DESIGN AND ANALYSIS OF CERTIFICATE-FREE UNDENIABLE SIGNATURE SCHEMES

ROUZBEH BEHNIA

MASTER OF SCIENCE
(INFORMATION TECHNOLOGY)

MULTIMEDIA UNIVERSITY

APRIL 2013

DESIGN AND ANALYSIS OF CERTIFICATE-FREE UNDENIABLE SIGNATURE SCHEMES

BY

ROUZBEH BEHNIA

B.IT. (Hons), Multimedia University, Malaysia

THESIS SUBMITTED IN FULFILMENT OF THE

REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

(by Research)

in the

Faculty of Information Science and Technology

MULTIMEDIA UNIVERSITY
MALAYSIA

April 2013

DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this Thesis has been submitted in support of any application for any other degree or qualification on this or any other university or institution of learning.

Rouzbeh Behnia

ACKNOWLEDGEMENTS

First and foremost, I offer my utmost gratitude to my supervisor Prof. Dr. Heng Swee Huay for her continuous support and immense knowledge. She provided me with the chance to continue my studies and patiently supported me through my path from the very beginning. Undoubtedly, without her help and guidance this thesis would not have been written. I also want to thank my co-supervisor Dr. Gan Che Sheng for his guidance, understanding and encouragement through this journey.

I wish to thank all the FIST faculty members who supported me during my time in FIST. I would use this opportunity to thank Mr. Jin Zhe, Mr. Tan Syh Yuan and Mr. Chin Ji Jian for their helpful discussions throughout this research. I also want to thank Ms. Lim Lian Tze for preparing the template of this thesis.

I would like to acknowledge the FRGS grant (FRGS/2/2010/TK/MMU/02/03) and Multimedia University Graduate Research Assistant (GRA) scheme for the financial support.

Last but not least, I would like to express my sincerest gratitude to my parents and my sister and brother for always being there for me and offering their unconditional love when I needed it the most.

To my beloved parents

ABSTRACT

The main focus of this thesis is on the design and analysis of undeniable signature schemes in certificate-free settings, namely, identity-based setting and certificateless setting. Undeniable signature is a special type of digital signatures which is not universally verifiable.

Identity-based cryptography overcomes the costly issues in traditional public key cryptography by computing the users' public keys directly from their publicly available information. However, identity-based systems suffer from an inherent private key escrow problem. Certificateless cryptography was later proposed to bridge between identity-based cryptography and traditional public key cryptography by eliminating the use of certificates while addressing the private key escrow problem at the same time.

Firstly, two attacks are mounted on an efficient identity-based undeniable signature scheme. A provably secure and efficient identity-based undeniable signature scheme with short signature is then proposed.

Secondly, cryptanalysis is presented on a newly proposed efficient certificateless undeniable signature scheme. More precisely, security flaws are found on the invisibility and non-impersonation properties of the scheme. A revised scheme is then proffered which tackles both of the attacks while enjoys from an equally efficient Sign algorithm. Independently, a provably secure certificateless undeniable signature scheme which is more efficient than the only existing scheme that is secure in the strong security model is proposed.

Lastly, in our effort in proposing certificateless undeniable signature schemes with additional features, the security model of convertible certificateless undeniable signature schemes is formally defined and an instance of such schemes is presented for the first time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

The active invention course of highly capable hand-held devices along with the manifestation of Internet, wireless networks and mobile telecommunications technology has metamorphosed our life in many aspects. Meetings take place between a group of people who are thousand miles away in real time and paying bills, online shopping, online banking and so on are just a few clicks away.

Historically, communication was mostly paper-based, whereas information was mainly stored on paper and transmitted by means of letters in mail. Today, most of the information is being stored and transmitted digitally, leading to a high risk of fraud, forgery, inception and so forth. The rapid growth of Internet-based business has become a fruitful target for cyber criminals. Accordingly, measures have been deployed to thwart such threats. Information security provides many contrivances according to the type and level of protection needed.

Cryptography provides information security. It is the study to design and employ mathematical techniques and approaches to propose algorithms and protocols in order to establish secure communication in an exposed environment. Following are the main objectives of cryptography through which security is achieved.

1. Confidentiality: Preserving the secrecy of data from being accessed/read from all but the authorised parties.
2. Data Integrity: Protecting data from any unauthorised tampering or modification attempt. More specifically, data integrity ensures the detection of any unauthorised modification of the original data.

3. Authentication: Providing users with mechanisms to prove their identification.
4. Non-repudiation: Measures to ensure that users are prevented from denying their antecedent engagements or commitments.

