	31 ms	0.024 ms
YASHE (by LN14a [LN14])	LTV12 [LATV12]	C/C++	$w = 2^{32}$	q=130	16 ms	15 ms	18 ms	0.7 ms
FV (by LN14a [LN14])	BV11 [BV11]	C/C++	$w = 2^{32}$	q=130	34 ms	16 ms	59 ms	1.4 ms
RC14 [RC14]	LTV12 [LATV12]	Matlab	$n = 2^{10}$	t=1	12 ms	3.36 ms	100 ms	0.56 ms

depth FHE implementations are given at Table 7.4. Since the performance of the state of the art was unsatisfactory, as an early attempt, a relatively simpler FHE, which allows only a few homomorphic multiplication operations was implemented in [NLV11]. Later, this performance was improved by Bos et al. [BLLN13] due to the new method to evaluate the homomorphic multiplication operation. Moreover, unlike [NLV11], in [BLLN13] the underlying scheme was implemented in C programming language to avoid the unwelcome overhead due to the computer algebra system. Then, a similar performances with [BLLN13] is obtained. Recently, a significant improvement is made by using double-CRT in the representation of ciphertexts and used parallelism to accelerate the implementation in Matlab [RC14].

"Real world" complex FHE implementations

In contrast to above schemes, which are either proof of concept or small-depth implementations, the authors in [GHS12] implemented FHE for the first time to evaluate the circuit complex enough for a real life application. In [GHS12] Gentry et al. implemented a variant of BGV scheme proposed in [SV14][8], which is a leveled FHE without bootstrapping, in order to evaluate AES circuit homomorphically. Actually, the idea of homomorphic evaluation of AES is first discussed in [NLV11] with the following scenario. A client first sends the key of AES by en-

[8]Later updated in [BGV14].

crypting with FHE, $FHE(K)$. Then, the client uploads the data by encrypting with AES only, $AES_K(m)$. When the cloud wants to evaluate the data homomorphically, it computes $FHE(AES_K(m))$ and decrypts AES homomorphically (blindfold) to obtain $FHE(m)$. After that, the cloud can compute every homomorphic operation on the data encrypted with FHE. The comparison of such more complex "real world" FHE implementations are presented in Table 7.5. A realization of how to achieve SIMD (single-instruction multiple-data) operations using homomorphic evaluation of AES is proposed by Smart and Vercauteren [SV14]. Later, some works [CLT13, MS13, CLT14, DHS16] also improved the performances of the homomorphic evaluation of AES circuit by applying the recent improvements and optimizations in theoretical side. In addition to the use of AES circuit to evaluate homomorphically, lightweight block ciphers such as Prince [DSES14], SIMON [LN14], and LowMC [ARS+15] are also proposed. In [MS13], Mella and Susella estimated the cost of some of the symmetric cryptographic primitives such as AES-128, SHA-256 hash function, Salsa20 stream cipher, and KECCAK sponge function. They concluded that AES is best suited for the homomorphic evaluation because of its low number of rounds and absence of integer operations and logical ANDs in its internals. However, in [MS13], only AES-128 is implemented.

Publicly available FHE implementations

Although all aforementioned implementations are published in the literature, unfortunately, only a few of them are publicly available to researchers. Some of the publicly available implementations are listed in Table 7.6. From publicly available implementations, HElib [HS14b] is the most important and widely utilized one. HElib implements the BGV scheme [BGV11] with Smart-Vercauteren ciphertext packing techniques and some new optimizations. The design and implementation of

Table 7.5: "Real world" complex FHE implementations

Scheme			Platform	Parameters		Running Times		
Implemented Scheme	**Base scheme**	**Circuit**	**Reported Specs**	λ	**AND depth**	**total evaluation time**	**number of parallel enc**	**relative time**
GHS12 (original)(packed) [GHS12]	BGV11 [BGV11]	AES	Intel Xeon CPU @ 2.0 Ghz with 256GB RAM	80	40	48 hours	54	37 min
GHS12 (original)(byte-sliced) [GHS12]						65 hours	720	5 min
CLT13 (byte-wise) [CCK+13]	DGHV10 [VDGHV10]	AES	Intel Core i7 @ 3.4Ghz with 32GB RAM	72	40	18.3 hours	33	33 min
CLT13 (state-wise) [CCK+13]						113 hours	531	12 min 46 s
CLT14 (state-wise) [CLT14]	DGHV10 [VDGHV10]	AES	Intel Xeon E5-2690 @ 2.9 GHz	80	40	102 hours	1875	3 min 15 s
CLT14 (state-wise) [CLT14]				72		3 h 35 min	569	23 s
LN14a (YASHE) [LN14]	LTV12 [LATV12]	SIMON	Intel Core i7-2600 @ 3.4 GHz[9]	128	34	1 h 10 min	2048	2.04 s
LN14a (FV) [LN14]	Bra12 [Bra12]					3 h 27 min	2048	6.06 s
DHS14 [DHS16]	LTV12 [LATV12]	AES	Intel Xeon @ 2.9 GHz	∼80	40	31 hours	2048	55 s
DSES14 [DSES14]	LTV12 [LATV12]	Prince	Intel Core i7 3770K @ 3.5 Ghz with 32 GB RAM[10]	130	30	57 min	1024	3.3 s
ARSTZ15 [ARS+15]	BGV11 [BGV11]	LowMC	Intel Haswell i7-4770K CPU @ 3.5 GHz with 16GB RAM	80	12	8 min	600	0.8 s
GHS12 (updated)(no bootstrapping) [GHS12]	BGV11 [BGV11]	AES	Intel Core i5-3320M at 2.6GHz with 4GB RAM[11]	80	40	4 min 12 s	120	2 s
GHS12 (updated)(with bootstrapping) [GHS12]						17 min 30 s	180	5.8 s

Table 7.6: Some publicly available FHE implementations

Name	**Scheme**	**Lang**	**Documentation**	**Libraries**
HElib [HS14b]	BGV [BGV11]	C++	Yes [HS13]	NTL, GMP
libScarab [PBS11a]	SV [SV10]	C	Yes [PBS11b]	GMP, FLINT, MPFR, MPIR
FHEW [DM14]	DM14 [DM15]	C++	Yes [DM15]	FFTW
TFHE [CGGI17]	CGGI16 [CGGI16]	C++	Yes [CGGI16]	FFTW
SEAL [LCP17]	FV12 [FV12b]	C++	Yes [CLP17]	No external dependency

HElib are documented in [HS13] and algorithms used in HElib are documented in [HS14a]. HElib is designed using low-level programming, which deals with the hardware constraints and components of the computer without using the functions and commands of a programming language and hence, defined as "assembly language for HE". It was implemented using GPL-licensed C++ library. Since December 2014, it supports bootstrapping [HS15] and since March 2015, it supports multi-threading. In an important extension, homomorphic evaluation of AES was implemented on top of HElib [GHS12] and included in the HElib source code in [HS14b].

[9]With hyper-threading turned off and over-clocking ('turbo boost') disabled.

[10]Only single thread is used.

[11]An Ubuntu 14.04 installed VM

Unfortunately, the usage of HElib is not easy because of the sophistication needed for its low-level implementation and parameter selection which effects both performance and security level. Another notable open source FHE implementation is libScarab [PBS11a]. To the best of our knowledge, libScarab [PBS11a] is the first open-source implementation of FHE. Its parameter selection is relatively easier than that of HElib, but it suffers from a lot of limitations. For instance, it does not implement modern techniques (e.g., modulus reduction and re-linearization techniques [BV14a]) to handle the noise level or it also does not support the SIMD techniques introduced in [SV14]. It implements Smart-Vercauteren's FHE scheme in [SV10] and documentation is provided in [PBS11b].

Another major implementation is introduced by Ducas and Micciancio and called "Fastest Homomorphic Encryption in the West" (FHEW) [DM14]. It is documented in [DM15]. It significantly improves the time required to bootstrap the ciphertext claiming homomorphic evaluation of a NAND gate "in less than a second". A NAND gate is functionally complete. Hence, any possible boolean circuits can be built using only NAND gates. In [DM15], the usage of ciphertext packing and SIMD techniques provides an amortized cost. However, in FHEW such performance is achieved using only a few hundred lines of code with the use of one additional library, FFTW [FJ05]. Later, the homomorphic computation cost of any binary gate [DM15] is increased by a factor of 50 by making some optimizations on the bootstrapping algorithm. The main improvement is based on the torus representation of LWE ciphertexts. This improved the cost of bootstrapping 10 times according to the best known bootstrapping in [DM14]. They also further improved the noise propagation overhead algorithms using some approximations. Finally, they also reduced the size of bootstrapping key from 1GB to 24MB by achieving the same security level.

More recently, another HE library called Simple Encrypted Arithmetic Library (SEAL) [LCP17] is released by Microsoft. The goal of releasing this library is explained as providing a well-documented HE library that can be easily used by both crypto experts and non-experts with no crypto background like practitioners in bioinformatics. The library does not have external dependencies like others and it includes automatic parameter selection and noise estimator tools, which makes it easier to use. Finally, the security estimates of two well-known LWE-based HE libraries, HElib and SEAL, against dual lattice attacks are revised in [Alb17]. It is shown that the parameters promising 80 bits of security actually gives an estimated cost of 68 bits for SEAL v2.0 and 62 bits for HElib. As a final note, we give the list of general-purpose HE libraries as follows: HEAAN implementing that supports fixed point arithmetics [CKKS16], a GPU-accelerated library cuHE [DDS17], a general lattice crypto library PALISADE [Roh17].

FHE hardware implementations and productions

The first known usage of FHE in a production environment is announced by Fujitsu Laboratories Ltd. [Ltd13]. Their reported implementation provides statistical calculations and biometric authentication by using FHE-based security. They improved an FHE by batching the string bits of data. The practical testing of this FHE implementation by Fujitsu is still pending as of this writing. Although the software only implementations are considered promising to obtain a practical FHE implementation, there is still a substantial gap between the achieved and the targeted performance. This gap led to new alternative research area in hardware implementations. The hardware solutions to accelerate both FHE and SWHE schemes mainly focused on three implementation platforms: Graphics Processing Unit (GPU), Application-Specific Integrated Circuit (ASIC), and Field-Programmable Gate Array (FPGA)

(A useful survey of hardware implementations of homomorphic schemes can be found in [MOO+14]). Although GPU is for graphical purposes, its highly parallel structure offers great promise over CPU for efficiency. Hence, it is suggested in some studies to use GPU order to improve the efficiency of homomorphic evaluation [DDS14, WCH14, WHC+15, DDS15, LLCP15]. One of the major barriers to a practical FHE is the noise growth in the homomorphic multiplication operation. This prompted researchers to find a solution that can deal with a large number of modular multiplications. Therefore, there are some works focusing particularly on this problem using the customized ASICs [DÖS13, WHEW14, DÖS15]. In spite of the potential of GPU and ASIC solutions, most of the proposed studies are based on the reconfigurable hardware, specifically FPGA. FPGA platforms offer not only Fast Fourier Transform (FFT), but also some optimization techniques such as number theoretic transformation (NTT) and fast modular polynomial reduction at hardware level. Such large and reconfigurable environment provided by FPGAs motivates many researchers to speed up the practicality of FHE schemes [CRPS12, WH13, CMO+13, MHM+13, CMV+15, CMO+14, MOHO14, CGRS14, SRJV+15, PNPM15, ÖDSS15].

In conclusion, some of the SWHE implementations (leveled-FHE) [GHS12] get closer to a tolerable performance. However, the bootstrapping techniques in FHE schemes need to be improved and the cost of homomorphic multiplications should be reduced to increase the performance.

7.4 Further Research Directions and Lessons Learned

Performance of any encryption scheme is evaluated with three different criteria: security, speed, and simplicity. First, an encryption scheme must be secure so that

an attacker can not obtain any type of information by using a reasonable amount of resources. Second, its efficiency must not disturb the user's comfort, i.e., it must be transparent to the users because users prefer usability against security. Lastly, if and only if an encryption scheme is understandable by the other area practitioners, they will implement the scheme for their applications and productions. If the existing FHE schemes are evaluated in terms of the three criteria, there is, though getting closer, still a substantial room for improvement in terms of all these criteria, especially for the speed performance.

Even though some of the nonstandard security assumptions (e.g., SSSP[12] [Lee11, HR11]) in the Gentry's original scheme are later removed, there are still some open security issues about the FHE schemes. First one is the circular security of FHE. Circular security (aka KDM security), as mentioned earlier, keeps its own secret key secure by encrypting it with the public key. All known FHE schemes use Gentry's blueprint bootstrapping technique to obtain an unlimited FHE scheme. So, the encryption of the secret key is also sent to the cloud to bootstrap the noisy ciphertexts and an eavesdropper can capture the encryption of secret key. Even though some SWHE and leveled-FHE schemes are proven as semantically secure, an unbounded FHE still has not been proven as semantically secure with respect to any function, so it does not guarantee that an adversary can not reveal the secret key from its encryption under the public key. This unfortunate situation is still open to be proven. Moreover, although some SWHE schemes [LMSV11] are proven as indistinguishable under non-adaptive chosen ciphertext attack (IND-CCA1), none of the unbounded FHE schemes is IND-CCA1 secure for now. (IND-CCA2 (adaptive) is not applicable

[12]Indeed, Moon Sung Lee showed that it is quite probable that SSSP challenges can be solved within two days [Lee11, HR11].

to FHE because FHE itself requires to be malleable.) In brief, FHE still needs to be studied extensively to prove that it is secure enough.

FHE allows an unlimited number of functions on encrypted data. However, limitations on the efficiency of the FHE schemes prompts researchers to find the SWHE schemes that can be good enough to use in real-life applications. Recently, homomorphic evaluation of one AES, which is a highly complex and nontrivial function, is reduced to 2 seconds [GHS12] and researchers are now focusing to improve this instead of trying to implement an FHE scheme, which is extremely slow for now.

The main process that increases the computational cost in FHE is the bootstrapping process. An unbounded FHE scheme that allows unlimited operations without bootstrapping is still an open problem.

Showing the existence of FHE instilled hope to solve other long waiting problems (applications) such as Functional Encryption (FE) (i.e., Identity-based encryption (IBE) and Attribute-based encryption (ABE)). Functional encryption basically controls the access over data while allowing computation on it according to the features of identity or attribute. The purpose of designing ABE or IBE based on FHE is to take the advantage of the functionality of two worlds. However, for now, there exists a few [GSW13, CM14, CM16, WWL15a]. Another fruitful application of FHE is multi-party computation (MPC) which allows the computation of the function with multiple inputs from different users while keeping the inputs hidden. Even though there exist a few FHE-based MPC protocols [DPSZ12, LATV12, CLO$^+$13, DPR16] proposing these powerful and useful tools, unfortunately, their performances are not yet comparable with the conventional MPC approaches [MGBF14, CMTB16, PH14, CMTB15] because of the computational cost of the existing FHE schemes. However, FHE does not require any interaction, which reduces the complexity of the communication protocol significantly. However,

there are still some gaps on how to realize those protocols. Furthermore, FHE itself can not perform a homomorphic evaluation on independently encrypted data, i.e., multi-key FHE. some primitive result to deal with this issue was presented in [LATV12]. However, the proposed scheme can only handle a bounded number of users. When the cloud and number of connected devices are considered, the restriction may not be feasible. Hence, a multi-key FHE with an unlimited number of users is another promising direction for future applications.

7.5 Conclusion

In today's always-on, Internet-centric world, the privacy of data plays a more significant role than ever before. For highly sensitive systems such as online retail and e-banking, it is crucial to protect users' accounts and assets from malicious third parties. Nonetheless, today's norm is to encrypt the data and share the keys with the service provider, cloud operator, etc. In this model, the control over the privacy of the sensitive data is lost. The users or service providers with the key have exclusive rights on the data. Untrusted providers, cloud operators can keep sensitive data and its identifying credentials of users long after the user ends the relationship with the services. One promising direction to preserve the privacy of the data is to utilize homomorphic encryption (HE) schemes. HE is a special kind of encryption scheme, which allows any third party to operate on the encrypted data without decrypting it in advance. Indeed, the idea of HE has been around for over 30 years; however, the first plausible and achievable *Fully Homomorphic Encryption* (FHE) scheme was introduced by Craig Gentry in 2009. Since then, different FHE schemes demonstrated that FHE still needs to be improved significantly to be practical on every platform as they are very expensive for real-life applications. Hence, in this chapter,

we surveyed the HE and FHE schemes. Specifically, starting from the basics of HE, the details of the well-known *Partially HE* (PHE) and *Somewhat HE* (SWHE), which are important pillars of achieving FHE, were presented. Then, after classifying FHE schemes in the literature under four different categories, we presented the major FHE schemes with this classification. Moreover, we articulated the implementations and the new improvements in Gentry-type FHE schemes. Finally, we discussed promising research directions as well as lessons learned for interested researchers.

CHAPTER 8

CURIE: POLICY-BASED PRIVACY-AWARE SECURE DATA EXCHANGE

8.1 Introduction

Inter-organizational data sharing is crucial to the advancement of many domains including security, health care, and finance. Previous works have shown the benefit of data sharing within distributed, collaborative, and federated learning [DCM+12, SCST17, APP+18]. Privacy-preserving machine learning offers data sharing among multiple members while avoiding the risks of disclosing the sensitive data (e.g., health-care records, personally identifiable information) [EESA+12]. For example, secure multiparty computation enables multiple members, each with its training dataset, to collaboratively learn a shared predictive model without revealing their datasets [MZ17]. These approaches solve the privacy concerns of members during model computation, yet do not consider the complex relationships such as regulations, competitive advantage, data sovereignty, and jurisdiction among members on private data sharing. Members want to be able to articulate and enforce their conflicting requirements on data sharing.

To illustrate such complex data sharing requirements, consider health care organizations that collaborate for a joint prediction model of diagnosis of patients experiencing blood clots (see Figure 8.1). Members wish to dictate their needs through their legal and political limitations as follows: U.S._1 is able to share its complete data for nation-wide members (U.S._2) [Ame17, Hea17], yet it is obliged to share the data of patients deployed in NATO countries with NATO members (UK) [fMS17]. However, U.S._1 wishes to acquire all patient data from other countries. UK is able to share and acquire complete data from NATO members, yet it desires to acquire

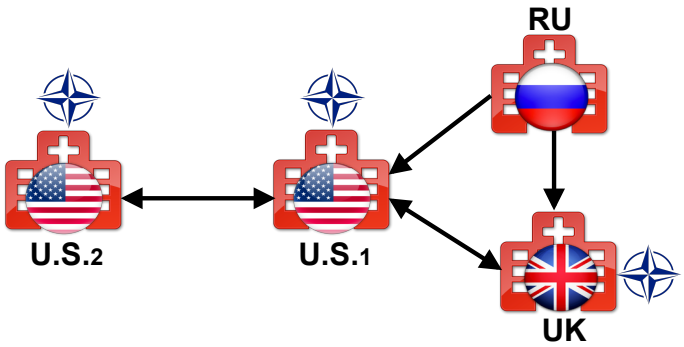


Figure 8.1: An illustration of data exchange requirements of countries learning a predictive model on their shared data. Arrows show the data requirements of countries.

only data of certain race groups from $U.S_1$ to increase its data diversity. RU wishes to share and acquire complete data from all members, yet members limit their data share to Russian citizens who live in their countries. Such complex data sharing requirements also commonly occur today in non-healthcare systems [BCD+09]. For instance, National Security Agency has varying restrictions on how human intelligence is shared with other countries; financial companies share data based on trust, and competition among each other.

In this chapter, we present a policy-based data exchange approach, called Curie, that allows secure data exchange among members that have such complex relationships. Members specify their requirements on data exchange using a policy language (CPL). The requirements defined with the use of CPL form the local data exchange policies of members. Local policies are defined separately for data sharing and data acquisition policies. This property allows asymmetric relations on data exchange. For example, a member does not necessarily have to acquire the data that the other members dictate to share. By using these two policies, members specify statements of who to share/acquire and what to share/acquire. The statements are defined using *conditional* and *selection* expressions. Selections allow members to filter data

and limit the data to be exchanged, whereas conditional expressions allow members to define logical statements. Another advanced property of CPL is predefined *data-dependent conditionals* for calculating the statistical metrics between member's data. For instance, members can define a conditional to compute the intersection size of data columns without disclosing their data. This allows members to define content-dependent conditional data exchange in their policies.

Once members have defined their local policies, they negotiate a sharing agreement. The guarantee provided by Curie is that all data exchanged among members will respect the agreement. The agreement is executed in a multi-party privacy-preserving prediction model enhanced with optional differential privacy guarantees. We begin in the next section by defining the analysis task and outlining the security and attacker models.

8.2 Problem Scope and Attacker Model

Problem Scope. We introduce Curie Policy Language (CPL) to express data exchange requirements of distributed members. Unlike the programming languages used for writing secure multiparty computation (MPC) [HKoS+10, RHH14] and the frameworks designed for privacy-preserving machine learning (ML) [LWN+15, O+16, BKLS18, EESA+12, MZ17], CPL is a policy language in a Backus Normal Form (BNF) notation to express the conflicting relationships of members on data sharing. Members can express data exchange requirements using the conditionals, selections, and secure pairwise data-dependent statistics. Curie then enforces the policy agreements in a shared predictive model through an MPC protocol that ensures members comply with the policies as negotiated.

We integrate Curie into 24 medical institutions. Without deployment of Curie, institutions compute warfarin dosage of a patient using a model computed on their local patient records. Curie allows institutions to construct various consortia wherein each member defines a data exchange policy for other members via CPL. This enables institutions to acquire the patient records based on regulations as well as the records that they need to improve the accuracy of their dose predictions. Curie implements a privacy-preserving dose model through homomorphic encryption (HE) to enforce the policy agreements of the members. We note that a centralized party in HE cannot provide a privacy-preserving model on negotiated data [VDJ10]. However, Curie implements a novel protocol that allows institutions to perform local computations by aggregating the intermediate results of the dose model. Additionally, Curie implements an optional differential private (DP) mechanism that allows institutions to perform differentially-private (DP) secure dose model. DP guarantees that no information leaks on the targeted individual (i.e., patient) with high confidence from the released dose model.

Threat Model. We consider a semi-honest adversary model. That is, members in a consortium runs the protocol exactly as specified, yet they try to learn the dataset inputs of the other members as much as possible from their views of the protocol. Additionally, we consider non-adaptive adversary wherein members cannot modify inputs of their dataset once the protocol on shared data is initiated.

8.3 Organizational Data Exchange

Depicted in Figure 8.2, Curie includes two independent parts: policy management and multiparty secure computation.

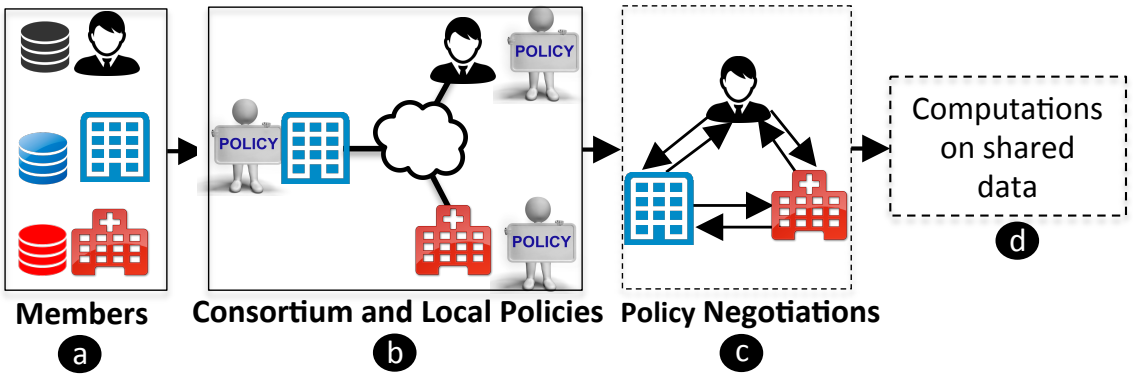


Figure 8.2: CURIE data exchange process in a collaborative learning setting. The dashed boxes show data remains confidential.

Policy Management. We define a *consortium* that is a group made up of two or more members–individuals, companies or governments (ⓐ). Members of a consortium aim to compute a predictive model m over their confidential data in a secure manner. For instance, data may be curated from medical history of patients or financial reports of companies with the objective of building an ML model. Moreover, each member wants to enforce a set of local constraints toward other consortium members to control their requirements on how and with whom they share their confidential data. These constraints define a member's interest, trust, regulations and data demands, and also impacts the accuracy of a model m. Thus, there is a need for connecting data needs of members to the privacy-preserving models. In CURIE, each member of a consortium defines a *local policy* (ⓑ). The local policy of a member dictates the requirements of data exchange as follows:

1. The member wishes to specify with whom to share and acquire data (*partnership requirement*).
2. The member wishes to define what data to share and acquire (*sharing and acquisition requirement*).

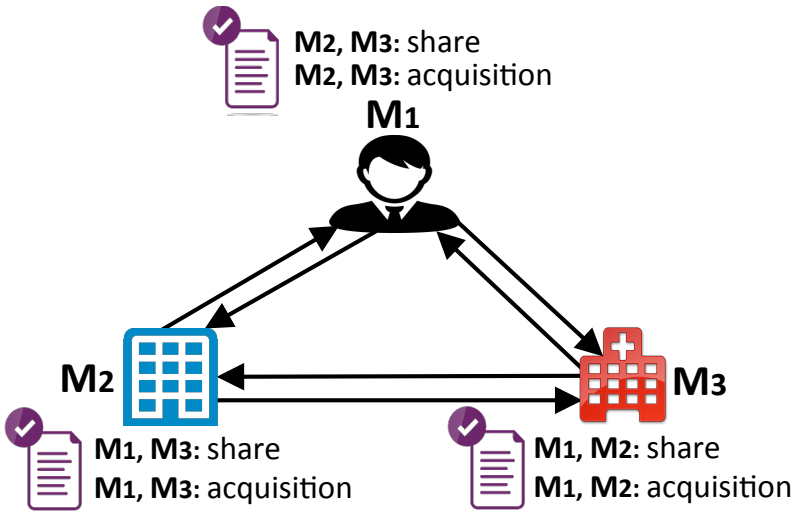


Figure 8.3: An example consortium of three members.

In this, the member wishes to refine its sharing and acquisition requirements to express the following:

1. The member wishes to dictate a set of conditions to restrict data sharing and select which data to be acquired (*conditional selective share and acquisition*); and
2. The member wishes to dictate conditionals based on the other member's data (*data-dependent conditionals*).

The policy of members need not be-nor are likely to be-symmetric. Local policy is defined with requirements for sharing and acquisition that is tailored to each partner member in the consortium–thus allowing each pairwise sharing to be unique. Here, the local policies are used to negotiate pairwise sharing within the consortium. To illustrate how members negotiate an agreement, consider the consortium of three members in Figure 8.3.

Each member initiates pairwise policy negotiations with other members to reconcile contradictions between acquisition and share policies (©). A member starts the negotiation by sending a request message including the acquisition policy defined for a member. When a member receives the acquisition policy, it reconciles the received

acquisition policy with its share policy specified for that member. Three negotiation outcomes are possible: the acquisition policy is entirely satisfied, partially satisfied with the intersection of acquisition and share policies or is an empty set. A member completes its negotiations after all of its acquisition policies for interested parties are negotiated.

Computations on Negotiated Data. Once members negotiate their policies (❹), Curie provides a multiparty data exchange device using secure multi-party computation techniques enhanced with (optional) differential privacy guarantees. This device ensures data and individual privacy. The guarantee provided by Curie is that all computations among members will respect their policies.

