

A THESIS SUBMITTED TO THE  
GRADUATE SCHOOL OF NATURAL  
AND APPLIED SCIENCES

# SECURE AND EFFICIENT BIOMETRIC AUTHENTICATION BASED ON ADVANCED CRYPTOGRAPHIC PRIMITIVES

ZİYA ALPER GENÇ



JULY 2015

ZİYA ALPER GENÇ

SECURE AND EFFICIENT BIOMETRIC AUTHENTICATION BASED ON ADVANCED CRYPTOGRAPHIC PRIMITIVES

2015

# Secure and Efficient Biometric Authentication Based on Advanced Cryptographic Primitives

A thesis submitted to the  
Graduate School of Natural and Applied Sciences

by

Ziya Alper GENÇ

in partial fulfillment for the  
degree of Master of Science

in

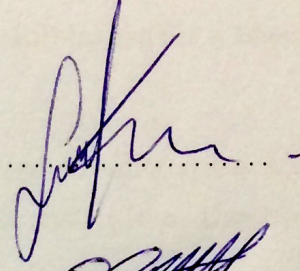
Cybersecurity Engineering



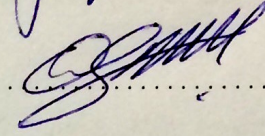
This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Cybersecurity Engineering.

**APPROVED BY:**

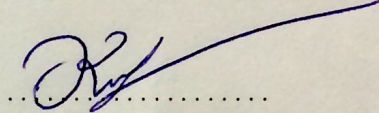
Dr. Mehmet Sabır Kiraz  
(Thesis Advisor)



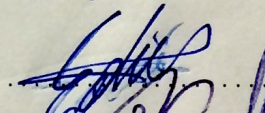
Dr. Osmanbey Uzunkol  
(Thesis Co-advisor)



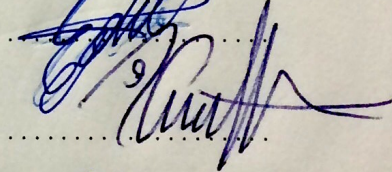
Dr. Kerem Kaşkaloğlu



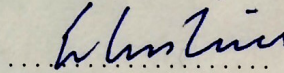
Yrd. Doç. Erdinç Öztürk



Dr. İsa Sertkaya



Prof. Dr. Tahsin Erkan Türe



This is to confirm that this thesis complies with all the standards set by the Graduate School of Natural and Applied Sciences of İstanbul Şehir University:

**DATE OF APPROVAL:**

2015-07-08

**SEAL/SIGNATURE:**



# Declaration of Authorship

I, Ziya Alper GENÇ, declare that this thesis titled, 'Secure and Efficient Biometric Authentication Based on Advanced Cryptographic Primitives' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Ziya

Date:

08.07.2015

*“No human investigation can be called real science if it cannot be demonstrated mathematically. ”*

Leonardo da Vinci

# Secure and Efficient Biometric Authentication Based on Advanced Cryptographic Primitives

Ziya Alper GENÇ

## Abstract

In this thesis, we study secure biometric authentication protocols and mitigation methods against password database breaches. Biometric identifiers such as fingerprint of a user is inherently unique and can be used to authenticate individuals. However, biometric identifiers cannot be replaced with new ones, once they are compromised. Therefore, biometric authentication systems that use biometric data in plain form raises security and privacy issues. In this thesis, we give a survey of state of the art protocols designed for secure biometric authentication. Hamming distance is generally used to compare biometric feature vectors. As a practical example, we analyze a cryptographic protocol of Bringer *et al.* (called SHADE) which aims to securely compute a Hamming distance computation based on Committed Oblivious Transfer protocol. We show that SHADE is in fact insecure in the malicious model. We mount different attacks to the protocol and introduce mitigation techniques that makes the protocol indeed secure. Furthermore, we also analyze the protocol from the efficiency perspective and show that the complexity of the protocol can be significantly improved.

Another important security problem in authentication is the password database breach. In the last few years, several password databases were breached and millions of user names and password hashes were released. With the recent technological advancements like using graphical-processing unit (GPU) in computation, it is comparably easier to invert password hashes. Once the password database has been breached and passwords have been recovered, no server can detect any illegitimate user authentication. In order to thwart these kinds of threats, Juels and Rivest developed the Honeywords system. In this system, each user is associated with multiple possible passwords but only one of them is genuine. This thesis will analyze the security of the Honeywords system as well as from functionality perspective. The authors point out that the Honeyword system cannot withstand active attacks, *i.e.*, code modification of the system. We introduce an enhanced model which solves this open problem. We propose some improvements for determining the number of Honeywords per user, generating typo-safe honeywords and managing old passwords. Finally, the security and efficiency analysis has been discussed.

**Keywords:** oblivious transfers, biometric authentication, shade, password database breach, honeywords

# Gelişmiş Kriptografik Ögelere Dayalı Güvenli ve Verimli Biyometrik Kimlik Doğrulama

Ziya Alper GENÇ

## ÖZ

Bu tezde biyometrik kimlik doğrulama protokolleri ve parola veritabanlarının çalınmasını engellemeye yönelik sistemleri inceledik. Parmak izi gibi biyometrik kimlik belirleyiciler kişisel olarak eşsizdir ve bu sayede kimlik doğrulamada kullanılabilirler. Ancak bir kez ele geçirildiklerinde yerlerine yenileri koyulamaz. Bu nedenle, biyometrik veriyi açık metin halinde kullanan biyometrik sistemler güvenlik ve mahremiyet problemi oluştururlar. Bu tezde son teknoloji ürünü güvenli biyometrik kimlik doğrulama protokolleri ile ilgili yaptığımız araştırmayı sunuyoruz. Biyometrik vasıf vektörleri karşılaştırmak için genellikle Hamming uzaklığı kullanılır. Pratik bir örnek olması için, Bringer ve arkadaşlarının geliştirdiği (SHADE adındaki) kriptografik protokolü analiz ettik. Bu protokol taahhütlü habersiz transfer protokolünü kullanarak Hamming uzaklığını güvenli bir şekilde hesaplamayı amaçlar. Biz ise, SHADE protokolünün kötü niyetli kullanıcılara karşı aslında güvenli olmadığını ispat ettik. Protokole farklı ataklar gerçekleştirdik protokolün gerçekten güvenli olması için bu saldırılara karşı koyma yöntemleri geliştirdik. Bir adım ileri giderek, protokolün verimliliğini inceledik ve protokolün karmaşıklığının iyileştirilebileceğini gösterdik.

Kimlik doğrulamada bir başka önemli problem ise parola veritabanlarının çalınmasıdır. Son birkaç yılda, birçok parola veritabanı çalındı ve milyonlarca kullanıcı adı ve parola özeti açığa çıkmıştır. Son teknolojik gelişmelerle birlikte, örn: grafik işlemci biriminin hesaplamada kullanılması, parola özetlerinin kırılması daha kolay hale gelmiştir. Parola veritabanı çalınp parolalar ele geçirilirse, hiçbir sunucu bunu tespit edemeyecektir. Bu tehditlere karşı, Juels ve Rivest Honeywords sistemini geliştirmişlerdir. Bu sistemde, her kullanıcı birden fazla olası parola ile ilişkilendirilir ancak bunlardan sadece bir tanesi gerçek paroladır. Bu tezde, Honeywords sistemi güvenlik ve fonksiyonellik açısından incelenecektir. Yazarlara göre Honeywords sistemi aktif saldırılara karşı koyamamaktadır, örn: sistemin kodlarının değiştirilemesi. Biz ise bu problemi çözen güçlendirilmiş Honeywords sistemini tanıtaacağız. Kişi başına honeywords sayısı, yanlış yazmaya karşı güvenli honeywords üretimi ve eski parola yönetimi konularına çözüm önereceğiz. Son olarak güvenlik ve verimlilik analizi yapacağız.

**Anahtar Sözcükler:** habersiz transfer, biyometrik kimlik doğrulama, parola veritabanı ifşası

*This thesis is dedicated to my mother Necla, my father Ahmet Murat  
my brother İsmail and my grandmother Nuriye.*



# Acknowledgments

I would like to express my gratitude to my thesis supervisor Dr. Mehmet Sabır Kiraz and my co-advisor Dr. Osmanbey Uzunkol for their guidance and constant encouragement. It is my fortune to work with these two good friends. I am also very grateful to Dr. Süleyman Kardaş for his scientific advice, knowledge, and many insightful discussions and suggestions. It has been an honor to work with them.

A special thanks goes to Birnur Ocaklı for her endless support and friendship. She has always been there for me with her sisterly hand whenever I needed it most. Her constant support has always kept me going ahead. I owe her a great deal of gratitude for always being there.

I am grateful to Dr. İsa Sertkaya for his encouragement and practical advice. I am also thankful to him for reading my thesis, commenting on my views, and helping me understand and enrich my ideas.

I would like to thank to the rest of my thesis committee: Prof. Dr. Tahsin Erkan Türe, Asst. Prof. Erdinç Öztürk and Dr. Kerem Kaşkaloğlu, for their encouragement, insightful comments, and hard questions.

I would also thank to TÜBİTAK BİLGEM UEKAE for its opportunities and contribution on my experience on Cryptography that let me complete my MS study.

Last but not least, I would like to thank my family: my mother Necla, my father Ahmet Murat, and my brother İsmail. A very special thanks goes to my dear grandmother Nuriye, who motivated and encouraged me to finish this scientific research during her lifetime. This thesis, indeed, is the result of her efforts, starting from my childhood. This thesis would never be completed without their support.

# Contents

<b>Abstract</b>	<b>iii</b>
<b>Öz</b>	<b>iv</b>
<b>Acknowledgments</b>	<b>vi</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>x</b>
<b>Abbreviations</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Related Work . . . . .	2
1.2 Contributions . . . . .	5
1.3 List of Publications . . . . .	7
1.4 Outline . . . . .	7
<b>2 Preliminaries</b>	<b>9</b>
2.1 Mathematical Background . . . . .	9
2.2 Authentication . . . . .	10
2.2.1 COT versus VOT . . . . .	11
2.3 Password Related Attacks . . . . .	12
2.4 Security and Privacy Model . . . . .	14
<b>3 Privacy Issue of the SHADE Protocol and Efficiency Improvements</b>	<b>16</b>
3.1 The Basic and the Full Scheme of SHADE . . . . .	16
3.1.1 The Basic Scheme . . . . .	17
3.1.2 The Full Scheme . . . . .	17
3.2 Security and Efficiency Analysis of SHADE . . . . .	18
3.2.1 Attack to the Full Scheme . . . . .	19
3.2.2 A Special Case: Apply the Attack to Biometric Authentication Systems . . . . .	21
3.2.3 Apply the Generic for Uniformly Distributed Inputs . . . . .	21
3.2.4 Our Solution for the Attack . . . . .	22
3.2.4.1 More Efficient Solution for Biometric Authentication . . . . .	23
3.2.5 Efficiency Enhancements . . . . .	23
3.2.5.1 Efficiency Improvement Using VOT . . . . .	23
3.3 Our Fixed and Improved Scheme . . . . .	24

---

3.4	Security Analysis of Our Scheme . . . . .	25
3.5	Complexity Analysis Of Our Fixed Protocol . . . . .	27
<b>4</b>	<b>Enhanced Honeywords System</b>	<b>29</b>
4.1	Honeywords System . . . . .	29
4.2	Improvements for Honeywords System . . . . .	31
4.2.1	Number of Honeywords . . . . .	31
4.2.2	Typo-Safe Honeyword Generation . . . . .	32
4.2.3	Old Passwords Problem . . . . .	34
4.3	Solution to an Open Problem: Active Attacks Against Honeywords System	34
4.3.1	Assumptions . . . . .	35
4.3.2	Adversarial Capabilities . . . . .	35
4.3.3	The Proposal . . . . .	36
4.3.4	Security Analysis of the New Proposed Model . . . . .	37
<b>5</b>	<b>Conclusion</b>	<b>39</b>
	<b>Bibliography</b>	<b>41</b>

# List of Figures

2.1	Verifiable Oblivious Transfer . . . . .	11
2.2	Committed Oblivious Transfer . . . . .	12
2.3	Password Related Attacks . . . . .	12
2.4	Communication over an unsecured channel . . . . .	13
3.1	Our Improved Scheme . . . . .	25
4.1	Standard system vs Honeywords system . . . . .	30
4.2	Login schema of a system using honeywords. . . . .	30
4.3	Typo-Safe Honeyword Generation Algorithm. . . . .	32
4.4	Sign up schema of an enhanced honeywords system. . . . .	37
4.5	Password change schema of an enhanced honeywords system. . . . .	37

# List of Tables

3.1 Complexity Comparison . . . . .	28
-------------------------------------	----

# Abbreviations

<b>COT</b>	<b>C</b> ommitted <b>O</b> blivious <b>T</b> ransfer
<b>COTCD</b>	<b>C</b> ommitted <b>O</b> blivious <b>T</b> ransfer with <b>C</b> onstant <b>D</b> ifference
<b>DB</b>	<b>D</b> ata <b>B</b> ase
<b>GPU</b>	<b>G</b> raphical <b>P</b> rocessing <b>U</b> nit
<b>MD5</b>	<b>M</b> essage <b>D</b> igest <b>5</b>
<b>OPE</b>	<b>O</b> blivious <b>P</b> olynomial <b>E</b> valuation
<b>OT</b>	<b>O</b> blivious <b>T</b> ransfer
<b>SCiFI</b>	<b>S</b> ecure <b>C</b> omputation of <b>F</b> ace <b>I</b> dentification
<b>SHA1</b>	<b>S</b> ecure <b>H</b> ash <b>A</b> lgorithm <b>1</b>
<b>SHADE</b>	<b>S</b> ecure <b>H</b> AMming <b>D</b> istanc <b>E</b> <b>C</b> omputation from <b>O</b> T
<b>SMS</b>	<b>S</b> hort <b>M</b> essage <b>S</b> ervice
<b>VOT</b>	<b>V</b> erifiable <b>O</b> blivious <b>T</b> ransfer

# Chapter 1

## Introduction

Several commercial organizations have invested in secure electronic authentication systems to reliably verify identity of individuals. Biometric authentication systems are receiving a lot of public attention and becoming a crucial solution to many authentication and identity management problems because of cost-effective improvements in sensor technologies and in efficiency of matching algorithms [1]. Biometric data (i.e. templates) of a user is inherently unique. This uniqueness provides assurance to individuals to be securely authenticated for accessing an environment provided that the biometric data is kept as a secret. The biometric data cannot be directly used with conventional encryption techniques because the data itself is inherently noisy [2]. Namely, whenever two samples of data are extracted from the same fingerprint, they will not be exactly the same. In this context, in order to eliminate the noisy nature of the biometric templates, several error correction techniques were proposed in the literature [3-5].