1.2 Public Key Cryptography

Before the seminal work of Diffie and Hellman (1976b), all the cryptographic schemes were implemented in symmetric key systems. In such systems, if two users (the sender and the recipient) wish to communicate in a secure manner, they must agree upon a pre-computed secret key called the shared secret key. The security of their communication would rely on an assumption that no one else has knowledge on the shared secret key. Consequently, the leakage of the shared secret key would lead in revealing all the secret information being transmitted between the users. In addition, if the users are located in geographically distant locations, transferring the shared secret key in a secure manner is sometimes excessively expensive or even impossible. Lastly, for every system user to be able to communicate with other users in a secure manner, she would need to maintain a large number of keys securely. These constraints are considered as the key distribution problem which is inherited in symmetric key cryptography and greatly limited the use of cryptography in the old days.

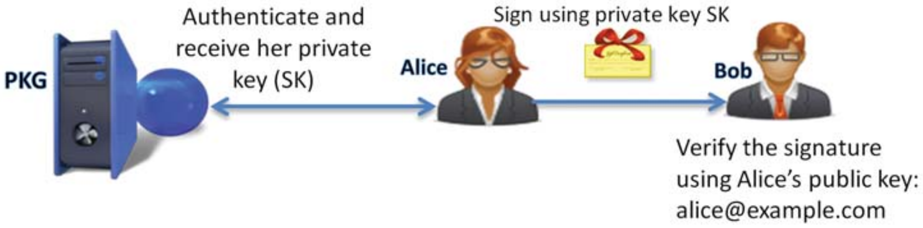
The introduction of the new paradigm in 1976 revolutionised the use of cryptographic schemes by addressing the inherent key distribution problem in symmetric key cryptography. In public key cryptography, which is also called asymmetric cryptography, each user has a pair of keys consisted of a public key and the corresponding private key. As the names convey, the private key should be kept secret and the public key is to be made public (e.g. published on a public bulletin). Messages are encrypted/verified using the user's pubic key while they can only be decrypted/signed using the recipient's/signer's private key.

In the system proposed by Diffie and Hellman (1976b), the user (Alice) computes her key pair and publishes her public key in the system. Before any entity attempts to perform any cryptographic operation (e.g. encrypt message or verify signatures) using Alice's public key, he needs to verify its authenticity. The authentication of the users' public keys is transferred in the form of signed certificates issued by a semi-trusted third party called the Certificate Authority (CA). Throughout this thesis, this type of public key cryptography is referred to as traditional public key cryptography to be easily differentiated from other types of public key cryptography.

1.2.1 Identity-Based Cryptography

The concept of identity-based cryptography was put forth by Shamir (1985) in order to address the costly infrastructure needed in traditional public key cryptography. Identity-based cryptography is a public key cryptosystem where the public key of the user is efficiently computed from her publicly available information (e.g. an IP address of a network host, an email address associated with a user) that can uniquely identify her in the system. Hence, the one-to-one mapping eliminates the need of certificates to verify the authenticity of the users' public keys. Identity-based systems rely on an authoritative server or a Trusted Third Party (TTP) called the Private Key Generator (PKG) which is solely needed to generate and deliver the users' private keys. The PKG uses the master secret key to compute the private key of the users based on their publicly available information. Through this approach, implicit certification takes place where the users would not need to receive and verify the authenticity of public keys via signed certificates (explicit certification). Figure 1.1 below illustrates the idea behind identity-based cryptography where the user Alice first authenticates herself to the PKG, receives her private key and uses her private key to issue signatures. On the other side, the verifier Bob can compute Alice's public key from her publicly available information (her email address in this case) and verify the validity of the signature.

The first successful implementation of identity-based cryptography was proffered by the seminal work of Boneh and Franklin (2001), in which they employed bilinear pairing over elliptic curves.

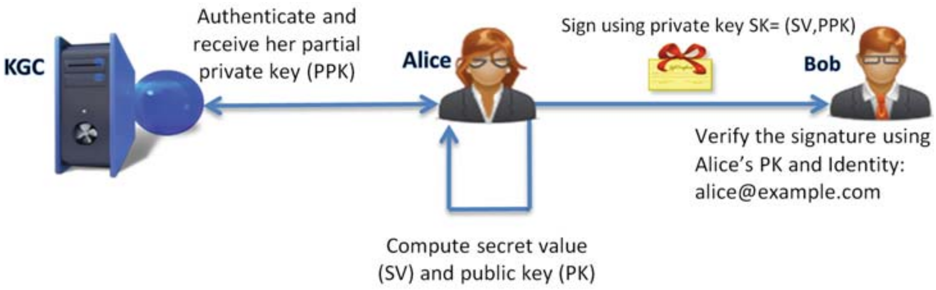


PKG
Authenticate and receive her private key (SK)
Alice
Sign using private key SK
Bob
Verify the signature using Alice's public key: alice@example.com