To ensure data privacy, Curie includes cryptographic primitives such as Homomorphic Encryption (HE) and garbled circuits from the secure multi-party computation literature that allows members to perform computations on negotiated data with no disclosed data from any single member. At the end of the secure computation, all of the parties obtain a final predictive model based on their policy negotiations. To ensure the privacy of the individuals in the dataset, which the final model is computed on, Curie integrates Differential Privacy (DP). DP protects against an attacker who tries to extract a particular individual's data in the dataset from the final computed model at the end of the secure computation protocol.

8.4 Curie Policy Description Language

We now illustrate the format and semantics of the Curie Policy Language (CPL). Turning to the example consortium in Figure 8.3 established with three members, each member defines its requirements for other members on a dataset having the

columns of age, race, genotype, and weight (see Table 8.1). The criteria defined by members are used throughout to construct their local policies.

Share and Acquisition Clauses. CURIE policies are collections of clauses. The collection of clauses for partners defines the local policy of a member. The clauses allow each member to dictate a member specific policy for each other member. Clauses have the following structure:

$$\langle \text{clause tag} \rangle : \langle \text{members} \rangle : \langle \text{conditionals} \rangle :: \langle \text{selections} \rangle;$$

Clause tags are reference names for policy entries. *Share* and *acquire* are two reserved tags. Those clauses are comprised of three parts. The first part, *members*, defines a list of members with whom to share and acquire. This can be a single member or a comma-separated list of members. An empty member entry matches all members. The second part, *conditionals*, is a list of conditions controlling when this clause will be executed. A condition is a Boolean function which expresses whether the share or acquire is allowed or not. For instance, a member may define a condition where the data size is greater than a specific value. Only if all conditions listed in conditionals are true, then this clause is executed. Last part, *selections*, states what to share or acquire. It can be a list of filters on a member's data. For instance, a member may define a filter on a column of a dataset to limit acquisition to a subset of the dataset. More complex selections can be assigned using member defined sub-clauses. A sub-clause has the following structure:

$$\langle \text{tag} \rangle : \langle \text{conditionals} \rangle :: \langle \text{selections} \rangle;$$

where *tag* is the name of sub-clause; conditionals is, as explained above, a list of conditions stating whether this clause will be executed; selections is a list of filters or a reference to a new sub-clause. Complex data selection can be addressed with nested sub-clauses.

Consortia member: M_1	
M_2–	desires to acquire complete data of users who are older than 25
M_2–	shares its complete data
M_3–	desires to acquire Asian users such that the Jaccard similarity of its age column and M_3's age column is greater than 0.3
M_3–	shares its complete data
Consortia member: M_2	
M_1–	desires to acquire complete data
M_1–	limits its share to EU and NATO citizen users if M_1 is both NATO and EU member and located in North America. Otherwise, it shares only White users
M_3–	desires to acquire complete data if M_3 is a NATO member
M_3–	shares its complete data
Consortia member: M_3	
M_1–	desires to acquire complete data of users having genotype 'A/A'
M_1–	share complete data if intersection size of its and M_1's genotype column is less than 10. Otherwise, it shares data of users that weigh more than 100 pounds
M_2–	desires to acquire complete data
M_2–	shares complete data if M_2 is EU member and its data size is greater than 1K

Table 8.1: An example of member's data exchange requirements.

CPL allows members to define multiple clauses. For instance, a member may share a distinct subset of data for different conditions. CPL evaluates multiple clauses in a top-down order. When conditionals of a clause evaluate to false, it moves to the next clause until a clause is matched or it reaches end of the policy file.

Conditionals and Selections. We present the use of conditionals and selections through policies with examples. Their format and semantics are detailed. Consider an example of two members, M_1 and M_2, within a consortium. They define their local policies as:

⟨selections⟩ ::= <filters> | <tag>
<filters> ::= <filter> [',' <filters>]
<filter> ::= <var> <operation> <value> | ''

@M_1 acquire : M_2 : :: s_1 ;
share : M_2 : :: ;
@M_2 acquire : M_1 : :: ;
share : M_1 : c_1, c_2 :: fine-select ;
fine-select : c_3 :: s_2 ;
fine-select : :: s_3 ;

where c_1, c_2 and c_3 are conditionals, s_1, s_2 and s_3 are selections and fine-select is a tag defined by M_2.

The acquire clause of M_1 states that data is requested from M_2 after it applies s_1 selection (e.g., age > 25) to its data. In contrast, its share clause allows complete share of its data if M_2 requests. On the other hand, the acquisition clause of M_1 dictates requesting complete data from M_2. However, M_2 allows data sharing if the acquisition clause issued by M_1 holds $c_1 \wedge c_2$ conditions (e.g., is both NATO and EU member). Then, M_2 delegates selection to member-defined fine-select sub-clauses. fine-select states that if the request satisfies the c_3 condition (located in North America) then the request is met with the data that is selected by the s_2 selection (e.g., limits share of its data to NATO and EU member country citizens). Otherwise, it shares data that is specified by selection s_3 (White users).

CPL supports selections through filters. A filter contains zero or more operations over data inputs describing the share and acquisition criteria to be enforced. Operations are defined as keywords or symbols such as $<$, $>$, $=$, *in*, *like*, and so on. Selections and filters are defined in CPL as follows:
Selections are executed when conditionals evaluated to be true. Conditionals can be consortium and dataset-specific. For instance, a member may require other members

⟨conditionals⟩ ::= <var>'='<value> [',' <conditionals>]
| 'evaluate' '(' <data_ref> ',' <alg_arg> ',' <thshold_-
arg> ')' [',' <conditionals>] | ''

to be in a particular country or to be in an alliance such as NATO and to have their dataset size greater than a particular value. Such conditionals do not require any data exchange between members to be evaluated. However, members may want to incorporate a relation between their data and other member's data into their policies as detailed next.

Data-dependent Conditionals. A member's decision on whether to share or to acquire data can depend on other member's data. Simply put, one example of a data-dependent conditional among two members could be whether the intersection size of the two sets (e.g., a specific column of a dataset) is not too high. Considering such knowledge, a member can make a conditional decision about share or acquisition of that data. For instance, consider a list of private IP addresses used for blacklisting the domains. If a member knows that the intersection size is close to zero, then the member may dictate an acquire clause to request complete features from that member based on IP addresses [FDCB15].

CPL defines an evaluate keyword for data-dependent conditionals through functions on data. Data-dependent conditionals take the following form:

A member that uses the data-dependent conditionals defines a reference data (data_ref) required for a such computation, an algorithm (alg_arg) and a threshold (thshold_arg) that is compared with the output of the computation. CPL includes four algorithms for data-dependent conditionals (see Table 8.2). To be brief, intersection size measures the size of the overlap between two sets; Jaccard index is a statistic measure of similarity between sets; Pearson correlation is a statistical measure of how much two sets are linearly dependent; and Cosine similarity is a measure

Pairwise alg.	**Output**	**Private protocol**	**Proof**
Intersection size	$\lvert\mathcal{D}_i \cap \mathcal{D}_j\rvert$	Intersection cardinality	[DCGT12]
Jaccard index	$(\lvert\mathcal{D}_i \cap D_j\rvert)/(\lvert\mathcal{D}_i \cup \mathcal{D}_j\rvert)$	Jaccard similarity	[BDCG12]
Pearson correlation	$(COV(\mathcal{D}_i, \mathcal{D}_j))/(\sigma_{\mathcal{D}_i}\sigma_{\mathcal{D}_j})$	Garbled circuits	[H$^+$11]
Cosine similarity	$(\mathcal{D}_i\mathcal{D}_j)/(\lVert\mathcal{D}_i\rVert\lVert\mathcal{D}_j\rVert)$	Garbled circuits	[H$^+$11]

Table 8.2: CPL data-dependent conditional algorithms. Two members of a consortium use the conditionals to compute the pairwise statistics. The members then use the output of the algorithm to determine whether to acquire or share data from another party. ($\mathcal{D}_i$ and $\mathcal{D}_j$ are the inputs of a dataset, and σ is std. deviation).

of similarity between two vectors. Each algorithm is based on a different assumption about the underlying reference data. However, central to all of them is to privately (without leaking any sensitive data) measure a relation between two members' data to offer an effective data exchange. We note that these algorithms are found to be effective in capturing input relations in datasets [FDCB15, GPGMP16].

Data-dependent conditionals are implemented through private protocols (as defined in Table 8.2). These protocols are implemented with the cryptographic tools of garbled circuits and private functions. Protocols preserve the confidentiality of data. That is, each member gets the output indicated in Table 8.2 without revealing their sensitive data in plain text. After the private protocol terminates, the output of the algorithm is compared with a threshold value set by the requester. If the output is below the threshold value, the conditional is evaluated to true. Turning to above example M_3 joins the consortium. M_1 and M_2 extend their local policies for M_3:

@M_1 acquire : M_3 : evaluate(local data, 'Jaccard', 0.3) :: race=Asian;
share : M_3 : :: ;
@M_2 acquire : M_3 : M_3 in $NATO :: ;
share : M_3 : :: ;
@M_3 acquire : M_1 : :: Genotype = 'A/A' ;
share : M_1 : evaluate(local data,'intersection size', 10) :: ;
share : M_1 : :: weight>150 ;
acquire : M_2 : :: ;
share : M_2 : M_2 in $EU, size(data)> 1K :: ;

The acquire clause of M_1 defines a data-dependent conditional for M_3. It defines a Jaccard measure on its local data through evaluate keyword and sets its threshold value equal to 0.3. M_3 agrees to share its local data with M_1 if intersection size of its local data is less then 10. Otherwise, it consults the next share clause defined for M_1 which states that an individual's weight greater than 150 pounds will be shared. All other share and acquire clauses are trivial. Members agree to share and acquire complete data based on data size (data size > 1K), alliance membership (e.g., NATO or EU member) and inputs (e.g., genotype).

Putting pieces together, CPL allows members independently define a data exchange policy with share and acquire clauses. The policies are dictated through conditionals and selections. This allows members to dictate policies in complex and asymmetric relationships. Defined in Section 8.3, CPL provides members to dictate partnership, share, acquisition, and data-dependent conditionals.

Policy Negotiation and Conflicts. Data exchange between members is governed by matching share and acquire clauses in each member's respective policies. Both share and acquire clauses state conditions and selections on the data exchanged.

Policy ID	Consortium Name	Policy Definition	Acquisition Policy	Share Policy
P.1	Single Source	Each member uses its local patient dataset to learn warfarin dose model.	✗	✗
P.2	Nation-wide	Members in the same country establish a consortium based on state and country laws.	✓	✓
P.3	Regional	Members in the same continent establish a consortium.	✓	✓
P.4	NATO-EU	NATO and EU members establish a consortium independently based on their mutual agreements.	✓	✓
P.5	Global	Members exchange their complete data to build the warfarin dose model.	✓	✓

Table 8.3: Consortia constructed among members. Acquisition and share policies of members for each consortium are studied in Section 8.7.

Consider two example local policies with a share clause $@m_2$ $(share : m_1 : c_1 :: s_1)$ and matching acquire clause $@m_1$ $(acquire : m_2 : c_2 : s_2)$. Curie's negotiation algorithm respects both autonomy of the data owner and the needs of the requester. It conservatively negotiates share and acquire clauses such that it will return the *intersection* of respective data sets in resulting policy assignment. The resolved policy in this example is $share : m_1 : c_1 \wedge c_2 :: s_1 \wedge s_2$ which states that the data exchange from m_2 to m_1 is subject to both c_1 and c_2 conditionals and resulting sharing has s_1 and s_2 selections on m_2's data. This authoritative negotiation makes sure no member's data is shared beyond its explicit intent, regardless how the other members' policies are defined. This is because negotiation fulfilling the criteria for each clause is based on the union of logical expressions defined in two policies. Each member runs the negotiation algorithm for members found in their member list. After all members terminate their negotiations, the negotiated policy is enforced in computations.

8.5 Deployment of Curie

To validate Curie in a real application, we integrated Curie into 24 medical institutions. Each institution wants to compute a warfarin dose model on the distributed dataset without disclosing the patient health-care records. Without deployment of Curie, institutions compute warfarin dosage of a patient using a model computed on their local patient data. Curie first enables institutions to negotiate their data

exchange requirements through CPL. In this, Curie allows members to construct various consortia wherein each member defines a data exchange policy for other members. The next step is to compute a privacy-preserving dose model such that each party does not learn any information about the patient's records of other medical institutions and respects the policy negotiated. Curie implements a secure dose protocol through homomorphic encryption (HE) to enforce the policy agreements of the members. We next present the deployment of Curie to institutions (Section 8.5.1) and integration of policy agreements in warfarin dose model (Section 8.5.2).

8.5.1 Deployment Setup

Warfarin- known as the brand name Coumadin is a widely prescribed (over 20 million times each year in the United States) anticoagulant medication. It is mainly used to treat (or prevent) blood clots (thrombosis) in veins or arteries. Taking high-dose warfarin causes thin blood which may result in intracranial and extracranial bleeding. Taking low doses causes thick blood which may result in embolism and stroke. Current clinical practices suggest a *fixed* initial dose of 5 or 10 mg/day. Patients regularly have a blood test to check how long it takes for blood to clot (international normalized ratio (INR)). Based on the INR, subsequent doses are adjusted to maintain the patient's INR at the desired level. Therefore, it is important to predict the proper warfarin dose for the patients.

Consortium Members. 24 medical institutions from nine countries and four continents individually collected the largest patient data for predicting *personalized* warfarin dose. Members collect 68 inputs from patients' genotypic, demographic, background information, yet a long study concluded that eight inputs are sufficient for proper prescriptions [Int09].

Warfarin Dose Prediction Model. To determine the proper personalized warfarin dosage, a long line of work concluded with an algorithm of an ordinary linear regression model [Int09]. The model is a function $f : \mathcal{X} \to \mathcal{Y}$ that aim at predicting targets of warfarin dose $y \in \mathcal{Y}$ given a set of patient inputs $x \in \mathcal{X}$. We represent the patient dataset of each member $\mathcal{D}_i = \{(x_i, y_i)\}_{i=1}^n$, and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$. The loss function penalizes deviations between true dose and predictions. Learning is then searching for a dose model f minimizing the average loss:

$$\mathcal{L}(\mathcal{D}, f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i). \tag{8.1}$$

The dose model reduces to minimizing the average loss $\mathcal{L}(\mathcal{D}, f)$ with respect to the parameters of the model f. The model is linear, i.e., $f(x) = \alpha^\top x + \beta$, and the loss function is the squared loss $\ell(f(x), y) = (f(x) - y)^2$. The dose model gives as well or better results than other more complex numerical methods and outperforms fixed-dose approach[1] [Int09]. We re-implemented the algorithm in Python by direct translation from the authors' implementation and found that the accuracy of our implementation has no statistically significant difference.

Consortia and Member Policies. We define consortia among medical institutions that they state partnerships for data exchange. Table 8.3 summarizes the consortia. The consortia are defined based on statute and regulations between members, as well as regional, and national partnerships are studied based on their countries [fMS17, Ame17, Hea17, Rep17]. For example, NATO allied medical support doctrine allows strategic relationships that are otherwise not obtainable by

[1]The model has been released online `http://www.warfarindosing.org` to help doctors and other clinicians for predicting ideal dose of warfarin.

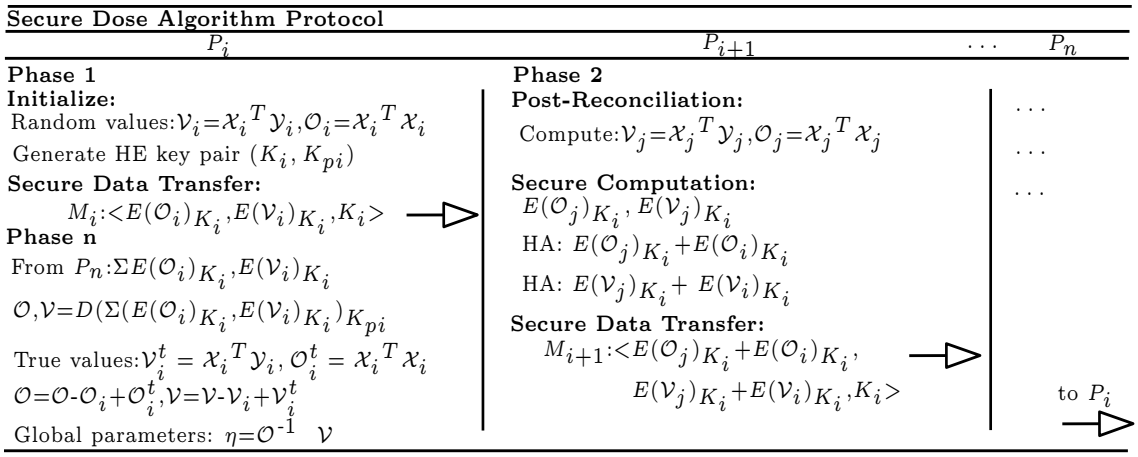


Figure 8.4: Secure dose algorithm protocol: Member (P_i) starts the protocol, the procedures and message flow among members are highlighted in boldface. At the final phase, P_i is able to compute the dose model coefficients from the negotiated data.

non-NATO members. Each member in a consortium exchanges data with other members based on its CPL policy. Various acquisition and share policies of CPL are studied via conditionals and selections in Section 8.7. We note that policy construction is a subjective enterprise. Depending on the nature and constraints of a given environment, any number of policies are appropriate. Such is the promise of policy defined behavior; alternate interpretations leading to other application requirements can be addressed through CPL.

8.5.2 Privacy-preserving Dose Prediction Model

The computation of *local dose* model of a medical institution is straightforward: a member calculates the dose model through Equation 8.2 with the use of patient data collected locally. To implement a privacy-preserving dose model among consortia members of medical institutions, we define the dose prediction formula stated in Equation 8.1 in a matrix form by minimizing with maximum likelihood estimation:

$$\beta = (\mathcal{X}^\intercal \mathcal{X})^{-1} \mathcal{X}^\intercal \mathcal{Y}, \tag{8.2}$$

where $\mathcal{X}$ is the input matrix, $\mathcal{Y}$ is the dose matrix, and β is the coefficients of the dose model.

Curie allows members to collaboratively learn a dose model without disclosing their patient records and guarantees data sharing complies with the policy as negotiated. As shown in Equation 8.4, each member translates its negotiated data into neutral input matrices [WKT11]. Particularly, patient samples to be exchanged by each member are computed as an input matrix $\mathcal{X}_0, \ldots, \mathcal{X}_n$ and dose matrix $\mathcal{Y}_0, \ldots, \mathcal{Y}_n$. The transformation defines each member's *local statistics* $\mathcal{O}_i = \mathcal{X}^\intercal \mathcal{X}$ and $\mathcal{V}_i = \mathcal{X}^\intercal \mathcal{Y}$. Local statistics is the output of the negotiation of each member in a consortium. The aggregation of the local statistics corresponds to a *negotiated dataset* which is the exact amount that a member negotiates to obtain from other members in a consortium. Curie constructs the dose algorithm of the negotiated dataset as a concatenation of members' local statistics as follows:

$$\mathcal{X}^\intercal \mathcal{X} = \Big[\mathcal{X}_1^\intercal | \ldots | \mathcal{X}_n^\intercal\Big] \Big[\mathcal{X}_1 | \ldots | \mathcal{X}_n\Big]^\intercal = \sum_{i=1}^{n} \mathcal{X}_i^\intercal \mathcal{X}_i = \sum_{i=1}^{n} \mathcal{V}_i = \mathcal{V} \tag{8.3}$$

$$\mathcal{X}^\intercal \mathcal{Y} = \Big[\mathcal{X}_1^\intercal | \ldots | \mathcal{X}_n^\intercal\Big] \Big[\mathcal{Y}_1 | \ldots | \mathcal{X}_n\Big]^\intercal = \sum_{i=1}^{n} \mathcal{X}_i^\intercal \mathcal{Y}_i = \sum_{i=1}^{n} \mathcal{O}_i = \mathcal{O} \tag{8.4}$$

In Equation 8.4, a member computes model coefficients using the sum of other members local statistics. The local statistics includes $m \times m$ constant matrices where m is the number inputs (independent of number of dataset size). Using this observation, a party computes the coefficients of the negotiated dataset:

$$\eta^{(negotiated)} = (\mathcal{X}^\intercal \mathcal{X})^{-1} \mathcal{X}^\intercal \mathcal{Y} = \mathcal{O}^{-1} \mathcal{V} \tag{8.5}$$

In Equation 8.5, while the accuracy objective of the dose model is guaranteed using the coefficients obtained from the sum of local statistics, the exchange of clear statistics among parties may leak information about members' data. A member can infer knowledge about the distribution of each input of other members from matrices of $\mathcal{O}_i$ and $\mathcal{V}_i$ [EESA+12]. Furthermore, an adversary may sniff data traffic to control and modify exchanged messages. To solve these problems, we use homomorphic encryption (HE) that allows computation on ciphertexts [AAUC18]. HE allows members to perform the computation of joint of function without requiring additional communication complexity other than the data exchange. We note that HE itself cannot preserve the confidentiality of data from multiple parties in centralized settings [VDJ10]. However, CURIE implements a distributed privacy-preserving multi-party dose model, as shown in Figure 8.4.

To illustrate, we consider an example session of n members authorized for data exchange in a consortium. In this example, a ring topology is used for secure group communication (i.e., P_i talks to P_{i+1}, and similarly P_n talks to P_i). P_1 initially generates a pair of encryption keys using the homomorphic cryptosystem and broadcasts the public key to the members in its member list. P_1 then generates random $\mathcal{V}_i$, $\mathcal{O}_i$ and encrypts them $E(\mathcal{O}_i)_{K_i}$ and $E(\mathcal{V}_i)_{K_i}$ using its public key K_i. It starts the session by sending them to the next member in the ring. When next member receives the encrypted message, it adds its local $\mathcal{V}_i$ and $\mathcal{O}_i$ matrices through homomorphic addition to the output of its policy reconciliation for P_1 and passes to the next member. Remaining members take the similar steps. Secure computation executes one round per member in which the computation for the particular member visits other members. This allows CURIE to enforce HE on shared data of a particular member in each round uses and does not suffer insecurities associated with centralized HE constructions [VDJ10].

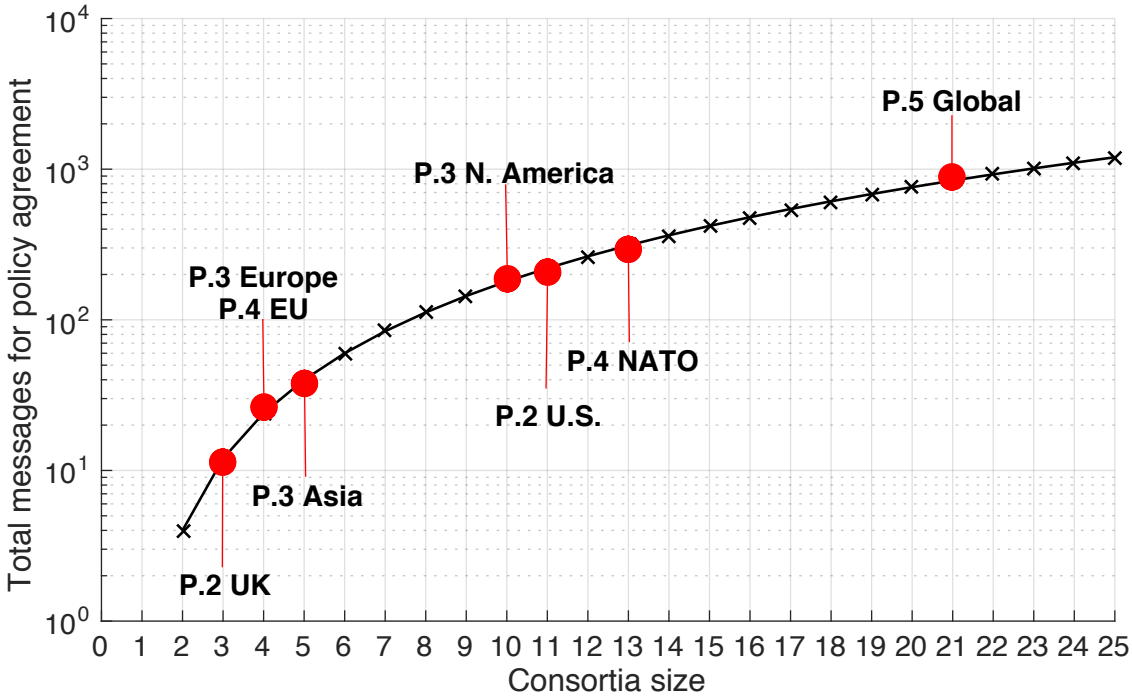


Figure 8.5: CPL negotiation cost - Costs associated with a number of varying members in a consortium. Each member defines asymmetric share and acquisition policy for other members. The number of members in warfarin consortia is marked with red circles.

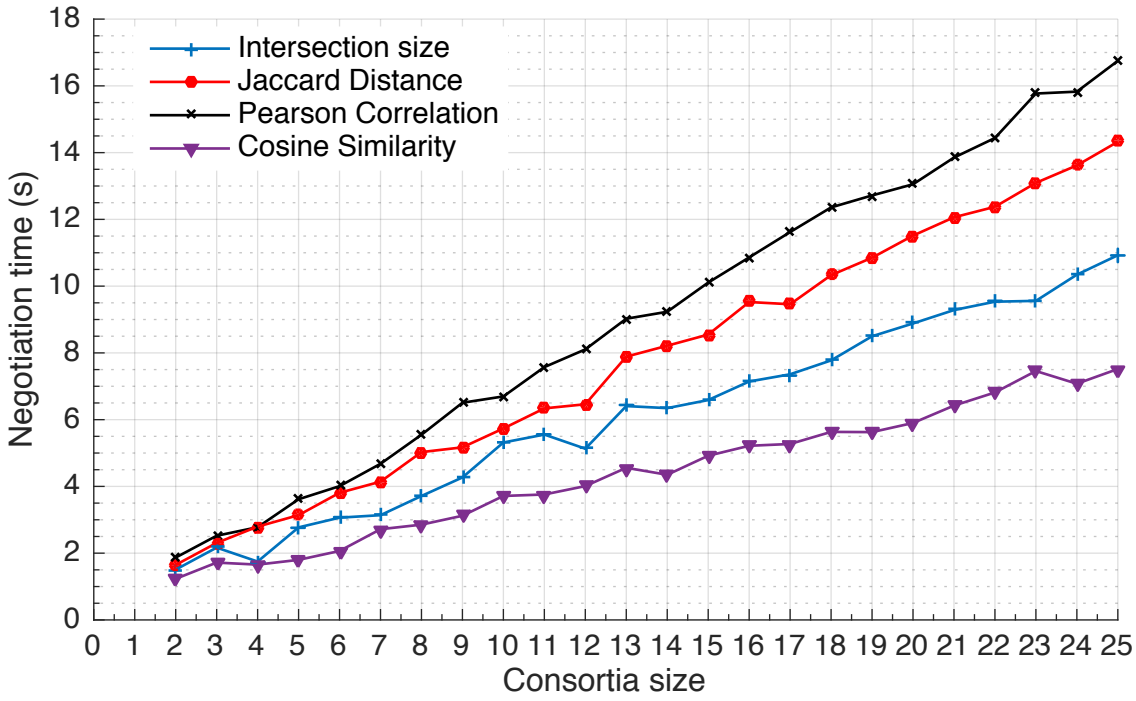


Figure 8.6: CPL selections and data-dependent conditional costs - Costs associated with varying members and algorithms. All consortia members agree on policy including a different data-dependent conditional and selections over one input of having 200 samples.