Biometric authentication over an insecure network raises more security and privacy issues. The primary security issue is the protection of the plain biometric templates against a malicious adversary because they cannot be replaced with new ones, once they are compromised. The common biometric authentication system is as follows: For each user, the biometric template is stored in a database during the *enrollment* phase. In the *verification* phase a new fresh acquisition of a user is compared to the template of the same individual stored in the database. The verification phase can either be processed within a smart card (i.e, on-card matching), or in a system outside the card (i.e, off-card matching) [6]. Since the biometric template is not necessarily transferred to outside

environment, the on-card matching technique protects the template. In both techniques, authentication protocols should not expose the biometric template without the user's agreement. In order to ensure privacy of the user, the biometric template should be stored in an encrypted form in a database and no one, including the server, can learn any information on the biometric data in plain form. But still, it should be possible to verify whether a user is authentic [7].

The use of passwords is one of the most common methods for user authentication [8]. However, many users whether they are aware or not, choose either weak passwords or common words that can be easily guessed by using a dictionary attack [9, 10]. Although authentication systems lock user accounts after a small number of unsuccessful login attempts, the adversaries are frequently able to obtain hashed password databases. Lately, there has been several data breaches in which millions of user names and password hashes have been obtained by malicious adversaries [11–15]. Furthermore, the attackers can easily obtain the original passwords by mounting a dictionary attack on password hashes.

There exist several state of the art techniques that increase the success chance of brute force attacks. As an example, Weir *et al.* developed a password cracking algorithm which uses probabilistic, context-free grammars [16]. Kelley *et al.* recently showed [17] that using Weir's attack, one billion guess is enough to crack % 40.3 of the passwords that comply with the "basic8" policy, *i.e.*, all passwords must have at least 8 characters. In the meantime, parallel processing capabilities of GPUs have been increased dramatically. For example, using oclHashcat, a free password recovery software, cracking speed of hashes has reached 8.5 billion/sec for MD5 and 2.7 billion/sec for SHA1 on a single GPU [18]. These advancements make it necessary to develop new and effective security countermeasures.

## 1.1 Related Work

There has been a large amount of research done on the security and efficiency of the biometric authentication systems. In this section, we review the most recent works for biometric authentication.



Hamming distance together with Oblivious Transfers is one of the most elegant tools used in biometric authentication systems. For example, Jarrous and Pinkas propose the *bin*HDOT protocol [19] to compute Hamming distance based on 1-out-of-2 Committed Oblivious Transfer with Constant Difference (COTCD) of Jarecki and Shmatikov [20] and Oblivious Polynomial Evaluation (OPE) of Hazay and Lindell [21]. The protocol also uses commitments and zero-knowledge proofs to guarantee that each party follows the protocol. This protocol provides full security in the malicious model. One OPE protocol and  $n$  COTCDs are invoked to compute the Hamming distance between two strings of  $n$  bits.

The SCiFI (Secure Computation of Face Identification) of Osadchy *et al.* is the first secure face identification system which is well suited for real-life applications [22]. SCiFI system consists of two parts: a client and a server. The server prepares a face recognition database that contains representations of face images. This computation is done offline. In the verification phase, a client prepares her face representation and then a cryptographic protocol which uses Paillier encryption and Oblivious Transfer running between the server and the client. The authors implemented a complete SCiFI system in which a face is represented with a string of 900 bits. The authors designed the system by aiming the minimal online overhead: the most significant requirement for computing Hamming distance between this length of bit strings is 8 invocations of 1-out-of-2 OTs.

Bringer *et al.* [23] used biometric authentication/identification for access control. Note that it is important to securely store the biometric template on the server and using conventional encryption schemes for securing the biometric template can provide a strong protection. Note also that conventional cryptography requires an exact match while biometrics always have a threshold value, therefore biometric authentication over the encrypted domain is a challenging task. In [23], a cryptographic scheme is given for biometric identification over an encrypted domain which uses Bloom Filters with Storage and Locality-Sensitive Hashing. Their paper is interesting because it proposes the first biometric authentication/identification scheme over encrypted binary templates which is stored in the server's database.

In another paper, Bringer *et al.* [24] proposed a security model for biometric-based authentication protocols, relying on the Goldwasser-Micali cryptosystem [25]. This system allows the biometric match to be performed in the encrypted domain in such a way that

the server cannot identify which user is authenticating. The proposed system requires storage of biometric templates in plain form. In order to protect the privacy, the system ensures that the biometric feature stored in the database cannot be explicitly linked to any identity, but the DB only verifies whether the received data belongs to an identity in the database.

Erkin *et al.* [26] propose a privacy preserving face recognition system on encrypted messages which is based on the standard Eigenface recognition system [27]. In their protocol design, they utilized semantically secure Paillier homomorphic public-key encryption schemes and Damgård, Geisler and Krøigaard (DGK) cryptosystem [28, 29]. Later, Sadeghi *et al.* make an improvement over the efficiency of this system [30] by merging the eigenface recognition algorithm using homomorphic encryption and Yao's garbled circuits. Their protocol improves the scheme proposed by Erkin *et al.* significantly since it has only a constant number of rounds and most of the computation and communication is performed during the pre-computation phase. Schneider and Zohner [31] provide an improvement over [30] and [22] by using the GMW protocol [32].

Tuyls *et al.* [33] propose a template protection scheme for fingerprint based authentication in order to protect biometric data. During the enrollment phase, client's biometric features  $X$  is extracted, the Helper Data [34]  $W$  is computed (that is required by the error-correction mechanism), a one-way hash function  $\mathcal{H}$  is applied to  $S$  and the data (client,  $W$ ,  $\mathcal{H}(S)$ ) is stored on the server. Here,  $S$  is a randomly chosen secret value such that  $G(X, W)=S$  for a shielding function  $G$  [35]. During the verification phase, after client's noisy biometric data  $\bar{X}$  is extracted, the server sends  $W$  back to the sensor. The sensor computes  $\bar{S} = G(\bar{X}, W)$  and  $\mathcal{H}(\bar{S})$ . Then, the server compares  $\mathcal{H}(S)$  with  $\mathcal{H}(\bar{S})$ , and grants access if the results are equal. The Helper Data is sent over the public channel, i.e. an adversary may obtain  $W$ . Tuyls *et al.* however designed the system in such a way that the adversary obtains minimal information about  $X$  by capturing  $W$ .

Kulkarni *et al.* [36] propose a biometric authentication scheme based on Iris Matching. Their scheme uses the *somewhat* homomorphic encryption scheme of Boneh *et al.* [37] which allows an arbitrary number of additions of ciphertexts but supports only one multiplication operation between the ciphertexts. The scheme is based on Paillier encryption and bilinear pairings. This scheme consists of two phases: Enrollment phase and Verification phase. During the Enrollment phase, necessary keys are first generated

by the server and then sent to the client securely. Secondly, the client's biometric data is XORed with the key, and a mask value is XORed with a mask key. Both XORed values are sent to the server. During the Verification (authentication) phase, the client sends an encryption of the authenticated biometric data to compute the distance. The protocol is proven to be secure in the semi-honest model.

Kerschbaum *et al.* [38] propose an authentication scheme in a different setting. In particular, they assume that there are two parties where each of them has a fingerprint template. They would like to learn whether the templates match, i.e. generated from the same fingerprint. However, they do not want to reveal the templates if there is no match. Their protocol is secure only in the semi-honest model using secure multi-party computation as a building block.

Barni *et al.* propose a privacy preserving authentication scheme for finger-code templates by using homomorphic encryption which is secure only in the semi-honest model [39, 40]. Their protocol allows the use of the Euclidean distances to compare fingerprints in such a way that the biometric data is reduced for computing a smaller encrypted value that is sent to the server.

## 1.2 Contributions

The main contributions of the thesis are twofold. In the first part, we revisit the Hamming distance computation protocol SHADE of Bringer *et al.* [41]. We show that SHADE is in fact insecure in the malicious model [47]. More precisely, we show that the full scheme has a severe weakness allowing any malicious adversary to violate soundness property of the protocol, *i.e.*, a different value of Hamming distance from the actual one.

The protocol flaw resides in the method used for validation of the inputs of a user. Using zero-knowledge proofs, the protocol aims to force the user to submit valid inputs, i.e. pairs of integers  $(x, y)$  that differ by 1. The method succeeds at checking the difference, however, it fails at validation of the pairs, i.e. a malicious party can submit bogus pairs  $(\tilde{x}, \tilde{y})$  and can pass the verification steps without being detected. Since SHADE computes the Hamming distance by using the outputs of COT, a verifier would compute an incorrect Hamming distance. We would like to highlight that any fake Hamming distance can be set in advance. As a practical example for biometric authentication, we

show that a malicious adversary can pass the authentication by running the algorithm at most  $O(n)$  times (instead of running  $O(2^n)$  times, where  $n$  is the input length.). Last but not least, an adversary with knowledge of the distribution of inputs can mount a more powerful attack. Note that this attack is of independent interest and may be applied to other schemes.

In order to eliminate this severe weakness, we propose a new method for input validation. This way, we remove the fault in the protocol and enhance the security of it. We also show that the computational complexity of the fixed protocol is comparable with the insecure protocol. Moreover, we optimize the new input validation method for biometric authentication systems. We prove the security of our protocol using the ideal/real simulation paradigm in the standard model [42–44] and [45].

Lastly, we consider the efficiency of the protocol and show that running a COT is not necessary in the full scheme of the protocol. We show that VOT is sufficient instead of using complete COT protocol which contains additional commitments and zero-knowledge proofs [46]. This leads to a considerable improvement in the computational complexity of the protocol.

In the second part of this thesis, we analyze the security of the honeywords system as well as from functionality perspective. The original Honeywords system suggest some small number of honeywords per user, in general this number is 20. However, an authentication system can have some users more important than others. Thus, these important users must have more honeywords than the regular users. Following this idea, we suggested a flexible abstract method for determining the number of honeywords per user [48].

There was a problem regarding typo safety that a legitimate user may enter a honeyword instead of her password. In this case, the system will be alarmed and maybe an unnecessary safety procedures will be called. In order to avoid this false alarms, we design an algorithm for checking the typo-safety of the generated honeywords. Also, we suggested an idea for handling old passwords as they can be used by adversaries to predict the current password if they are compromised.

Finally, we introduce an enhanced honeywords model which solves the open problem of active attacks highlighted by Juels and Rivest. We proposed built-in accounts that

will work as a probabilistic watchdog system. We also design an enhanced Honeywords system which utilizes the Short Message Service as a second factor of authentication.

### 1.3 List of Publications

This thesis is partly based upon the following publications:

[47] M. S. Kiraz, Z. A. Genc, S. Kardas. Security and efficiency analysis of the Hamming distance computation protocol based on oblivious transfer. *Security and Communication Networks*, 2015. doi = 10.1002/sec.1329.

[48] Z. A. Genc, S. Kardas, M. S. Kiraz. Examination of a New Defense Mechanism: Honeywords. Cryptology ePrint Archive, Report 2013/696, 2013. <http://eprint.iacr.org/>.

### 1.4 Outline

This thesis is organized into five chapters. The first chapter provides brief introduction on secure biometric authentication and password based authentication. Then we give a literature review on secure biometric authentication methods, followed by contributions of this thesis.

**Chapter 2** introduces the cryptological concepts used throughout the thesis. We classify the password related attacks in this chapter. The security and privacy model is also given here.

**Chapter 3** is solely dedicated on SHADE protocol of Bringer *et. al.* We start by describing the two versions of the protocol, the basic scheme which is secure in the semi honest setting and the full scheme which aims full security in malicious case. We show that the full scheme is in fact insecure in the malicious case. We optimize the attack for different scenarios. Then we fix the protocol and give a security proof. Furthermore, we increase the efficiency of the protocol without decreasing the security level.

**Chapter 4** is focused on one serious problem: password database breach. We analyze the security and efficiency of the Honeywords protocol of Juels and Rivest. We find that the Honeywords system provides a good defense against password database breaches. Nevertheless, we propose improvements about the number of honeywords per user, typo-safe honeyword generation and old passwords management problem. Moreover, we introduce the Enhanced Honeywords protocol which is a solution to withstand active adversaries.

**Chapter 5** summarizes and concludes the thesis.

## Chapter 2

# Preliminaries

### 2.1 Mathematical Background

In this section we give the definitions of the cryptographic mechanisms used in this thesis.

We use the symbol  $\mathbb{Z}$  to represent the integers. For any prime number  $p$ ,  $\mathbb{Z}_p$  denotes the field of integers modulo  $p$ .  $\mathbb{Z}_p^*$  denotes the set of units in  $\mathbb{Z}_p$ , i.e.,  $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$ .

**Definition 2.1.** (Discrete Logarithm) Let  $G$  be a finite cyclic group of order  $n$ . Let  $\alpha$  be a generator of  $G$ , and let  $\beta \in G$ . The *discrete logarithm of  $\beta$  to the base  $\alpha$* , denoted  $\log_\alpha \beta$ , is the unique integer  $x$ ,  $0 \leq x \leq n - 1$ , such that  $\beta = \alpha^x$ .