Figure 1.1: Identity-Based Cryptography

1.2.2 Certificateless Cryptography

The notion of certificateless cryptography was first proposed by Al-Riyami and Paterson (2003). The underlying idea in certificateless cryptography is that the private key of the user is consisted of two parts: an identity-based private key (partial private key) which is generated by a semi-trusted third party called the Key Generation Centre (KGC), and a random value (secret value) which is chosen and kept secret by the user. The corresponding public key has to be computed (by the user herself) and made available in the system. Similar to identity-based cryptography, implicit certification takes place where the KGC uses the master secret key to generate the partial private key of the user.



KGC
Authenticate and receive her partial private key (PPK)
Alice
Sign using private key SK= (SV,PPK)
Bob
Verify the signature using Alice's PK and Identity: alice@example.com
Compute secret value (SV) and public key (PK)

Figure 1.2: Certificateless Cryptography

Figure 1.2 above depicts the underlying idea in certificateless systems. The user Alice computes her secret value and public key before authenticating herself to the KGC and receiving her partial private key. She would then use both her secret value and partial private key to form the signature. Bob as the verifier would retrieve

Alice's public key (e.g. from a public bulletin board) and uses her publicly available information to verify the validity of the signature.

1.2.3 Strengths and Weaknesses of Different Public Key Cryptosystems

In the following, we provide a brief comparison between traditional public key cryptography, identity-based cryptography and certificateless cryptography by highlighting their strengths and weaknesses.

The main distinguishing factor between traditional public key systems and identity-based and certificateless systems is that in the former, the certification takes place explicitly i.e. signed certificates are used to bind public keys and users; while identity-based and certificateless systems are developed based on the concept of implicit certification. In traditional public key cryptography, the user generates her own set of key pair (private and public key) and requests for a certificate on her public key from the CA (explicit certification). However, the cost of implementing and maintenance of the infrastructure needed for issuing and managing these certificates would be a critical issue when such systems are employed in a large scale. In order to tackle this problem, in identity-based systems, the public key of the user is directly derived from her identifying information and her private key is computed by the PKG based on the same identifying information using the master secret key. Hence, public keys are self-certified and any user in the system can compute other users' public keys on her own, without needing to query for it from any directory or entity. However, the knowledge of the PKG over the users' private keys introduces the private key escrow problem and therefore, it would be disastrous if the PKG is compromised. The private key escrow problem can be addressed to a certain point by having more than one PKG in place to issue users' private keys i.e. by use of hierarchical identity-based cryptography (Horwitz & Lynn, 2002). However, such methods are less efficient since they require superfluous communication and infrastructure support. Another issue of identity-based systems is the need for a secure channel to deliver the users' private keys in a secure manner.

Certificateless cryptography overcomes the costly issues in traditional public key cryptography while addressing the private key escrow problem in identity-based cryptography. Since the KGC only supplies a portion of the users' private keys (i.e. partial private key), the compromise of the master secret key is less disastrous than in identity-based cryptography. On the downside, certificateless systems do not possess the easy public key feature of identity-based systems since public keys are computed based on the secret values and a secure channel is still needed to deliver the users' partial private keys securely and confidentially.

Table 1.1 below is inspired by the work proposed by Yap, Heng, and Goi (2006) and provides an instant overview and comparison between traditional public key cryptography (TPKC), identity-based cryptography (IDC) and certificateless cryptography (CLC).

Table 1.1: Public Key Cryptography Variations

	TPKC	**IDC**	**CLC**
TTP	CA	PKG	KGC
Private Key Escrow	No	Yes	No
Certification	Explicit	Implicit	Implicit
Secure Channel with TTP	No	Yes	Yes
Easy Public Key	No	Yes	No
Trust Level	3	1	2/3

In Table 1.1, the term trust level refers to the level of trust on the TTP (i.e. CA, PKG, and KGC). Based on Girault (1991), the level of trust can be categorised in the following levels:

- **Level 1:** The trusted authority can easily compute the users' private keys and therefore, it can impersonate the users without the chance of being detected.
- **Level 2:** The authority is unable to compute the users' private keys. However, it can still impersonate the users by generating false guarantee (e.g. false public key in certificateless system) without being detected.

- **Level 3:** The authority is unable to compute the users' private keys, and if it does so, it can be easily detected.