At the final stage of the protocol, P_1 receives the sum statistics of $\mathcal{O}_i$ and $\mathcal{V}_i$ from P_n. P_1 decrypts the sum of the statistics using its private key and then subtracts the initial random values of $\mathcal{V}_i$, $\mathcal{O}_i$ and adds its true values used for computation of the local dose model coefficients. The final result $\mathcal{O}$ and $\mathcal{V}$ is the coefficients of the dose model that respects P_1's policy negotiations. Other consortium members similarly start the protocol and compute the coefficients.

8.6 Security Analysis of the Dose Algorithm

We present security and privacy guarantees of the dose algorithm provided to all members through the share of encrypted integrated statistics, ($\mathcal{O}_i = \mathcal{X}^\intercal\mathcal{X}$ and $\mathcal{V}_i = \mathcal{X}^\intercal\mathcal{Y}$ matrices). Since all data exchange among parties is encrypted through the use of HE, the security of the algorithm against any adversary outside the authorized parties is based on the underlying HE cryptosystem.

An adversary not involving session initiator. Assume for now that a session initiator does not collude with other parties. Loosely speaking, since all computations are performed on the encrypted data, none of the parties learn anything about other parties' input.

We consider a party P_{i+1} in Figure 8.4. The party P_{i+1} has the public key generated by the session initiator K_i, the encryption of local statistics of previous parties $M_i = (E(\mathcal{O}_i)_K, E(\mathcal{V}_i)_K)$. Its input is $(\mathcal{V}_{i+1}, \mathcal{O}_{i+1})$ and its output is $M_{i+1} = (E(\mathcal{O}_i + \mathcal{O}_{i+1}), E(\mathcal{V}_i + \mathcal{V}_{i+1}))$. A simulator S selects random values for its own inputs $(\mathcal{V}'_{i+1}, \mathcal{O}'_{i+1})$ and encrypts them using the public key published by the session initiator. Then, the simulator S performs the homomorphic operation on the received message M_i and outputs $M'_{i+1} = (E(\mathcal{O}_i + \mathcal{O}'_{i+1})_K, E(\mathcal{V}_i + \mathcal{V}'_{i+1})_K)$. Here, we assume the underlying HE is semantically secure. Therefore, the output

of the simulator M'_{i+1} is computationally indistinguishable from output of the real execution of the protocol M_{i+1} for every input pairs. Therefore, using the definition in [Gol09] the protocol privately computes the function in the presence of one semi-honest corrupted party. The extension to multi-corrupted semi-honest adversaries is straightforward as the only difference is the view of a subset of parties having many encrypted messages. Since the semantic security of the underlying HE is hold for any pair of these many encrypted messages, no information leaks about the corresponding plaintexts.

Adversary involving session initiator. We consider the case when the session initiator is corrupted. The corrupted parties including session initiator can infer the input of an honest party if the predecessor (previous party) and successor (next party) of an honest party are both corrupted. We consider the possible cases for data leakage: (1) *2-party:* The session initiator is corrupted, and another party is honest. In this case, predecessor and successor of the honest party are both the corrupted session initiator. Therefore, the input of honest party is learned by the corrupted party, (2) *3-party*: A corrupted session initiator is either predecessor or successor; thus it can learn inputs of the one of the honest party only if another party is corrupted, and (3) *n-party ($n > 3$)*: To learn an honest party's input, at least two parties must be corrupted and placed in previous and next of the honest party.

While the individual raw data of members does not leak, the risk of inappropriate disclosures from local summary statistics exists in some extreme cases [EESA+12]. Consider the exchange of plain matrix $V_i = X^\intercal Y$ among two parties; a party may use the extreme values found in V_i to identify particular patients. For instance, in dose algorithm, taking inducers such as Rifadin and Dilantin could indicate high dose prescriptions. If the values of V_i are high, then a party may infer a patient that

takes enzyme inducers and the presence of high dosage warfarin intake. Similarly, exchange of $O_i = X^\intercal X$ may leak information about the number of observations and represent the number of 0s or 1s in a column. For instance, for the former first entry in the matrix, $X^\intercal X$, gives the total number of patients. For the latter, $(X^\intercal X)_{j,j}$ gives the number of 1s in the column. This type information lets a party infer knowledge, particularly when binary inputs (e.g., use of the medicine) are used.

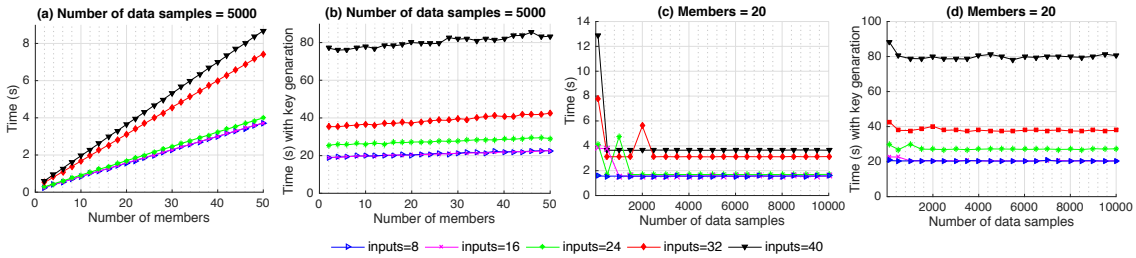


Figure 8.7: CPL performance on privacy-preserving and differential private protocol - All members define an asymmetric share and acquisition policy through selections and conditionals. The agreements of CPL policies between consortia members are studied with the different number of consortia members, data samples, and input size. (Std. dev. of ten runs is ± 3.6 and ± 0.3 sec. with and without homomorphic key generation.)

8.7 Evaluation

This section details the operation of the Curie through policies. We show how flexible data exchange policies are implemented and operated. We focus on the following questions:

1. What are the performance trade-offs in configuring CPL?
2. Can members reliably use Curie to integrate various policies?
3. Do members improve the accuracy of dose predictions with the use of CPL?

The answers to the first two questions are addressed in Section 8.7.1, and the last question is answered in Section 8.7.2. As detailed throughout, Curie allows 50

members to compute the privacy-preserving model using 5K data samples with 40 inputs in less than a minute. We also show how an algorithm with flexible data exchange policies can improve–often substantially–the accuracy of the warfarin dose model accuracy.

Experimental Setup. The experiments were performed on a cluster of machines with 32 GB of maximum memory and 16-core Intel Xeon CPU at 1.90 GHz, where we use one core to get a lower bound estimate. Each member is simulated in a server that stores its data. Secure computation protocols of Curie are implemented using the open-source HElib library [HS14b]. We set the security parameter of HElib as 128 bits. Multiplication level is optimized per member to increase the number of allowed homomorphic operations without decryption failure and to reduce the computation time.

We validate the accuracy of dose model in various consortia defined in Table 8.3 with members defining different data exchange policies. The dataset used in our experiments contains 5700 patient records from 21 members. Dose model accuracy of each member is validated with Mean Absolute Percentage Error (MAPE). MAPE measures the percentage of how far predicted dosages are away from true dosage. Lower values indicate better quality of treatment.

8.7.1 Performance Evaluation

We present the costs associated with various Curie mechanisms. We illustrate the cost of the CPL in policy negotiations, in the use of data-dependent conditionals, and in the dose algorithm.

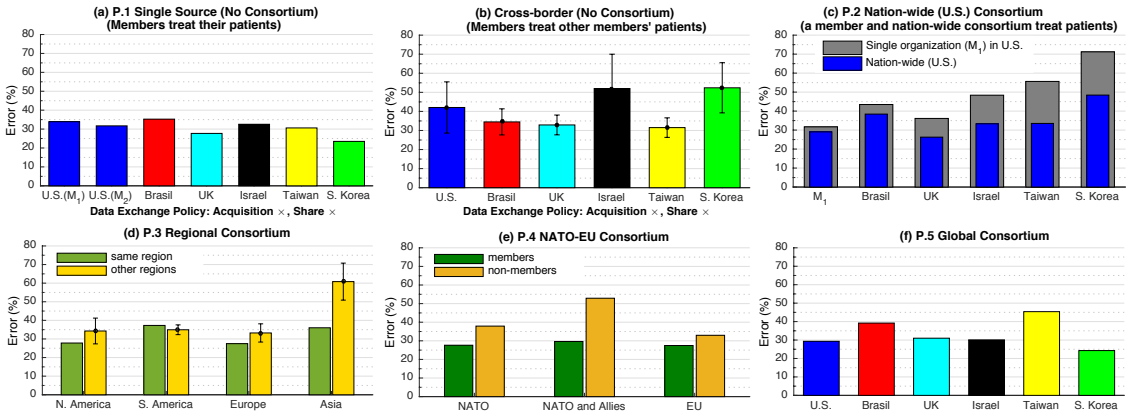


Figure 8.8: The implication of policies on model accuracy - errors are validated in various consortia through data exchange policies. Figure 6(c-f): The local acquisition policies of members comply with the sharing policy within a consortium (i.e., members acquire complete data of the consortia members. Std. devs. of errors are within %5, if not illustrated).

CPL Benchmarks

Our first set of experiments characterize the policy construction and negotiation costs. Various consortia and policies are instrumented to analyze the overhead of the number of messages and time required to compute the CPL selections and data-dependent conditionals. All the costs not specific to the policies are excluded in measurements (e.g., network latency). The benchmark results are summarized in Figure 8.5 and 8.6 and discussed below.

Figure 8.5 shows the number of messages for policy construction required for different consortia size. The number of members in warfarin study is also labeled. For instance, NATO consortium has 13 members; ten members from U.S. and three from UK. The experiments illustrate the upper bound results wherein each member defines a different share and acquisition policy for other members (i.e., asymmetric relations). In this, each member sends acquisition policy request to consortium members. After a member gets the acquisition request, it reconciles with its share policy and output of negotiation message is returned. The number of messages asso-

ciated with varying number of selections and conditionals dictated by the members does not require any additional messages. For instance, the acquisition request of a member includes arguments when conditionals are defined (e.g., reference data and a threshold value for data-dependent conditionals such as pairwise Jaccard distance), and the result is returned with the negotiation output message. However, the use of the selections and data-dependent conditionals brings additional processing cost as detailed next.

Figure 8.6 shows the costs associated with the use of CPL selection and data-dependent conditionals. All the members dictate data-dependent conditionals and selections on a single input. The members input size for the data-dependent conditional computations is set to 200 real values. This is the average number of inputs found in members' dataset. Since selections and conditionals reconcile contradictions between acquisition and share policies, they do not require any additional computation overhead and yield a processing time of milliseconds. However, the time associated with varying data-dependent conditionals depend on the protocol of associated secure pairwise algorithm. In our experiments, cosine similarity and intersection size exhibited shorter computation time than Pearson correlation and Jaccard distance. Overall, we found that 25 members compute the metrics less than 18 seconds. Note that the results serve as an upper bound that all members define a set of selections and a data-dependent conditional on one input.

Dose Model Benchmarks

Our second series of experiments characterize the impact of CPL on the average time of computing privacy-preserving dose model with varying number of members and dataset sizes. Though the warfarin study includes eight inputs, evaluations are repeated with the input size of 8, 16, 24, 32, and 40 through various dataset sample

sizes for completeness. The input and sample size together represents the total dataset shared for a member as a result of the policy agreements. Our experiments show that 80% of computation overhead is attributed to HE key generation. The cost of the differential privacy takes microseconds, as the members can calculate the (optional) differential private algorithm model at the end of the secure dose protocol. Computations are instrumented to classify the overheads incurred by key generation, encryption, decryption, and evaluation. We next present the costs with and without key generation to study the impact of the number of members and data size.

Figure 8.7 (a-b) presents the computation cost with varying number of members. Each member's dataset includes 5000 data samples which acquired as a result of the policy negotiations. Figure 8.7 (a) presents the cost of the total computation time excluding HE key generation. There is a linear increase in time with the growing number of members. This is the fundamental cost of encryption and evaluation operations dominated by matrix encryption and addition. To profile the generation of key cost, in Figure 8.7 (b), we conducted similar experiments. Each input size cost increases because of the key generation overhead. The increase is quadratic as a number of slots (plaintext elements) are set to square of input size not to lose any data during input conversion. It is important to note that the cost is independent of the member size because a member generates the key only once in a computation of a consortium. We note that the time overhead of key generation is not a limiting factor as members may generate keys before a consortium is established.

In Figure 8.7 (c-d), we show the costs associated with different data samples. The number of members in a consortium is set to 20. Similar to the previous experiments, the key generation dominates the computation costs. Our experiments also reported no relationship between the cost and number of samples. That is, even though the

Member	Agreement of policy negotiations
U.S.	[(Race="Asian")∨(EVALUATE(age))∨(height <160) ∨(weight <65)∨(CYP2C9 IN (2*/*2, 2*/*3)∨(Amiodarone="Y")∨(Enzyme="Y")]
Brasil	[(Race="Asian")∨(height <165)∨(CYP2C9 IN (2*/*2, 2*/*3)∨EVALUATE (Amiodarone)∨(Enzyme="Y")]
UK	[(Race≠"White")∨(age BETWEEN 20-29 AND >80)∨(height<165)∨(60<weight <100)∨EVALUATE(CYP2C9)∨(Amiodarone="Y"), (Enzyme="Y")]
Israel	[(Race≠ "White")∨(height <160cm)∨(weight <60)∨(CYP2C9=3*/*3)∨(Amiodarone="Y")∨(Enzyme Inducer ="Y")]
Taiwan	[(Race=All)∨(age BETWEEN 20-29)∨(height >170)∨(weight >65)∨(CYP2C9 IN (1*/*2, 2*/*2, 2*/*3, 3*/*3)∨(VK0RC1="G/G")∨(Amiodarone="Y")∨(Enzyme="Y")]
S. Korea	[(Race=All)∨ (age BETWEEN 20-29)∨(height >165)∨(weight >60)∨(CYP2C9 IN (1*/*2, 2*/*2, 2*/*3, 3*/*3)∨(VK0RC1="G/G")∨(Amiodarone="Y")∨(Enzyme="Y")]

Table 8.4: An exploration of CPL policies in the global consortium (illustrated as a plain language): Each member defines asymmetric local policy based on its data diversity. The agreement of share and acquisition policies are depicted as a policy clause in a single row. The agreement result of each member for other members is not presented for brevity.

size of the data samples increases, the overhead is amortized over the operations on the local statistics of the computations (which is the square matrix of the input size in the warfarin dataset); thus the time of computing dose algorithm converges to the number of dataset inputs. This explains the similar trends observed in plots.

8.7.2 Effectiveness of Policies

We validate the performance of privacy-preserving dose model quantitatively and qualitatively. For the warfarin study, these are translated to the following questions: How do policies impact the accuracy of members' warfarin dose prediction? (Section 8.7.2), and Does policies help to prevent the adverse impacts of dose errors on patient health? (Section 8.7.2).

Implications of CPL on Model Accuracy

In our first set of experiments, we validate how well a member prescribe warfarin dose for its local patients and patient's of the consortium members without using CPL. These results are used as a baseline for comparison of varying consortia and data exchange policies throughout. Figure 8.8 (a) sought to identify the local algorithm errors (**P.1**). The errors significantly differ between countries and for the members of the same country (depicted as M_1 and M_2 in the U.S.). The low results are

due to having homogeneous data; all the inputs in these countries have similar traits. For instance, similar age and ethnicity found in a dataset produce over-fitted computation results for its local patients. These findings are validated with use of local algorithms for treatment of other countries' patients. As illustrated in Figure 8.8 (b), the dose errors yield significantly high for particular countries' patients. The results indicate that improvements in dose predictions of local patients and members' patients lay in the creation of data exchange policies to increase the patient diversity.

The next experiments measure the impact of CPL in nation-wide (**P.2**), regional (**P.3**), NATO-EU (**P.4**) and global (**P.5**) consortia. Each member creates a local acquisition policy to acquire the complete data of consortia members (i.e., the acquisition policy of a consortium member complies with the share policy of the requested member). We make three major observations. First, varying partnerships yield different dose accuracy. For instance, members of nation-wide consortium get better dose accuracy than their local results. This result is validated through nationwide consortia and a single member (M_1) in United States (see Figure 8.8 (c)). Second, supporting previous findings, all regional (excluding Asia) and NATO-EU policies decrease the error for both treatment of their patients and the other countries' patients (see Figure 8.8 (d-e)). However, Asia consortium results in unexpected dose errors for the treatment of other regions' patients. This is because nation-wide, regional, and NATO-EU policies include patient population having different characteristics; thus the data obtained through policy negotiations better generalize to the dosages. In contrast, Asia collaboration lacks large enough White and Black groups. Third, the global consortium results in higher dose errors when evaluated for particular countries such as Brazil and Taiwan (see Figure 8.8 (f)). To conclude, while CPL is effective in reducing dose error of a member, the results highlight the

need for the systematic use of CPL through selections and conditionals to obtain better results.

In these experiments, each member dictates a different acquisition policy based on its racial groups. Members aim at having an ideal patient population uniformity. To do so, each member defines a local acquisition policy and negotiates it with other members. Each member sets its share policy to conditionals of being in the same consortium and data size greater than 200; thus, the policy of each member is asymmetric. Table 8.4 shows the simplified notation of the policy agreements in the global consortium. For instance, a member having a small number of white patients defines selections to solely acquire that group and a member having large enough patients for all genotypes sets data-dependent conditionals to obtain patient inputs that are not similar in its data samples (e.g., acquires different genotypes). Figure 8.9 presents a subset of results on dose errors per patient race. The errors of the other races yield similar for each member. The results without CPL conditionals and selections are plotted as a dashed line for comparison. We find that members can improve the dose accuracy with the use of policies. We note that the use of different data-dependent conditionals defined in evaluate does not result in statistically significant accuracy gain.

Implications of CPL on Patient Health

We examine the impact of the dose errors found in the previous section to better quantify the effectiveness of policies on patient health.

To identify the adverse effects of warfarin, we use a clinical study to evaluate the clinical relevance of prediction errors [CLM17] and a medical guide to identify the consequences of over- and under-prescriptions [FA17]. We define errors that are inside and outside of the warfarin safety window, and the under- or over pre-

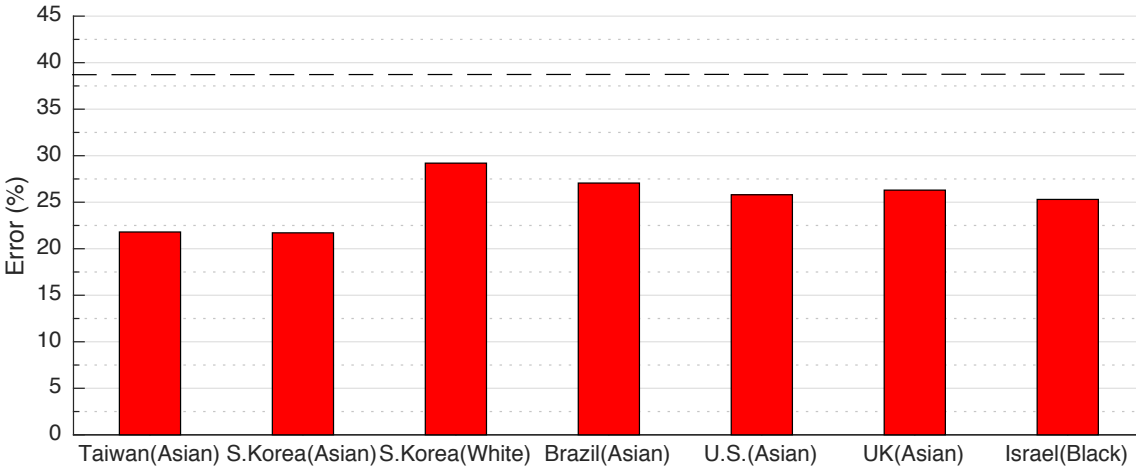


Figure 8.9: Dose accuracy of members using CPL policies defined in Table 8.4. Members construct a model per race after they reconcile the policies. The dashed line is the average error found without the use of conditionals and selections in policies.

scriptions. We consider weekly errors for each patient because using weekly values eliminates the errors posed by the initial (daily) dose. The weekly dose is in the safety window if an estimated dose falls within 20% of its corresponding clinically-deduced value [Int09, K$^+$13]. The deviations falling outside of the safety window is an under- or over prescriptions, and cause health-related risks.

Table 8.5 presents the percentage of patients falls in safety window, over- and -under prescriptions with varying policies of a member. We find that use of CPL increases the number of patients in the safety window. For instance, a member has 43.4% patient with using its local data (single source model), and the member increases the percentage of patients in a safety window with varying consortia and policies, for instance, it is 52.4% in the nation-wide consortium. We conclude that CPL might be useful in preventing errors that introduce health-related risks.

Consortium	U	SW	O	Selections	Conditionals
Single Source	37.7%	43.4%	18.8%	✗	✗
Nation-wide	18.9%	52.3%	28.8%	✓	✓
NATO	19.3%	51.5%	29.2%	✓	✓
Regional	19%	51.3%	29.7%	✓	✓
Global	21.2%	46.8%	32%	✓	✓

Table 8.5: Impact of policies on health-related risks: Results are from a global consortium patients using policy agreement of a member located in the U.S. The member uses the policy defined in Table 8.4. (U: Under-prescription, SW: Safety Window, O: Over-prescription)

8.8 Limitations and Discussion

One requirement for correctly interpreting the CPL policies is a shared schema for solving the compatibility issues among members. For instance, members may interpret the data columns (e.g., column names and types) differently or may not have the information about consortium members (e.g., membership status of an alliance). CPL implements a shared schema describing column names, their types, and explanations of data fields as well as consortium-specific information. Members can negotiate the schema similar to the policy negotiations and revise the schema based on the schema of a negotiation initiator.

CPL provides a set of data-dependent statistical functions (e.g., cosine similarity) to compute pairwise statistics among member's local data. However, there might be a need for other functions that help members decide their data exchange policies. For example, data exchange among finance companies may require calculating the similarity between data distributions. Future work will investigate the integration of different data-dependent statistics into CPL.

Lastly, we did not focus much on the reasons of policy impacts on the prediction success of the dose algorithm and its adverse outcomes on patient health over time. While our evaluation results showed that members could express both complex rela-

tions and constraints on the data exchange through CPL policies, members require establishing true partnerships to improve the prediction model accuracy. While this explanation matches both our intuition and the experimental results, a further domain-specific formal analysis is needed. We plan to pursue this in future work.

8.9 Conclusions

In this chapter, we presented Curie which provides a novel policy language called CPL to define the specifications of data exchange requirements securely for use in collaborative learning settings. Members can assert who and what to exchange separately for data sharing and data acquisition policies. This allows members to efficiently dictate their policies in complex and asymmetric relationships through selections, conditionals, and pairwise data-dependent statistics. We validated Curie in an example real-world healthcare application through varying policies of consortia members. A secure multi-party and (optional) differentially-private model is implemented to illustrate the policy/performance trade-offs. Curie allowed 50 different members to efficiently compute a privacy-preserving model using 5K data samples with 40 inputs in less than a minute. We also showed how an algorithm with effective use of data exchange policies could improve the accuracy of the dose prediction model.

CHAPTER 9

ACHIEVING SECURE AND DIFFERENTIALLY PRIVATE COMPUTATIONS IN MULTIPARTY SETTINGS

9.1 Introduction

Secure and private computation of statistical models is increasingly used in different operational settings from healthcare [KHK+16, CAA+19] to finance [BTW12] and security sensitive applications [FDCB15]. Given the distributed nature of these applications, security and privacy are mostly achieved by utilizing Secure Multiparty Computation (SMC). SMC allows distributed parties to jointly compute an agreed function over their private inputs without revealing those inputs to other parties. Each party learns the final result, but no other information. However, SMC has a major privacy concern for a targeted individual as it does not guarantee that the final result of distributed computation would not leak any information about an individual in a sensitive dataset. Privacy of individuals and their data can be easily violated. [EESA+12, NS08, GKS08]. Therefore, there is a need for a mechanism, where individual parties do not see each others' inputs and further can not infer their data from the final constructed model. Indeed, combining SMC with Differential Privacy (DP) could solve this privacy problem as DP introduces sufficient noise into the final result to prevent any leakage about a single individual.

However, combining SMC with DP is not a trivial task. In an ideal case, a trusted data collector[1] can collect the data, aggregate them and add calibrated noise to the results of the queries (predictions) (Centralized DP (CDP) in Fig. 9.1). However, a trusted party does not exist in many real life scenarios. This technique would easily

[1]A data collector is either one of the parties or a third party. Every discussion here applies to both of the types.

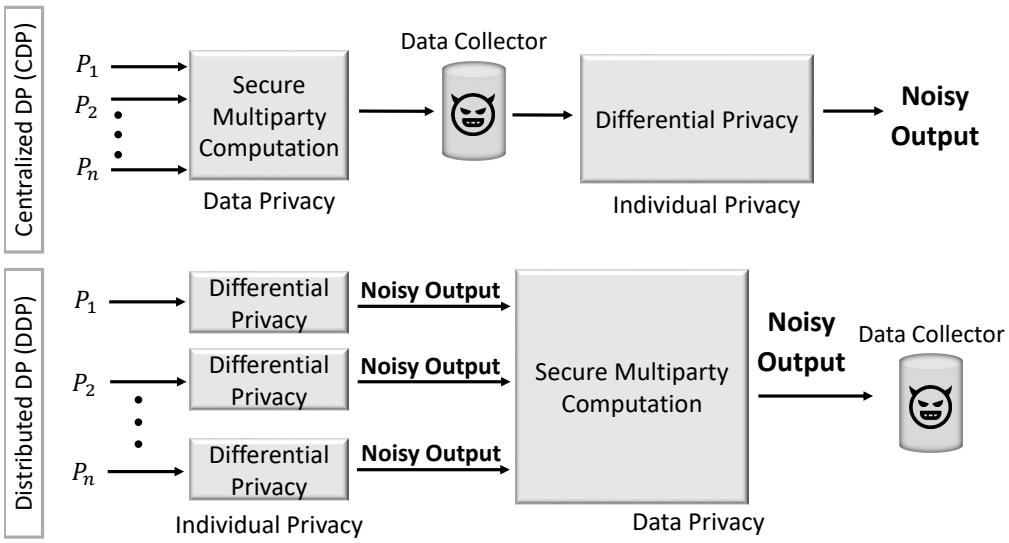


Figure 9.1: Illustration of secure multiparty computation with distributed and centralized differential privacy methods.

leak the model of the sensitive data to an untrusted data collector who collects the final model of the data. Even for scenarios with a trusted data collector, relying on the centralized entity makes it a single point of failure for the entire data collection mechanism.

On the other hand, another mechanism involves applying a data sanitization technique (Distributed DP (DDP) in Fig. 9.1) directly on the local data held by the parties. In this case, the untrusted data collector can not infer individuals' data since sufficient noise is injected by DP to hide the individuals' data. However, this mechanism requires a meticulous analysis since it may lead to a divergent or excessive amount of accumulated noise due to DP at the data collector end. As such, this process may lead to a significant accuracy loss in the final models, which may cause catastrophic consequences in, for example, the healthcare domain. Therefore, enabling distributed differential privacy on local data with differential privacy guarantees on final results is a challenging problem.