**Definition 2.2.** (Discrete Logarithm Problem) The *discrete logarithm problem* (DLP) is the following: given a finite cyclic group  $G$  of order  $n$  and an element  $\beta \in G$ , find the integer  $x$ ,  $0 \leq x \leq n - 1$ , such that  $\alpha^x \equiv \beta$ .

**Definition 2.3.** (The ElGamal Cryptosystem) The ElGamal cryptosystem is a public key encryption scheme based on the discrete logarithm problem. This scheme consists of a tuple (**Key Generation**, **Encryption**, **Decryption**) defined as the following:

- **Key Generation:** Select  $a \in_R \{1, \dots, q\}$  and compute  $h = g^a$ .
- **Encryption:**
  1. Select  $r \in_R \{1, \dots, q\}$ .
  2. Compute  $(c_1, c_2) = (g^r, m \cdot B^r)$ .

- **Decryption:** Compute  $c_1^{-a} \cdot c_2$

**Definition 2.4.** (Commitment Schemes) Commitment scheme is a two party protocol between committer and a receiver. The protocol take place in two phases:

- Commit phase: The chooses a value  $m$  and a random value  $r$ , computes  $c = \text{Commit}(m, r)$  and sends  $c$  to the receiver.
- Reveal phase: The committer sends  $m$  and  $r$  to the receiver where  $m$  is the value in the commitment and  $r$  is the randomness used to compute  $c$ . Then the receiver computes the commitment and checks whether  $m$  and  $r$  commits to  $c$ .

Throughout this thesis, we will use the notation  $\text{Commit}_P(m, r)$  where  $P$  is the committer,  $m$  is the message to be committed and  $r$  is the randomness. When it is clear from the context, we shall omit  $P$  or  $r$ .

A secure commitment scheme must satisfy certain properties. First, a commitment scheme must be *hiding*, *i.e.*, the receiver cannot learn any information from the commitment  $c$ , until the point of revealing. Second, the commitment scheme must have *binding* property. That is, it should be impossible to change the committed value  $m$  unless  $c$  is changed, *i.e.*, find another value  $m'$  that commits to the same commitment value  $c$ . In other words, hiding property protects committer from malicious receivers that may try to learn  $m$  before reveal phase. On the other hand, the binding property prevents malicious verifiers from changing the committed value  $m$  to another value  $m'$  after the commitment has done.

## 2.2 Authentication

Authentication is a process in which a party proves its identity to an authority.

An authentication system consist of two parts:

- **Registration:** In this part, a party identifies itself and gives information required by the authority. This information will later be used to verify the identity of the party and must be kept secret by both sides. The type of information is dependent of the verification type used by the authority.



- **Verification:** Once a party is registered, its identity can be verified. The party submits its credentials to the authority. The authority then verifies the party based on the prior knowledge gained in the Registration part.

Basically, authentication methods is classified into three groups based on the type of information which is used for verification.

- Methods based on **something that you know**: A party proves its identity by showing an information that it knows. A typical example of this information is the login credentials that one submits to an e-mail server.
- Methods based on **something that you have**: In this kind of verification, a party uses *something* to prove its identity. User tokens and Mobile SMS based One Time Passwords (OTP) are widely used in the finance sector.
- Methods based on **something that you have**: This authentication method uses the characteristic information of a party. This information is called biometric information and includes but not limited to fingerprint and iris pattern.

### 2.2.1 COT versus VOT

Verifiable Oblivious Transfer (like COT) [49] is also a natural combination of  $\binom{2}{1}$ -OT and commitments. Let  $\text{Commit}_S$  and  $\text{Commit}_C$  be commitments by Sender and Chooser respectively. In a VOT protocol, the Sender has  $(x_0, x_1)$ , the Chooser has  $y \in \{0, 1\}$  and the commitments  $\text{Commit}_S(x_0)$ ,  $\text{Commit}_S(x_1)$ ,  $\text{Commit}_C(y)$  are common input. At the end of the protocol the Chooser learns  $x_y$  and the sender has no output. Note that the difference with COT is that commitment to the output  $x_y$  is not computed, i.e., VOT is defined if the  $\text{Commit}_C(x_y)$  is not required as output. The functionality of VOT is illustrated in Figure 2.1.

**Definition 2.5.** (Verifiable Oblivious Transfers)

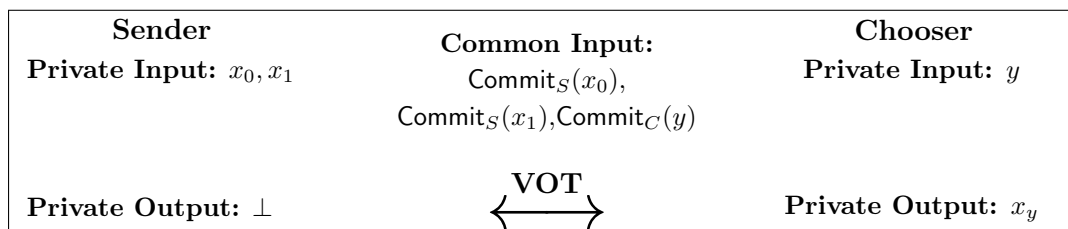


FIGURE 2.1: Verifiable Oblivious Transfer

We note here the two main aspects of COT vs. VOT:

$$\text{What to transfer} \begin{cases} \text{bits} & x_0, x_1 \in \{0, 1\} \\ \text{strings} & x_0, x_1 \in \{0, 1\}^k \end{cases}$$

$$\text{Committed Output} \begin{cases} \text{yes} \rightarrow \mathbf{COT} \\ \text{no} \rightarrow \mathbf{VOT} \end{cases}$$

We show that the basic protocol in [41] does not have to use COT in the case that the server computes the result (i.e., VOT is already sufficient because it is not necessary to compute the final commitment.).

**Definition 2.6.** (Committed Oblivious Transfers)

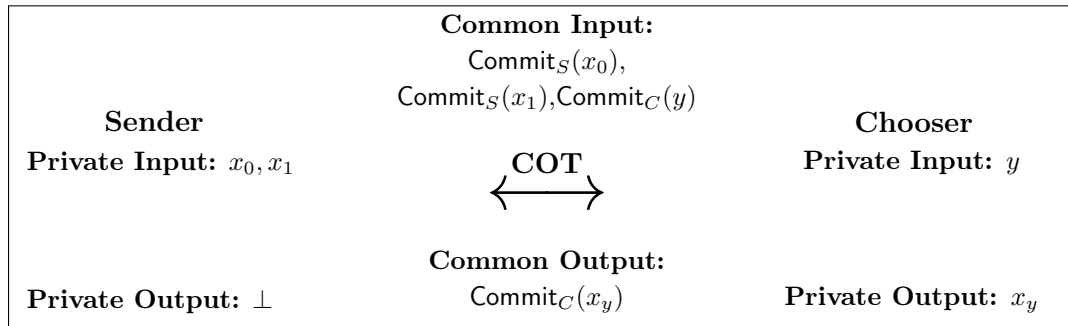


FIGURE 2.2: Committed Oblivious Transfer

## 2.3 Password Related Attacks

There are numerous attacks to obtain a user's password. Seven of these techniques are depicted in Figure 2.3.

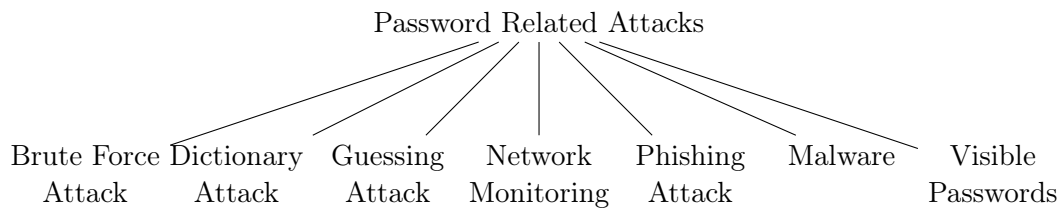


FIGURE 2.3: Password Related Attacks

Password attacks can be classified as follows:

- **Brute force attack:** In (offline version of) this scenario, the adversary steals the password hash file. She creates a set which contains the presumed characters that appear in a

password. She then creates a combination of characters from this set, computes its hash and compares the hash with the password hash. This process continues until she finds a match [50].

In order to understand the practicality of this attack consider the following scenario. A user created a password of length eight which consists of only lowercase English letters and digits, *i.e.*, a-z and 0-9. The required time to crack this password can be computed as follows

$$Time(inseconds) = \frac{\text{Password space length}}{\text{Crack speed}}$$

Applying the above formula, we find that using only one GPU, an adversary can crack the SHA1 hash of that password in

$$\frac{(26 + 10)^8}{2.7 \times 10^9 \text{ per second}} \cong 1045 \text{ seconds}$$

The above computation is based on [18].

- **Dictionary attack:** In this attack, an adversary computes the hash of words from a list that consists of strings which are typically derived from a dictionary. She compares this hash with the password hash. The intention is to try words which are more likely than a random string to be the password.
- **Guessing attack:** Many users choose weak passwords such that an adversary can find out the passwords of some users of a system by trying common passwords while attempting to login to that system [51, 52]. Spafford suggests good password choice should avoid common words and names [53].
- **Network monitoring:** If the communication between the user and the system is unsecured, *i.e.*, unencrypted, an adversary may monitor the network traffic and obtain the passwords or interrupt the traffic while a user is entering or creating her password [54]. This attack is also called man-in-the-middle-attack [55]. Most websites use TLS protocol to mitigate this attack [56].

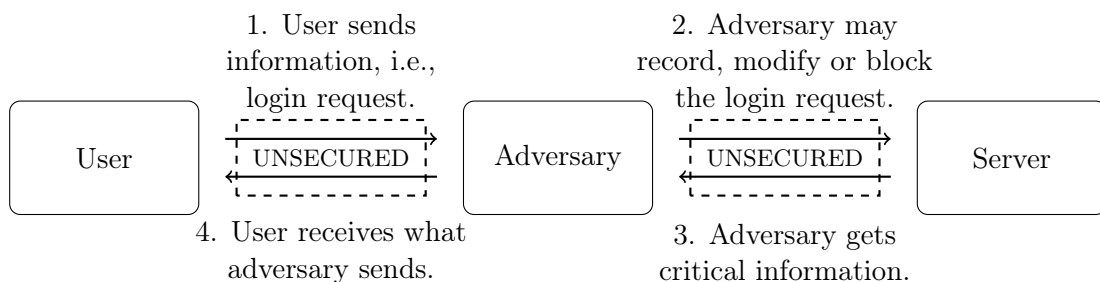


FIGURE 2.4: Communication over an unsecured channel

- **Phishing attack:** A user can be fooled to submit her login information to a web page which seems resembles the original system's login screen but is run by the adversary [57]. In addition to obtaining user credentials, this attack is often used to steal credit card information.
- **Malware:** A Trojan program can capture the key strokes and send this information to the adversary [58]. There are some advanced forms of malware that can steal the login information from messenger type software some of which does not keep the login information encrypted [59]. Sun *et al.* propose oPass which uses a user's cellphone and Short Message Service (SMS) to prevent password stealing [60].
- **Visible passwords:** A password that is written to a stickie can be seen by an adversary. He can also observe a user while she enters her password (shoulder surfing). Kumar *et al.* propose EyePassword, gaze-based password entry, to overcome direct observation [61].

## 2.4 Security and Privacy Model

We adopt the standard simulation-based definition of ideal/real security paradigm in the standard model which is already highlighted in [42–44] and [45]. In simulation-based security, the view of a protocol execution in a real setting is compared (a statistical/computational indistinguishable manner) as if the computation is executed in an ideal setting where the parties send inputs to an ideal functionality  $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$  that performs the computation and returns its result.

In an ideal setting, the parties send their inputs  $x$  and  $y$  to an ideal functionality  $\mathcal{F}$  who computes  $\mathcal{F}(x, y)$  (which is the output of the Hamming distance in our setting) and sends  $\mathcal{F}_1(x, y)$  to the first party and  $\mathcal{F}_2(x, y)$  to the second party ( $\mathcal{F}_1(x, y)$  or  $\mathcal{F}_2(x, y)$  can be  $\perp$  if only one party is required to learn the output). Note that the adversary, who controls one of the parties, can choose to send any input to the functionality  $\mathcal{F}$ , while the honest party always sends its specified input. In a real execution of a protocol  $\Pi_{\mathcal{F}}$  for a functionality  $\mathcal{F}$ , one of the parties is assumed to be corrupted under the complete control of an adversary  $\mathcal{A}$ . Note that we assume that the adversary  $\mathcal{A}$  corrupts one of the two parties at the beginning of the protocol execution and is fixed throughout the computation (as it is known as static adversary model).

Informally, a protocol  $\Pi_{\mathcal{F}}$  is secure if for every real-model adversary  $\mathcal{A}$  interacting with an honest party running the protocol, there exists an ideal-model adversary  $\mathcal{S}$  interacting with the trusted party computing  $f$ , such that the output of the adversary and the honest party in the real model is computationally indistinguishable from the output of simulator and the honest party in the ideal model. More formally,

**Definition 2.7.** (Simulation-based security) Let  $\mathcal{F}$  and the protocol  $\Pi_{\mathcal{F}}$  be as above. We say that the protocol  $\Pi_{\mathcal{F}}$  securely computes the ideal functionality  $\mathcal{F}$  if for any probabilistic polynomial-time real-world adversary  $\mathcal{A}$ , there exists a probabilistic polynomial-time ideal-model adversary  $\mathcal{S}$  (called the simulator) such that

$$\text{REAL}_{\Pi_{\mathcal{F}}, \mathcal{A}}(x, y)_{x, y \text{ s.t. } |x|=|y|} \approx \text{IDEAL}_{\mathcal{F}, \mathcal{S}}(x, y)_{x, y \text{ s.t. } |x|=|y|}$$

Note that the above definition implies that the parties already know the input lengths (by the requirement that  $|x| = |y|$ ).