In this thesis, we refer to identity-based and certificateless systems as certificate-free systems since the need of signed certificates on the users' public keys is eliminated in such systems.

1.2.4 Digital Signatures

Digital signature is one of the many cryptographic primitives which is provided by public key cryptography. The notion of digital signature was first introduced by Diffie and Hellman (1976a). Digital signatures are analogous to handwritten signatures on digital data. Generally, in such schemes, the signer uses her private key to generate a signature on some message and the verifier uses the signer's public key to verify the signature. Therefore, when a signature passes the verification step, the verifier is convinced that the signature was indeed signed by the signer (source authentication). Moreover, digital signatures preserve the integrity of the message. More precisely, if the signed message is tampered in any possible way, then, the signature would be invalidated. Another main property of digital signatures is non-repudiation which ensures that the signer is not able to deny the validity of her signatures.

Digital signatures are publicly verifiable. In other words, any user with knowledge on the public key of the signer is able to verify the validity of the signatures. The notion of digital signature with all the above properties is referred to as ordinary digital signature throughout this thesis.

1.3 Motivation

The course of research on undeniable signature schemes in traditional public key cryptography has been going on since the introduction of such schemes (Chaum & van Antwerpen, 1989), and many practical examples of undeniable signature schemes with different levels of security and special features have been proposed to the literature (Boyar, Chaum, Damgård, & Pedersen, 1991; Chaum, 1995; Chaum, van Heijst,

& Pfitzmann, 1992; Galbraith & Mao, 2003; Kurosawa & Heng, 2005; Kurosawa & Takagi, 2006; Phong, Kurosawa, & Ogata, 2010). However, the research on undeniable signature schemes in certificate-free systems has been surprisingly slow. The first identity-based undeniable signature scheme was proposed by Han, Yeung, and Wang (2003) and later it was shown to be insecure by Zhang, Safavi-Naini, and Susilo (2005). Libert and Quisquater (2004) proposed the first provably secure identity-based undeniable signature scheme. Later Duan (2008) proposed the first certificateless undeniable signature scheme and in the same year, Wu, Mu, Susilo, and Huang (2008) proposed a convertible identity-based undeniable signature scheme. Very recently, Zhao and Ye (2012) proposed an efficient certificateless undeniable signature scheme to the literature.

To the best of our knowledge, only five secure certificate-free undeniable signature schemes exist in the literature, in which three are identity-based and the other two are certificateless. Due to the limited research that has been done in the area, and especially in certificateless undeniable signature schemes, the focus of this thesis is to analyse the security and the structure of the existing schemes and to design new certificate-free undeniable signature schemes with efficiency advantages and extra features which are premier to the existing ones.

1.4 Objectives

The main focus of this thesis is on the design of certificate-free undeniable signature schemes with better security, improved efficiency or additional features. Moreover, we also aim to analyse the security of the existing certificate-free undeniable signature schemes. We briefly summarise the objectives as follows.

- To cryptanalyse the existing weak identity-based undeniable signature schemes and put forth new efficient and provably secure schemes.
- To cryptanalyse the existing weak certificateless undeniable signature schemes and propose new provably secure certificateless undeniable signature schemes with improved efficiency and additional features.

1.5 Organisation of the Thesis

In the following, we highlight the contribution of each of the remaining chapters of this thesis.

- In Chapter 2, we first review the concept of provable security and provide a preliminary review on the mathematical tools and some useful definitions which are going to be used throughout this thesis. Next, we provide a focused study on the existing certificate-free undeniable signature schemes. Along that line, we define the notions of identity-based undeniable signature schemes and certificateless undeniable signature schemes and analyse the structure and the features of the existing certificate-free undeniable signature schemes in the literature.
- In Chapter 3, we analyse the security of the most efficient identity-based undeniable signature scheme in the literature (Chow, 2005) and highlight two weaknesses in the structure of the scheme. By exploiting the weaknesses, we mount two attacks on the unforgeability and non-transferability of the scheme. We then put forth a new provably secure identity-based undeniable signature scheme which is more efficient than all the existing secure schemes in the literature. Lastly, we formally prove the security of the new scheme based on the hardness of some well-known mathematical assumptions.
- In Chapter 4, we focus on certificateless undeniable signature schemes. We start by analysing the recently proposed efficient certificateless undeniable signature scheme (Zhao & Ye, 2012), and discover two flaws in its structure. We show that the proposed scheme is not secure by mounting two attacks on its invisibility and non-impersonation properties. Next, we come up with a revised scheme which overcomes both of the flaws while is equally efficient as the original scheme (Zhao & Ye, 2012). Similar to the original scheme, the revised scheme is only secure in the weak security model. Independently, we proffer a new certificateless undeniable signature scheme which is secure in the strong security model. Comparing to the only existing certificateless undeniable signature scheme which is secure in the strong security model (Duan, 2008), our scheme is more efficient in all aspects (i.e. signature generation, proof genera-

tion and proof verification). Finally, we show that our scheme is secure in the random oracle model by relating its security to some well-studied mathematical assumptions.