In this chapter, we are motivated to provide a solution to this problem. Specifically, we propose a novel protocol for achieving Secure Multiparty Distributed Differentially Private (SM-DDP) computations on sensitive data. The protocol provides the guarantees of both SMC and DP. SMC is provided through Homomorphic Encryption (HE) [Gen09] while DP is provided via Functional Mechanism (FM) [ZZX+12]. An important characteristic of FM is that it injects noise into the feature matrices (i.e., coefficients of objective function), which can be computed independently by each party in a multiparty computational environment. We explore this feature of FM and apply it to linear regression using our SM-DDP protocol, but it can be applied to the computation of any statistical model function that allows independent calculation from the local statistics. We show that the accumulated noise in our protocol is still bounded and convergent by using the infinite divisibility property of Laplacian distribution [McN02]. Finally, we evaluated SM-DDP protocol's computational efficacy on linear regression using two real-world datasets. We compare our results with the use of Centralized DP (CDP) in a multiparty setting as in Fig. 9.1. The intuition is that the distributed setting of DP (DDP), which is proposed in this paper, would cause a greater accuracy loss than the typical client-server setting of SMC systems. However, we show exactly same trade-off can be achieved using the SM-DDP protocol that is presented in Fig. 9.3. The extensive evaluation results indicate that the proposed SM-DDP protocol yields minimal computational overhead—less than a minute for 20 parties with 32 attributes and 10K samples. The individual parties obtain better accuracy than that would be obtained from a single party model. Finally, SM-DDP is scalable while providing security and privacy guarantees.

9.2 Linear Models

In this section, we start by introducing the linear models. We, then, show how to compute linear regression in a distributed fashion.

9.2.1 Background

Assume a database D consists of n observations $\{x_i, y_i\}_{i=1}^{n}$, where x_i is a vector of d attributes (i.e., $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and y_i is a scalar response. The aim is to find a *model function* $f : X \to Y$ that can predict $y_i \in Y$ as close as its actual value using the attributes $x_i \in X$. The type of the regression model is decided by the type of the model function. For instance, in linear regression, the model function is simply a straight line. Model function f takes model coefficients $w = (w_1, w_2, \ldots, w_d)$ and x_i as inputs and outputs a prediction for the value of y_i. The deviations between predicted value and the actual response value are calculated through a *loss function* $\ell : Y \times Y \to \mathbb{R}$. The global value of w over the training data D is calculated by the objective function. We denote the objective function by $\mathcal{L}$ and it is calculated as follows:

$$\mathcal{L}(f, D) = \sum_{i=1}^{n} \ell(f(x_i, w), y_i). \tag{9.1}$$

9.2.2 Distributed Linear Regression

Regression is a statistical approach that explores the relationships between a set of independent variables called *attributes* and one dependent variable called *response*. In regression, the relationship between the attributes and the response is modeled using a prediction function.

In linear regression, L_2-norm of the objective function (i.e., $\ell(f(x_i, w), y_i) = (w \cdot x_i - y_i)^2$) that is minimized in the matrix form as follows:

$$w^* = \arg\min_w \mathcal{L}(f, D) = \arg\min_w \sum_{i=1}^{m} (w \cdot x_i - y_i)^2, \tag{9.2}$$

where m is the number of tuples in the database. To calculate the regression in a distributed way, we represent the regression objective by minimizing with the *Maximum likelihood Estimation* (MLE). MLE allows us to obtain the global solution of the Equation 9.2 as follows[2]:

$$w^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}. \tag{9.3}$$

We characterize the model parameter w of each party using three parameters:

$$\mathcal{P}_i = \mathbf{X}_i^\top \mathbf{X}_i, \mathcal{V}_i = \mathbf{X}_i^\top \mathbf{Y}_i, \mathcal{O}_i = \mathbf{Y}_i^\top \mathbf{Y}_i \tag{9.4}$$

Each party computes its *local statistics* $< \mathcal{P}_i, \mathcal{V}_i, \mathcal{O}_i >$ and shares with other parties. Then, the global values of $\mathcal{P}$,$\mathcal{V}$ and $\mathcal{O}$ are computed using the shared local statistics as follows:

$$\mathcal{P} = \mathbf{X}^\top \mathbf{X} = \begin{bmatrix} X_{i_1}^\top | ... | X_{i_n}^\top \end{bmatrix} \begin{bmatrix} X_{i_1} \\ \vdots \\ X_{i_n} \end{bmatrix} = \sum_{k=1}^{n} \mathbf{X}_{i_k}^\top \mathbf{X}_{i_k} = \sum_{k=1}^{n} \mathcal{P}_k$$

$$\mathcal{V} = \mathbf{X}^\top \mathbf{Y} = \begin{bmatrix} X_{i_1}^\top | ... | X_{i_n}^\top \end{bmatrix} \begin{bmatrix} Y_{i_1} \\ \vdots \\ Y_{i_n} \end{bmatrix} = \sum_{k=1}^{n} \mathbf{X}_{i_k}^\top \mathbf{Y}_{i_k} = \sum_{k=1}^{n} \mathcal{V}_k$$

[2]A unique solution only exists if $(\mathbf{X}^\top \mathbf{X})^{-1}$ is non-singular. In other cases, there are techniques for solving Equation 9.2 [Myu03]; however, it is out of the scope of this paper.

$$\mathcal{O} = \mathbf{Y}^\top \mathbf{Y} = \begin{bmatrix} Y_{i_1}^\top | ... | Y_{i_n}^\top \end{bmatrix} \begin{bmatrix} Y_{i_1} \\ \vdots \\ Y_{i_n} \end{bmatrix} = \sum_{k=1}^{n} \mathbf{Y}_{i_k}^\top \mathbf{Y}_{i_k} = \sum_{k=1}^{n} \mathcal{O}_k,$$

where n is the number of parties in the collaboration. Using this, the global coefficients can be computed as follows:

$$w^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y} = \mathcal{P}^{-1} \mathcal{V}. \tag{9.5}$$

In order to calculate the error of the global function, we rewrite the objective function in Equation 9.2 in terms of the local statistics (i.e., matrix form) as follows:

$$\begin{aligned} \sum_{i=1}^{m} (w \cdot x_i - y_i)^2 &= (\mathbf{X}w - \mathbf{Y})^\top (\mathbf{X}w - \mathbf{Y}) \\ &= ||(\mathbf{X}w - \mathbf{Y})||^2 \\ &= w^\top \mathbf{X}^\top \mathbf{X} w - 2w^\top \mathbf{X}^\top \mathbf{Y} + \mathbf{Y}^\top \mathbf{Y} \\ &= w^\top \mathcal{P} w - 2w^\top \mathcal{V} + \mathcal{O}, \end{aligned} \tag{9.6}$$

where $|| \cdot ||$ denotes the Euclidean norm. We note that even though we do not need $\mathcal{O}$ to calculate the global coefficients, it is used for computing the error of the model.

9.3 Technical Preliminaries

Preserving the privacy of the users and data is a long-studied problem in the area of cryptography [SCR+11, DPSZ12, Dan15, CMS11, SKLR04, DKM+06]. As a result of these long-term studies, there are several theoretically well-studied tools that can be employed to protect the data and user privacy such as Secure Multiparty Computation (SMC) [DPSZ12] and Differential Privacy (DP) [Dwo08]. In this section, we

introduce the essentials of the secure computation and differential privacy primitives to understand the implementation of SM-DDP algorithms. Particularly, we introduce Homomorphic Encryption (HE) to provide SMC and Functional Mechanism (FM) to provide DP guarantees.

9.3.1 Secure Multiparty Computation

SMC allows the computation of a function with multiple inputs from different users while keeping the users' inputs hidden from each other. For instance, each party P_i in a n-party environment holds input x_i learns nothing but the output $f(x_1, ..., x_n)$ of a computation. In the literature, SMC schemes are mostly achieved via either the Yao's garbled circuits [Yao82] or Homomorphic Encryption (HE) [Gen09]. In the following, we use HE to provide guarantees of secure computation.

Homomorphic Encryption (HE)- HE provides an ability to evaluate the functions directly on the encrypted data while keeping the data confidential. The primary advantage of the HE is that it does not require any interaction between the parties other than the data exchange. That is, there is no additional communication complexity. However, it may introduce computational overhead on large plaintexts. Recent works improved its performance significantly by introducing new techniques like single instruction, multiple data (SIMD) operations [SV14] or using different mathematical assumptions like learning with errors LWE [BGV14, BV14a] (see [AAUC18] for a recent survey about HE).

An HE scheme is primarily characterized by four operations: key generation ($KeyGen$), encryption (Enc), decryption (Dec), and evaluation ($Eval$). $KeyGen$ is the operation that is used to generate a secret and public key pair for the asymmetric version of HE or a single key for the symmetric version. $KeyGen$, Enc and

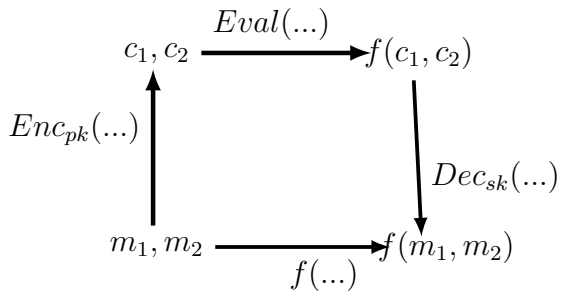


Figure 9.2: HE operations of encryption, evaluation, and decryption (pk is the public key, sk is the secret key, and f is the function desired to be computed).

Dec are similar to the ones used in conventional encryption schemes. However, $Eval$ is an HE-specific operation, which takes ciphertexts as input and outputs a ciphertext corresponding to a functioned plaintext. Fig. 9.2 illustrates a commutative diagram depicting the relationship among the four major operations. The simplified version of the diagram shows only one homomorphic encryption with two ciphertexts [Gen14].

9.3.2 Differential Privacy (DP)

DP is a statistical disclosure control technique ensuring that the outputs of queries do not leak information about the individuals found in a dataset. It injects a certain amount of noise into the replies of the queries so that while it is not possible to infer an individual-level leak, the output of the query is still "almost" the same. In other words, query results of a data release algorithm for two closely similar data sets give the same answer. The formal definition of $\epsilon-$differential privacy is formulated as follows [DR14]:

Definition 1. *A randomized algorithm $\mathcal{M}$ is ϵ-differentially private if for all data sets D and D' differing on at most one element and all $S \subseteq Range(\mathcal{M})$,*

$$Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \times Pr[\mathcal{M}(D') \in S], \tag{9.7}$$

where Range($\mathcal{M}$) shows all possible outputs of the function (query), f.

The definition states that two adjacent sets D and D', which differs at most one element, act approximately the same against a query[3] defined by a given mechanism M. ϵ can be considered as the degree of the privacy guarantee and the amount of information which can be learned from a result of a single query is bounded by $\exp(\epsilon)$. Since ϵ is too small, its guarantee is preserved for consecutive queries. Differential privacy works on the release mechanism and does not modify data or the format of the data in any way.

The parameter ϵ, called *privacy budget*, is the main parameter to tune the balance between privacy and accuracy. Decreasing ϵ increases the privacy guarantees while decreasing the accuracy. The common mechanism to control the amount of noise that needs to be added is *Laplace Mechanism* (LM). In this case, the noise is drawn from a Laplace Distribution. The probability density function of LM is as follows:

$$Lap(x|b) = \frac{1}{2b} exp\left(-\frac{|x|}{b}\right), \tag{9.8}$$

for scale b and center 0. It is shown that LM preserves ϵ-differential privacy [DR14].

Definition 2. *Given any function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$, the mechanism is a Laplace Mechanism $\mathcal{M}$ if:*

$$\mathcal{M}(x) = f(x) + \eta, \tag{9.9}$$

where $x \in \mathcal{X}$ and η is a vector of independent and identically distributed random variables drawn from Lap($\Delta f/\epsilon$).

In addition to the ϵ, *sensitivity* is another important parameter in DP to determine the optimum noise amount. It is defined as follows:

[3]The queries or functions correspond to the predictions in the statistical models.

Algorithm 1 [Z+12] Functional Mechanism (D, $\mathcal{L}$, ϵ)

Input: Let $\mathcal{L}(f, D) = \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \sum_{i=1}^{n} \lambda_{\phi_i} \phi(w)$

1: Set $\Delta = 2 \max_{w} \sum_{i=1}^{n} ||\lambda_{\phi_i}||_1$
2: **for** each $j \in \{0, ..., J\}$ **do**
3: **for** each $\phi \in \Phi_j$ **do**
4: $\lambda_\phi = \sum_{i=1}^{n} \lambda_{\phi_i} + Lap(\frac{\Delta}{\epsilon})$ ▷ *noise inject*
5: **end for**
6: **end for**
7: Compute new $w^* = \arg\min_{w} \mathcal{L}(f, D)$ ▷ *optimize*
8: **return** w^*

Definition 3. *For a function $f : D \rightarrow R^k$, sensitivity of f is*

$$\Delta f = \max_{D,D'} \parallel f(D) - f(D') \parallel \tag{9.10}$$

for all D, D' differing in at most one element.

The sensitivity shows the maximum number of elements that can change in two different queries.

Functional Mechanism (FM)- FM is an algorithm that is used to provide differential privacy guarantees for a set of linear models [ZZX+12]. It is an extension of the Laplace Mechanism. The goal of the algorithm is injecting the noise to the polynomial coefficients of a model's objective function. This is accomplished with the mechanism of *objective perturbation* [CMS11]. The optimization of the noisy objective function gives new model parameters that ensure the ϵ-privacy of each element in a database. Algorithm 1 [ZZX+12] presents the functional mechanism.

As illustrated in Algorithm 1, FM takes a dataset D, the polynomial representation of the objective function L, and the privacy budget ϵ as inputs and it returns the differentially private model coefficients w^*. It firstly injects noise drawn from

a Laplacian distribution ($Lap(\frac{\Delta}{\epsilon})$) into all the coefficients λ_{ϕ_i} of the polynomial representation of the objective function and then the optimization is performed using noisy coefficients. It is shown that it satisfies ϵ-differential privacy [ZZX+12] i.e., the predictions using w^* does not leak any information about an individual in the database data. For example, if we have a quadratic objective function in the matrix form of $w^\top \mathcal{P} w + w^\top \mathcal{V} + \mathcal{O}$, where $\mathcal{P}$, $\mathcal{V}$, and $\mathcal{O}$ are the coefficients of the polynomial representation of the objective function. FM firstly injects noise into the coefficients, which results in $w^\top \mathcal{P}^* w + w^\top \mathcal{V}^* + \mathcal{O}^*$. Then, the optimization problem (i.e., $w^* = \arg\min_w \mathcal{L}(f, D)$) is solved using $\mathcal{P}^*$, $\mathcal{V}^*$, and $\mathcal{O}^*$.

9.4 Secure and Differentially-private Distributed Computations

In this section, we propose a novel protocol for secure multiparty distributed and differentially private (SM-DDP) computations through the use of homomorphic encryption (HM) and functional mechanism (FM). We evaluate its application to linear regression and discuss its extension to the logistic regression that can be used in supervised classification.

Consider n parties $P_1, \ldots, P_n$, where each has private horizontally distributed database $D_1, \ldots, D_n$. Each database consists of a certain number of tuples in the format of $t_i = (x_i, y_i)$. The parties would like to jointly build a linear model of the pooled database $f(D)$, where $D = \cup_{i=1}^{n} D_i$ so that the security guarantees of both SMC and DP are preserved. Before running the protocol, each party in the collaboration agrees on the function to be computed and compute a collection of local statistics $M_i = (L_{i_1}, \ldots, L_{i_t})$. We assume the linear model can be computed using the local statistics generated by each party independently i.e.,

Algorithm 2 Computation of Linear Regression using SM-DDP protocol

Input: Each party holds a database in the format of $D_i = (x_i, y_i)_{i=1}^{n}$ i.e., horizontally partitioned
The global privacy budget ε.
Output: The differentially private global regression model of $D = \cup_{i=1}^{n} D_i$

Setup: Runs at the party P_i (DC)

1: $(pk_i, sk_i) \leftarrow KeyGen()$ ▷ generate the key pair of HE
2: η_{max}, $\eta_{min} \leftarrow ComputeMinMax(D)$ ▷ calculate the global max and min of each attribute via [43]
3: $\Delta \leftarrow 2(d+1)^2$ ▷ calculate the global sensitivity, d is the number of attributes

Secure Regression Protocol: each party P_j runs locally

Input: Received aggregate statistics for all previous parties as:
ξ: $E_{pk_i}(\sum_{k=1}^{j-1} \mathcal{P}_k^*)$
κ: $E_{pk_i}(\sum_{k=1}^{j-1} \mathcal{V}_k^*)$
δ: $E_{pk_i}(\sum_{k=1}^{j-1} O_k^*)$

4: $D_j^{norm} \leftarrow (D_j - \eta_{min})/(\eta_{max} - \eta_{min})$ ▷ perform min-max normalization
5: $\mathcal{P}_j \leftarrow \mathbf{X}_j^\top \mathbf{X}_j$, $\mathcal{V}_j \leftarrow \mathbf{X}_j^\top \mathbf{Y}_j$, and $O_j \leftarrow \mathbf{Y}_j^\top \mathbf{Y}_j$ ▷ compute local statistics
6: $\varepsilon_i \leftarrow \alpha\varepsilon$ ▷ compute its share from the global privacy budget
7: $(\mathcal{P}_j^*, \mathcal{V}_j^*, O_j^*) \leftarrow FM.NoiseInject(\mathcal{P}_j, \mathcal{V}_j, O_j)$ ▷ apply FM noise injection
8: $C_j^* = \left(E_{pk_i}(\mathcal{P}_i^*), E_{pk_i}(\mathcal{V}_i^*), E_{pk_i}(O_i^*)\right)$ ▷ perform encryption
▷ add its own encrypted local statistics to the received aggregate statistics
9: $E_{pk_i}(\sum_{k=1}^{j} \mathcal{P}_k^*) \leftarrow E_{pk_i}(\mathcal{P}_j^*) + \xi$
10: $E_{pk_i}(\sum_{k=1}^{j} \mathcal{V}_k^*) \leftarrow E_{pk_i}(\mathcal{V}_j^*) + \kappa$
11: $E_{pk_i}(\sum_{k=1}^{j} O_k^*) \leftarrow E_{pk_i}(O_j^*) + \delta$
12: **Send(** $E_{pk_i}(\sum_{k=1}^{j} \mathcal{P}_k^*)$, $E_{pk_i}(\sum_{k=1}^{j} \mathcal{V}_k^*)$, $E_{pk_i}(\sum_{k=1}^{j} O_k^*)$ **)** to P_{j+1} ▷ send updated aggregate statistics to the next party.

Reconstruction: runs at the party P_i (DC)

Input: Received aggregate statistics for all parties as:
ξ: $E_{pk_i}(\sum_{k=1}^{n} \mathcal{P}_k^*)$
κ: $E_{pk_i}(\sum_{k=1}^{n} \mathcal{V}_k^*)$
δ: $E_{pk_i}(\sum_{k=1}^{n} O_k^*)$

13: $\mathcal{P}^* \leftarrow D_{sk_i}\left(\xi\right)$ ▷ acquire the cleartext
14: $\mathcal{V}^* \leftarrow D_{sk_i}\left(\kappa\right)$ ▷ acquire the cleartext
15: $O^* \leftarrow D_{sk_i}\left(\delta\right)$ ▷ acquire the cleartext
16: $(\mathcal{P}^*, \mathcal{V}^*, O^*) \leftarrow FM.Optimize(\mathcal{P}^*, \mathcal{V}^*, O^*)$ ▷ apply optimization
17: $w^* \leftarrow \mathcal{P}^{*-1}\mathcal{V}^*$ (i.e., $w^* = \arg\min_w w^\top \mathcal{P}^* w + w^\top \mathcal{V}^* + O^*$) ▷ compute the global parameters
18: $Err \leftarrow w^{*\top} \mathcal{P}^* w^* + w^{*\top} \mathcal{V}^* + O^*$
19: **Publish(** w^*, Err **)** to all parties.
▷ Use of Model:
20: $f(x_i, w^*) \leftarrow \sum_{i=1}^{n} x_i w^*{}_i$ for an input $x_i \in \mathbf{X}_i$ ▷ computes the normalized predictions
21: $y_{pred} \leftarrow f(x_i, w^*)(\eta_{max} - \eta_{min}) + \eta_{min}$ ▷ perform de-normalization to get actual values

$\eta_{global} = f(M_1, \ldots, M_i, \ldots, M_n)$. We define the guarantees and goals of our protocol as follows:

- *Individual privacy:* No information leaks about the individuals in the private databases held by the parties, i.e., tuples t_i is not leaked.
- *Data privacy:* Information about the statistics of the data does not leak in the databases held by the parties, i.e., the statistics about the data M_i is not leaked.
- *Correctness:* The parties receive the correct output of the model.

We note that using SMC only would violate the individual privacy while using DP only violates the data privacy. In our combined protocol, we achieve individual privacy through FM and data privacy through HE and since all operations in the protocol are deterministic, the correctness is satisfied by design. We note that we assume there is a secure channel between parties to exchange messages.

Fig. 9.3 illustrates our protocol to be able to perform SM-DDP computations. It is initiated by one of the parties called *data collector* (DC). In the setup phase, DC generates a key pair (pk_i, sk_i) and computes its own local statistics M_i independent from other parties. Then, in the next phase, DC applies DP by injecting (adding) noise drawn from a random distribution that satisfies ϵ-differential privacy into its local statistics. The encryption of the noisy local statistics is transmitted to the next party P_{i+1}. The next party P_{i+1} also computes its local statistics and injects noise into them. The result is encrypted with pk_i and the function is evaluated homomorphically with the inputs of parties P_i and P_{i+1}. The protocol is continuous in the same way, where parties are located in a ring topology. At the final step, the securely evaluated function result is used by the party P_i which decrypts it with sk_i. In the end, P_i reveals the differentially private global model.

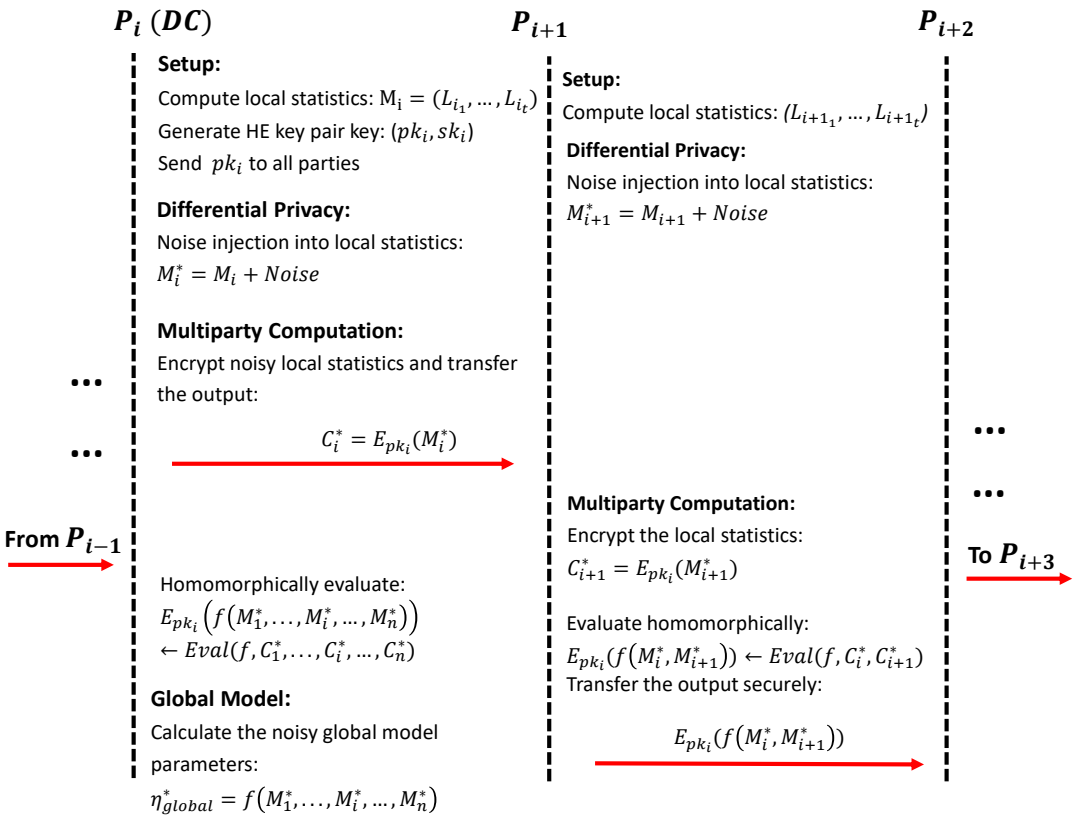


Figure 9.3: Secure Multiparty Distributed Differentially Private (SM-DDP) protocol for the computation of a linear model coefficients. The parties create a ring topology and the Data Collector (DC) initiates the protocol. The protocol can be applied to any statistical model function that allows independent calculation of local statistics.

9.4.1 Case Study: Linear Regression

In this subsection, we show how to compute linear regression using our protocol proposed in Fig. 9.3. Particularly, we use functional mechanism shown in Algorithm 1 by splitting it into two parts: $NoiseInject()$ and $Optimize()$. In $NoiseInject()$, the noise drawn from Laplacian distribution (Equation 9.8) is injected into each coefficient of the polynomial representation of the objective function. Then, in $Optimize()$, the optimization problem of the objective function is solved by applying regularization and spectral trimming introduced in [ZZX+12] in order to avoid unbounded noisy objective function. Moreover, in FM, it is assumed that $\sqrt{\sum_{i=1}^{d} x_{id}^2} \leq 1$. Therefore, a secure maximum computation is performed to calculate η_{min} and η_{max} in setup phase of Algorithm 2, where η_{min} (resp. η_{max}) is vector consists of global minimum (resp. maximum) of each attribute. Before applying FM, each party normalizes its database using the global maximum and minimum values. This guarantees that the local sensitivity of the parties is always same as the global sensitivity as we focus on the horizontally distributed data.

Algorithm 2 illustrates the computation of linear regression algorithm using the protocol presented in Fig. 9.3. In linear regression, the global model is calculated by simply aggregating locally calculated noisy statistics. While aggregating the local statistics, the noise of each party is aggregated as well. Therefore, it is necessary to make sure the final model will not violate ϵ-differential privacy nor cause an unbounded noise. Particularly, the noise is injected to each coefficient as follows:

$$\mathcal{P}_i{}^* = \mathcal{P}_i + Lap\Big(\frac{\Delta}{\epsilon_i}\Big). \tag{9.11}$$

Then, when DC computes the global model, the local statistics are summed up as follows:

$$\mathcal{P}^* = \sum_{i=1}^{n} \mathcal{P}_i{}^* = \sum_{i=1}^{n} \Big(\mathcal{P}_i + Lap\Big(\frac{\Delta}{\epsilon_i}\Big)\Big) = \mathcal{P} + \sum_{i=1}^{n} Lap\Big(\frac{\Delta}{\epsilon_i}\Big). \tag{9.12}$$

Moreover, $\mathcal{V}^*$ and $\mathcal{O}^*$ can be computed similarly. In all $\mathcal{P}^*$, $\mathcal{V}^*$, and $\mathcal{O}^*$, the noise term is $\sum_{i=1}^{n} Lap\big(\frac{\Delta}{\epsilon_i}\big)$. In order to make sure that the accumulated noise is also Laplacian distribution, we use the following theorem.