Note also that VOT and COT protocols are used as subprotocols. In [62, 63], it is shown that it is sufficient to analyze the security of a protocol in a hybrid model in which the parties interact with each other and assumed to have access to a trusted third party that computes a VOT (resp. COT) protocol for them. Thus, in the security analysis of our protocol the simulator plays the role of the trusted third party for VOT (resp. COT) functionality when simulating the corrupted party. Roughly speaking, in the hybrid model, parties run an arbitrary protocol like in the real model, but have access to a trusted third party that computes a functionality (in our case VOT or COT) like in the ideal model. A protocol is secure if any attack on the real model can be carried out in the hybrid model.

## Chapter 3

# Privacy Issue of the SHADE Protocol and Efficiency Improvements

In this chapter, we focus on SHADE protocol of Bringer *et al.* [41], analyze the scheme from both security and efficiency perspective. We first describe the two versions of the protocol, the basic scheme which is secure in the semi honest setting and the full scheme which aims full security in malicious case. We show that the full scheme is in fact insecure in the malicious case, as demonstrated in [47]. We optimize the attack for different scenarios. Then we fix the protocol and give a security proof. Furthermore, we increase the efficiency of the protocol without decreasing the security level.

### 3.1 The Basic and the Full Scheme of SHADE

In this section, we briefly describe the basic and the full scheme of SHADE protocol [41] used for computation of Hamming distance between two bit strings. The basic scheme uses oblivious transfer (OT) and provides full security when the parties are semi-honest and one-sided security in the malicious model. The full scheme uses committed oblivious transfer (COT) [64] and zero-knowledge proofs of knowledge [46] to compute the Hamming distance in malicious model. Each scheme has two options to select the party which computes and outputs the result meaning that each party may act as a server and the other as a client.

### 3.1.1 The Basic Scheme

The basic scheme is designed to provide secure and efficient method for computing the Hamming distance between two bit strings in semi-honest model. The intuition behind this protocol is that if both parties are semi-honest, the OT protocols are sufficient to preserve privacy.

The basic scheme in [41] which is secure against semi-honest adversaries is as follows:

Two parties  $P_1$  and  $P_2$  are willing to compute the Hamming distance of their private inputs  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_n\}$ , respectively. At the first step,  $P_1$  randomly picks  $r_1, \dots, r_n \in_R \mathbb{Z}_{n+1}$  and computes  $R = \sum_{i=1}^n r_i$ . For  $i = 1, \dots, n$ , the parties run an OT protocol in which  $P_1$  acts as the sender and  $P_2$  acts as the receiver. More precisely,  $P_1$  inputs  $(r_i + x_i, r_i + \tilde{x}_i)$  where  $\tilde{x}_i = 1 - x_i$  and  $P_2$  inputs  $y_i$ . At the end of the OT protocol,  $P_2$  receives  $t_i = (r_i + x_i)$  if  $y_i = 0$  and  $(t_i = r_i + \tilde{x}_i)$  otherwise. Next,  $P_2$  computes  $T = \sum_{i=1}^n t_i$ . In the last step,

- **1<sup>st</sup> Option:**  $P_2$  sends  $T$  to  $P_1$ . Next,  $P_1$  outputs  $T - R$ .
- **2<sup>nd</sup> Option:**  $P_1$  sends  $R$  to  $P_2$ . Next,  $P_2$  outputs  $T - R$ .

The privacy is still guaranteed in the presence of semi-honest adversaries as they proved in Section 6 of [41]. Furthermore, the efficiency of the basic scheme of Bringer *et al.* [41] was further improved in [65]. The authors also mention that the basic scheme can be optimized by using the state of the art techniques, i.e. extended oblivious transfer, as first proposed by Ishai *et al.* in [66] and later improved in [67]. This technique leads to an efficient construction which extends  $k$  OTs to  $n$  OTs ( $k < n$ ) in the random oracle model that is secure against only semi-honest adversaries (note that hash functions can be replaced with RO model in the real case).

### 3.1.2 The Full Scheme

The full scheme of Bringer *et al.* considers the case where the parties are assumed to be malicious. Note that running OT protocol does not prevent a party from modifying her input. Secondly, the receiver may send a different value than the actual OT output that she computes. In order to prevent such scenarios, the authors propose to use the 1-out-of-2 Committed Oblivious Transfer (COT) protocol of Kiraz *et al.* presented in [64] (see Figure 2.2). Though, in Section 3.2, we show that the idea of input validation for  $P_1$  is not sufficient and can be exploited with success.

Before we proceed, let's continue with the description of the full scheme (refer to [41] for more details).

- At the first step of the protocol,  $P_2$  commits to her input bits  $y_i$ 's and proves in zero-knowledge [46] that each  $y_i$  is either equal to 0 or equal to 1.
- At the same time,  $P_1$  generates random elements  $r_i$ 's from the plaintext space of the commitment scheme and computes  $R = \sum_{i=1}^n r_i$ . Next, she commits to  $a_i$  and  $b_i$  where  $(a_i, b_i) = (r_i + x_i, r_i + \tilde{x}_i)$ <sup>1</sup>. Let's denote  $\text{Commit}(M)$  for the commitment functionality of a message  $M$ <sup>2</sup>.  $P_1$  publishes the commitments  $A_i = \text{Commit}(a_i)$  and  $B_i = \text{Commit}(b_i)$ . Furthermore, using these commitments she proves that  $a_i$  and  $b_i$  differ by 1 for each  $i$ .
- Next, the COT protocol is run for each  $i$ . At the end of each COT,  $P_2$  receives  $t_i = r_i + (x_i \oplus y_i)$  and both parties receive  $C_i = \text{Commit}(t_i)$ . When all the COTs are run,  $P_2$  computes the sum  $T = \sum_{i=1}^n t_i$ .
- At this point, there are two options:
  - **1<sup>st</sup> Option:**  $P_2$  computes  $C = C_1 \cdots C_n$ , and because of the underlying homomorphic property we have  $\text{Commit}(T) = C$  [41].  $P_2$  sends  $T$  to  $P_1$  and proves in zero-knowledge that  $C$  indeed commits to  $T$ .  $P_1$  also computes  $C = C_1 \cdots C_n$  and verifies the proof. If all verifications are successful,  $P_1$  outputs  $T - R$ .
  - **2<sup>nd</sup> Option:**  $P_1$  computes  $K = \text{Commit}(2R + n) = A_1 \cdots A_n \cdot B_1 \cdots B_n$ .  $P_1$  sends  $R$  to  $P_2$  and proves in zero-knowledge that  $K$  indeed commits to  $2R + n$ .  $P_2$  computes  $K = A_1 \cdots A_n \cdot B_1 \cdots B_n$  and verifies that  $K = \text{Commit}(2R + n)$ . If all verifications are successful,  $P_2$  outputs  $T - R$ .

The authors in [41] claim that the above scheme is fully secure against malicious adversaries. However, in the next section we show that a malicious  $P_1$  can easily break the correctness property of the scheme.

### 3.2 Security and Efficiency Analysis of SHADE

We are now ready to describe the protocol flaw of the full scheme in detail. The security flaw is due to the proof for validation of  $P_1$ 's input bits. The flaw allows a malicious  $P_1$  to change the Hamming distance between her input and  $P_2$ 's input. In the next section, we propose a solution to fix the flaw by designing a new proof for validation. We show that the complexity of the new proof for the validation of  $P_1$ 's input bits for biometric authentication systems is significantly reduced.

<sup>1</sup>The commit functionality of [64] is basically a (2,2)-threshold homomorphic encryption scheme (e.g., ElGamal [68], Paillier [69]). Let  $(pk_{P_1, P_2}, (sk_{P_1}, sk_{P_2}))$  denote public and private key pairs of the encryption scheme where  $pk_{P_1, P_2}$  is the common public key, and  $sk_{P_1}, sk_{P_2}$  are the corresponding private key shares of  $P_1$  and  $P_2$ , respectively.

<sup>2</sup>Note that because of the underlying encryption scheme  $\text{Commit}$  includes randomness and public key, and we hide them for the sake of simplicity.



Furthermore, we also analyze the protocol from the efficiency perspective and show that the complexity of the protocol can be significantly improved. COT protocol is basically designed as a sub-protocol in order to prevent possible malicious behaviors between sender and receiver, where the committed output of COT is expected to be used in further parts of the system. However, the committed outputs of COT are not used in the case that  $P_1$  computes the Hamming distance. Hence, we point out that Verifiable Oblivious Transfer is sufficient in the case that  $P_1$  computes the Hamming distance. This eliminates to compute  $n$  commitments together with the zero-knowledge proofs (for each run of COT protocol). In this way, we improve the efficiency of the protocol by using VOT instead of COT when  $P_1$  is the server.

### 3.2.1 Attack to the Full Scheme

The protocol is insecure in the case where  $P_1$  is malicious. This is because  $P_1$  is free in the sense that she can commit to any pair such that the absolute value of the difference of the encrypted values is 1, i.e.  $P_1$  proves that  $|b_i - a_i| = 1$  where the pair  $(a_i, b_i)$  is supposed to be  $(r_i + x_i, r_i + \tilde{x}_i)$ . However, a malicious  $P_1$  may choose invalid pairs in a special way together with the proofs that difference between each pair is equal to 1. Our attack uses the fact that at the end of each COT,  $P_2$  receives either  $t_i = r_i + g$  or  $t_i = r_i + h$  and computes the sum  $T = \sum_{i=1}^n t_i$ , where  $g, h$  are within the finite cyclic group. Note that  $g$  is expected to be equal to  $x_i$  and  $h$  to  $\tilde{x}_i$ . However, with a careful choosing of  $g$ 's and  $h$ 's, some  $g$ 's can be neutralized by some  $h$ 's in this sum. Hence, the soundness property of the protocol can be violated. In fact, the security proof of [41] does not explicitly use the zero-knowledge proof of the statement leading to the flaw in their security analysis.

Before we describe the attack it is important to highlight that the underlying COT scheme uses threshold ElGamal encryption as a commitment mechanism, i.e.  $\text{Commit}(x_i) = \text{Enc}(x_i)$  where  $x_i \in G$  where  $G$  is a large finite cyclic group (of a prime order) [64]. This guarantees the existence of the inverse of  $n$ .

Without loss of generality assume that  $\#0$ 's in  $P_2$ 's input  $Y$  is  $\ell$  (i.e.,  $\#1$ 's in  $Y$  is  $n - \ell$ ). A predetermined fake Hamming distance can be computed with the knowledge of  $\#0$ 's (similarly  $\#1$ 's) in  $P_2$  as follows: a malicious  $P_1$  uses  $(a_i, b_i) = (r_i + g, r_i + h)$  for an arbitrary Hamming distance  $\text{HD} = \ell g + (n - \ell)h$  such that  $g - h = 1$ , where  $g, h$  are the group elements. Then,

$$\text{HD} = \ell g + (n - \ell)(g - 1) = ng - n + \ell.$$

For an example, if a malicious  $P_1$  desires Hamming distance HD to be 0 then she chooses  $g = 1 - \ell n^{-1}$ . Next,  $h = g - 1 = -\ell n^{-1}$ . Hence,  $P_1$  may use  $(a_i, b_i) = (r_i + (1 - \ell n^{-1}), r_i - \ell n^{-1})$  as input. To be more concrete, the attack is given as follows:

- $P_2$  commits to her inputs  $y_i$ 's and proves that each  $y_i$  is either 0 or 1.  $P_1$  then generates random  $r_i$ 's and computes  $R = \sum_{i=1}^n r_i$ .
- Next, instead of following the protocol,  $P_1$  computes  $(a_i, b_i) = (r_i + (1 - \ell n^{-1}), r_i - \ell n^{-1})$  and publishes  $A_i = \text{Commit}(a_i)$  and  $B_i = \text{Commit}(b_i)$ . Note that for each  $i$ ,  $|b_i - a_i| = 1$  and hence, the proofs pass successfully.
- At the end of each COT,  $P_2$  receives either  $t_i = r_i + (1 - \ell n^{-1})$  or  $t_i = r_i - \ell n^{-1}$ . After COTs are run,  $P_2$  computes the sum

$$\begin{aligned}
T &= \sum_{i=1}^n t_i \\
&= \sum_{i|y_i=0} (r_i + (1 - \ell n^{-1})) + \sum_{i|y_i=1} (r_i - \ell n^{-1}) \\
&= \ell(1 - \ell n^{-1}) + (n - \ell)(-\ell n^{-1}) + \sum_{i=1}^n r_i \\
&= \sum_{i=1}^n r_i \\
&= R.
\end{aligned}$$

Therefore, the Hamming distance  $d_H(X, Y) = T - R$  is equal to 0. We stress that the weakness in the scheme is destructive as we prove that a relatively insignificant information leakage causes computation of a completely inaccurate result. Namely, without knowledge of the real  $X$ ,  $P_1$  fools  $P_2$  into outputting an incorrect Hamming distance value without being detected. Furthermore, a malicious  $P_1$  with the prior knowledge of  $\ell$  is capable of manipulating HD by computing the values  $g$  and  $h$  using the above-mentioned equation. This is interesting because Hamming distance is not necessarily equal to 0 or 1. For example, in [70], the authors propose a privacy-preserving protocol for iris-based authentication using Yao's garbled circuits. They show that Hamming distance between two iris codes owned by the same person is rarely close to 0 (and similarly it is rarely close to  $n$  for different persons). Therefore, the scalability feature of our attack can be easily adopted to various general settings.

In this part, we propose the most general case and in the next section we give a practical attack for biometric authentication schemes reducing the computational complexity of an attacker from  $O(2^n)$  to  $O(n)$ , where  $n$  is the input length. Namely, an attacker without any prior knowledge can authenticate herself using only  $n$  trials instead of  $2^n$ .