- In Chapter 5, we propose the first convertible certificateless undeniable signature schemes. First, we formalise the security models of convertible undeniable signature schemes in a certificateless paradigm for the first time. Then, we put forth our concrete scheme and discuss about its additional features. Lastly, we prove the security of our scheme in the random oracle model by relying its security on the hardness of some well-known mathematical problems.
- In Chapter 6, we summarise the findings of this thesis and highlight the possible directions of the future research.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Provable Security

The concept of provable security was first proposed by Goldwasser and Micali (1984) where they provided semantic security for public key encryption. Since its introduction, provable security has become an undetachable component of modern cryptography. Informally, provable security implies definitions that make use of mathematical techniques to analyse the security of cryptographic schemes. The goal in establishing provable security is to rely the security of a cryptographic scheme on some well-known mathematical assumptions. As it is depicted in Figure 2.1 below, the process starts by formulating a real world scenario where the adversary $\mathcal{A}$ uses all its power to break the scheme by requesting assistance from an algorithm $\mathcal{C}$ (challenger). The goal is to prove that if $\mathcal{A}$ is able to break the scheme, then $\mathcal{C}$ can use $\mathcal{A}$ as its subroutine to solve a well-studied mathematical problem (e.g. computing discrete logarithms in finite fields). In order to assure the correctness of the proofs, we need to specify the computation power of the adversary and define (in detail) the goals of the adversary.

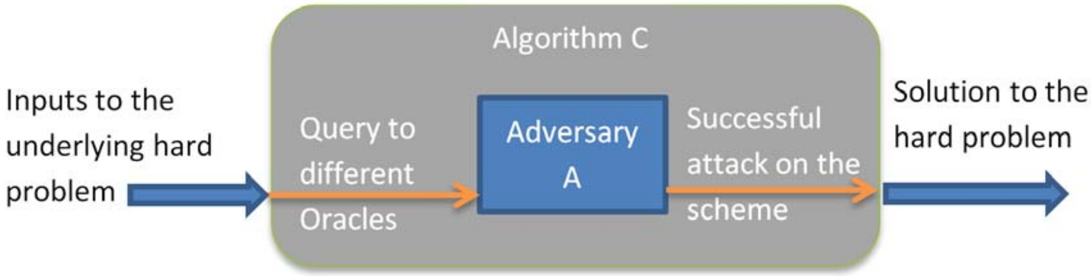

Algorithm C
Inputs to the underlying hard problem
Query to different Oracles
Adversary A
Successful attack on the scheme
Solution to the hard problem

Figure 2.1: Provable Security

The main part in provable security is identifying and analysing all the possible attacks in order to relate their difficulty to the hardness of mathematical problems. However, provable security does not assure that the scheme is completely unbreakable (Canetti, Goldreich, & Halevi, 2004; Neven, 2004), some possible attacks on provably secure schemes are:

- Solving the underlying mathematical assumption
- Breach in the security proof
- Breaking a sub-component

Nonetheless, provable security is still the most accepted approach to analyse the security of cryptographic schemes (Bellare, Boldyreva, & Palacio, 2004; Canetti et al., 2004).

The random oracle model was first envisioned by Bellare and Rogaway (1993) where they replaced cryptographic hash functions with imaginary random functions called random oracles. Random oracles are the idealised version of cryptographic hash functions, wherein they return a random value of a desired length whenever they are queried with a new value. In practice, random oracles are to be replaced by well-known cryptographic hash functions (e.g. SHA-2, MD5, etc.). Employing the random oracle model enables the design of more efficient cryptographic schemes comparing to the ones devised in the standard model.

However, the security of schemes developed in the random oracle model is not as rigid as the schemes proved secure in the standard model, the use of random oracles provides us with an assurance that the scheme itself is sound, whereas the only possible weaknesses may be due to the underlying hash functions instantiated in the real world. Albeit, the dispute exists on the security assurances provided by random oracles (Bellare et al., 2004; Canetti, 1997; Canetti et al., 2004; Goldwasser & Kalai, 2003; Maurer, Renner, & Holenstein, 2004; Nielsen, 2002), it is still considered as a hugely successful tool in developing efficient schemes.