Theorem 1. *Let Y, Y_1, Y_2... be non-degenerate and symmetric i.i.d. random variables with variance $\sigma^2 > 0$, and let ν_p be a geometric random variable with mean $1/p$, independent of the Y_i's. Then, the following statements are equivalent (Proof is given in [McN02]):*

(i) Y is stable with respect to geometric summation, i.e., there exist constants $a_p > 0$ and $b_p \in \mathbb{R}$, such that

$$a_p \sum_{i=1}^{\nu_p} (Y_i + b_p) = Y \quad \forall p \in (0, 1) \tag{9.13}$$

(ii) Y possesses the Laplace distribution with mean zero and variance ν_2. Moreover, the constants a_p and b_p must be of the form: $a_p = \sqrt{p}$, $b_p = 0$

From the theorem above, a Laplace distribution can be calculated by summing up several Laplace distributions in a certain form. In other words, the sequence of partial sums, $a_p \sum_{i=1}^{\nu_p}(Y_i + b_p)$ converges to a Laplace distribution under beta-distributed a_p. We addressed requirements of the theorem in Algorithm 2 by multiplying the noise distribution of local parties with a number drawn from the geometric distribution i.e., $a_p \sum_{i=1}^{n} Lap\big(\frac{\Delta}{\epsilon_i}\big)$, where a_p is a geometric random variable.

9.5 Performance Evaluation

In this section, we give the experimental results for the application of our SM-DDP protocol to linear regression. Table 9.1 presents the notations used throughout the experiments. We first demonstrate how we set the parameters that are introduced in the distributed setting. Particularly, the success probability of the geometric

random variable p in Equation 9.13 and α introduced in Algorithm 2 is investigated. After experimentally tuning these two parameters, we test the final protocol with a different dataset without random sampling directly as it is collected. During evaluation, we focus on the following questions: (i) Can we obtain a differentially private global linear regression model from differentially private local statistics? (ii) Does our approach support up to 100 parties? (iii) How long does it take to complete the protocol? (iv) Does it guarantee the security and privacy of both data and individuals? We analyzed and discussed each of these questions in Sections 9.5.1-9.5.4.

Dataset- We used two real-world datasets to evaluate the algorithms of our protocol. Both datasets include highly sensitive data. The first dataset is *Integrated Public Use Microdata Series* (IPUMS) [II17]. It contains 370K decennial census records of people living in the US with 14 attributes, 7 of which are demographic information and the rest are working hours per week, the number of years residing in the current location, the number of children, the number of automobiles, and the annual income. The attributes are used to predict the *annual income* of a person. The second dataset is the warfarin dataset collected by the International Warfarin Pharmacogenetics Consortium (IWPC) [Int09]. The dataset contains clinical and genetic data of patients to predict the stable therapeutic dose of warfarin. Clinical data includes demographics, background, and phenotypic attributes. Genetic data includes genotype variants of CYP2C9 (*1, *2 and *3) and VKORC1 (one of seven single nucleotide polymorphisms in linkage disequilibrium). 21 sites in 9 countries and four continents contributed to the dataset. We used a subset of this dataset wherein patient samples include no missing attributes. Overall, we used 1400 complete patient samples from seven medical institutions. We used IPUMS dataset to experimentally set the parameters of our protocol and we tested the final protocol

Table 9.1: Abbreviations and notations used in experiments

Notation	Description	Range
DDP	Distributed Differential Privacy	-
NoDP	No Differential Privacy	-
CDP	Centralized Differential Privacy	-
ϵ	global privacy budget	{0.1,0.2,0.4,0.8,1.6,3.2,6.4,12.8}
ϵ_i	local privacy budget	$\epsilon_i = \alpha\epsilon$
α	local privacy ratio i.e., $\alpha = \epsilon_i/\epsilon$	{1,10,100}
p	success probability of the geometric random variable, a_p	{0.1,0.5,0.9}
n	number of parties	[1,100]
L	number of levels in HElib	{4,6}
nslots	number of slots in HElib	calculated by HElib
s	minimum of *nslots*	$\{8^2, 16^2, 24^2, 32^2, 40^2\}$

with the IWPC dataset, where each party corresponds to a medical institution in the dataset.

Evaluation Metrics- We applied stratified cross validation to split the dataset into training and test sets. To evaluate the model's prediction accuracy, we used *Mean Squared Error* (MSE) as it is a commonly used metric for linear regression analysis. It is calculated as $\frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)$, which gives the average of the squared errors between actual (y_i) and predicted ($\hat{y_i}$) values in n data samples. The lower values of MSE shows better predictions. Finally, it is worth mentioning that all the experiments show 100 independent runs and their average is reported in this work.

Experimental Setup- To evaluate the computational overhead, we used open-source HE library (HElib) [], which implements BGV homomorphic cryptosystem [BGV14] and we ran experiments on 16-core Intel Xeon CPU at 1.90 GHz running Linux Server. In BGV, a prior level L should be set before initiating the computation. In addition to the level L, HElib also has a parameter *nslots* which defines a number of slots for the utilization of SIMD techniques [SV10, SV14]. HElib allows encrypting multiple messages at one time through its SIMD features by pack-

ing the messages into the independent slots of an array. We note that the parameter L affects not only the number of allowed homomorphic operation but also all the other timings and the key size. Therefore, the parameter L should be optimized so that the minimum L is set without failure of the decryption. To do so, we first calculated the table of a number of homomorphic operations for each level L and we used the minimum level for each number of the party.

Furthermore, in our experiments, the data encrypted is the local statistics i.e., not the raw data. The size of the local statistics is considered the same for all the parties. The homomorphic operation computed for linear regression is the element-wise matrix addition. To take advantage of HElib library SIMD features, we converted matrices into arrays and the parameter of minimum number for *nslots* was set to the length of the array for each statistics. This prevents data loss during the conversion. We did not utilize any multi-threading technique during our experiments to see the lower bound of the performance of our protocol. Thus, our results are lower bound and can be improved with the use of any multi-threading technique.

9.5.1 Accuracy Analysis

We evaluate the accuracy-privacy trade-off of distributed evaluation of differential privacy on linear regression. Specifically, we compare our results with the centralized approach. In *Centralized Differential Privacy* (CDP), the accuracy of the regression depends only on the global privacy budget ϵ. However, in *Distributed Differential Privacy* (DDP), each party has its own local privacy budget ϵ_i and DDP is applied independently by each party. We note that this is a particular property of FM. In FM, data is first normalized and the optimum noise amount is only determined by the number of the attributes which is same for all parties. Therefore, the size and the range of the local statistics are same for all the parties; it does not depend

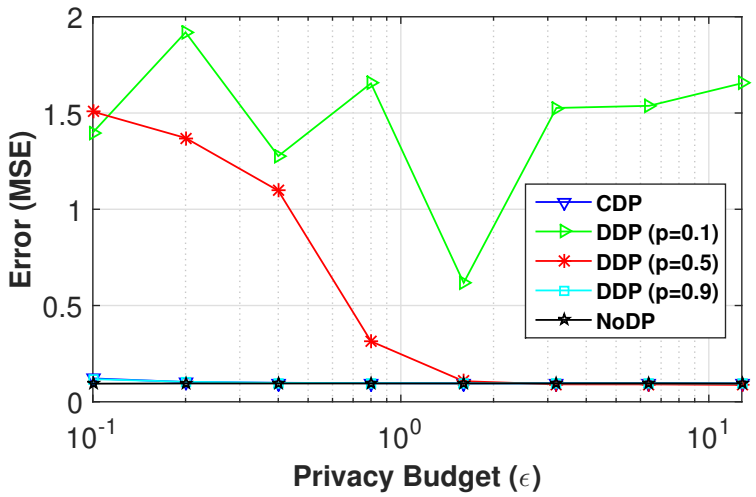


Figure 9.4: Tuning p . Variation of error is tested for several values of p. As a result, $p = 0.1$ is not stable or convergent; $p = 0.5$ is convergent, but error is much higher than CDP for especially small ϵ values. Hence, we chose $p = 0.9$ as the best case.

on the number of tuples in the local database. Since all parties are identical, we choose the same local privacy budget ϵ_i for all the parties. Finally, in our fist three experiments (Fig. 9.4, 9.5, and 9.7), we used IPUMS dataset and split it into parties using random sampling methods. In the last experiment, we used IWPC dataset for accuracy evaluation. We split the dataset based on the given medical institutions (See Fig. 9.6)

The first set of experiments was conducted to analyze the optimum value of p, which is a parameter of geometric random variable a_p given in Equation 9.13. In theory, a_p is required to obtain a Laplace distribution in the global model, thereby it is required to be able to satisfy ϵ-differential private model. To present the impact of the parameter p on the accumulated global noise, we kept the party number constant for several values of p and various ϵ values ($\epsilon_i = \epsilon$). To do so, each party multiplies the noise drawn from Laplace distribution with a random variable a_p, which is a geometric random variable with success probability p. We compared the error rates of CDP, DDP, and NoDP algorithms in terms of MSE.

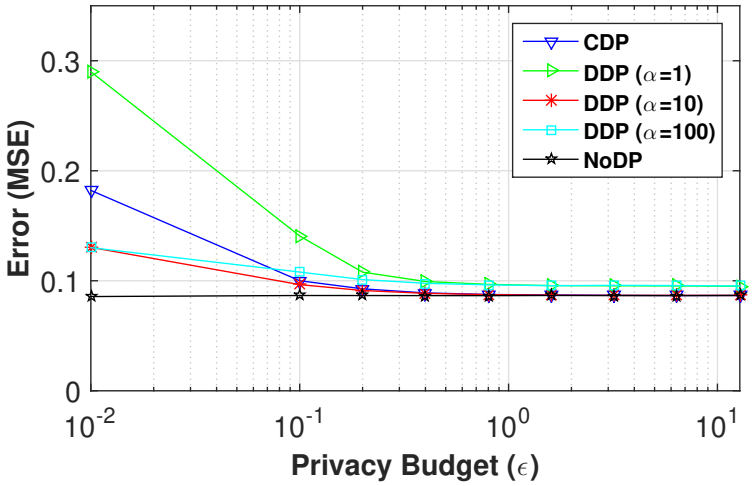


Figure 9.5: Tuning ϵ_i. Variation of error is tested for several values of local privacy budget ϵ_i for $\alpha = \epsilon_i/\epsilon$. For $\alpha = 1$, error is too high for small ϵ values. For $\alpha = 10$, error is lower than CDP and and it converging to the value as NoDP. For $\alpha = 1$, error is low, but it converges to a value higher than NoDP. Hence, we chose $\alpha = 10$ as the best case.

Fig. 9.4 illustrates the error and privacy budget trade-off for various values of p. We varied p from $\{0.1, 0.5, 0.9\}$. We found that DDP with $p = 0.1$ does not converge to a value while increasing the value of ϵ. However, $p = 0.5$ and $p = 0.9$ converges to the same value as NoDP as it is desired and when p is 0.9, it gives similar results to CDP. In the sequel, we tuned $p = 0.9$ and used it in our experiments.

In the second set of experiments, we were interested in finding the optimal local privacy budget ϵ_i for a predetermined global privacy budget. In other words, we assume all parties agree on a global privacy budget according to the sensitivity of the dataset, which was indeed calculated by the number of attributes. We denote the ratio of local privacy budget to the global privacy budget as α, i.e., $\alpha = \epsilon_i/\epsilon$. We first tried the value of α less than 1, the result of DDP was much worse than CDP. This is because smaller ϵ_i means more noise injected locally by each party than the centralized approach. This noise decreases the accuracy significantly. Therefore, we changed α from $\{1, 10, 100\}$ and compared the results with CDP and NoDP

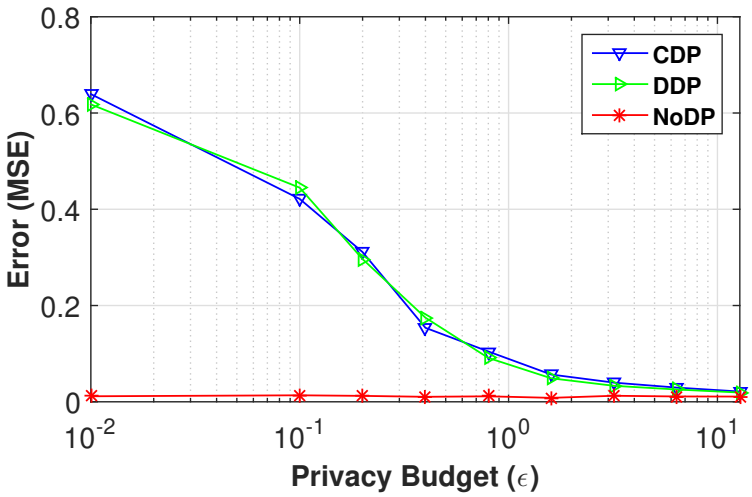


Figure 9.6: A real test: Warfarin dataset with 7 parties with $\epsilon_i = n\epsilon$ and $p = 0.9$. Exactly the same trade-off as the centralized differential privacy is obtained.

mechanisms. The results are presented in Fig. 9.5. We found that if α is the number of parties, which is 10 in this experiment, the plot gets closer to CDP and the error is converging to NoDP, which is the desired case. Therefore, in the rest of experiments, we set $\alpha = n$, where n is the number of parties.

So far, we tuned the parameters of our approach experimentally. Now, in our last experiment, we evaluated the efficiency of our protocol using the dataset (IWPC dataset) collected from multi sources. We applied DP locally on each party's dataset and calculated the global model and error. Our goal was to see the feasibility of our approach in a real case and test the feasibility of our approach.

In this experiment, we set $\epsilon_i = n\epsilon$, $p = 0.9$ as we found in earlier experiments. We compared the performance of CDP, DDP, and NoDP algorithms. Fig. 9.6 shows MSE rates for varying ϵ. We found that the same trade-off with CDP can be achieved by applying DP while training the classifiers locally. We note the DDP is also converging to the error of NoDP when ϵ approaches infinity as desired.

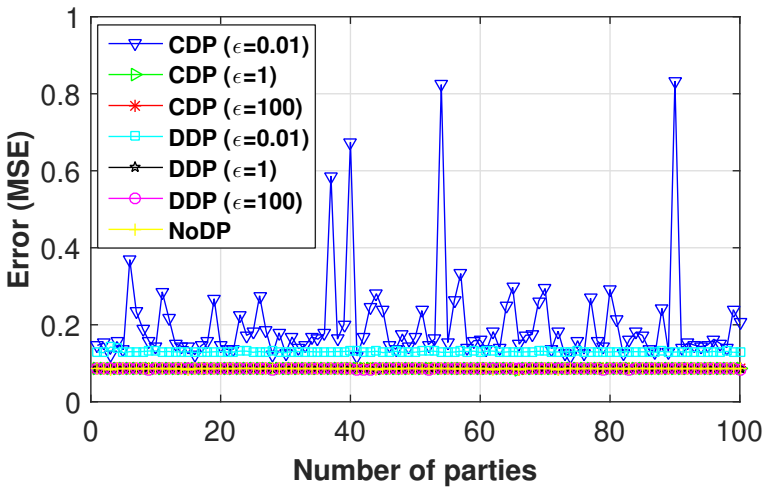


Figure 9.7: Impact of number of parties in the collaboration for $\epsilon_i = n\epsilon$ and $p = 0.9$.

9.5.2 Scalability Analysis

In this set of experiments, we evaluated the scalability of our proposed protocol. We set $\epsilon_i = n\epsilon$, where n is the number of parties; as we found $\alpha = n$ is optimum and for a different number of parties, we split the dataset into the number of parties (n) by using random sub-sampling. Then, each party applies DP locally, but we note that the pooled dataset is still the same.

Laplace distribution is infinitely divisible [McN02]. Therefore, the accumulated error of global model should not be affected by the number of parties. We ran the analysis for some users ranging from 1 to 100 and present the results in Fig. 9.7. The results demonstrated an interesting point, which is when $\epsilon = 0.01$, even though CDP is not stable, DDP is. On the other hand, when ϵ is 1 or 100, the error rate stays the same even for 100 parties. This means our protocol is scalable even for 100 parties.

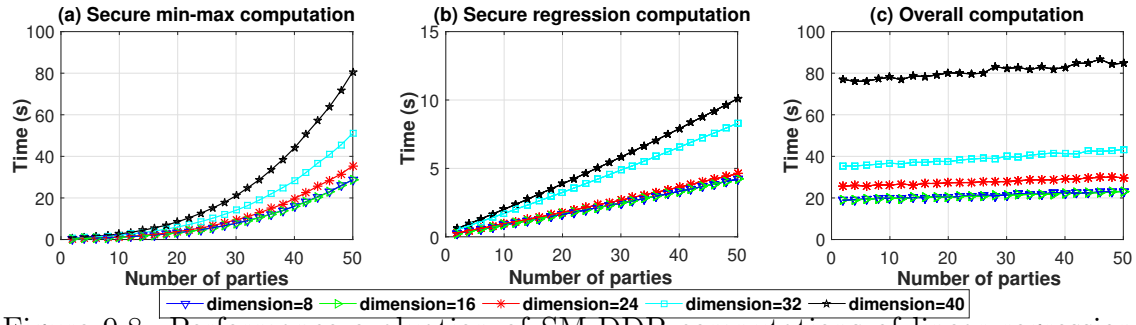


Figure 9.8: Performance evaluation of SM-DDP computations of linear regression algorithm.

9.5.3 Computational Overhead Analysis

In this subsection, we evaluate the computational overhead of linear regression presented in Algorithm 2. We found that DP algorithms do not introduce computational overhead. Therefore, we only evaluate the computational overhead of our SMC algorithm, which consists of three main parts: Key generation of HE, min-max, and regression computation.

Fig. 9.8 shows the computation time for different dimension sizes. Fig. 9.8a presents the time for secure computation of finding global min-max of each attribute. It increases quadratically with the number of parties. However, this algorithm runs at the setup phase, so it is performed before initiating the computations. There are two interesting results worth to note. First, the time of secure regression computation increases linearly as a number of parties in the collaboration increases, but with a different slope for dimension, which is illustrated in Fig. 9.8. The reason for the linear increase is that the number of encryptions and homomorphic evaluations are directly scaled by the number of parties in the group. Second, similar results hold for the overall computation time (see Fig. 9.8c), but as a minor change since the key generation time shifts the lines in the y-axis and also increases the scale. However, similar to the secure min-max computation, the execution of the key generation algorithm does not require all parties in the group to be online since it occurs in the

setup phase. On the other hand, we also note that size of the local database of each party does not have an impact on the total computational time since parties only share the local statistics, which is dependent on the attribute size, instead of the raw data. As can be seen in Fig. 9.8c, the overall computation of the protocol including both offline and online phases for 20 parties with 32 attributes and 10K samples is less than a minute. Hence, our SM-DDP protocol yields minimal computational overhead.

9.5.4 Security and Privacy Analysis

In this section, we discuss the security and privacy guarantees of SM-DDP protocol given in Fig. 9.3. As all the communication among the parties is encrypted, the security of the algorithm is simply reduced to the security of underlying HE scheme. A leak can occur only if DC is corrupted since the data is encrypted using the public key generated by DC. However, even in this case, DC will only obtain the noisy local statistics, not the raw data, and at the end of the protocol, DC has only control over the aggregated data while reconstructing the global model and it can not know which party contributed to the result. While the protocol is running, the view of all the other parties consists of homomorphically encrypted data. Therefore, if the given homomorphic encryption scheme is semantically secure, the parties can not distinguish the corresponding plaintexts. So, the computation is private even in the presence of an honest, but curious adversary model presented in [Gol09]. Therefore, data privacy is preserved.

On the other hand, we both showed theoretically (Section 9.4.1) and experimentally (Fig. 9.6), a differentially private global model can be obtained through the locally applied DP. Therefore, it is not possible that an untrusted data collector can

infer information about the individuals. Furthermore, the collaboration comes with a price as the local parties used ϵ_i instead of ϵ. Therefore, the local privacy guarantee is decreased by α (i.e., ϵ_i is increased by α), even though the global model's guarantee is still the same, meaning that data privacy against an untrusted DC is still preserved and the local privacy guarantee is important only if the underlying SMC is bypassed. Finally, since we set α as the number of parties in the collaboration, each party should take this into consideration while deciding on the global privacy budget.

9.6 Discussion

The preceding analysis showed how to achieve secure multiparty computation and differential privacy in distributed settings focusing on linear regression on horizontally distributed data. That is, parties do not see each others' inputs and further can not infer individuals' data from the final constructed model. A limitation of our algorithm is that we assume parties do not collaborate to learn a target party's input. However, if the party that generates the key pair conspires with the parties that are neighbors of a target in the ring topology, the noisy local statistics (ξ, κ, δ) of the victim can be extracted. More generally, this is known as *active corruption*, where the data collector is an active attacker and has control over the other corrupted parties. Our protocol in Fig. 9.3 achieves only a collusion threshold of 1, but the distributed DP algorithm that we present here can easily be adapted to work with recent solutions in SMC such as [DPSZ12], which is secure in the presence of an active adversary corrupting up to $n-1$ of the n parties. To extend our work with these more secure SMC schemes, it suffices to use the noisy output of the functional

mechanism instead of using the local statistics directly as input to the underlying SMC algorithm.

In our evaluation, we used HElib, an implementation of the fully homomorphic operation, to compute generic results. It supports both addition and multiplication; however, while computing the linear regression coefficients, we only used the addition operation. The performance of secure computation can be improved by using other libraries such as Paillier cryptosystem [Pai99a], which is only additively homomorphic cryptosystem.

Finally, our algorithms can be easily extended to other algorithms such as logistic regression in a supervised classification setting. In logistic regression, each party independently computes a score vector u_i and information matrix $\mathcal{I}_i$. Instead of injecting noise to the local statistics as in linear regression, noise can be injected into u_i and $\mathcal{I}_i$ vectors. However, the optimization of objective function differs in logistic regression as it requires several iterations. Fortunately, there exist some techniques that let implementing the iterations for computing the secure multi-site logistic regression [EESA+12]. Combining this secure multi-site logistic regression algorithm with FM would solve this issue. We defer the detailed application of this method to future work.

9.7 Conclusion

In this chapter, we proposed a novel Secure Multiparty Distributed Differentially Private (SM-DDP) protocol to achieve private computations in a multiparty environment as an application in linear regression. Using homomorphic encryption and functional mechanism, we first presented a protocol to provide the guarantees of secure multiparty computation and differential privacy. Then, we built the algorithms

that would allow distributed parties to compute a global model while preserving the privacy of their data and individuals found in the dataset. Any statistical model function that can be independently calculated by sharing the local statistics of the parties can be computed through this protocol. Finally, we evaluated the performance of the proposed protocol on two datasets, namely, warfarin dose and budget predictions. Our findings show that a party can achieve individual-level privacy via our proposed protocol for distributed differential privacy, which is independently applied by each party in a distributed fashion. Moreover, the experiment results demonstrated that the proposed SM-DDP protocol is both feasible and scalable that is its computational overhead is minimal and overall computation time is sub-linear with the number of parties. Indeed, SM-DDP protocol provides security and privacy guarantees while being feasible and scalable.

CHAPTER 10

CONCLUSIONS AND FUTURE WORK

10.1 Conclusions

In this dissertation, we introduced several privacy-aware security solutions. We split them under three categories: 1) Alternative Complementary Authentication Methods 2) Smart Home User Privacy, and 3) Secure Data Exchange Methods. Among alternative authentication methods, in WACA, we used the motion sensors of wrist-worn wearables, such as to capture the typing behavior of computer users. Captured sensor data is used to design a CA system. This system has an advantage of capturing user behavior seamlessly, but also it may have more challenges as the smartwatches are still not mature enough to be used in a real-life application. Besides, we tested WACA against more powerful active attacks such as imitation and statistical attacks. Our experiments showed that the active attacker has the same success rate as the zero-effort attackers, which are used as a base for biometric-based studies. Moreover, the literature lacks a biometric-based continuous authentication protocol. In PACA, we designed a privacy-aware continuous authentication protocol using the noise-tolerant secure template matching method called $\mathsf{NTT\text{-}Sec\text{-}}\mathbb{R}$. This method allows template comparison of noisy biometric data by transforming the feature vectors in an irreversible way. This provides privacy guarantees without relying on any trusted party or any long-term keys. Furthermore, for the security and privacy analysis, we tested our protocol against eight different attacks, which are known in biometric-based authentication systems. In PINTA, we used a hybrid behavioral profile of the user as a second factor in the authentication system. Since these features may include sensitive information, we utilize Fuzzy Hashing and Homomorphic Encryption to provide the authentication in a privacy-preserving. Our

approach has only shown a slight reduction in the performance compared to the traditional MAF techniques.

For the smart home user privacy, in Peek-a-boo, we discovered a multi-stage privacy attack, wherein every stage attacker reveals information related to the devices and the user activities at a smart home using the machine learning-based methods. The advantage of this method can work even on encrypted traffic as only the meta-data of the network traffic is used. However, it has the limitation that the attacker should be within the radio frequency range to capture the pairwise network traffic.

Finally, we designed a policy-based privacy-aware secure data exchange approach. For this, we first investigated the state-of-the art HE schemes. Then, we defined a policy language called Curie Policy Language (CPL). CPL allows each party in a group to define their requirements on the data to be exchanged. Moreover, we introduced a method for achieving secure and differentially private computation at the same time in multiparty settings. Our method allows distributed parties to make computations while the parties learn nothing about each other's data, but the final result. For this, we combined Homomorphic Encryption and Differential Privacy. For homomorphic encryption, we used an FHE scheme [HS14b] and for the differential privacy, we utilized a method called Functional Mechanism, which allows the addition of noise on the local data directly.

10.2 Future Work

The studies in this dissertation aim to use the ubiquitous of IoT and smart devices for alternative complementary authentication systems offering a better security-usability trade-off than the existing systems while additionally protecting the privacy of the sensitive user information.

In WACA, we only utilized the typing behavior of the user, while it would be interesting to see the effect of covering more activities of the user. This may offer both better usability as it may decrease the false rejections of genuine users and also increase security by reducing the vulnerable window of time. Moreover, an alternative design can be a multimodal system utilizing both keystroke-dynamics captured by keyboards and the keystroke-dynamics captured from motion sensors. Such an approach would combine the best of traditional and modern approaches. In PACA, we designed a privacy-aware continuous authentication protocol and also proposed an actual system using the WACA. However, as a future work, our protocol with other biometric-based authentication methods where the feature vectors are fixed-size real-valued vectors can be tested. In PINTA, we used a long period of data (30 minutes), while future works would propose methods for decreasing the data collection time to get the same or better performance.

In Peek-a-boo, we designed a multi-stage privacy attack on smart home users. However, the same approach can also be used for the authentication of smart home users. Such an approach would provide an unobtrusive way of authenticating the smart home users, maybe even in a continuous way. However, the heterogeneity of smart home devices and variability of the authentication accuracy would be the possible challenges that needs to be handled.

Finally, in Curie, an interesting future work would be to investigate the use of Curie in other collaborative learning settings exploring different statistics for data-dependent conditionals and explore its performance trade-offs by integrating it into other off-the-shelf secure computation frameworks. And, a future work of our differentially private secure computation study can extend the algorithms outside the linear models and investigate the accuracy and performance trade-offs of other algorithms. Similarly, it would also be interesting to compare the performance of

Laplacian mechanism used in FM with other DP mechanisms such as Exponential Mechanism [MT07] and Sample-and-aggregate [NRS07].

REFERENCES

[AAAU16] Abbas Acar, Hidayet Aksu, Kemal Akkaya, and A. Selcuk Uluagac. Waca: Wearable-assisted continuous authentication framework with motion sensors (poster). In *USENIX Security '16 Poster Session*. `https://www.usenix.org/conference/usenixsecurity16/poster-session`, 2016.

[AAUA18] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya. Waca: Wearable-assisted continuous authentication. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 264–269, May 2018.

[AAUA20] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya. A usable and robust continuous authentication framework using wearables. *IEEE Transactions on Mobile Computing*, pages 1–1, 2020.