### 3.2.2 A Special Case: Apply the Attack to Biometric Authentication Systems

In the previous section, we described the most general case, i.e., for any system that uses the proposed Hamming distance protocol. We now apply the proposed attack as a practical example on biometric authentication systems with full success. Note that the matching procedure for fingerprint, palm print or iris actually measures the Hamming distance between the two bit-strings  $X$  and  $Y$  that encode the biometric sample and template (e.g., [7, 36, 71]).

The attack basically consists of  $n$  runs of the proposed attack method to successfully authenticate to the system, where  $n$  is the input length. In general, for an  $n$ -bit string  $Y = (y_1, \dots, y_n)$ , an attacker must roughly try  $2^n$  search for  $X$  to pass the authentication successfully and it is infeasible for large  $n$ . However, using the proposed attack a corrupted  $P_1$  can authenticate the system after at most  $n$  trials (because the number of 0s or 1s in  $Y$  is between 0 and  $n$ , i.e.,  $0 \leq \ell \leq n$ ). More precisely, starting  $\ell = 1$  until  $\ell = n$  a corrupted  $P_1$  runs the proposed attack method, and because  $0 \leq \ell \leq n$  the authentication is successful with at most  $n$  trials (without any knowledge of the real input  $X$ ).

### 3.2.3 Apply the Generic for Uniformly Distributed Inputs

This attack can also be directly applied to uniformly distributed bit strings  $X$  and  $Y$ . In this scenario the input bit-strings of  $P_2$  (which is generated from a biometric template) is expected to be independent and identically distributed. That is, there are nearly equal number of zeros and ones in an input bit string. Below, we show that this fact easily allows an adversary to minimize the Hamming distance and successfully deceive a verifier:

1.  $P_2$  commits to her inputs  $y_i$ 's and proves that each  $y_i$  is either 0 or 1.
2.  $P_1$  picks random  $r_i$ 's and computes  $R = \sum_{i=1}^n r_i$ .
3. Instead of computing  $(a_i, b_i) = (r_i + x_i, r_i + \tilde{x}_i)$ ,  $P_1$  computes  $(a_i, b_i) = (r_i - 2^{-1}, r_i + 2^{-1})$  in order to make the commitments  $A_i = \text{Commit}_{P_{1,i}}(a_i)$  and  $B_i = \text{Commit}_{P_{1,i}}(b_i)$ . The authors in [41] uses homomorphic encryption as the commitment mechanism. Since those cryptosystems work in a group of prime order, the multiplicative inverse of 2 always exists, i.e.  $P_1$  can commit to  $(a_i, b_i) = (r_i - 2^{-1}, r_i + 2^{-1})$ . Next  $P_1$  proves that  $|b_i - a_i| = 1$  which always holds. Note that  $P_1$  does not prove the validity of her input, i.e, she does not prove that the  $x_i$ 's are equal to either 0 or 1.
4. COTs are run, and in one half of the COTs (because of the uniform distributed inputs),  $P_2$  receives  $t_i = r_i - 2^{-1}$  and  $t_i = r_i + 2^{-1}$  in the other half.

5.  $P_2$  computes  $T = \sum_{i=1}^n t_i$ . Since  $y_i$ 's are equally distributed, i.e. the numbers of 0s and 1s in  $\{y_1, \dots, y_n\}$  are nearly equal,  $P_2$  computes  $T = \left( \sum_i r_i + 2^{-1} \right) + \left( \sum_i r_i - 2^{-1} \right) = \sum_{i=1}^n r_i = R \pm k2^{-1}$  for some small  $k \ll n$ .
6. Using the  $2^{nd}$  option,  $K = \text{Commit}_{P_{2,i}}(2R + n) = A_1 \cdots A_n \cdot B_1 \cdots B_n$ .
7.  $P_1$  sends  $R$  and the proof that  $K$  commits to  $2R + n$  to  $P_2$ .
8.  $P_2$  computes  $d_H(X, Y) = T - R = k$  where  $k \ll n$  and successfully authenticates  $P_1$  since  $k$  will be less than the threshold value.

### 3.2.4 Our Solution for the Attack

The weakness of the full scheme is due to the zero-knowledge proof of a wrong statement used for validation of the input pairs  $\{(a_i, b_i), \forall i = 1, \dots, n\}$ . A malicious  $P_1$  can easily exploit this weakness as described in the previous section. Therefore, logical statements of zero-knowledge proofs should be carefully checked against these kinds of adversarial behaviors.

As a security fix, we modify the step in which  $P_1$  generates random  $r_i$  values. Namely, after generating each  $r_i$ ,  $P_1$  computes and publishes  $A_i = \text{Commit}(r_i + x_i)$ ,  $B_i = \text{Commit}(r_i + \bar{x}_i)$  and  $R_i = \text{Commit}(r_i)$ . Next,  $P_1$  sends the zero-knowledge proof of the following statement

$$((a_i - r_i) = 0 \vee (b_i - r_i) = 0) \wedge |b_i - a_i| = 1$$

that is equivalent to

$$(a_i + b_i - 2r_i = 1) \wedge |b_i - a_i| = 1$$

using the commitments  $A_i, B_i$  and  $R_i$ . This new statement contains one more relation than the one in the original proof of [41]. Although the computation cost of the protocol is slightly increased, the validation process now assures the security of the protocol.

Note that if the new statement  $(a_i + b_i - 2r_i = 1) \wedge |b_i - a_i| = 1$  is true then only one of the following two cases can occur:

$$a_i = b_i + 1 \Rightarrow 2b_i + 1 - 2r_i = 1 \Rightarrow b_i = r_i, a_i = r_i + 1$$

$$b_i = a_i + 1 \Rightarrow 2a_i + 1 - 2r_i = 1 \Rightarrow a_i = r_i, b_i = r_i + 1$$

In Section 3.4 we provide the security analysis of the improved scheme.

### 3.2.4.1 More Efficient Solution for Biometric Authentication

Biometric authentication systems are designed to tolerate a small level of errors. In general, the measure process is not perfect in most environments and thus, instead of exact match, a biometric system authenticates a party that matches with a small error to prevent false negatives.

The authentication process must also have a small complexity to compute the result in the fastest way. Therefore each party must prove nothing more than the necessary and sufficient data for validation of her input.

These motivations lead us to design a more efficient proof that can be used in the biometric authentication systems. Namely, after generating and publishing the commitments to  $a_i, b_i, r_i$  as in the previous section,  $P_1$  sends the proof of:

$$a_i + b_i - 2r_i = 1.$$

The above relation has a smaller complexity than  $|b_i - a_i| = 1$  while it still provides higher security. This input validation method is an efficient solution for our attack in the case of biometric authentication. Note that an adversary may input  $(a_i, b_i) = (r_i - 2^{-1}, r_i + 2^{-1})$  and pass the validation but its Hamming distance is  $\frac{n}{2}$  which is the expected value of Hamming distance between two random inputs with length  $n$ .

## 3.2.5 Efficiency Enhancements

In this section, we present some improvements for the efficiency of the protocol. First, we reduce the computational complexity of the protocol using VOT instead of COT without sacrificing the security. Namely, COT is not necessary in the case where  $P_2$  computes the final Hamming distance. Next we reduce the complexity of the proof for the validity of  $P_1$ 's inputs in the case of biometric authentication.

### 3.2.5.1 Efficiency Improvement Using VOT

In this section, we point out a computational complexity reduction. Note that COT is run for the malicious case in [41]. COT requires the receiver to obtain the output together with its commitment to this value. In the beginning of the protocol, the input of  $P_1$  is an  $n$ -bit string  $X = (x_1, \dots, x_n)$  and the input of  $P_2$  is an  $n$ -bit string  $Y = (y_1, \dots, y_n)$ . After running the protocol there are two options:

- $P_1$  obtains the Hamming distance  $d_H(X, Y)$  and  $P_2$  obtains nothing

- $P_2$  obtains the Hamming distance  $d_H(X, Y)$  and  $P_1$  obtains nothing

In case  $P_2$  computes the Hamming distance, the committed values from the output of COT is not used. In such case, these commitments are not necessary to be computed, and therefore VOT is sufficient to use. We realized this observation after writing the COT protocol explicitly with the overall protocol instead of using as a black box. If  $P_1$  computes the Hamming distance COT is still necessary to use.

### 3.3 Our Fixed and Improved Scheme

We made the modifications to the full scheme of [41] in order to fix the security weakness described in Section 3.2 and improve the efficiency of the protocol as mentioned in Section 3.2.5. Now, we give the corrected scheme with all details:

#### Inputs:

- $P_1$  inputs an  $n$ -bit string  $X = (x_1, \dots, x_n)$
- $P_2$  inputs an  $n$ -bit string  $Y = (y_1, \dots, y_n)$

#### Outputs:

- 1<sup>st</sup> Option:  $P_1$  obtains  $d_H(X, Y)$  and  $P_2$  obtains nothing
- 2<sup>nd</sup> Option:  $P_2$  obtains  $d_H(X, Y)$  and  $P_1$  obtains nothing

#### Protocol:

1.  $P_2$  commits to her inputs  $y_i$ 's and proves that each of  $y_i$  is either 0 or 1.
2.  $P_1$  generates random  $r_i$ 's from the plaintext space of Commit and computes  $R = \sum_{i=1}^n r_i$ .
3.  $P_1$  commits to  $(a_i, b_i, r_i) = (r_i + x_i, r_i + \tilde{x}_i, r_i)$ .  $P_1$  publishes  $A_i = \text{Commit}(a_i)$ ,  $B_i = \text{Commit}(b_i)$  and  $R_i = \text{Commit}(r_i)$ .
4.  $P_1$  proves that  $(|a_i - r_i| = 0 \vee |b_i - r_i| = 0) \wedge |b_i - a_i| = 1$  using  $A_i, B_i$  and  $R_i$ .
5. For each  $i = 1, \dots, n$ , a COT is run where
  - $P_1$  acts as the sender and  $P_2$  as the receiver.
  - $P_2$ 's selection bit is  $y_i$ .
  - $P_1$ 's input bit is  $(a_i, b_i)$ .
  - The output obtained by  $P_2$  is  $t_i = r_i + (x_i \oplus y_i)$ .

- Both parties obtain  $C_i = \text{Commit}(t_i)$ .
6.  $P_2$  computes  $T = \sum_{i=1}^n t_i$
  7. **1<sup>st</sup> Option: Run VOT**
    - (a)  $P_1$  computes  $K = \text{Commit}(2R + n) = A_1 \cdots A_n \cdot B_1 \cdots B_n$ .
    - (b)  $P_1$  sends  $R$  to  $P_2$  and proves that  $K$  commits to  $2R + n$ .
    - (c)  $P_2$  computes  $K = A_1 \cdots A_n \cdot B_1 \cdots B_n$  and checks that  $K = \text{Commit}(2R + n)$ .
    - (d) If all verifications are successful,  $P_2$  outputs  $T - R$ .
  8. **2<sup>nd</sup> Option: Run COT**
    - (a)  $P_2$  computes  $C = \text{Commit}(T) = C_1 \cdots C_n$ .
    - (b)  $P_2$  sends  $T$  to  $P_1$  and proves that  $C$  commits to  $T$ .
    - (c)  $P_1$  computes  $C = C_1 \cdots C_n$  and verifies the proof.
    - (d) If all verifications are successful,  $P_1$  outputs  $T - R$ .



FIGURE 3.1: Our Improved Scheme

### 3.4 Security Analysis of Our Scheme

A cryptographic protocol is secure if the view of an adversary in a real protocol execution can be generated from the information the adversary has (i.e., its input and output). In this section, we

proved the security of the proposed protocol by constructing a simulator, which is given only the input and output of the “corrupted” party, and generating a view that is indistinguishable from the view of the adversary in a real protocol execution [42–45]. This implies that the adversary learns no information from the real protocol because it could generate anything from what it sees in such an execution by itself.

**Theorem 3.1.** *The proposed protocol, which is shown in Figure 3.1, is secure in the presence of static malicious adversaries.*

*Proof.* We show that given a party is corrupted, there exists a simulator that can produce a view to the adversary that is statistically indistinguishable from the view in the real protocol execution based on its private decryption share as well as public information.

*Case-1- $P_1$  is corrupted.* Let  $\mathcal{A}_{P_1}$  be an adversary corrupting  $P_1$ . We construct a simulator  $\mathcal{S}_{P_1}$  and show that the view of the adversary  $\mathcal{A}_{P_1}$  in the simulation with  $\mathcal{S}_{P_1}$  is statistically close to its view in a hybrid execution of the protocol with a trusted party running the VOT (resp. COT) protocol. Since we assume that the VOT (resp. COT) protocol is secure, we analyze the security of the protocol in the hybrid model with a trusted party computing the VOT (resp. COT) functionality. Note that the simulator  $\mathcal{S}_{P_1}$  knows  $X, sk_{P_1}$  for the 1<sup>st</sup> option where VOT is run (in the 2<sup>nd</sup> the simulator also knows  $d_H(X, Y)$ ). The simulator proceeds as follows:

1.  $\mathcal{S}_{P_1}$  picks arbitrary  $\tilde{Y} = \tilde{y}_1 \dots \tilde{y}_n$  and computes  $\text{Commit}_{P_{2,i}}$ .  $\mathcal{S}_{P_1}$  can simulate the proofs since it knows the committed input values  $\tilde{y}_i$ 's and  $sk_{P_1}$ .
2. In case of VOT is run:
  - (a)  $\mathcal{S}_{P_1}$  first extracts the input of  $R_{P_1}$  from VOT functionality in the hybrid model, then sends the input to the trusted party and learns the output value  $\tilde{t}_i$ .
  - (b)  $\mathcal{S}_{P_1}$  computes  $\tilde{T} = \sum_{i=1}^n \tilde{t}_i$  and computes  $\text{Commit}_{P_{2,i}}(2R + n) = \prod_{i=1}^n A_i B_i$  as in the real protocol.