2.2 Mathematical Background

In this section, we provide an introduction on bilinear pairing and some well-studied mathematical assumptions which are going to be used throughout this thesis.

2.2.1 Bilinear Pairing

The introduction of identity-based cryptography (Shamir, 1985) created a promising line of research in implementing such systems. However, it was until the seminal work of Boneh and Franklin (2001), where they successfully implemented such systems by making use of bilinear pairing on elliptic curves. Since then, a huge number of identity-based and certificateless cryptosystems have been proposed using the same primitive. Here, we provide a brief introduction on bilinear pairing.

Let $\mathbb{G}_1$ denote an additive cyclic group of prime order q with P as its generator, and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order (i.e. $|\mathbb{G}_1| = |\mathbb{G}_2| = q$). An admissible bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is given which is to satisfy the following properties:

1. **Bilinearity:** For every $P, Q, R \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$ we have:
   a) $e(P, Q+R) = e(P, Q)e(P, R)$
   b) $e(aP, bQ) = e(P, Q)^{ab}$ and $e(aP, bQ) = e(abP, Q)$
2. **Non-degeneracy:** There exist P and $Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$.
3. **Computability:** e is efficiently computable.

2.2.2 Mathematical Assumptions

Here, we provide a quick overview on some well-studied mathematical assumptions which will be used in this thesis.

Definition 2.1. ***Discrete Logarithm (DL) problem:*** *Given a tuple (P, aP), for P as a random generator of $\mathbb{G}_1$ and a random selection of $a \in \mathbb{Z}_q$, the DL problem is to compute a.*

Definition 2.2. ***Computational Diffie-Hellman (CDH) problem:*** *Given a tuple (P, aP, bP), for P as a random generator of $\mathbb{G}_1$ and a random selection of $a, b \in \mathbb{Z}_q$, the CDH problem is to compute abP.*

Definition 2.3. ***Decisional Diffie-Hellman (DDH) problem:*** *Given a tuple (P, aP, bP, cP), for P as a random generator of $\mathbb{G}_1$ and a random selection of $a, b \in \mathbb{Z}_q$, the DDH problem is to decide if $c = ab$.*

Definition 2.4. ***Bilinear Diffie-Hellman (BDH) problem:*** *Given a tuple (P, aP, bP, cP), for P as a random generator of $\mathbb{G}_1$ and a random selection of $a, b, c \in \mathbb{Z}_q$, the BDH problem is to compute $e(P,P)^{abc}$.*

Definition 2.5. ***Decisional Bilinear Diffie-Hellman (DBDH) problem:*** *Given a tuple (P, aP, bP, cP, h), for P as a random generator of $\mathbb{G}_1$, $h \in \mathbb{G}_2$, and a random selection of $a, b, c \in \mathbb{Z}_q$, the DBDH problem is to decide if $h = e(P,P)^{abc}$.*

Definition 2.6. ***3-Decisional Diffie-Hellman (3-DDH) problem:*** *Given a tuple (P, aP, bP, cP, Z), for P as a random generator of $\mathbb{G}_1$, $Z \in \mathbb{G}_1$, and a random selection of $a, b, c \in \mathbb{Z}_q$, the DBDH problem is to decide if $Z = abcP$.*

The CDH problem is considered to be as hard as the DL problem (Shoup, 1997). The DBDH problem was introduced by Cheon and Lee (2002). As it is shown in the same paper (Cheon & Lee, 2002), the DBDH problem is not harder than the DDH problem. However, there is no probabilistic polynomial time (PPT) algorithm known to solve the DBDH assumption so far. The DDH problem can be easily solved in pairing-based schemes. The 3-DDH problem (Laguillaumie, Paillier, & Vergnaud, 2005; Laguillaumie & Vergnaud, 2005) is definitely no harder than the DDH problem, but it seems intractable and can be used to achieve the privacy required in pairing-based undeniable signature schemes.

2.3 Security Notions for Digital Signature Schemes

As it was presented in the first proposal of digital signature scheme (Goldwasser, Micali, & Rivest, 1988), there are two distinctive types of attacks on digital signatures, namely, key only attack and message attack. In the former, the adversary $\mathcal{A}$ is assumed to only have knowledge on the signer's public key and the latter refers to the attacks where $\mathcal{A}$ is allowed to query and obtain some signatures corresponding to either known or chosen messages before its finial attempt to break the scheme. Message attacks are further categorised in the following categories (Goldwasser et al., 1988), depending on how the messages for which the adversary $\mathcal{A}$ can see the corresponding signatures are selected.