[AAUC18] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, 51(4):79:1–79:35, July 2018.

[AB12] NETRESEC AB. Netresec splitcap - a fast pcap file splitter. `http://www.netresec.com/?page=SplitCap`, 2012. [Online; accessed 2020-3-20].

[ABC+15] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A Reuter, and Martin Strand. A guide to fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2015:1192, 2015.

[ABD16] Martin Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched ntru assumptions. In *Annual Cryptology Conference*, pages 153–178. Springer, 2016.

[ACA+17] A. Acar, Z. B. Celik, H. Aksu, A. S. Uluagac, and P. McDaniel. Achieving secure and differentially private computations in multiparty settings. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*, pages 49–59, Aug 2017.

[AFA+18] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and A Selcuk Uluagac. Peek-a-boo: I see your smart

home activities, even encrypted! *arXiv preprint arXiv:1808.02741*, abs/1808.02741, 2018.

[AFFP11] Martin Albrecht, Pooya Farshim, Jean-Charles Faugere, and Ludovic Perret. Polly cracker, revisited. *Advances in Cryptology–ASIACRYPT 2011*, pages 179–196, 2011.

[AGKL19] Kevin Atighehchi, Loubna Ghammam, Koray Karabina, and Patrick Lacharme. A cryptanalysis of two cancelable biometric schemes based on index-of-max hashing. *arXiv preprint arXiv:1910.01389*, 2019.

[AGS14] Nitesh Aggarwal, Cp Gupta, and Iti Sharma. Fully homomorphic symmetric scheme without bootstrapping. In *Cloud Computing and Internet of Things (CCIOT), 2014 International Conference on*, pages 14–17. IEEE, 2014.

[AHM+15] Giuseppe Ateniese, Briland Hitaj, Luigi Vincenzo Mancini, Nino Vincenzo Verde, and Antonio Villani. No place to hide that bytes won't reveal: Sniffing location-based encrypted traffic to track a user's position. In *International Conference on Network and System Security*, pages 46–59. Springer, 2015.

[AHPW15] Yoshinori Aono, Takuya Hayashi, Le Trieu Phong, and Lihua Wang. Fast and secure linear regression and biometric authentication with security update. *IACR Cryptology ePrint Archive*, 2015, 2015.

[AHR+19] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. Keeping the smart home private with smart(er) iot traffic shaping. In *Proceedings on Privacy Enhancing Technologies*, pages 128–148, 2019.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108. ACM, 1996.

[Aki09] Mufutau Akinwande. Advances in homomorphic cryptosystems. *Journal of Universal Computer Science*, 15(3):506–522, 2009.

[AKK19] Shoukat Ali, Koray Karabina, and Emrah Karagoz. Biometric data transformation for cryptographic domains and its application: poster. In *Proceedings of the 12th Conference on Security and Privacy in*

Wireless and Mobile Networks, WiSec 2019, Miami, Florida, USA, May 15-17, 2019., pages 304–305, 2019.

[AKP13] Frederik Armknecht, Stefan Katzenbeisser, and Andreas Peter. Group homomorphic encryption: characterizations, impossibility results, and applications. *Designs, codes and cryptography*, 67(2):209–232, 2013.

[Alb17] Martin R Albrecht. On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 103–129. Springer, 2017.

[ALB+19] Abbas Acar, Wenyi Liu, Raheem Bayeh, Kemal Akkaya, and Arif Selcuk Uluagac. A privacy-preserving multifactor authentication system. *Wiley Security and Privacy*, 2(5):e88, 2019.

[ALUK19] Abbas Acar, Long Lu, A Selcuk Uluagac, and Engin Kirda. An analysis of malware trends in enterprise networks. In *International Conference on Information Security*, pages 360–380. Springer, Cham, 2019.

[AM14] Aysajan Abidin and Aikaterini Mitrokotsa. Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-lwe. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 60–65. IEEE, 2014.

[Ame17] American Recovery and Reinvestment Act of 2009. `https://en.wikipedia.org/wiki/American_Recovery_and_Reinvestment_Act_of_2009`, 2017. [Online; accessed 01-June-2018].

[AMFF+13] Carlos Aguilar-Melchor, Simon Fau, Caroline Fontaine, Guy Gogniat, and Renaud Sirdey. Recent advances in homomorphic encryption: A possible future for signal processing in the encrypted domain. *Signal Processing Magazine, IEEE*, 30(2):108–117, 2013.

[AMN18] Dariush Abbasinezhad-Mood and Morteza Nikooghadam. Design and extensive hardware performance analysis of an efficient pairwise key generation scheme for smart grid. *International Journal of Communication Systems*, 31(5):e3507, 2018. e3507 IJCS-17-0460.R2.

[AMSS08] Antonia Azzini, Stefania Marrara, Roberto Sassi, and Fabio Scotti. A fuzzy approach to multimodal biometric continuous authentication. *Fuzzy Optimization and Decision Making*, 7(3):243–256, 2008.

[Ana17] IoT Analytics. State of the smart home market. `https://iot-analytics.com/wp/wp-content/uploads/2017/12/StateofSmartHomeMarket2017-vf.pdf`, 2017.

[APP+18] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

[ARF17] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*, 2017.

[ARS+15] Martin R Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for mpc and fhe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 430–454. Springer, 2015.

[ARS+17] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *arXiv preprint arXiv:1708.05044*, 2017.

[AS14] S Sobitha Ahila and KL Shunmuganathan. State of art in homomorphic encryption schemes. *International Journal of Engineering Research and Applications*, 4(2):37–43, 2014.

[ASBS12] Adam J Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M Smith. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 41–50. ACM, 2012.

[ASL15] Margit Antal, László Zsolt Szabó, and Izabella László. Keystroke dynamics on android platform. *Procedia Technology*, 19:820–826, 2015.

[ASNM05] Russell Ang, Rei Safavi-Naini, and Luke McAven. Cancelable key-based fingerprint templates. In *Australasian conference on information security and privacy*, pages 242–252. Springer, 2005.

[ASP13] Jacob Alperin-Sheriff and Chris Peikert. Practical bootstrapping in quasilinear time. In *Advances in Cryptology–CRYPTO 2013*, pages 1–20. Springer, 2013.

[ASP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In *Advances in Cryptology–CRYPTO 2014*, pages 297–314. Springer, 2014.

[AUT12] RSA SecurID WORLD'S LEADING TWO-FACTOR AUTHENTICATION. Identity and access management. `http://www.emc.com/security/rsa-securid.htm`, 2012. [Online; accessed 2020-3-20].

[BA09] Marina Blanton and Mehrdad Aliasgari. Secure computation of biometric matching. *Department of Computer Science and Engineering, University of Notre Dame, Tech. Rep*, 3:2009, 2009.

[BA11] M. Blanton and M. Aliasgari. On the (non-)reusability of fuzzy sketches and extractors and security in the computational setting. In *Proceedings of the International Conference on Security and Cryptography*, pages 68–77, July 2011.

[BBC+10] M. Barni, T. Bianchi, D. Catalano, M. Raimondo, R. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva. Privacy-preserving fingercode authentication. *ACM Workshop on Multimedia and Security*, pages 231–240, 2010.

[BBCdS08] Manuel Barbosa, Thierry Brouard, Stéphane Cauchie, and Simão Melo de Sousa. Secure biometric authentication with improved accuracy. In Yi Mu, Willy Susilo, and Jennifer Seberry, editors, *Information Security and Privacy*, pages 21–36, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[BBP+18] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. Iotsense: Behavioral fingerprinting of iot devices. *arXiv preprint arXiv:1804.03852*, 2018.

[BCD+09] Peter Bogetoft, Dan Lund Christensen, Ivan Damgård, Martin Geisler, Thomas Jakobsen, Mikkel Krøigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael Schwartzbach, and Tomas Toft. Secure multiparty computation goes live. In Roger Dingledine and Philippe Golle, editors, *Financial Cryp-*

tography and Data Security, pages 325–343, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[BCE+18] Attaullah Buriro, Bruno Crispo, Mojtaba Eskandri, Sandeep Gupta, Athar Mahboob, and Rutger Van Acker. S nap a uth: A gesture-based unobtrusive smartwatch user authentication scheme. In *International Workshop on Emerging Technologies for Authorization and Authentication*, pages 30–37. Springer, 2018.

[BCF+13] Arman Boehm, Dongqu Chen, Mario Frank, Ling Huang, Cynthia Kuo, Tihomir Lolic, Ivan Martinovic, and Dawn Song. Safe: Secure authentication with face and eyes. In *Privacy and Security in Mobile Systems (PRISMS), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[BCH+09] Ulrich Bayer, Paolo Milani Comparetti, Clemens Hlauschek, Christopher Kruegel, and Engin Kirda. Scalable, behavior-based malware clustering. *Network and Distributed System Security Symposium (NDSS)*, 9:8–11, 2009.

[BCI+07] J. Bringer, H. Chabanne, M. Izabachène, D. Pointcheval, Q. Tang, and S. Zimmer. An Application of the Goldwasser-Micali cryptosystem to biometric authentication. *Information Security and Privacy, Lecture Notes in Computer Science*, 4586:96–106, 2007.

[BCK08] Julien Bringer, Hervé Chabanne, and Bruno Kindarji. The best of both worlds: Applying secure sketches to cancelable biometrics. *Science of Computer Programming*, 74(1):43 – 51, 2008. Special Issue on Security and Trust.

[BCK+15] Julien Bringer, Hervé Chabanne, Firas Kraïem, Roch Lescuyer, and Eduardo Soria-Vázquez. Some applications of verifiable computation to biometric verification. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*, pages 1–6. IEEE, 2015.

[BDCG12] Carlo Blundo, Emiliano De Cristofaro, and Paolo Gasti. Espresso: efficient privacy-preserving evaluation of sample set similarity. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 89–103. Springer, 2012.

[BDNP08] Assaf Ben-David, Noam Nisan, and Benny Pinkas. Fairplaymp: a system for secure multi-party computation. In *Computer and communications security*, 2008.

[BEM19] Simon Birnbach, Simon Eberz, and Ivan Martinovic. Peeves: Physical event verification in smart homes. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 1455–1467, New York, NY, USA, 2019. ACM.

[Ben87] Josh Daniel Cohen Benaloh. *Verifiable Secret-Ballot Elections*. PhD thesis, Yale University, USA, 1987. AAI8809191.

[Ben94] Josh Benaloh. Dense probabilistic encryption. In *Proceedings of the workshop on selected areas of cryptography*, pages 120–128, 1994.

[Beu14] KM Beunder. Design of continuous authentication using face recognition. In *20th Twente Student Conference on IT*, pages 1–8, 2014.

[BG04] Qian Bao and Ping Guo. Comparative studies on similarity measures for remote sensing image retrieval. In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 1112–1116, 2004.

[BG11] M. Blanton and P. Gasti. Secure and efficient protocols for iris and fingerprint identification. *ESORICS'11, European Symposium on Research in Computer Security*, pages 190–209, 2011.

[BGJT14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 1–16, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[BGK+15] David Guy Brizan, Adam Goodkind, Patrick Koch, Kiran Balagani, Vir V Phoha, and Andrew Rosenberg. Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics. *International Journal of Human-Computer Studies*, 82:57–68, 2015.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of cryptography*, pages 325–341. Springer, 2005.

[BGV11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277, 2011. `http://eprint.iacr.org/`.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory*, 6(3):13:1–13:36, July 2014.

[BHOS12] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pages 553–567, 2012.

[BHVOS12] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 553–567, 2012.

[BHvOS15] Joseph Bonneau, Cormac Herley, Paul C van Oorschot, and Frank Stajano. Passwords and the evolution of imperfect authentication. *Communications of the ACM*, 58(7):78–87, 2015.

[BKLS18] D. Bogdanov, L. Kamm, S. Laur, and V. Sokk. Rmind: A tool for cryptographically secure statistical analysis. *IEEE Transactions on Dependable and Secure Computing*, 15(3):481–495, May 2018.

[BLLN13] Joppe W Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In *Cryptography and Coding*, pages 45–64. Springer, 2013.

[Bor] Mark Borgerding. Kiss fft - a mixed-radix fast fourier transform based up on the principle, "keep it simple, stupid.". `https://github.com/mborgerding/kissfft`. [Online; accessed 2019-07-01].

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, page 82–91, New York, NY, USA, 2004. Association for Computing Machinery.

[BPTG15] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine Learning Classification over Encrypted Data. In *Network*

and Distributed System Security Symposium (NDSS), pages 1–14, 2015.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in cryptology–crypto 2012*, pages 868–886. Springer, 2012.

[BSAU18] Leonardo Babun, Amit Kumar Sikder, Abbas Acar, and A Selcuk Uluagac. Iotdots: A digital forensics framework for smart environments. *arXiv preprint arXiv:1809.00745*, 2018.

[BSAU19] Leonardo Babun, Amit K Sikder, Abbas Acar, and A Selcuk Uluagac. A digital forensics framework for smart settings: poster. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 332–333, 2019.

[BSPvdM12] Rafael Ramos Regis Barbosa, Ramin Sadre, Aiko Pras, and Remco van de Meent. Simpleweb/university of twente traffic traces data repository. `http://traces.simpleweb.org`, 2012. [Online; accessed 2020-3-20].

[BSSB06] Abhilasha Bhargav-Spantzel, Anna Squicciarini, and Elisa Bertino. Privacy preserving multi-factor authentication with biometrics. *Proceedings of the second ACM workshop on Digital identity management*, pages 63–72, 2006.

[BST14] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473, Oct 2014.

[BTB17] Trevor Bihl, Michael Temple, and Kenneth Bauer. An optimization framework for generalized relevance learning vector quantization with application to z-wave device fingerprinting. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, pages 1–9, 2017.

[BTW12] Dan Bogdanov, Riivo Talviste, and Jan Willemson. Deploying secure multi-party computation for financial data analysis. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security*, pages 57–64, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 505–524, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[BV14a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[BV14b] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based fhe as secure as pke. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 1–12. ACM, 2014.

[BVACM18] Attaullah Buriro, Rutger Van Acker, Bruno Crispo, and Athar Mahboob. Airsign: A gesture-based smartwatch user authentication. In *2018 International Carnahan Conference on Security Technology (ICCST)*, pages 1–5. IEEE, 2018.

[BW12] Salil P Banerjee and Damon L Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[CA13] Chris Clifton and Balamurugan Anandan. Challenges and opportunities for security with differential privacy. In Aditya Bagchi and Indrakshi Ray, editors, *Information Systems Security*, pages 1–13, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[CAA+19] Z. Berkay Celik, Abbas Acar, Hidayet Aksu, Ryan Sheatsley, Patrick McDaniel, and A. Selcuk Uluagac. Curie: Policy-based secure data exchange. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, CODASPY '19, page 121–132, New York, NY, USA, 2019. Association for Computing Machinery.

[Car03] Cassandra M Carrillo. Continuous biometric authentication for authorized aircraft personnel: A proposed design. Master's thesis, NAVAL POSTGRADUATE SCHOOL Monterey, California, June 2003. Master's thesis.

[CBH14] Liquan Chen, Hongmei Ben, and Jie Huang. An encryption depth optimization scheme for fully homomorphic encryption. In *Identification, Information and Knowledge in the Internet of Things (IIKI), 2014 International Conference on*, pages 137–141. IEEE, 2014.

[CBN18] Maximilian Christ, Nils Braun, and Julius Neuffer. tsfresh - automatic extraction of relevant features from time series. `https://github.com/blue-yonder/tsfresh`, 2018. [Online; accessed 2020-3-20].

[CBS+18] Z. Berkay Celik, Leonardo Babun, Amit K. Sikder, Hidayet Aksu, Gang Tan, Patrick McDaniel, and A. Selcuk Uluagac. Sensitive information tracking in commodity iot. In *USENIX Security Symposium*, Baltimore, MD, August 2018.

[CBW07] Su-Jeong Choi, Simon R Blackburn, and Peter R Wild. Cryptanalysis of a homomorphic public-key cryptosystem over a finite group. *Journal of Mathematical Cryptology*, 1(4):351, 2007.

[CC12] Liang Cai and Hao Chen. On the practicality of motion based keystroke inference attack. In Stefan Katzenbeisser, Edgar Weippl, L. Jean Camp, Melanie Volkamer, Mike Reiter, and Xinwen Zhang, editors, *Trust and Trustworthy Computing*, pages 273–290, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[CCK+13] Jung Hee Cheon, Jean-Sébastien Coron, Jinsu Kim, Moon Sung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2013*, pages 315–335. Springer, 2013.

[CDK+12] Alexei Czeskis, Michael Dietz, Tadayoshi Kohno, Dan Wallach, and Dirk Balfanz. Strengthening user authentication through opportunistic cryptographic identity assertions. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 404–414, New York, NY, USA, 2012. Association for Computing Machinery.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 3–33. Springer, 2016.

[CGGI17] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Tfhe: Fast fully homomorphic encryption library over the torus. `https://github.com/tfhe/tfhe`, 2017. [Online; accessed September-2017].

[CGRS14] David Bruce Cousins, John Golusky, Kurt Rohloff, and Daniel Sumorok. An fpga co-processor implementation of homomorphic encryption. In *High Performance Extreme Computing Conference (HPEC), 2014 IEEE*, pages 1–6. IEEE, 2014.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. *Transactions on Emerging Telecommunications Technologies*, 8(5):481–490, 1997.

[CHLR16] Jung Hee Cheon, Hyunsook Hong, Moon Sung Lee, and Hansol Ryu. The polynomial approximate common divisor problem and its application to the fully homomorphic encryption. *Information Sciences*, 326:41–58, 2016.

[CKKS16] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers (heann). `https://github.com/kimandrik/HEAAN`, 2016. [Online; accessed September-2015].

[CKN06] Jung Hee Cheon, Woo-Hwan Kim, and Hyun Soo Nam. Known-plaintext cryptanalysis of the domingo-ferrer algebraic privacy homomorphism scheme. *Information Processing Letters*, 97(3):118–123, 2006.

[CLBR16] Bogdan Copos, Karl Levitt, Matt Bishop, and Jeff Rowe. Is anybody home? inferring activity from smart home network traffic. In *Security and Privacy Workshops (SPW), 2016 IEEE*, pages 245–251. IEEE, 2016.

[CLM17] Z. B. Celik, D. Lopez-Paz, and P. McDaniel. Patient-driven privacy control through generalized distillation. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*, pages 1–12, Aug 2017.

[CLO+13] Ashish Choudhury, Jake Loftus, Emmanuela Orsini, Arpita Patra, and Nigel P Smart. Between a rock and a hard place: Interpolating between mpc and fhe. In *Advances in Cryptology-ASIACRYPT 2013*, pages 221–240. Springer, 2013.

[CLP17] Hao Chen, Kim Laine, and Rachel Player. Simple encrypted arithmetic library. `https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/sealmanual_v2.2.pdf`, 2017. [Online; accessed September-2015].

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Batch fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2013/036, 2013. http://eprint.iacr.org/.

[CLT14] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Scale-invariant fully homomorphic encryption over the integers. In *Public-Key Cryptography–PKC 2014*, pages 311–328. Springer, 2014.

[Clu07] Chaos Computer Club. Fingerprint recognition at the supermarket as insecure as biometrics in passports. https://ccc.de/en/updates/2007/umsonst-im-supermarkt, 2007. [Online; accessed 20-March-2020].

[Clu13] Chaos Computer Club. Chaos computer club breaks apple touchid. http://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid/, 2013. [Online; accessed 20-March-2020].

[CM14] Michael Clear and Ciarán McGoldrick. Bootstrappable identity-based fully homomorphic encryption. In *Cryptology and Network Security*, pages 1–19. Springer, 2014.

[CM15] Michael Clear and Ciarán McGoldrick. Multi-identity and multi-key leveled fhe from learning with errors. In *Annual Cryptology Conference*, pages 630–656. Springer, 2015.

[CM16] Michael Clear and Ciarán McGoldrick. Attribute-based fully homomorphic encryption with a bounded number of inputs. In *International Conference on Cryptology in Africa*, pages 307–324. Springer, 2016.

[CMNT11a] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology–CRYPTO 2011*, pages 487–504. Springer, 2011.

[CMNT11b] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In Phillip Rogaway, editor, *Advances in Cryptology CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2011.

[CMO+13] Xiaolin Cao, Ciara Moore, Máire O'Neill, Elizabeth O'Sullivan, and Neil Hanley. Accelerating fully homomorphic encryption over the integers with super-size hardware multiplier and modular reduction. *IACR Cryptology ePrint Archive*, 2013:616, 2013.

[CMO+14] Xiaolin Cao, Ciara Moore, Máire O'Neill, Neil Hanley, and Elizabeth O'Sullivan. High-speed fully homomorphic encryption over the integers. In *Financial Cryptography and Data Security*, pages 169–180. Springer, 2014.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 2011.

[CMSV16] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde. Analyzing android encrypted network traffic to identify user actions. *IEEE Transactions on Information Forensics and Security*, 11(1):114–125, Jan 2016.

[CMTB15] Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing secure two-party computation as a black box. In *Cryptology and Network Security*, pages 214–222. Springer, 2015.

[CMTB16] Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure outsourced garbled circuit evaluation for mobile devices. *Journal of Computer Security*, 24(Preprint):137–180, 2016.

[CMV+15] Donald Donglong Chen, Nele Mentens, Frederik Vercauteren, Sujoy Sinha Roy, Ray CC Cheung, Derek Pao, and Ingrid Verbauwhede. High-speed polynomial multiplication architecture for ring-lwe and she cryptosystems. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 62(1):157–166, 2015.

[CN02] Mark D Corner and Brian D Noble. Zero-interaction authentication. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 1–11. ACM, 2002.

[CN12] Yuanmi Chen and Phong Q Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In *Advances in Cryptology–EUROCRYPT 2012*, pages 502–519. Springer, 2012.

[CNT12] Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2012*, pages 446–464. Springer, 2012.

[Coc20] Tom Cocagne. Minimal c implementation of the secure remote password protocol (version 6a). `https://github.com/cocagne/csrp`, 2020. [Online; accessed 20-March-2020].

[CRPS12] David Bruce Cousins, Kathrin Rohloff, Chris Peikert, and Richard Schantz. An update on sipher (scalable implementation of primitives for homomorphic encryption)—fpga implementation using simulink. In *High Performance Extreme Computing (HPEC), 2012 IEEE Conference on*, pages 1–5. IEEE, 2012.

[CWZX14] Zhigang Chen, Jian Wang, ZengNian Zhang, and Song Xinxia. A fully homomorphic encryption scheme with better key size. *Communications, China*, 11(9):82–92, 2014.

[CZJJ12] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 605–616. ACM, 2012.

[CZY+15] Jun Chen, Guang Zhu, Jin Yang, Qingshen Jing, Peng Bai, Weiqing Yang, Xuewei Qi, Yuanjie Su, and Zhong Lin Wang. Personalized keystroke dynamics for self-powered human–machine interfacing. *ACS nano*, 9(1):105–116, 2015.

[Dan15] Fida K. Dankar. Privacy preserving linear regression on distributed databases. *Trans. Data Privacy*, 8(1):3–28, December 2015.

[DCGT12] Emiliano De Cristofaro, Paolo Gasti, and Gene Tsudik. Fast and private computation of cardinality of set intersection and union. In Josef Pieprzyk, Ahmad-Reza Sadeghi, and Mark Manulis, editors, *Cryptology and Network Security*, pages 218–231, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[DCM+12] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou,

and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1223–1231. Curran Associates, Inc., 2012.

[DCRS12] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 332–346. IEEE, 2012.

[DDS14] Wei Dai, Yarkın Doröz, and Berk Sunar. Accelerating ntru based homomorphic encryption using gpus. In *High Performance Extreme Computing Conference (HPEC), 2014 IEEE*, pages 1–6. IEEE, 2014.

[DDS15] Wei Dai, Yarkın Doröz, and Berk Sunar. Accelerating swhe based pirs using gpus. In *International Conference on Financial Cryptography and Data Security*, pages 160–171. Springer, 2015.

[DDS17] Wei Dai, Yarkın Doröz, and Berk Sunar. cuhe: Homomorphic and fast. `https://github.com/vernamlab/cuHE`, 2017. [Online; accessed September-2015].

[DF02] Josep Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism*. In *Information security*, pages 471–483. Springer, 2002.

[DH76] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[DHC04] Wenliang Du, Yunghsiang S Han, and Shigang Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *SIAM International Conference on Data Mining*, 2004.

[DHS16] Yarkın Doröz, Yin Hu, and Berk Sunar. Homomorphic aes evaluation using the modified ltv scheme. *Designs, Codes and Cryptography*, 80(2):333–358, 2016.

[Dic16] Ben Dickson. The many ways your password can be stolen or bypassed. `https://bdtechtalks.com/2016/02/12/the-many-ways-your-password-can-be-stolen-or-bypassed/`, February 2016. [Online; accessed 20-March-2020].

[Dis17] Steve Dispensa. Enhanced multi factor authentication, September 12 2017. US Patent 9,762,576.

[DJ01] Ivan Damgård and Mads Jurik. A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system. In *Public Key Cryptography*, pages 119–136. Springer, 2001.

[DJ17] Asish Kumar Dalai and Sanjay Kumar Jena. Wdtf: A technique for wireless device type fingerprinting. *Wireless Personal Communications*, 97(2):1911–1928, 2017.

[DJW13] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 429–438, Oct 2013.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[dL17] Christian de Looper. The 12 best smart home devices you need to live like the jetsons. `http://www.businessinsider.com/best-smart-home`, 2017. [Online; accessed 2020-1-20].

[DLT+19] Shuaike Dong, Zhou Li, Di Tang, Jiongyi Chen, Menghan Sun, and Kehuan Zhang. Your smart home can't keep a secret: Towards automated fingerprinting of iot traffic with neural networks. *arXiv preprint arXiv:1909.00104*, 2019.

[DM09] Nguyen Minh Duc and Bui Quang Minh. Your face is not your password face authentication bypassing lenovo–asus–toshiba. `https://blackhat.com/presentations/bh-dc-09/Nguyen/BlackHat-DC-09-Nguyen-Face-not-your-password.pdf`, 2009. [Online; accessed 20-March-2020].

[DM14] Léo Ducas and Daniele Micciancio. A fully homomorphic encryption library. `https://github.com/lducas/FHEW`, 2014. [Online; accessed December-2015].

[DM15] Léo Ducas and Daniele Micciancio. Fhew: Bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology–EUROCRYPT 2015*, pages 617–640. Springer, 2015.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[DÖS13] Yarkın Doröz, Erdinç Öztürk, and Berk Sunar. Evaluating the hardware performance of a million-bit multiplier. In *Digital System Design (DSD), 2013 Euromicro Conference on*, pages 955–962. IEEE, 2013.

[DÖS15] Yarkın Doröz, Erdinç Öztürk, and Berk Sunar. Accelerating fully homomorphic encryption in hardware. *IEEE Transactions on Computers*, 64(6):1509–1521, 2015.

[DÖSS15] Yarkın Doröz, Erdinç Öztürk, Erkay Savaş, and Berk Sunar. Accelerating ltv based homomorphic encryption in reconfigurable hardware. In *Cryptographic Hardware and Embedded Systems–CHES 2015*, pages 185–204. Springer, 2015.

[DPCM16] Aveek K Das, Parth H Pathak, Chen-Nee Chuah, and Prasant Mohapatra. Uncovering privacy leakage in ble network traffic of wearable fitness trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 99–104. ACM, 2016.