In case of COT is run:

- (a)  $\mathcal{S}_{P_1}$  first extracts the input of  $\mathcal{R}_{P_1}$  from COT functionality in the hybrid model, then sends the input to the trusted party and learns the output value  $\tilde{t}_i$  and  $\tilde{C}_i = \text{Commit}(\tilde{t}_i) \forall i = 1, \dots, n$ .
- (b)  $\mathcal{S}_{P_1}$  computes  $\tilde{T} = \sum_{i=1}^n \tilde{t}_i$  and  $\text{Commit}(\tilde{T}) = \prod_{i=1}^n \tilde{C}_i$  as in the real protocol.
- (c)  $\mathcal{S}_{P_1}$  can simulate the proof since it knows the committed input value  $\tilde{T}$ 's,  $d_H(X, Y)$  and  $sk_{P_1}$ .



Consequently, each step of the proposed authentication protocol for the simulator is simulated and this completes the simulation for the malicious verifier. The transcript is consistent and statistically indistinguishable from the verifier's view when interacting with honest  $P_2$ .

*Case-2- $P_2$  is corrupted.* Let  $\mathcal{A}_{P_2}$  be an adversary corrupting  $P_2$ , we construct a simulator  $\mathcal{S}_{P_2}$  as follows. Since we assume that the COT (resp. VOT) protocol is secure, we analyze the security of the protocol in the hybrid model with a trusted party computing the COT (resp. VOT) functionality. Note that the simulator  $\mathcal{S}_{P_2}$  knows  $Y = y_1 \dots y_n, sk_{P_2}$  and  $d_H(X, Y)$  for the 1<sup>st</sup> option where VOT is run (in the 2<sup>nd</sup> the simulator does not know  $d_H(X, Y)$ ). The simulator proceeds as follows:

1.  $\mathcal{S}_{P_2}$  picks arbitrary  $\tilde{X} = \tilde{x}_1 \dots \tilde{x}_n$ .
2.  $\mathcal{S}_{P_2}$  picks  $\tilde{r}_i \in_R \mathbb{Z}_q^*$  and computes  $\tilde{R}_{P_2} = \sum_{i=1}^n \tilde{r}_i$ . Next,  $\mathcal{S}_{P_2}$  computes  $(\tilde{a}_i, \tilde{b}_i) = (\tilde{r}_i + \tilde{x}_i, \tilde{r}_i + \tilde{x}_i) \forall i = 1 \dots n$ .  $\mathcal{S}_{P_2}$  computes  $\tilde{A}_i, \tilde{B}_i$  and  $\tilde{R}_i$  as in the real protocol.  $\mathcal{S}_{P_2}$  can again simulate the proofs since he knows the committed input values and  $sk_{P_2}$ .
3. In case VOT is run:
  - (a)  $\mathcal{S}_{P_2}$  first extracts the input of  $\mathcal{R}_{P_1}$  from VOT functionality in the hybrid model and then sends the input to the trusted party.  $\mathcal{S}_{P_2}$  next computes  $\tilde{K} = \text{Commit}_{P_{2,i}}(2\tilde{R} + n)$ .  $\mathcal{S}_{P_2}$  can simulate the proof since it knows the committed input value  $R$ ,  $d_H(X, Y)$  and  $sk_{P_2}$ .

In case COT is run:

- (a)  $\mathcal{S}_{P_2}$  first extracts the input of  $\mathcal{R}_{P_1}$  from COT functionality in the hybrid model and then sends the input to the trusted party and learn  $C_i \forall i = 1, \dots, n$ .  $\mathcal{S}_{P_2}$  computes  $\text{Commit}(\tilde{T}) = \prod_{i=1}^n \tilde{C}_i$ .

Consequently, each step of the proposed authentication protocol for the simulator is simulated and this completes the simulation for the malicious verifier. The transcript is consistent and statistically indistinguishable from the verifier's view when interacting with honest  $P_1$ .  $\square$

### 3.5 Complexity Analysis Of Our Fixed Protocol

In this section, we analyze the computational complexity of our fixed protocol and compare it with the full scheme of Bringer *et al.* [41]. In our protocol, the number of invoked zero-knowledge proofs and multiplication of ciphertexts remain the same. However, we improved the efficiency of the protocol significantly by replacing  $n$  COTs with  $n$  VOTs in the second option of the protocol

where  $P_2$  computes the final Hamming distance. In this way, we show that  $n$  commitments,  $2n$  partial decryptions and  $2n$  ZK proofs can be removed. The number of commitments of  $P_1$  is increased from  $2n$  to  $3n$  in order to guarantee the validity of  $P_1$ 's inputs. This is the price that should be paid to make the protocol secure. The complexity comparison of the full scheme of Bringer *et al.* [41] and our fixed protocol is illustrated in Figure 3.1.

TABLE 3.1: Complexity Comparison

	Scheme of Bringer <i>et al.</i>		Our Fixed Scheme	
	$P_1$	$P_2$	$P_1$	$P_2$
Commitments	$2n$	$n$	$3n$	$n$
ZK proofs	$n$			
OTs	$n$ COTs		1 <sup>st</sup> opt: $n$ COTs 2 <sup>nd</sup> opt: $n$ VOTs	
Multiplication of ciphertexts	1 <sup>st</sup> opt: $n$ 2 <sup>nd</sup> opt: $2n$			

Our analysis shows that the additional cost of the security fix is only  $n$  commitments made by  $P_1$ , independent of the party which computes the final Hamming distance. However, in the case that  $P_2$  computes the final Hamming distance, the computational savings that can be achieved by replacing the  $n$  COTs with  $n$  VOTs are far larger. In general, a COT protocol requires one more flow than a VOT protocol in which the chooser recommitments to its received value and proves that the new commitment equals to her previous committed input. In particular, the full scheme in [41] uses the COT scheme of [64] where each run of a COT protocol requires one commitment, two partial decryption of a ciphertext and two zero-knowledge proofs in addition to a VOT protocol. As a result, we avoid unnecessary use of two zero-knowledge proofs and two partial decryptions. Consequently, we improve the efficiency of the protocol significantly while we establish the security of the protocol.

## Chapter 4

# Enhanced Honeywords System

In this chapter, we analyze the security and efficiency of the Honeywords protocol of Juels and Rivest. We find that the Honeywords system provides a good defense against password database breaches. Nevertheless, we propose improvements about the number of honeywords per user, typo-safe honeyword generation and old passwords management problem. Moreover, we introduce the Enhanced Honeywords protocol which is a solution to withstand active adversaries [48].

### 4.1 Honeywords System

Juels and Rivest propose a method for detecting password breaches and improving the security of hashed passwords. The proposed system, *Honeywords*, designed against brute-force and dictionary attacks, where an adversary has stolen the file of user names and associated password hashes from a server (see Figure 2.3) [72]. The adversary has also obtained salt values and other required parameters for computing the hash function. In this scenarios, the adversary can make a brute-force or a dictionary search to find one or more user's password (*i.e.*, the adversary can crack most of the hashes). The authors also assume that authentication can only be handled using passwords while logging into the server and the adversary leaves the system after stealing the password hashes, *i.e.*, does not monitor submitted passwords.

In [72], Juels and Rivest proposed the idea of changing the structure of the password file in such a way that each user is associated with a set of possible passwords, called *sweetwords* in which only one of them is real. The false passwords are called *honeywords*. As soon as a honeyword is submitted in the login process, it is detected that the password file has been stolen and the adversary has computed the inverse of a hash from that file. Hence, in this way the system can easily detect malicious login attempts. More concretely, the honeyword system works roughly as follows.

Let  $u_i$ ,  $p_i$  and  $\mathcal{H}()$  denote the  $i^{\text{th}}$  user name, her  $i^{\text{th}}$  password and the hash function of the system, respectively. As demonstrated in Figure 4.1, the system adds honeywords' hashes to this file at random positions. Thus, an adversary who has cracked the password hashes will see randomly ordered sweetwords  $w_{i,j}$  of user  $u_i$  where  $1 \leq i \leq m$  and  $1 \leq j \leq n$ .

Username	Password's Hash	→	Username	Password's Hash
$u_1$	$H(p_1)$		$u_1$	$H(w_{1,1}), H(w_{1,2}), \dots, H(w_{1,n})$
$u_2$	$H(p_2)$		$u_2$	$H(w_{2,1}), H(w_{2,2}), \dots, H(w_{2,n})$
$\dots$	$\dots$		$\dots$	$\dots$
$u_m$	$H(p_m)$		$u_m$	$H(w_{m,1}), H(w_{m,2}), \dots, H(w_{m,n})$

FIGURE 4.1: Password database of a standard system is on the left and password database of a honeword system is on the right, where  $m, n$  denotes the number of users and sweetwords, respectively.

When the user  $u_i$  sends a login request, the login server will determine her order among the users, and the order of the submitted string among her sweetwords. If the submitted value is not equal to any of the sweetwords of the user, then the login server handles this situation as a wrong password submission. Otherwise, the login server sends a message of the form  $\text{Check}(i, j)$  to a secure server for  $i^{\text{th}}$  user and her  $j^{\text{th}}$  sweetword. This message has the following meaning: “Is the  $i^{\text{th}}$  user's password in the  $j^{\text{th}}$  position among her sweetwords?”. The secure server, which is called “honeychecker”, will determine whether the submitted order is correct or not. If the order of the submitted sweetword is wrong, then honeychecker will raise an alarm or take an action that is previously chosen as illustrated in Figure 4.2. Note that the honeychecker cannot know anything about the users' passwords or honeywords because passwords and honeywords are never sent to the honeychecker. The honeychecker maintains a single database that contains for each user only the order of the true password among the user's sweetwords.

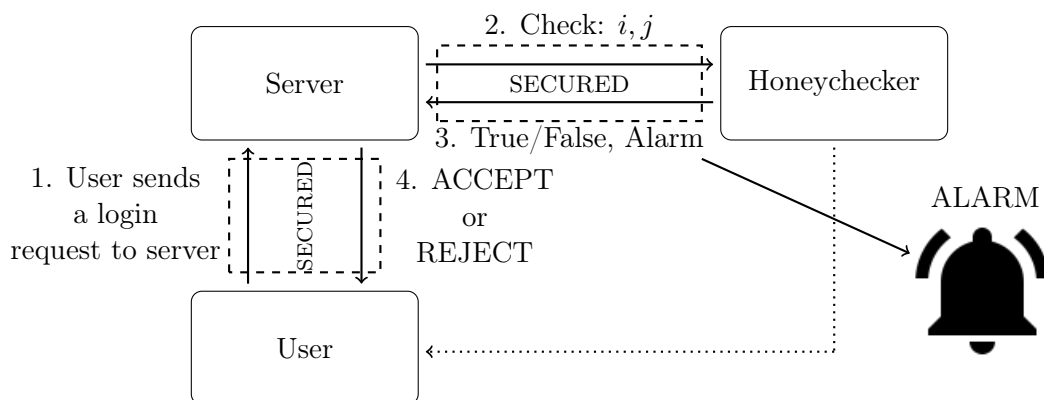


FIGURE 4.2: Login schema of a system using honeywords.

The adversary can still steal the file of hashed passwords and invert the hashes. Nonetheless, if the honeywords are carefully generated, *i.e.*, honeywords cannot be distinguished from the real password, the adversary cannot tell which sweetword is the real password. Since it is more likely

to submit a honeyword rather than the real password, and with high probability this will prevent the adversary from logging into the system.

A closely related work to honeywords is the *Kamouflage* system of Bojinov *et al.* [73] but that work differs from honeywords. In that system, password list of the user is placed with another lists that contain honeywords. When the user tries to access the password list, she provides a master password, which is then transformed into the index of the correct list. There is no need for a server in Kamouflage system although the authors in [73] note that servers might be used to empower the ability of detection of compromise.

## 4.2 Improvements for Honeywords System

In this section, we propose our practical improvements for the honeyword system which consist of four distinct solutions. The first three solutions are proposed to make the system more robust in the case that the adversary obtains the password hashes and leaves the system. Our last solution is related to an open problem mentioned in [72], *i.e.* where we deal with an active attack scenario to the honeyword system.

### 4.2.1 Number of Honeywords

Rivest and Juels recommends a small integer  $k = 20$  for the number of honeywords per-user. They note that, though, the number of honeywords does not need to be a system wide parameter. But how do we assess a user's importance and determine an appropriate number for honeywords of her? And, more importantly, how should we maintain this number for each user?

Instead of having constant number of honeywords per-user, the number of users' honeywords should be dynamic once there is an active attack. Namely, the system should generate more honeywords for users who were previously attacked. Our suggestion comes from the following fact: A user whom honeyword is submitted is more likely to be the target of an adversary than users whose honeywords are never submitted. The system should be setup in such a way that the password of this user is reset, her new honeywords are generated and the honeychecker is updated accordingly. The honeywords should be renewed after every attack and in order to decrease the success probability of the attacker. The number of honeywords for each user should be bounded with a certain security level in order to prevent denial-of-service (DoS) attacks.

This technique will deter the adversary to attack the same user again, because the success chance of the adversary will decrease in each unsuccessful attack.

### 4.2.2 Typo-Safe Honeyword Generation

The honeyword generation method called “*chaffing-by-tweaking*” tweaks the selected character positions of the password to obtain a honeyword [72]. This technique is easy to implement on the existing systems since it does not require any change in the login screen. However, since the honeywords differ from the password in a few characters, a legitimate user may submit a honeyword mistakenly and set off an alarm.

There is another honeyword generation algorithm called “*take-a-tail*” which generates honeywords by adding random -generally three digit- integers at the end of the password. As the authors [72] propose, error detection codes can be used to detect typos. Namely, difference of two tails is required to be a multiple of a small prime  $q$  greater than 10, *i.e.*,  $q = 13$ .