1. **Known message attack:** $\mathcal{A}$ is provided with a set of signatures corresponding to messages $m_1, m_2, \dots, m_i$; whereby, messages are known, but were not selected by $\mathcal{A}$.
2. **Generic chosen message attack:** $\mathcal{A}$ prepares a list of messages $m_1, m_2, \dots, m_i$ (prior to the attack) and requests for the corresponding signatures. The list of messages are fixed and independent of the signer's public key (i.e. m_i is chosen uniformly at random). Such an attack is generic since the list was prepared regardless of the signer's public key, and hence, the same list could have been submitted to any signer in the system. Note that the attack is non-adaptive, since the list of messages was fixed before any signature was received from the signer.
3. **Direct chosen message attack:** Similar to the above attack, except that the list of messages was formed after receiving the signer's public key but before obtaining any signature from the signer (i.e. the attack is non-adaptive). Note that the attack is directed against the target signer.
4. **Adaptive chosen message attack:** The target signer would be treated as an oracle where $\mathcal{A}$ is allowed to query messages which not only dependent on the signer's public key, but also on the signatures that were obtained previously.

In the same paper, Goldwasser et al. (1988) categorised the adversary goals based on their severity as follows.

- Extracting the private key of the signer is the strongest attack that $\mathcal{A}$ can initiate. This attack is called total break.

- Constructing a polynomial time algorithm capable of generating valid signature for any arbitrary message. This attack is called universal forgery.
- Computing a valid signature for a particular message selected by $\mathcal{A}$. This attack is called selective forgery.
- Forge a signature for at least one message. $\mathcal{A}$ does not need to have any control on the message neither does the message need to have any meaningful content. This attack is called existential forgery which is the weakest goal that $\mathcal{A}$ can attain.

Strong digital signature schemes should be secure against the strongest attacks (i.e. adaptive chosen message attack) which are initiated to attain the weakest adversarial goal (i.e. existential forgery).

2.4 Undeniable Signature Scheme

Ordinary digital signatures are publicly verifiable, more precisely, once the signer generates the signature, it can be verified by any party in the system. However, this property may not be desirable in situations where protecting the privacy of the signer is a concern (e.g. when two business parties sign a secret contract). The notion of undeniable signature (Chaum & van Antwerpen, 1989) was proposed to suite the signer's need in such situations. More specifically, undeniable signature schemes bridge between authentication and the privacy of the signer by allowing only the authorised parties to verify the validity/invalidity of the signatures. The validity or invalidity of an undeniable signature can only be verified with the consent and cooperation of its signer in a non-transferable manner. In order to address non-repudiation, undeniable signature schemes are equipped with an additional protocol (i.e. Disavowal protocol) which enables the signer to deny the validity of invalid signatures in court. Software licensing is one of the main applications of undeniable signatures (Chaum & van Antwerpen, 1989), the software vendor can incorporate an undeniable signature into the software and validate the software correctness and authenticity only to the paying customers. Therefore, the software will be protected from piracy since the pirate is not able to prove the correctness of the software. For the other main applications of undeniable signature schemes we can name e-cash (Sakurai & Miyazaki, 2000), e-voting (Boyd & Foo, 1998) and e-auction (Gao, Yao, Xie, & Wei, 2011).

2.4.1 Definition of Undeniable Signature Schemes

Basically, the structure of an undeniable signature scheme consists of two PPT algorithms which are key generation and sign and two protocols which are Confirmation and Disavowal (Chaum & van Antwerpen, 1989; Chaum, 1991). However, the number of the algorithms/protocols would be variable depending on the setting that the scheme is to be developed in and the extra features it provides.

Key Generation: A probabilistic algorithm that on the input of security parameter generates the user's key pair (sk, pk).

Sign: Provided a message m and the private key of the signer sk, the signer generates an undeniable signature σ on m.

Confirmation: A two-party protocol through which the signer convinces the verifier about the validity of a message-signature pair (m, σ).

Disavowal: Similar to the above protocol, except that the signer convinces the verifier about the invalidity of a message-signature pair (m, σ).

2.4.2 Security Notions of Undeniable Signature Schemes

Here, we discuss the main security notions of undeniable signature schemes in detail. An undeniable signature scheme is said to be secure if it meets all the security notions as follows.

Unforgeability: The notion of unforgeability of undeniable signature scheme is quite similar to the notion of existential unforgeability in adaptive chosen message attack (Goldwasser et al., 1988) of ordinary digital signatures. The only variation in defining this notion in the context of undeniable signature schemes is that in addition to the sign oracle, the adversary also has access to the Confirmation and Disavowal oracles.