[DPR16] Ivan Damgård, Antigoni Polychroniadou, and Vanishree Rao. Adaptively secure multi-party computation from lwe (via equivocal fhe). In *Public-Key Cryptography–PKC 2016*, pages 208–233. Springer, 2016.

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *Advances in Cryptology–CRYPTO 2012*, pages 643–662. Springer, 2012.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[DRS04] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Advances*

in Cryptology - Eurocrypt 2004, Lecture Notes in Computer Science, 3027:523–540, 2004. Updated by Y. Dodis, L. Reyzin, and A. Smith in 2008.

[DS16] Yarkin Doröz and Berk Sunar. Flattening ntru for evaluation key free homomorphic encryption. *IACR Cryptology ePrint Archive*, 2016:315, 2016.

[DSES14] Yarkın Doröz, Aria Shahverdi, Thomas Eisenbarth, and Berk Sunar. Toward practical homomorphic evaluation of block ciphers using prince. In *Financial Cryptography and Data Security*, pages 208–220. Springer, 2014.

[dT12] Telecommunication Networks Group Politecnico di Torino. Tstat - tcp statistic and analysis tool. `http://tstat.tlc.polito.it/index.shtml`, 2012. [Online; accessed 2020-3-20].

[Dwo08] Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[DZC+16] L. Duan, Y. Zhang, S. Chen, S. Zhao, S. Wang, D. Liu, R. P. Liu, B. Cheng, and J. Chen. Automated policy combination for secure data sharing in cross-organizational collaborations. *IEEE Access*, 4:3454–3468, 2016.

[EESA+12] Khaled El Emam, Saeed Samet, Luk Arbuckle, Robyn Tamblyn, Craig Earle, and Murat Kantarcioglu. A secure distributed logistic regression protocol for the detection of rare adverse drug events. *Journal of the American Medical Informatics Association*, 20(3):453–461, 08 2012.

[EHKM11] David Evans, Yan Huang, Jonathan Katz, and Lior Malka. Efficient privacy-preserving biometric identification. In *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, 2011.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in cryptology*, pages 10–18. Springer, 1985.

[ERLM15] Simon Eberz, Kasper Bonne Rasmussen, Vincent Lenders, and Ivan Martinovic. Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics. In *The Network and Distributed System Security Symposium (NDSS)*, pages 1–13, 2015.

[ERLM17] Simon Eberz, Kasper B. Rasmussen, Vincent Lenders, and Ivan Martinovic. Evaluating behavioral biometrics for continuous authentication: Challenges and metrics. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, page 386–399, New York, NY, USA, 2017. Association for Computing Machinery.

[F+14] Matthew Fredrikson et al. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *The 23rd USENIX Security Symposium*, pages 1–17, 2014.

[FA17] U.S. Food and Drug Administration. Medication guide, Caumadin (warfarin sodium). http://www.fda.gov, 2017. [Online; accessed 01-June-2018].

[FAD06] Feng Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, Sep. 2006.

[FBM+13] Mario Frank, Ralf Biedert, Eugene Ma, Ivan Martinovic, and Dawn Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE transactions on information forensics and security*, 8(1):136–148, 2013.

[FDCA11] Michael Fairhurst and Márjory Da Costa-Abreu. Using keystroke dynamics for gender identification in social network environment. *4th International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011)*, pages 1–9, 2011.

[FDCB15] Julien Freudiger, Emiliano De Cristofaro, and Alejandro E. Brito. Controlled data sharing for collaborative predictive blacklisting. In Magnus Almgren, Vincenzo Gulisano, and Federico Maggi, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 327–349, Cham, 2015. Springer International Publishing.

[Fea16a] Kassem Fawaz and et al. Protecting Privacy of BLE Device Users. In *the Proceedings of the 25th USENIX Security Symposium*, pages 1–18, 2016.

[Fea16b] David Formby and et al. Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems. In *The Network and Distributed System Security Symposium (NDSS)*, pages 1–17, 2016.

[FFS17] Huan Feng, Kassem Fawaz, and Kang G Shin. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 343–355. ACM, 2017.

[FG07] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007:15, 2007.

[FJ05] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[FK94] Michael Fellows and Neal Koblitz. Combinatorial cryptosystems galore! *Contemporary Mathematics*, 168:51–51, 1994.

[FLE14] Michael Fairhurst, Cheng Li, and Meryem Erbilek. Exploiting biometric measurements for prediction of emotional state: A preliminary study for healthcare applications using keystroke analysis. In *Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings, 2014 IEEE Workshop on*, pages 74–79. IEEE, 2014.

[FLY14] Y. Feng, M-H. Lim, and P. Yuen. Masquerade attack on transform-based binary-template protection based on perceptron learning. *Pattern Recognition*, 47:3019–3033, 2014.

[fMS17] NATO Standard Allied Joint Doctrine for Medical Support. `http://www.nato.int`, 2017. [Online; accessed 01-June-2018].

[FV12a] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[FV12b] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

[Gal02] Steven D Galbraith. Elliptic curve paillier schemes. *Journal of Cryptology*, 15(2):129–138, 2002.

[GD98] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14):2627–2636, 1998.

[Gen09] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[Gen10] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *Advances in Cryptology–CRYPTO 2010*, pages 116–137. Springer, 2010.

[Gen14] Craig Gentry. Computing on the edge of chaos: Structure and randomness in encrypted computation. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 21, page 106, 2014.

[GGB13] Sathya Govindarajan, Paolo Gasti, and Kiran S Balagani. Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8. IEEE, 2013.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology—CRYPTO'97*, pages 112–131. Springer, 1997.

[GGM16] Steven D Galbraith, Shishay W Gebregiyorgis, and Sean Murphy. Algorithms for the approximate common divisor problem. *LMS Journal of Computation and Mathematics*, 19(A):58–72, 2016.

[GH11] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology–EUROCRYPT 2011*, pages 129–148. Springer, 2011.

[GHPS12] Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P Smart. Ring switching in bgv-style homomorphic encryption. In *Security and Cryptography for Networks*, pages 19–37. Springer, 2012.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P Smart. Homomorphic evaluation of the aes circuit. In *Advances in Cryptology–CRYPTO 2012*, pages 850–867. Springer, 2012.

[GHV10] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. A simple bgn-type cryptosystem from lwe. In *Advances in Cryptology–EUROCRYPT 2010*, pages 506–522. Springer, 2010.

[Gjø04] Kristian Gjøsteen. *Subgroup membership problems and public key cryptosystems*. PhD thesis, Fakultet for informasjonsteknologi, matematikk og elektroteknikk, 2004.

[GKS08] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 265–273, New York, NY, USA, 2008. Association for Computing Machinery.

[GLN13] Thore Graepel, Kristin Lauter, and Michael Naehrig. Ml confidential: Machine learning on encrypted data. In Taekyoung Kwon, Mun-Kyu Lee, and Daesung Kwon, editors, *Information Security and Cryptology – ICISC 2012*, pages 1–21, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377. ACM, 1982.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GOC12] Cristiano Giuffrida, Stefano Ortolani, and Bruno Crispo. Memoirs of a browser: a cross-browser detection model for privacy-breaching extensions. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 10–11, 2012.

[Gol09] Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge university press, 2009.

[Goo12] Google. About two-step verification. `https://www.google.com/landing/2step/`, 2012. [Online; accessed 20-March-2020].

[Goo16] Google. Google eyes behavioural solution for continuous authentication. `http://www.planetbiometrics.com/article-details/i/4512/`, May 2016. [Online; accessed 20-March-2020].

[GP06] Dima Grigoriev and Ilia Ponomarenko. Homomorphic public-key cryptosystems and encrypting boolean circuits. *Applicable Algebra in Engineering, Communication and Computing*, 17(3-4):239–255, 2006.

[GPGMP16] Roberto Garrido-Pelaz, Lorena González-Manzano, and Sergio Pastrana. Shall we collaborate? a model to analyse the benefits of information sharing. In *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, WISCS '16, page 15–24, New York, NY, USA, 2016. Association for Computing Machinery.

[GR12] Romain Giot and Christophe Rosenberger. A new soft biometric approach for keystroke dynamics based on gender recognition. *International Journal of Information Technology and Management*, 11(1-2):35–49, 2012.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*, pages 75–92. Springer, 2013.

[GU13] E. Grosse and M. Upadhyay. Authentication at scale. *IEEE Security Privacy*, 11(1):15–22, Jan 2013.

[GXS13] Slawomir Goryczka, Li Xiong, and Vaidy Sunderam. Secure multiparty aggregation with differential privacy: A comparative study. In

Proceedings of the Joint EDBT/ICDT 2013 Workshops, EDBT '13, page 155–163, New York, NY, USA, 2013. Association for Computing Machinery.

[H+11] Yan Huang et al. Faster Secure Two-Party Computation Using Garbled Circuits. In *USENIX Security Symposium*, pages 1–16, 2011.

[HCS+18] Jun Han, Albert Jin Chung, Manal Kumar Sinha, Madhumitha Harishankar, Shijia Pan, Hae Young Noh, Pei Zhang, and Patrick Tague. Do you feel what i hear? enabling autonomous iot device pairing using different sensor types. In *Do You Feel What I Hear? Enabling Autonomous IoT Device Pairing using Different Sensor Types*, page 0. IEEE, 2018.

[Hea17] Health Information Technology for Economic and Clinical Health Act. `https://en.wikipedia.org`, 2017. [Online; accessed 01-June-2018].

[HFN11] Rob Hall, Stephen E Fienberg, and Yuval Nardi. Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27:669–691, 2011.

[HJP13] W. Hart, F. Johansson, and S. Pancratz. FLINT: Fast Library for Number Theory. Version 2.4.0, `http://flintlib.org`, 2013. [Online; accessed 2020-3-20].

[HKoS+10] Wilko Henecka, Stefan K ögl, Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Tasty: Tool for automating secure two-party computations. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, page 451–462, New York, NY, USA, 2010. Association for Computing Machinery.

[HP14] Darko Hrestak and Stjepan Picek. Homomorphic encryption in the cloud. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pages 1400–1404. IEEE, 2014.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *Algorithmic number theory*, pages 267–288. Springer, 1998.

[HPSS08] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[HR11] Shai Halevi and Nalini K. Ratha. Public challenges for fully-homomorphic encryption. `http://researcher.watson.ibm.com/researcher/view_group.php?id=1548`, 2011. [Online; accessed March-2016].

[HS13] Shai Halevi and Victor Shoup. Design and implementation of a homomorphic-encryption library. *IBM Research (Manuscript)*, pages 1–46, 2013.

[HS14a] Shai Halevi and Victor Shoup. Algorithms in helib. In *Advances in Cryptology–CRYPTO 2014*, pages 554–571. Springer, 2014.

[HS14b] Shai Halevi and Victor Shoup. An implementation of homomorphic encryption. `https://github.com/shaih/HElib`, 2014. [Online; accessed 01-January-2017].

[HS15] Shai Halevi and Victor Shoup. Bootstrapping for helib. In *Advances in Cryptology–EUROCRYPT 2015*, pages 641–670. Springer, 2015.

[HSU+16] O Huhta, P Shrestha, S Udar, M Juuti, N Saxena, and N Asokan. Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks. In *The Network and Distributed System Security Symposium (NDSS)*, pages 1–14, 2016.

[IET08] IETF. Transport layer security (tls). `http://datatracker.ietf.org/wg/tls/`, 2008. [Online; accessed 2020-3-20].

[iF96] Josep Domingo i Ferrer. A new privacy homomorphism and applications. *Information Processing Letters*, 60(5):277–282, 1996.

[II17] IPUMS-International. Harmonized international census data for social science and health research. `https://international.ipums.org/international/`, 2017. [Online; accessed March-2017].

[Int09] International Warfarin Pharmacogenetics Consortium. Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data. *The New England Journal of Medicine*, 360:1–12, 2009.

[Int18] Mordor Intelligence. Iot sensor market size - segmented by type (pressure sensor, temperature sensor, proximity sensor), end-user industry (healthcare, automotive, consumer electronics), and region - growth, trends, and forecast (2018 - 2023). `https://www.mordorintelligence.com/industry-reports/iot-sensor-market`, 2018.

[IP07] Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In *Theory of Cryptography*, pages 575–594. Springer, 2007.

[Jag12] Tibor Jager. *The Generic Composite Residuosity Problem*, pages 49–56. Vieweg+Teubner Verlag, Wiesbaden, 2012.

[JFF19] Pierre-Marie Junges, Jérôme François, and Olivier Festor. Passive inference of user actions through iot gateway encrypted traffic analysis. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 7–12. IEEE, 2019.

[JL10] ZQ John Lu. The elements of statistical learning: data mining, inference, and prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(3):693–694, 2010.

[JLG04] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, 37(11):2245 – 2255, 2004.

[JNN08] Anil K Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric template security. *EURASIP Journal on advances in signal processing*, 2008:113, 2008.

[JS06] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38:237–257, 2006.

[JT13] Prateek Jain and Abhradeep Thakurta. Differentially private learning with kernels. *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1–9, 2013.

[JW99] A. Juels and M. Wattenberg. A fuzzy commitment scheme. *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[JY11] Zach Jorgensen and Ting Yu. On mouse dynamics as a behavioral biometric for authentication. *Proc. of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 476–482, 2011.

[K+13] Stephen E Kimmel et al. A pharmacogenetic versus a Clinical Algorithm for Warfarin Dosing. *New England Journal of Medicine*, 369:1–11, 2013.

[Kas] Jacob Kastrenakes. Facebook stored hundreds of millions of passwords in plain text. `https://www.theverge.com/2019/3/21/18275837/facebook-plain-text-password-storage-hundreds-millions-users`. [Online; accessed 2020-1-20].

[KC16] Koray Karabina and Onur Canpolat. A new cryptographic primitive for noise tolerant template security. *Pattern Recognition Letters*, 80:70 – 75, 2016.

[KCB03] Manesh Kokare, BN Chatterji, and PK Biswas. Comparison of similarity metrics for texture image retrieval. In *TENCON 2003. Conference on convergent technologies for Asia-Pacific region*, volume 2, pages 571–575. IEEE, 2003.

[KCZ+06] Adams Kong, King-Hong Cheung, David Zhang, Mohamed Kamel, and Jane You. An analysis of biohashing and its variants. *Pattern recognition*, 39(7):1359–1368, 2006.

[KDPP16] Georgios Kambourakis, Dimitrios Damopoulos, Dimitrios Papamartzivanos, and Emmanouil Pavlidakis. Introducing touchstroke: keystroke-based authentication system for smartphones. *Security and Communication Networks*, 9(6):542–554, 2016.

[Kev90] S McCURLEY Kevin. The discrete logarithm problem. *Cryptology and computational number theory*, 42:49, 1990.

[KHK+16] E Kimura, K Hamada, R Kikuchi, K Chida, K Okamoto, S Manabe, T Kuroda, Y Matsumura, T Takeda, and N Mihara. Evaluation of secure computation in a distributed healthcare setting. *Studies in health technology and informatics*, 228:152–156, 2016.

[KJ06] Hang-Bong Kang and Myung-Ho Ju. Multi-modal feature integration for secure authentication. In *International Conference on Intelligent Computing*, pages 1191–1200. Springer, 2006.

[KL87] LF Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[KLSR05] Alan F Karr, Xiaodong Lin, Ashish P Sanil, and Jerome P Reiter. Secure regression on distributed databases. *Journal of Computational and Graphical Statistics*, 14(2):263–279, 2005.

[KLSR09] Alan F Karr, Xiaodong Lin, Ashish P Sanil, and Jerome P Reiter. Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, 25:125–138, 2009.

[KLYC13] Jinsu Kim, Moon Sung Lee, Aaram Yun, and Jung Hee Cheon. Crt-based fully homomorphic encryption over the integers. *IACR Cryptology ePrint Archive*, 2013:57, 2013.

[KM09a] Kevin S Killourhy and Roy A Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *Dependable Systems & Networks, 2009.*, pages 125–134. IEEE, 2009.

[KM09b] K.S. Killourhy and R.A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. *Dependable Systems Networks, 2009. DSN '09. IEEE/IFIP International Conference on*, pages 125–134, 2009.

[KM12] Kevin Killourhy and Roy Maxion. Keystroke dynamics: Benchmark data set. `http://www.cs.cmu.edu/~keystroke`, 2012. [Online; accessed 2020-3-20].

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *focs*, page 364. IEEE, 1997.

[Kor06] Jesse Kornblum. Identifying almost identical files using context triggered piecewise hashing. *Digital Investigation*, 3, Supplement(0):91 – 97, 2006. The Proc. of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[Kor18] Jesse Kornblum. ssdeep python wrapper. `https://github.com/DinoTools/python-ssdeep`, 2018. [Online; accessed 2020-3-20].

[KPDD09] Sanjay Kanade, Dijana Petrovska-Delacrétaz, and Bernadette Dorizzi. Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 120–127. IEEE, 2009.

[Kra05] Hugo Krawczyk. Hmqv: A high-performance secure diffie-hellman protocol. In *Annual International Cryptology Conference*, pages 546–566. Springer, 2005.

[Kra18] H. Krawczyk. The opaque asymmetric pake protocol draft-krawczyk-cfrg-opaque-00. `https://tools.ietf.org/html/draft-krawczyk-cfrg-opaque-00`, 2018. [Online; accessed 2020-3-20].

[KSC10] Stan Kurkovsky, Ewa Syta, and Bernardo Casano. Continuous rfid-enabled authentication and its privacy implications. In *2010 IEEE International Symposium on Technology and Society*, pages 103–110. IEEE, 2010.

[KTX07] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit cryptosystems based on lattice problems. In *Public Key Cryptography–PKC 2007*, pages 315–329. Springer, 2007.

[Kum04] M. Kumar. New remote user authentication scheme using smart cards. *Consumer Electronics, IEEE Transactions on*, 50(2):597–600, 2004.

[KV10] Karl Kümmel and Claus Vielhauer. Reverse-engineer methods on a biometric hash algorithm for dynamic handwriting. In *Proceedings of the 12th ACM workshop on Multimedia and security*, pages 67–72. ACM, 2010.

[KYSR09] Geraldine Kwang, Roland HC Yap, Terence Sim, and Rajiv Ramnath. An usability study of continuous biometrics authentication. In *International Conference on Biometrics*, pages 828–837. Springer, 2009.

[LATV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homo-

morphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234. ACM, 2012.

[LCP17] Kim Laine, Hao Chen, and Rachel Player. Simple encrypted arithmetic library. `https://sealcrypto.codeplex.com/`, 2017. [Online; accessed September-2015].

[LdVMPT09] Françoise Levy-dit Vehel, Maria Grazia Marinari, Ludovic Perret, and Carlo Traverso. A survey on polly cracker systems. *Gröbner Bases, Coding, and Cryptography*, pages 285–305, 2009.

[LdVP04] Françoise Levy-dit Vehel and Ludovic Perret. A polly cracker system based on satisfiability. *Coding, Cryptography and Combinatorics*, pages 177–192, 2004.

[LDW+18] Chris Xiaoxuan Lu, Bowen Du, Hongkai Wen, Sen Wang, Andrew Markham, Ivan Martinovic, Yiran Shen, and Niki Trigoni. Snoopy: sniffing your smartwatch passwords via deep sequence learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):152, 2018.

[Le03] Van-Ly Le. *Polly two-a public key cryptosystem based on Polly cracker.* PhD thesis, Ruhr University Bochum, Germany, 2003.

[Lee11] Moon Sung Lee. On the sparse subset sum problem from gentry-halevi's implementation of fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2011:567, 2011.

[LL06] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted http connections. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 255–263. ACM, 2006.

[LL17] Wei-Han Lee and Ruby B Lee. Sensor-based implicit authentication of smartphone users. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 309–320. IEEE, 2017.

[LLCP15] Moon Sung Lee, Yongje Lee, Jung Hee Cheon, and Yunheung Paek. Accelerating bootstrapping in fhew using gpus. In *Application-specific Systems, Architectures and Processors (ASAP), 2015 IEEE 26th International Conference on*, pages 128–135. IEEE, 2015.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LMSV11] Jake Loftus, Alexander May, Nigel P Smart, and Frederik Vercauteren. On cca-secure somewhat homomorphic encryption. In *Selected Areas in Cryptography*, pages 55–72. Springer, 2011.

[LN14] Tancrède Lepoint and Michael Naehrig. A comparison of the homomorphic encryption schemes fv and yashe. In *Progress in Cryptology–AFRICACRYPT 2014*, pages 318–335. Springer, 2014.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43, 2013.

[LS96] Henry George Liddell and Robert Scott. *An intermediate Greek-English lexicon: founded upon the seventh edition of Liddell and Scott's Greek-English lexicon*. Harper & Brothers, 1896.

[Ltd13] Fujitsu Laboratories Ltd. Fujitsu develops world's first homomorphic encryption technology that enables statistical calculations and biometric authentication. `http://www.fujitsu.com/global/about/resources/news/press-releases/2013/0828-01.html`, 2013. press release dated August 5, 2013.

[LUB14a] W. Liu, A. S. Uluagac, and R. Beyah. Maca: A privacy-preserving multi-factor cloud authentication system utilizing big data. *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 518–523, 2014.

[LUB14b] Wenyi Liu, A Selcuk Uluagac, and Raheem Beyah. Maca: A privacy-preserving multi-factor cloud authentication system utilizing big data. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 518–523. IEEE, 2014.

[LW15] Bingxin Liu and Huapeng Wu. Efficient architecture and implementation for ntruencrypt system. In *Circuits and Systems (MWSCAS), 2015 IEEE 58th International Midwest Symposium on*, pages 1–4. IEEE, 2015.

[LWN+15] C. Liu, X. S. Wang, K. Nayak, Y. Huang, and E. Shi. Oblivm: A programming framework for secure computation. In *2015 IEEE Symposium on Security and Privacy*, pages 359–376, May 2015.

[LXZ+16] Huaxin Li, Zheyu Xu, Haojin Zhu, Di Ma, Shuai Li, and Kai Xing. Demographics inference through wi-fi network traffic analysis. In *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, pages 1–9. IEEE, 2016.

[MBS+17] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. Detection of unauthorized iot devices using machine learning techniques. *arXiv preprint arXiv:1709.04647*, 2017.

[MCG08] Carlos Aguilar Melchor, Guilhem Castagnos, and Philippe Gaborit. Lattice-based homomorphic encryption of vector spaces. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 1858–1862. IEEE, 2008.

[McM13] Robert McMillan. Apple finally reveals how long siri keeps your data. `http://www.wired.com/2013/04/siri-two-years/`, April 2013. [Online; accessed 2020-1-20].

[McN02] Alexander J McNeil. The laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance. *Journal of the American Statistical Association*, 97(460):1210–1212, 2002.

[MGBF14] Benjamin Mood, Debayan Gupta, Kevin Butler, and Joan Feigenbaum. Reuse it or lose it: More efficient secure computation through reuse of encrypted values. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 582–596. ACM, 2014.

[MGH10] Carlos Aguilar Melchor, Philippe Gaborit, and Javier Herranz. Additively homomorphic encryption with d-operand multiplications. In *Advances in Cryptology–CRYPTO 2010*, pages 138–154. Springer, 2010.

[MHM+13] Ciara Moore, Neil Hanley, John McAllister, Máire O'Neill, Elizabeth O'Sullivan, and Xiaolin Cao. Targeting fpga dsp slices for a large

integer multiplier for integer based fhe. In *Financial Cryptography and Data Security*, pages 226–237. Springer, 2013.

[Mic19] Microsoft. Turning two-step verification on or off for your microsoft account. `https://support.microsoft.com/en-us/help/4028586/microsoft-account-turning-two-step-verification-on-or-off`, 2019. [Online; accessed 2020-3-20].

[Mik12] Michal Mikuš. Experiments with the plaintext space in gentry's somewhat homomorphic scheme. *Tatra Mountains Mathematical Publications*, 53(1):147–154, 2012.

[Min68] Hermann Minkowski. *Geometrie der zahlen*, volume 40. B. G. Teubner, 1968.

[MMC+14] Shrirang Mare, Andrés Molina Markham, Cory Cornelius, Ronald Peterson, and David Kotz. Zebra: zero-effort bilateral recurring authentication. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 705–720, 2014.

[MMH+17] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 2177–2184. IEEE, 2017.

[MOHO14] Ciara Moore, Maire O'Neill, Neil Hanley, and Elizabeth O'Sullivan. Accelerating integer-based fully homomorphic encryption using comba multiplication. In *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*, pages 1–6. IEEE, 2014.

[Mon94] Peter L Montgomery. A survey of modern integer factorization algorithms. *CWI quarterly*, 7(4):337–366, 1994.

[MOO+14] Ciara Moore, Maire O'Neill, Elizabeth O'Sullivan, Yarkın Doröz, and Berk Sunar. Practical homomorphic encryption: A survey. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 2792–2795. IEEE, 2014.

[MP06] Patrick McDaniel and Atul Prakash. Methods and limitations of security policy reconciliation. *ACM Trans. Inf. Syst. Secur.*, 9(3):259–291, August 2006.

[MR97] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56, 1997.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[MR18] WA. Microsoft Research, Redmond. Simple Encrypted Arithmetic Library (release 3.1.0). `https://github.com/Microsoft/SEAL`, December 2018. [Online; accessed 2020-3-20].

[MRRT17] Ivan Martinovic, Kasper Rasmussen, Marc Roeschlin, and Gene Tsudik. Authentication using pulse-response biometrics. *Commun. ACM*, 60(2):108–115, January 2017.

[MS13] Silvia Mella and Ruggero Susella. On the homomorphic computation of symmetric cryptographic primitives. In *Cryptography and Coding*, pages 28–44. Springer, 2013.

[MT07] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103, Oct 2007.

[MT16] Javid Maghsoudi and Charles C Tappert. A behavioral biometrics user authentication study using motion data from android smartphones. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 184–187. IEEE, 2016.

[MVBC12] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 323–336. ACM, 2012.

[Myu03] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47(1):90 – 100, 2003.

[MZ17] Payman Mohassel and Yupeng Zhang. SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 1–20. IEEE, 2017.

[NJ15] K. Nandakumar and A. K. Jain. Biometric template protection: Bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, 32(5):88–100, Sep. 2015.

[NK15] Koji Nuida and Kaoru Kurosawa. (batch) fully homomorphic encryption over integers for non-binary message spaces. In *Advances in Cryptology–EUROCRYPT 2015*, pages 537–555. Springer, 2015.

[NLV11] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM, 2011.

[NMM+19] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A. Sadeghi. DÏot: A federated self-learning anomaly detection system for iot. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 756–767, July 2019.

[NMX+16] I. Natgunanathan, A. Mehmood, Y. Xiang, G. Beliakov, and J. Yearwood. Protection of privacy in biometric data. *IEEE Access*, 4:880–892, 2016.

[NNJ07] K. Nandakumar, A. Nagar, and A. Jain. Hardening fingerprint fuzzy vault using password. *Advances in Biometrics, Lecture Notes in Computer Science*, 4642:927–937, 2007.

[NNJ10] A. Nagar, K. Nandakumar, and A. Jain. Biometric template transformation: A Security analysis. *Proceedings of SPIE, Electronic Imaging, Media Forensics and Security II*, 7541:1–15, 2010. Invited paper.

[NPVB05] Nam Thanh Nguyen, Dinh Q Phung, Svetha Venkatesh, and Hung Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 955–960. IEEE, 2005.

[NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.

[NS98] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *Proceedings of the 5th ACM conference on Computer and communications security*, pages 59–66. ACM, 1998.

[NS08] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, May 2008.

[O+16] Olga Ohrimenko et al. Oblivious Multi-Party Machine Learning on Trusted Processors. In *USENIX Security Symposium*, pages 1–19, 2016.