We generalize this idea to all tweaking methods as follows. First, a *honeyword* is generated by honeyword generation function,  $\text{Gen}(\text{password}, \text{genParams})$  where *password* is a user’s password and *genParams* denotes honeyword generation parameters of the system. Once a *honeyword* is generated, a new function  $\text{Eval}(\text{password}, \text{honeyword})$  evaluates the typo-safety of the honeyword considering the users keyboard scheme. In this setting, a honeyword which contains a character that is close to, *i.e.*, right or left to, the corresponding character of user’s password gets a lower score. If the honeyword’s typo-safe score is lower than *threshold*, *i.e.*, the minimum required distance to the password, then the generation procedure generates a new honeyword. Otherwise, the procedure outputs the honeyword. The pseudocode of the algorithm is given in Figure 4.3.

---

**Algorithm 4.2.1:** TYPO-SAFE(*password*)

---

```

global threshold, genParams
honeyword  $\leftarrow$   $\text{Gen}(\text{password}, \text{genParams})$ 
while  $\text{Eval}(\text{honeyword}, \text{password}) < \text{threshold}$ 
  do honeyword  $\leftarrow$   $\text{Gen}(\text{password}, \text{genParams})$ 
output (honeyword)

```

---

FIGURE 4.3: Typo-Safe Honeyword Generation Algorithm.

We give an implementation of  $\text{Eval}()$  function for the English (en-US) keyboard layout. It is assumed that the password contains only lower case letters (a-z) and digits (0-9). The implemented function handles honeywords which is generated from the same set. We also assume that the  $\text{Gen}$  function generates honeywords that has the same character length with the password.

---

```

private double  $\text{Eval}(\text{string } \text{honeyword}, \text{string } \text{password})$ 
{

```

```
List<List<char>>> rows = new List<List<char>>>();

rows.Add("1234567890".ToList<char>());
rows.Add("qwertyuiop".ToList<char>());
rows.Add("asdfghjkl".ToList<char>());
rows.Add("zxcvbnm".ToList<char>());

double distance = 0;

for (int i=0; i<password.Length; i++)
{
    int px = 0; int py = 0;
    int hx = 0; int hy = 0;

    for (int j=0; j <rows.Count; j++)
    {
        if (rows[j].IndexOf(password[i]) >= 0)
        {
            px = rows[j].IndexOf(password[i]);
            py = j;
        }

        if (rows[j].IndexOf(honeyword[i]) >= 0)
        {
            hx = rows[j].IndexOf(honeyword[i]);
            hy = j;
        }
    }

    distance += Math.Sqrt(Math.Pow((px - hx), 2) + Math.Pow((py - hy), 2));
}

return distance;
}
```

---

Eval() function computes the distance of a honeyword to a password using Euclidean Distance between two keys of characters on the keyboard. The characters are modeled as points in the  $x$ - $y$  plane and the  $y$  axis increases downwards. The origin is the digit 1, the left most character

which can appear in the password. The distance is computed as follows: for each character in the password, the Euclidean Distance to the corresponding character in the honeyword is computed and added to the previous sum.

### 4.2.3 Old Passwords Problem

Most clients use the same passwords on different systems. An old password of a user on some system may be the current password of that user on another system. Thus, taking advanced security countermeasures may not guarantee the safety of the passwords. Namely, an adversary may attack to a weaker system that the targeted user have an account on it and obtain her old passwords and submit them on a more secure system. Juels and Rivest give an effective solution to this problem where instead of storing old passwords per-user basis the system will store all user's old passwords in a list anonymously. When a password is created, system checks whether this list contains the password. If it is in the list, the system will not allow that password to be used. However, this solution will not be user-friendly since it is rather strange to forbid to use a password just because of somebody else used it before.

Juels and Rivest also propose to encrypt and keep old passwords per-user basis on the actual system and keep the encryption keys in the honeychecker. When needed, the system asks the honeychecker for that user's old passwords' key. This seems to be a good solution. However, this method increases the complexity of the system because the honeychecker has to perform more computation, needs more storage, and accepts new type of commands which contradicts the simplicity of the honeychecker.

We offer an another to solve this issue. In our solution the system generates honeywords for old passwords, "*old honeywords*", as well. The system will generate old honeywords and keep their hashes with old passwords' hashes per-user basis in random order. Note that in this setting, old honeywords and old passwords are stored randomly and the honeychecker does not know any information about the old passwords.

## 4.3 Solution to an Open Problem: Active Attacks Against Honeywords System

The honeyword system is only designed to withstand off-line attacks. In this scenario we assume, as the authors mentioned in [72], that the adversary has only stolen the password hashes but did not compromise the system on a persistent basis, *i.e.*, the adversary obtained the password hashes and left the system. However, Juels and Rivest mentioned about a problem which we



believe is still open: How can a honeyword system be best designed to withstand active attacks, *e.g.*, code modification, of the system (or the honeychecker)?

The question is very reasonable as the adversary who has access the password hashes may also gain other permissions like administrator rights. In that case, the system is assumed to be corrupted and therefore may behave arbitrarily.

### 4.3.1 Assumptions

In our proposal, we assume that the login server and the honeychecker cannot be compromised at the same time (which is a trivial assumption as otherwise the honeyword scheme will not be secure at all). We also assume that the administrators of login server and honeychecker do not cooperate.

### 4.3.2 Adversarial Capabilities

In our model, we classify adversarial attack scenarios into two classes.

- The adversary has compromised the login server and has gained administrator rights. She can now modify the codes in the login server as well as other components on it.
- The adversary has compromised the honeychecker. She can now modify the codes of the honeychecker as well as other components on it.

In the first case, the attacker has gained administrator rights and has system wide effects. Thus she can send any message (or request) to honeychecker. Note that the honeychecker understands only two type of messages: **Check** and **Set**. We previously described the **Check** message. The **Set** message is of the form  $\text{Set}(i, j)$  which is understood by the honeychecker as the  $i^{\text{th}}$  user's password is in the  $j^{\text{th}}$  position among her sweetwords. Using this advantage, the adversary can attack as follows. She inverts the hash of a sweetword of  $i^{\text{th}}$  user, say the  $j^{\text{th}}$  sweetword. Then she sends a message  $\text{Set}(i, j)$  to the honeychecker as she controls the login server. Now she can impersonate the  $i^{\text{th}}$  user and the honeychecker cannot detect it.

In the second case, the adversary has administrator rights on the honeychecker. In this case, the most important attack that the adversary can mount is a DoS attack. The attacker can allow an illegitimate login request or disallow a legitimate login request by randomly returning a **True** or **False** message.

### 4.3.3 The Proposal

In [72], the login server is assumed to be honest and therefore does not maliciously use any **Set** message. However, in one of our attack scenarios the login server is assumed to be malicious. Thus, we need to enhance the overall honeyword system in such a way that the login server cannot transmit malicious **Set** messages without being detected. **Set** messages are sent to honeychecker by the login server if

- A user signs up, *i.e.*, she creates her password for the first time.
- A user changes her password.

We need a secure channel to communicate with the user whose password's order is being set. However, the honeychecker does not have any communication information of the user. One of the design principles of honeychecker is that compromise of only the honeychecker should not reduce the security level of the whole system. Therefore, in our proposal, we give honeychecker the minimal information about the user. This information must be enough to communicate with the user to validate her with a fair confidence. In today's world, the most common way of this communication can be done on Short Message Service (SMS).

In our model, when a user registers to the system, the login server asks her to enter registration information including mobile number and updates the honeychecker. In order to accomplish this, we overload the **Set** function as follows:

- $\text{Set}(i, j)$
- $\text{Set}(i, j, phn)$

The first version of **Set** function is the same as the **Set** function in the original honeywords system. It is called when a user changes her password. The second version of **Set** function takes an extra parameter: *phn*, the user's mobile number. It is called when the user registers to the system. The honeychecker will add this information to its database and will communicate with the client when needed. During registration, the honeychecker sends a random code to the client via an SMS message. Then, the client uses this code in the registration phase to verify that the phone number is a valid one. The summary of this enhanced honeyword system is depicted in Figure 4.4.

If the user would like to change her password, she will send a password change request to the login server. The login server will send a **Set** message to the honeychecker. The honeychecker generates a random code and sends it to the client via SMS. The client sends back the code to the server and it sends the code with the required update messages to the honeychecker in order to validate

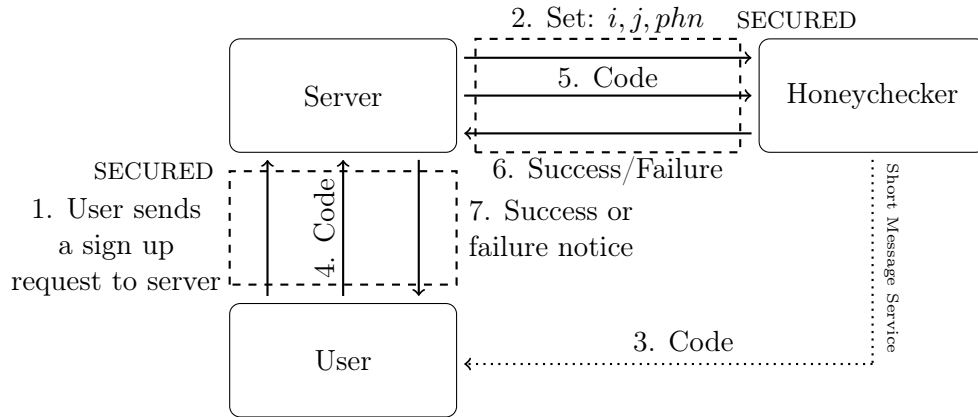


FIGURE 4.4: Sign up schema of an enhanced honeywords system.

the origin of the request. The password change scheme of an enhanced honeyword system is depicted in Figure 4.5.

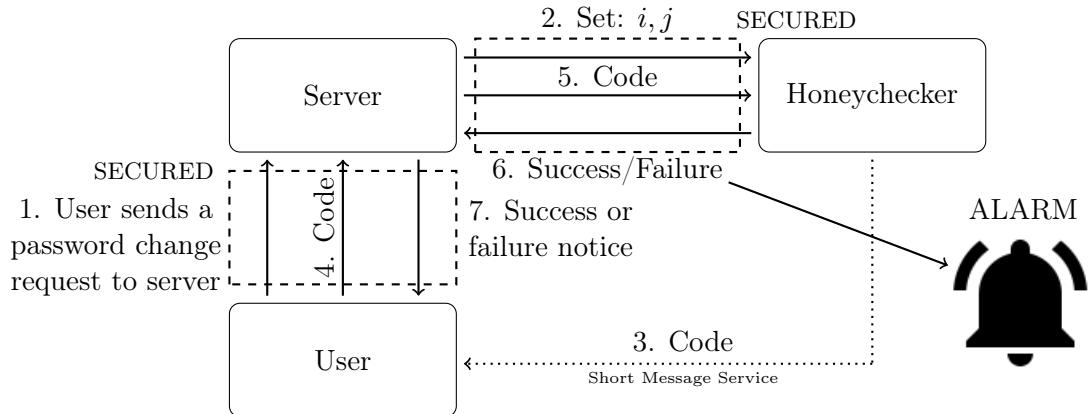


FIGURE 4.5: Password change schema of an enhanced honeywords system.

If a variety of invalid update requests are sent to the honeychecker, the honeychecker will assume this situation as an attack and it sets on an alarm. The alarm must be sent to the system owner or to the administrator.

#### 4.3.4 Security Analysis of the New Proposed Model

If the adversary gains administrator rights on login server, she can send Set messages which she desires. In this case, the honeychecker will confirm the update request by asking the user. A dishonest login server will fail to change the order of a user's password in this scenario.

The adversary can also send repetitive Check messages to find the order of a user's password. We suggest that the honeychecker counts and monitors the check requests and decides whether the login server is compromised or not. High number of check requests means that the (malicious) login server is making a brute-force search to find the correct order of password of each user.

An adversary may also attack honeychecker and may gain administrator rights on it. She can modify the honeychecker to send arbitrary results to login server after **Check** messages. Our suggestion for fighting with this attack is creating some number of dummy accounts to test the honesty of the honeychecker. The login server will send valid and invalid login requests with these accounts frequently. If the honeychecker is infected, *i.e.*, compromised, it will not take the correct action in response to these requests. Thus the adversary will be detected.

Hence, our enhanced model of honeyword system is more robust to active attacks than the primitive system designed in [72].

## Chapter 5

# Conclusion

In this thesis, we studied secure biometric authentication protocols and mitigation methods against password breaches. We first showed that the SHADE protocol of Bringer *et al.* is not secure in the malicious case. In our attack, we show that an adversary without having any prior knowledge can make the verifier compute an incorrect Hamming distance. In the case of biometric authentication systems, a malicious user can easily authenticate without any information about the honest party. Namely, the complexity of the security of the system is reduced from  $O(2^n)$  to  $O(n)$ , where  $n$  is the input length. Moreover, we fixed the protocol by placing a robust method for input validation without adding a significant cost. We also enhanced the efficiency of their protocol significantly by showing that Verifiable Oblivious Transfer (VOT) is sufficient to use instead of Committed Oblivious Transfer (COT) in the second option of the full scheme. The VOT reduction avoids the unnecessary computation of one commitment, two zero-knowledge proofs and two partial decryptions of the ciphertext for each bit of the input.

This thesis also examined the Honeywords system which is designed to mitigate against password breaches. We found that the honeyword system provides powerful defense in the scenario where an adversary steals the file of password hashes and inverts most of the hashes. Namely, even if the adversary has broken all the hashes in the password file, he cannot login to the system without being detected. Hacking the honeychecker has also no benefit to the adversary since there is no information about a user's password or honeyword in the honeychecker. The order of the true password is meaningless without obtaining the file of password hashes. On the other hand, honeyword system is still not a complete and effective solution for the password management problem. In particular, the following scenarios should also be considered in order to have a robust and secure system:

- An adversary can manage to observe the submitted passwords in real time.

- An adversary can steal the sweetwords of a user and submit to another systems which does not use honeywords.