[ÖDSS15] E Öztürk, Yarkın Doröz, Berk Sunar, and E Savaş. Accelerating somewhat homomorphic evaluation using fpgas. Technical report, Cryptology ePrint Archive, Report 2015/294, 2015.

[OER19] TJ OConnor, William Enck, and Bradley Reaves. Blinded and confused: uncovering systemic flaws in device telemetry for smart-home internet of things. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 140–150. ACM, 2019.

[OHD+12] Emmanuel Owusu, Jun Han, Sauvik Das, Adrian Perrig, and Joy Zhang. Accessory: Password inference using accelerometers on smartphones. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, HotMobile '12, New York, NY, USA, 2012. Association for Computing Machinery.

[OMM+19] TJ OConnor, Reham Mohamed, Markus Miettinen, William Enck, Bradley Reaves, and Ahmad-Reza Sadeghi. Homesnitch: behavior transparency and control for smart home iot devices. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 128–138. ACM, 2019.

[Osb] Charlie Osborne. Symantec sacks staff for issuing unauthorized google certificates. `https://www.zdnet.com/article/symantec-sacks-staff-for-issuing-unauthorized-google-certificates/`. [Online; accessed 2020-1-20].

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology—EUROCRYPT'98*, pages 308–318. Springer, 1998.

[OYKU10] Naoki Ogura, Go Yamamoto, Tetsutaro Kobayashi, and Shigenori Uchiyama. An improvement of key generation algorithm for gentry's homomorphic encryption scheme. In *Advances in Information and Computer Security*, pages 70–83. Springer, 2010.

[PAD12] Pedro Silveira Pisa, Michel Abdalla, and Otto Carlos Duarte. Somewhat homomorphic encryption scheme for arithmetic operations on large integers. In *Global Information Infrastructure and Networking Symposium (GIIS), 2012*, pages 1–8. IEEE, 2012.

[Pai99a] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*, pages 223–238. Springer, 1999.

[Pai99b] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[PBS11a] Henning Perl, Michael Brenner, and Matthew Smith. An implementation of the fully homomorphic smart-vercauteren cryptosystem. `https://github.com/hcrypt-project/libScarab`, 2011. [Online; accessed December-2015].

[PBS11b] Henning Perl, Michael Brenner, and Matthew Smith. Poster: an implementation of the fully homomorphic smart-vercauteren cryptosystem. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 837–840. ACM, 2011.

[PCCB16] Vishal M Patel, Rama Chellappa, Deepak Chandra, and Brandon Barbello. Continuous user authentication on mobile devices: Recent progress and remaining challenges. *IEEE Signal Processing Magazine*, 33(4):49–61, 2016.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[PGR07] N. D. Phung, M. M. Gaber, and U. Rohm. Resource-aware online data mining in wireless sensor networks. In *2007 IEEE Symposium on Computational Intelligence and Data Mining*, pages 139–146, March 2007.

[PH14] Sriram N Premnath and Zygmunt J Haas. A practical, secure, and verifiable cloud computing for mobile systems. *Procedia Computer Science*, 34:474–483, 2014.

[PM17] Elena Pagnin and Aikaterini Mitrokotsa. Privacy-preserving biometric authentication: challenges and directions. *Security and Communication Networks*, 2017:1–9, 2017.

[PMZ+11] CelesteLyn Paul, Emile Morse, Aiping Zhang, Yee-Yin Choong, and Mary Theofanos. A field study of user behavior and perceptions in smartcard authentication. In *Human-Computer Interaction INTERACT 2011*, volume 6949 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2011.

[PNPM15] Thomas Pöppelmann, Michael Naehrig, Andrew Putnam, and Adrian Macias. Accelerating homomorphic evaluation on reconfigurable hardware. In *Cryptographic Hardware and Embedded Systems–CHES 2015*, pages 143–163. Springer, 2015.

[PPJ03] S. Prabhakar, S. Pankanti, and A. K. Jain. Biometric recognition: security and privacy concerns. *IEEE Security Privacy*, 1(2):33–42, March 2003.

[PPP+14] Payal V Parmar, Shraddha B Padhar, Shafika N Patel, Niyatee I Bhatt, and Rutvij H Jhaveri. Survey of various homomorphic encryption algorithms and schemes. *International Journal of Computer Applications*, 91(8):1–7, 2014.

[PRR10] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1876–1884. Curran Associates, Inc., 2010.

[PVG+11a] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html`, 2011. [Online; accessed 2020-1-20].

[PVG+11b] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_classif.html`, 2011. [Online; accessed 2020-1-20].

[Que12] AWE Multi-Factor Authentication Frequently-Asked Questions. `http://aws.amazon.com/cn/mfa/faqs/`, 2012. [Online; accessed 20-March-2020].

[RAD78] Ronald L Rivest, Len Adleman, and Michael L Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[RBBB14] Christian Rathgeb, Frank Breitinger, Christoph Busch, and Harald Baier. On application of bloom filters to iris biometrics. *IET Biometrics*, 3(4):207–218, 2014.

[RC14] Kurt Rohloff and David Bruce Cousins. A scalable implementation of fully homomorphic encryption built on ntru. In *Financial Cryptography and Data Security*, pages 221–234. Springer, 2014.

[RCB01] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3):614–634, 2001.

[RCCB07] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, April 2007.

[RDC+19] Jingjing Ren, Daniel J Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proceedings of the Internet Measurement Conference*, pages 267–279. ACM, 2019.

[Reg06] Oded Regev. Lattice-based cryptography. In *Advances in Cryptology-CRYPTO 2006*, pages 131–141. Springer, 2006.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[Rep17] European Commission Report. Overview of the National Laws on Electronic Health Records in the EU Member States. http://ec.europa.eu, 2017. [Online; accessed 01-June-2018].

[RHH14] A. Rastogi, M. A. Hammer, and M. Hicks. Wysteria: A programming language for generic, mixed-mode multiparty computations. In *2014 IEEE Symposium on Security and Privacy*, pages 655–670, May 2014.

[RJR+16] Ana C. Riekstin, Guilherme C. Januário, Bruno B. Rodrigues, Viviane T. Nascimento, Tereza C.M.B. Carvalho, and Catalin Meirosu. Orchestration of energy efficiency capabilities in networks. *Journal of Network and Computer Applications*, 59:74 – 87, 2016.

[RK12a] Y Govinda Ramaiah and G Vijaya Kumari. Efficient public key homomorphic encryption over integer plaintexts. In *Information Security and Intelligence Control (ISIC), 2012 International Conference on*, pages 123–128. IEEE, 2012.

[RK12b] Y Govinda Ramaiah and G Vijaya Kumari. Towards practical homomorphic encryption with efficient public key generation. *International Journal on Network Security*, 3(4):10, 2012.

[RKBT07] Ajita Rattani, Dakshina Ranjan Kisku, Manuele Bicego, and Massimo Tistarelli. Feature level fusion of face and fingerprint biometrics. In *2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6. IEEE, 2007.

[Roh17] Kurt Rohloff. The palisade lattice cryptography library. `https://git.njit.edu/palisade/PALISADE`, 2017. [Online; accessed September-2017].

[Rot11] Ron Rothblum. Homomorphic encryption: From private-key to public-key. In *Theory of cryptography*, pages 219–234. Springer, 2011.

[RP13] Ajita Rattani and Norman Poh. Biometric system design under zero and non-zero effort attacks. In *2013 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2013.

[RSA78] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RTWB16] C. Rathgeb, B. Tams, J. Wagner, and C. Busch. Unlinkable improved multi-biometric iris fuzzy vault. *EURASIP Journal on Information Security*, 26:1–16, 2016.

[RU10] Christian Rathgeb and Andreas Uhl. Iris-biometric hash generation for biometric database indexing. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2848–2851. IEEE, 2010.

[RU11] Christian Rathgeb and Andreas Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):3, 2011.

[RU12] C. Rathgeb and A. Uhl. Statistical attack against fuzzy commitment scheme. *IET Biometrics*, 1:94–104, 2012.

[SAA+18] Amit Kumar Sikder, Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, Kemal Akkaya, and Mauro Conti. Iot-enabled smart lighting systems for smart cities. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 639–645. IEEE, 2018.

[SAU17] Amit Kumar Sikder, Hidayet Aksu, and A Selcuk Uluagac. 6thsense: A context-aware sensor-based attack detector for smart devices. In *26th USENIX Security Symposium*, pages 1–19, 2017.

[SB07] W. Scheirer and T. Boult. Cracking fuzzy vaults and biometric encryption. *Proceedings of Biometrics Symposium*, pages 1–6, 2007.

[SBC+19] Amit Kumar Sikder, Leonardo Babun, Z Berkay Celik, Abbas Acar, Hidayet Aksu, Patrick McDaniel, Engin Kirda, and A Selcuk Uluagac. Multi-user multi-device-aware access control system for smart home. *arXiv preprint arXiv:1911.10186*, 2019.

[SBZD18] Mustafizur R Shahid, Gregory Blanc, Zonghua Zhang, and Hervé Debar. Iot devices recognition through network traffic analysis. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5187–5192. IEEE, 2018.

[SCG12] Chao Shen, Zhongmin Cai, and Xiaohong Guan. Continuous authentication for mouse dynamics: A pattern-growth approach. *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pages 1–12, 2012.

[SCR+11] Elaine Shi, HTH Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. Privacy-preserving aggregation of time-series data. In *Annual Network & Distributed System Security Symposium (NDSS)*, pages 1–17. Internet Society., 2011.

[SCST17] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4424–4434. Curran Associates, Inc., 2017.

[Sea16] Ioannis C Stylios et al. A review of continuous authentication using behavioral biometrics. In *Proceedings of the SouthEast European Design Automation, Computer Engineering, Computer Networks and Social Media Conference*, pages 72–79. ACM, 2016.

[SECK19] Ola Salman, Imad H Elhajj, Ali Chehab, and Ayman Kayssi. A machine learning based framework for iot device identification and abnormal traffic detection. *Transactions on Emerging Telecommunications Technologies*, page e3743, 2019.

[SFSM13] Tim Stöber, Mario Frank, Jens Schmitt, and Ivan Martinovic. Who do you sync you are?: smartphone fingerprinting via application behaviour. In *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*, pages 7–12. ACM, 2013.

[ŠGGB15] Jaroslav Šeděnka, Sathya Govindarajan, Paolo Gasti, and Kiran S Balagani. Secure outsourced biometric authentication with performance evaluation on smartphones. *IEEE Transactions on Information Forensics and Security*, 10(2):384–396, 2015.

[She12] Chao Shen. Mouse data for continuous authentication. `http://nskeylab.xjtu.edu.cn/people/cshen/`, April 2012. [Online; accessed September-2014].

[SIG+19] Saeed Samet, Mohd Tazim Ishraque, Mehdi Ghadamyari, Krishna Kakadiya, Yash Mistry, and Youssef Nakkabi. Touchmetric: a ma-

chine learning based continuous authentication feature testing mobile application. *International Journal of Information Technology*, 11(4):625–631, Dec 2019.

[Sil13] Alice Silverberg. Fully homomorphic encryption for mathematicians. *Women in Numbers 2: Research Directions in Number Theory*, 606:111, 2013.

[SKD+10] C. Strasburg, S. Krishnan, K. Dorman, S. Basu, and J. S. Wong. Masquerade detection in network environments. *2010 10th IEEE/IPSJ International Symposium on Applications and the Internet*, pages 38–44, July 2010.

[SKH+19] Muhammad Sajjad, Salman Khan, Tanveer Hussain, Khan Muhammad, Arun Kumar Sangaiah, Aniello Castiglione, Christian Esposito, and Sung Wook Baik. Cnn-based anti-spoofing two-tier multi-factor authentication system. *Pattern Recognition Letters*, 126:123 – 131, 2019. Robustness, Security and Regulation Aspects in Current Biometric Systems.

[SKLR04] Ashish P. Sanil, Alan F. Karr, Xiaodong Lin, and Jerome P. Reiter. Privacy preserving regression modelling via distributed computation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 677–682, New York, NY, USA, 2004. Association for Computing Machinery.

[SLM+16] Riccardo Spolaor, QianQian Li, Merylin Monaro, Mauro Conti, Luciano Gamberini, and Giuseppe Sartori. Biometric authentication methods on smartphones: A survey. *PsychNology Journal*, 14(2):87–98, 2016.

[SO19] Asma Salem and Mohammad S. Obaidat. A novel security scheme for behavioral authentication systems based on keystroke dynamics. *Wiley Security and Privacy*, 2(2):e64, 2019.

[SP12] M. Sujithra and G. Padmavathi. Next generation biometric security system: An approach for mobile device security. In *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, CCSEIT '12, page 377–381, New York, NY, USA, 2012. Association for Computing Machinery.

[SP13] Abdul Serwadda and Vir V Phoha. Examining a large keystroke biometrics dataset for statistical-attack openings. *ACM Transactions on Information and System Security (TISSEC)*, 16(2):8, 2013.

[SPW13] Abdul Serwadda, Vir V Phoha, and Zibo Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8, 2013.

[SRJV+15] Sujoy Sinha Roy, Kimmo Järvinen, Frederik Vercauteren, Vassil Dimitrov, and Ingrid Verbauwhede. Modular hardware architecture for somewhat homomorphic function evaluation. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems – CHES 2015*, pages 164–184, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[SRRM16] Ivo Sluganovic, Marc Roeschlin, Kasper B Rasmussen, and Ivan Martinovic. Using reflexive eye movements for fast challenge-response authentication. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1056–1067, 2016.

[SS10] Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In *Advances in Cryptology-ASIACRYPT 2010*, pages 377–394. Springer, 2010.

[SS11a] Peter Scholl and Nigel P Smart. Improved key generation for gentry's fully homomorphic encryption scheme. In *Cryptography and Coding*, pages 10–22. Springer, 2011.

[SS11b] Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *Advances in Cryptology–EUROCRYPT 2011*, pages 27–47. Springer, 2011.

[SS15] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, page 1310–1321, New York, NY, USA, 2015. Association for Computing Machinery.

[SSCG16] Valeriu-Daniel Stanciu, Riccardo Spolaor, Mauro Conti, and Cristiano Giuffrida. On the effectiveness of sensor-enhanced keystroke dynamics against statistical attacks. In *Proceedings of the Sixth ACM*

Conference on Data and Application Security and Privacy, pages 105–112, 2016.

[SSNS14] Nashad Ahmed Safa, Reihaneh Safavi-Naini, and Siamak F Shahandashti. Privacy-preserving implicit authentication. In *IFIP International Information Security Conference*, pages 471–484. Springer, 2014.

[SSNS15] Siamak F Shahandashti, Reihaneh Safavi-Naini, and Nashad Ahmed Safa. Reconciling user privacy and implicit authentication for mobile devices. *Computers & Security*, 53:215–233, 2015.

[SSW+02] Qixiang Sun, Daniel R Simon, Yi-Min Wang, Wilf Russell, Venkata N Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on*, pages 19–30. IEEE, 2002.

[SSW08] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. Protecting your daily in-home activity information from a wireless snooping attack. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 202–211. ACM, 2008.

[SSY12] Deian Stefan, Xiaokui Shu, and Danfeng Daphne Yao. Robustness of keystroke-dynamics based biometrics against synthetic forgeries. *computers & security*, 31(1):109–121, 2012.

[Ste10] Rainer Steinwandt. A ciphertext-only attack on polly two. *Applicable Algebra in Engineering, Communication and Computing*, 21(2):85–92, 2010.

[Sto10] A. Stoianov. Cryptographically secure biometrics. In B. V. K. Vijaya Kumar, Salil Prabhakar, and Arun A. Ross, editors, *Biometric Technology for Human Identification VII*, volume 7667, pages 107 – 118. International Society for Optics and Photonics, SPIE, 2010.

[STP09] K. Simoens, P. Tuyls, and B. Preneel. Privacy Weaknesses in Biometric Sketches. *Security and Privacy, 2009 30th IEEE Symposium on Security and Privacy*, pages 188–203, 2009.

[STU17] A. Smith, A. Thakurta, and J. Upadhyay. Is interaction necessary for distributed private learning? In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 58–77, May 2017.

[SV10] Nigel P Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography–PKC 2010*, pages 420–443. Springer, 2010.

[SV14] Nigel P Smart and Frederik Vercauteren. Fully homomorphic simd operations. *Designs, codes and cryptography*, 71(1):57–81, 2014.

[SWT01] Dawn Xiaodong Song, David Wagner, and Xuqing Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th USENIX Security Symposium*, volume 2001, pages 1–17, 2001.

[SYY99] T. Sander, A. Young, and M. Yung. Non-interactive cryptocomputing for nc1. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 554–566, 1999.

[Sø17] Thomas Søderholm. Eu gdpr: Privacy for connected medical devices. `https://blog.nordicsemi.com/getconnected/eu-gdpr-privacy-for-connected-medical-devices`, 2017. [Online; accessed 2020-1-20].

[Tam14] B. Tams. Decodability attack against the fuzzy commitment scheme with public feature transforms. `arxiv.org/abs/1406.1154.pdf`, 2014.

[TBA+19] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, AISec'19, page 1–11, New York, NY, USA, 2019. Association for Computing Machinery.

[TCCL14] Cheng-Jung Tasia, Ting-Yi Chang, Pei-Cheng Cheng, and Jyun-Hao Lin. Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. *Security and Communication Networks*, 7(4):750–758, 2014.

[TGC16] Mary Theofanos, Simson Garfinkel, and Yee-Yin Choong. Secure and usable enterprise authentication: Lessons from the field. *IEEE Security & Privacy*, 14(5):14–21, 2016.

[TGG13] Chee Meng Tey, Payas Gupta, and Debin Gao. I can be you: Questioning the use of keystroke dynamics as biometrics. *Annual Network*

and Distributed System Security Symposium 20th NDSS 2013, 24-27 February, pages 1–16, 2013.

[TGN06] A. Teoh, A. Goh, and D. Ngo. Random multispace quantization as an analytic mechanism for BioHashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1892–1901, 2006.

[TMM15] B. Tams, P. Mihăilescu, and A. Munk. Security considerations in minutiae-based fuzzy vaults. *IEEE Transactions on Information Forensics and Security*, 10(5):985–998, May 2015.

[TO13] Matthias Trojahn and Frank Ortmeier. Toward mobile authentication with keystroke dynamics on mobile phones and tablets. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 697–702. IEEE, 2013.

[TSCM16] Vincent F Taylor, Riccardo Spolaor, Mauro Conti, and Ivan Martinovic. Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 439–454. IEEE, 2016.

[TTY13] Pin Shen Teh, Andrew Beng Jin Teoh, and Shigang Yue. A survey of keystroke dynamics biometrics. *The Scientific World Journal*, 2013:1–24, 2013.

[TVMD20] Rahmadi Trimananda, Janus Varmarken, Athina Markopoulou, and Brian Demsky. Pingpong: Packet-level signatures for smart home device events. *Network and Distributed Systems Security (NDSS) Symposium 2020*, pages 1–18, 2020.

[Vai11] Vinod Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 5–16. IEEE, 2011.

[VČČD15] Petr Velan, Milan Čermák, Pavel Čeleda, and Martin Drašar. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management*, 25(5):355–374, 2015.

[VDGHV10] Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in cryptology–EUROCRYPT 2010*, pages 24–43. Springer, 2010.

[VDJ10] Marten Van Dijk and Ari Juels. On the Impossibility of Cryptography Alone for Privacy-preserving Cloud Computing. In *USENIX Hot Topics in Security*, pages 1–6, 2010.

[Ver12] VeriSign. Using verisign identity protection. `https://pci.qualys.com/static/help/merchant/account/vip/using_vip.htm`, 2012. [Online; accessed 2020-3-20].

[VL06] Le Van Ly. Polly two: a new algebraic polynomial-based public-key scheme. *Applicable Algebra in Engineering, Communication and Computing*, 17(3):267–283, 2006.

[Wag03] David Wagner. Cryptanalysis of an algebraic privacy homomorphism. In *Information Security*, pages 234–239. Springer, 2003.

[WCH14] Wei Wang, Zhilu Chen, and Xinming Huang. Accelerating leveled fully homomorphic encryption using gpu. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 2800–2803. IEEE, 2014.

[WDL+18] Changsheng Wu, Wenbo Ding, Ruiyuan Liu, Jiyu Wang, Aurelia C Wang, Jie Wang, Shengming Li, Yunlong Zi, and Zhong Lin Wang. Keystroke dynamics enabled authentication and identification using triboelectric nanogenerator array. *Materials Today*, 21(3):216–222, 2018.

[WH13] Wei Wang and Xinming Huang. Fpga implementation of a large-number multiplier for fully homomorphic encryption. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 2589–2592. IEEE, 2013.

[WHC+15] Wei Wang, Yin Hu, Lianmu Chen, Xinming Huang, and Berk Sunar. Exploring the feasibility of fully homomorphic encryption. *Computers, IEEE Transactions on*, 64(3):698–706, 2015.

[WHEW14] Wei Wang, Xinming Huang, Niall Emmart, and Charles Weems. Vlsi design of a large-number multiplier for fully homomorphic encryption.

Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 22(9):1879–1887, 2014.

[WKT11] Fang-Jing Wu, Yu-Fen Kao, and Yu-Chee Tseng. From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile Computing*, 7(4):397 – 413, 2011.

[WLRC15] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 155–166. ACM, 2015.

[WMM06] Charles V Wright, Fabian Monrose, and Gerald M Masson. On inferring application protocol behaviors in encrypted network traffic. *Journal of Machine Learning Research*, 7(Dec):2745–2769, 2006.

[WRDI12] Y. Wang, S. Rane, S. Draper, and P. Ishwar. A Theoretical analysis of authentication, privacy, and reusability across secure biometric systems. *IEEE Transactions on Information Forensics and Security*, 7:1825–1840, 2012.

[Wu07] Tom Wu. The stanford srp homepage. `http://srp.stanford.edu/`, 2007. [Online; accessed 2020-3-20].

[Wu15] David J Wu. Fully homomorphic encryption: Cryptography's holy grail. *XRDS: Crossroads, The ACM Magazine for Students*, 21(3):24–29, 2015.

[WWL12] Ting Wu, Hui Wang, and You-Ping Liu. Optimizations of brakerski's fully homomorphic encryption scheme. In *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pages 2000–2005. IEEE, 2012.

[WWL15a] Fuqun Wang, Kunpeng Wang, and Bao Li. An efficient leveled identity-based fhe. In *Network and System Security*, pages 303–315. Springer, 2015.

[WWL15b] Fuqun Wang, Kunpeng Wang, and Bao Li. Lwe-based fhe with better parameters. In *Advances in Information and Computer Security*, pages 175–192. Springer, 2015.

[WWZJ18] Guannan Wu, Jian Wang, Yongrong Zhang, and Shuai Jiang. A continuous identity authentication scheme based on physiological and behavioral characteristics. *Sensors*, 18(1):179, 2018.

[XBZ12] Zhi Xu, Kun Bai, and Sencun Zhu. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pages 113–124, 2012.

[Yao82] A. C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, Nov 1982.

[YG08] Roman V. Yampolskiy and Venu Govindaraju. Behavioural biometrics: a survey and classification. *Int. J. Biometrics*, 1(1):81–113, June 2008.

[YSK+13] Masaya Yasuda, Takeshi Shimoyama, Jun Kogure, Kazuhiro Yokoyama, and Takeshi Koshiba. Packed homomorphic encryption based on ideal lattices and its application to biometrics. In *International Conference on Availability, Reliability, and Security*, pages 55–74. Springer, 2013.

[YXWT12] Hao-Miao Yang, Qi Xia, Xiao-fen Wang, and Dian-hua Tang. A new somewhat homomorphic encryption scheme over integers. In *Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on*, pages 61–64. IEEE, 2012.

[ZBHW14] Nan Zheng, Kun Bai, Hai Huang, and Haining Wang. You are how you touch: User verification on smartphones via tapping behaviors. In *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pages 221–232. IEEE, 2014.

[Zha14] Zhenfei Zhang. *Revisiting fully homomorphic encryption schemes and their cryptographic primitives*. PhD thesis, University of Wollongong, 2014.

[ZPW11] Nan Zheng, Aaron Paloski, and Haining Wang. An efficient user verification system via mouse movements. *Proc. of the 18th ACM conference on Computer and communications security*, pages 139–150, 2011.

[ZW09] Kehuan Zhang and XiaoFeng Wang. Peeping tom in the neighborhood: Keystroke eavesdropping on multi-user systems. *USENIX Security Symposium*, 20:1–23, 2009.

[ZXJ+14] Xiaojun Zhang, Chunxiang Xu, Chunhua Jin, Run Xie, and Jining Zhao. Efficient fully homomorphic encryption from rlwe with an extension to a threshold encryption scheme. *Future Generation Computer Systems*, 36:180–186, 2014.

[ZYZW16] Tanping Zhou, Xiaoyuan Yang, Wei Zhang, and Liqiang Wu. Efficient fully homomorphic encryption with circularly secure key switching process. *Int. J. High Perform. Comput. Netw.*, 9(5/6):417–422, January 2016.

[ZZX+12] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. *Proc. VLDB Endow.*, 5(11):1364–1375, July 2012.

VITA

ABBAS ACAR

2015 B.S., Electrical and Electronics Engineering
Middle East Technical University University (METU)
Ankara, Turkey

2019 M.Sc., Electrical Engineering
Florida International University (FIU)
Miami, Florida

2020 Ph.D., Electrical and Computer Engineering
Florida International University (FIU)
Miami, Florida

PUBLICATIONS AND PRESENTATIONS

- A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya. 2020. A Usable and Robust Continuous Authentication Framework using Wearables. IEEE Transactions on Mobile Computing (2020), pages 1–1, 2020. `https://doi.org/10.1109/TMC.2020.2974941`.
- Abbas Acar, Wenyi Liu, Raheem Bayeh, Kemal Akkaya, and Arif Selcuk Uluagac. A privacy-preserving multifactor authentication system. Wiley Security and Privacy, 2(5):e88, 2019.
- Abbas Acar, Long Lu, Engin Kirda, and A. Selcuk Uluagac. An Analysis of Malware Trends in Enterprise Networks. In International Conference on Information Security, pages 360–380. Springer, Cham, 2019.
- Celik, Z. B., Acar, A., Aksu, H., Sheatsley, R., McDaniel, P., & Uluagac, A. S. (2019, March). Curie: Policy-based secure data exchange. In Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy (pp. 121-132).
- A. Acar, H. Aksu, K. Akkaya, and S. A. Uluagac, "Method for continuous user authentication with wearables," Sep. 11 2018, US Patent App. 15/674,133.
- Abbas Acar, Hidayet Aksu, Kemal Akkaya, and A. Selcuk Uluagac. Waca: Wearable-assisted continuous authentication. In 2018 IEEE Security and Privacy Workshops (SPW), pages 264–269, May 2018.
- Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. ACM Comput. Surv., 51(4):79:1–79:35, July 2018.

- Amit Kumar Sikder, Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, Kemal Akkaya, and Mauro Conti. Iot-enabled smart lighting systems for smart cities. In 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pages 639{645. IEEE, 2018.
- A. Acar, Z. B. Celik, H. Aksu, A. S. Uluagac, and P. McDaniel. Achieving secure and differentially private computations in multi-party settings. In 2017 IEEE Symposium on Privacy-Aware Computing (PAC), pages 49–59, Aug 2017.
- Abbas Acar, Hidayet Aksu, Kemal Akkaya, and A. Selcuk Uluagac. Waca: Wearable-assisted continuous authentication framework with motion sensors (poster). In USENIX Security '16 Poster Session. `https://www.usenix.org/conference/usenixsecurity16/poster-session`, 2016.