In the second part of the thesis, we revisited and analyzed the Honeywords system of Juels and Rivest [72] and then suggest some possible improvements for

- determining the number of honeywords of a user
- generating typo-safe honeywords
- managing old passwords

Furthermore, we introduced an enhanced Honeywords system as a solution to the active attacks problem. We conclude with the security analysis of the new model.

# Bibliography

- [1] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, Jan 2004. ISSN 1051-8215. doi: 10.1109/TCSVT.2003.818349.
- [2] T. A. M. Kevenaar, G. J. Schrijen, M. van der Veen, A. H. M. Akkermans, and F. Zuo. Face recognition with renewable and privacy preserving binary templates. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on*, pages 21–26, Oct 2005. doi: 10.1109/AUTOID.2005.24.
- [3] F. Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE Trans. Comput.*, 55(9):1081–1088, September 2006. ISSN 0018-9340. doi: 10.1109/TC.2006.138.
- [4] S. Kanade, D. Petrovska-Delacretaz, and B. 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, June 2009. doi: 10.1109/CVPR.2009.5206646.
- [5] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security, CCS '99*, pages 28–36, New York, NY, USA, 1999. ACM. ISBN 1-58113-148-8. doi: 10.1145/319709.319714.
- [6] A. Salaiwarakul and M. D. Ryan. Analysis of a biometric authentication protocol for signature creation application. In *Advances in Information and Computer Security*, volume 5312 of *Lecture Notes in Computer Science*, pages 231–245. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-89597-8. doi: 10.1007/978-3-540-89598-5\_16.
- [7] J. Bringer and H. Chabanne. An authentication protocol with encrypted biometric data. In *Progress in Cryptology – AFRICACRYPT 2008*, volume 5023 of *Lecture Notes in Computer Science*, pages 109–124. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-68159-5. doi: 10.1007/978-3-540-68164-9\_8.

- 
- [8] C. Braz and J.-M. Robert. Security and usability: The case of the user authentication methods. In *Proceedings of the 18th International Conference of the Association Francophone D'Interaction Homme-Machine*. ACM, 2006.
- [9] S. Furnell, P. Dowland, H. Illingworth, and P. Reynolds. Authentication and supervision: A survey of user attitudes. *Computers & Security*, 2000.
- [10] D. Florencio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on the World Wide Web*. Association for Computing Machinery, Inc., 2007.
- [11] R. Abelson and M. Goldstein. Millions of anthem customers targeted in cyberattack. *The New York Times*, 5 February 2015.
- [12] M. Backman. Home depot: 56 million cards exposed in breach. *CNN*, 18 September 2014.
- [13] D. Rushe. Jp morgan chase reveals massive data breach affecting 76m households. *The Guardian*, 3 October 2014.
- [14] J. L. Yang and A. Jayakumar. Target says up to 70 million more customers were hit by december data breach. *The Washington Post*, January 10, 2014.
- [15] M. Sparkes. ebay tells users to change passwords after 'cyber attack'. *The Telegraph*, 21 May 2014.
- [16] M. Weir, S. Aggarwal, B. d. Medeiros, and B. Glodek. Password cracking using probabilistic context-free grammars. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, SP '09, pages 391–405, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3633-0.
- [17] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *IEEE Symposium on Security and Privacy*, pages 523–537, 2012.
- [18] T. Hashcat. oclhashcat, 2015. URL <http://hashcat.net/oclhashcat/>.
- [19] A. Jarrow and B. Pinkas. Secure hamming distance based computation and its applications. In *Applied Cryptography and Network Security*, volume 5536 of *Lecture Notes in Computer Science*, pages 107–124. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01956-2. doi: 10.1007/978-3-642-01957-9\_7.
- [20] S. Jarecki and V. Shmatikov. Efficient two-party secure computation on committed inputs. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer*



- Science*, pages 97–114. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72539-8. doi: 10.1007/978-3-540-72540-4\_6.
- [21] C. Hazay and Y. Lindell. Efficient oblivious polynomial evaluation with simulation-based security. Cryptology ePrint Archive, Report 2009/459, 2009. <http://eprint.iacr.org/>.
- [22] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. Scifi - a system for secure face identification. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 239–254, May 2010. doi: 10.1109/SP.2010.39.
- [23] J. Bringer, H. Chabanne, and B. Kindarji. Identification with encrypted biometric data. *Security and Communication Networks*, 4(5):548–562, 2011. ISSN 1939-0122. doi: 10.1002/sec.206.
- [24] J. Bringer, H. Chabanne, M. Izabachéne, D. Pointcheval, Q. Tang, and S. Zimmer. An application of the goldwasser-micali cryptosystem to biometric authentication. In *Information Security and Privacy*, volume 4586 of *Lecture Notes in Computer Science*, pages 96–106. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-73457-4. doi: 10.1007/978-3-540-73458-1\_8.
- [25] S. Goldwasser and S. 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*, STOC '82, pages 365–377. ACM, 1982. ISBN 0-89791-070-2. doi: 10.1145/800070.802212.
- [26] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies*, volume 5672 of *Lecture Notes in Computer Science*, pages 235–253. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03167-0. doi: 10.1007/978-3-642-03168-7\_14.
- [27] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, January 1991. ISSN 0898-929X. doi: 10.1162/jocn.1991.3.1.71.
- [28] I. Damgård, M. Geisler, and M. Krøigaard. Efficient and secure comparison for on-line auctions. In *Information Security and Privacy*, volume 4586 of *Lecture Notes in Computer Science*, pages 416–430. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-73457-4. doi: 10.1007/978-3-540-73458-1\_30.
- [29] I. Damgård, M. Geisler, and M. Kroigard. A correction to efficient and secure comparison for on line auctions. *Int. J. Appl. Cryptol.*, 1(4):323–324, August 2009. ISSN 1753-0563. doi: 10.1504/IJACT.2009.028031.

- [30] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *Proceedings of the 12th International Conference on Information Security and Cryptology, ICISC'09*, pages 229–244. Springer-Verlag, 2010. ISBN 3-642-14422-5, 978-3-642-14422-6.
- [31] T. Schneider and M. Zohner. GMW vs. Yao? efficient secure two-party computation with low depth circuits. In A. Sadeghi, editor, *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, volume 7859 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2013.
- [32] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC '87*, pages 218–229, New York, NY, USA, 1987. ACM. ISBN 0-89791-221-7.
- [33] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis. Practical biometric authentication with template protection. In *Audio- and Video-Based Biometric Person Authentication*, volume 3546 of *Lecture Notes in Computer Science*, pages 436–446. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-27887-0. doi: 10.1007/11527923\_45.
- [34] P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In *Biometric Authentication*, volume 3087 of *Lecture Notes in Computer Science*, pages 158–170. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22499-0. doi: 10.1007/978-3-540-25976-3\_15.
- [35] J.-P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *Audio- and Video-Based Biometric Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*, pages 393–402. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40302-9. doi: 10.1007/3-540-44887-X\_47.
- [36] R. Kulkarni and A. Nambodiri. Secure hamming distance based biometric authentication. In *Biometrics (ICB), 2013 International Conference on Biometrics Compendium, IEEE*, pages 1–6, June 2013. doi: 10.1109/ICB.2013.6613008.
- [37] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of Cryptography*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-24573-5. doi: 10.1007/978-3-540-30576-7\_18.
- [38] F. Kerschbaum, M. J. Atallah, D. M'Raihi, and J. R. Rice. Private fingerprint verification without local storage. In *Biometric Authentication*, volume 3072 of *Lecture Notes in Computer Science*, pages 387–394. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-22146-3. doi: 10.1007/978-3-540-25948-0\_54.

- [39] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, A. Piva, and F. Scotti. A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1–7, Sept 2010. doi: 10.1109/BTAS.2010.5634527.
- [40] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva. Privacy-preserving fingercode authentication. In *Proceedings of the 12th ACM Workshop on Multimedia and Security, MM&Sec '10*, pages 231–240. ACM, 2010. ISBN 978-1-4503-0286-9. doi: 10.1145/1854229.1854270.
- [41] J. Bringer, H. Chabanne, and A. Patey. Shade: Secure hamming distance computation from oblivious transfer. In *Financial Cryptography and Data Security*, volume 7862 of *Lecture Notes in Computer Science*, pages 164–176. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-41319-3. doi: 10.1007/978-3-642-41320-9\_11.
- [42] R. Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000. ISSN 0933-2790. doi: 10.1007/s001459910006.
- [43] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521830842.
- [44] Y. Lindell and B. Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *Advances in Cryptology - EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 52–78. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-72539-8. doi: 10.1007/978-3-540-72540-4\_4.
- [45] A. Shelat and C.-H. Shen. Two-output secure computation with malicious adversaries. In *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology, EUROCRYPT'11*, pages 386–405. Springer-Verlag, 2011. ISBN 978-3-642-20464-7.
- [46] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology - CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer Berlin Heidelberg, 1994. ISBN 978-3-540-58333-2. doi: 10.1007/3-540-48658-5\_19.
- [47] M. S. Kiraz, Z. A. Genc, and S. Kardas. Security and efficiency analysis of the hamming distance computation protocol based on oblivious transfer. *Security and Communication Networks*, 2015. doi: 10.1002/sec.1329.
- [48] Z. A. Genc, S. Kardas, and M. S. Kiraz. Examination of a new defense mechanism: Honeywords. Cryptology ePrint Archive, Report 2013/696, 2013. <http://eprint.iacr.org/>.

- [49] M. S. Kiraz. *Secure and Fair Two-Party Computation*. PhD thesis, Technische Universiteit Eindhoven, the Netherlands, August 2008.
- [50] A. Conklin, G. Dietrich, and D. Walz. Password-based authentication: A system perspective. In *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7 - Volume 7*, HICSS '04, pages 70170.2–, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2056-1.
- [51] J. Bonneau. Guessing human-chosen secrets. Technical Report UCAM-CL-TR-819, University of Cambridge, Computer Laboratory, May 2012. URL <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-819.pdf>.
- [52] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE Symp. Security and Privacy*, 2012.
- [53] E. H. Spafford. Opus: preventing weak password choices. *Comput. Secur.*, 11(3):273–278, May 1992. ISSN 0167-4048.
- [54] P. G. Neumann. Risks of passwords. *Commun. ACM*, 37(4):126–, April 1994. ISSN 0001-0782.
- [55] National information assurance (ia) glossary, 2010.
- [56] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, RFC Editor, August 2008. URL <http://www.rfc-editor.org/info/rfc5246>.
- [57] A. van der Merwe, M. Loock, and M. Dabrowski. Characteristics and responsibilities involved in a phishing attack. In *Proceedings of the 4th international symposium on Information and communication technologies*, WISICT '05, pages 249–254. Trinity College Dublin, 2005. ISBN 1-59593-169-4.
- [58] D. Elser and M. Pekrul. Inside the password-stealing business: the who and how of identity theft, 2009.
- [59] J. Erasmus. Malware attacks: Anatomy of a malware attack. *Netw. Secur.*, 2009(1):4–7, January 2009. ISSN 1353-4858.
- [60] H.-M. Sun, Y.-H. Chen, and Y.-H. Lin. opass: A user authentication protocol resistant to password stealing and password reuse attacks. *Information Forensics and Security, IEEE Transactions on*, 7(2):651–663, 2012. ISSN 1556-6013.
- [61] M. Kumar, T. Garfinkel, D. Boneh, and T. Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *Proceedings of the 3rd symposium on Usable privacy and security*, SOUPS '07, pages 13–19, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-801-5.

- [62] R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 136–145, Oct 2001. doi: 10.1109/SFCS.2001.959888.
- [63] Y. Aumann and Y. Lindell. Security against covert adversaries: Efficient protocols for realistic adversaries. In *Theory of Cryptography*, volume 4392 of *Lecture Notes in Computer Science*, pages 137–156. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-70935-0. doi: 10.1007/978-3-540-70936-7\_8.
- [64] M. S. Kiraz, B. Schoenmakers, and J. Villegas. Efficient committed oblivious transfer of bit strings. In *Information Security*, volume 4779 of *Lecture Notes in Computer Science*, pages 130–144. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-75495-4. doi: 10.1007/978-3-540-75496-1\_9.
- [65] J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider, and M. Zohner. Gshade: Faster privacy-preserving distance computation and biometric identification. In *Proceedings of the 2Nd ACM Workshop on Information Hiding and Multimedia Security, IH&MMSec '14*, pages 187–198, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2647-6.
- [66] Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 145–161. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-40674-7. doi: 10.1007/978-3-540-45146-4\_9.
- [67] V. Kolesnikov and R. Kumaresan. Improved ot extension for transferring short secrets. In *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 54–70. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40083-4. doi: 10.1007/978-3-642-40084-1\_4.
- [68] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472, Jul 1985. ISSN 0018-9448. doi: 10.1109/TIT.1985.1057074.
- [69] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin Heidelberg, 1999.
- [70] Y. Luo, S.-c. S. Cheung, T. Pignata, R. Lazzeretti, and M. Barni. An efficient protocol for private iris-code matching by means of garbled circuits. *19th IEEE International Conference on Image Processing (ICIP'12)*, pages 2653–2656, 2012. <http://dx.doi.org/10.1109/ICIP.2012.6467444>.

- 
- [71] A. Baig, A. Bouridane, F. Kurugollu, and G. Qu. Fingerprint - iris fusion based identification system using a single hamming distance matcher. In *Bio-inspired Learning and Intelligent Systems for Security, 2009. BLISS '09.*, pages 9–12, Aug 2009. doi: 10.1109/BLISS.2009.4.
- [72] A. Juels and R. L. Rivest. Honeywords: Making password-cracking detectable, 2013. URL <http://people.csail.mit.edu/rivest/honeywords/>. Unpublished draft.
- [73] H. Bojinov, E. Bursztein, X. Boyen, and D. Boneh. Kamouflage: Loss-resistant password management. In *ESORICS*, pages 286–302, 2010.