Invisibility: The notion of invisibility was first introduced by Chaum et al. (1992). Essentially, this notion implies the inability of an adversary to distinguish between an undeniable signature and a random value (chosen randomly from a predefined signature space). Invisibility is the distinguishing factor of undeniable signatures from ordinary digital signatures. If the verifier is able to determine the validity of a message-signature pair without the help of the signer, then the signature is not any different than an ordinary digital signature.

Anonymity: The notion of anonymity was first introduced by Galbraith and Mao (2003) and it is considered as a variation of invisibility. Without loss of generality, the notion of anonymity implies, given a message-signature pair (m,σ), and public keys of two possible signers S_1 and S_2, the adversary should be unable to distinguish which signer issued the signature.

Non-Transferability: Non-transferability is a security notion which is driven from zero-knowledgeness property of both the Confirmation and Disavowal protocols in undeniable signature schemes. Intuitively, non-transferability refers to the inability of the verifier to transfer the proof of validity or invalidity of a message-signature pair to a third party. Informally, information that the verifier obtains from the Confirmation/Disavowal protocols should only be enough to convince the verifier about the validity/invalidity of a particular message-signature pair, and not to enable him to transfer the proof to a third party. As was formally defined by Monnerat and Serge (2006), a Confirmation/Disavowal protocol is non-transferable if the proof generated by the signer (using her private key) can be simulated by a PPT algorithm $\mathcal{R}$ using the private key of the verifier. The proof that is generated by $\mathcal{R}$ is indistinguishable from the one generated by the signer. Non-transferability is another factor that distinguishes undeniable signature schemes from ordinary digital signatures.

Non-Impersonation: Security notion against impersonation attack is yet another security notion introduced by Kurosawa and Heng (2005). Informally, this notion prevents the adversary from initiating either the Confirmation or Disavowal protocol on behalf of the signer with any third party.

2.5 Development of Certificate-Free Undeniable Signature Schemes

The successful implementation of identity-based systems by Boneh and Franklin (2001) gave rise to the development of many different certificate-free cryptographic schemes. Two years later, Al-Riyami and Paterson (2003) proposed the concept of certificateless cryptography to address the private key escrow problem in identity-based systems. In both of the systems, the need to implement the costly infrastructure to issue and manage certificates is eliminated by taking advantage of implicit certification where the TTP (either the PKG or the KGC) calculates the whole (in identity-based cryptography) or a part (in certificateless cryptography) of the user's private key.

The fact that certificate-free systems were much cheaper and easier to implement and manage, created a promising line of research of developing schemes in such settings. The development of certificate-free undeniable signature schemes was not an exception. To the best of our knowledge, certificate-free undeniable signature schemes that have been proposed to the literature to this day are either identity-based or certificateless. Table 2.1 below provides a quick overview on the existing certificate-free undeniable signature schemes.

Table 2.1: Certificate-Free Undeniable Signature Schemes

Schemes	**Underlying Assumptions**	**Paradigm**	**Signature Length (in bits)**
Han et al. (2003)	Broken (Zhang et al., 2005)	Identity-based	$2\|\mathbb{G}_1\| \approx 320$
Chow (2005)	Sketchy proof	Identity-based	$2\|\mathbb{G}_1\| \approx 320$
Libert and Quisquater (2004)	BDH - DBDH	Identity-based	$\|\mathbb{G}_2\| + \|r\| \approx 1124$
Wu et al. (2008)	CDH - DBDH	Identity-based	$\|\mathbb{G}_2\| + 2\|\mathbb{G}_1\| \approx 1344$
Duan (2008)	BDH - DBDH	Certificateless	$\|\mathbb{G}_2\| + \|r\| \approx 1124$
Zhao and Ye (2012)	CDH - 3-DDH	Certificateless	$2\|\mathbb{G}_1\| \approx 320$

In the above table, r is a 100 bit random value. In the following, we provide a detailed review on the development course of certificate-free undeniable signature scheme. We first start by reviewing the identity-based schemes and then we move on to the certificateless undeniable signature schemes that have been proposed to the literature to this day.

2.5.1 Identity-Based Undeniable Signature Schemes

Before reviewing the development of the existing identity-based undeniable signature schemes in the literature, we define the notion of identity-based undeniable signature schemes. It is evident that schemes with different security level and additional features (e.g. convertibility) may have various numbers of algorithms and protocols in their structure.

Setup: By inputting the security parameter k, the PKG generates its key pair (s, P_{Pub}). Whereby, s is the master secret key and P_{Pub} is the corresponding public key. The PKG also generates and publishes the system public parameters *params*.

Extract: Given a user identity ID, the PKG computes the private key of the user D_{ID} using the master secret key s. The PKG then sends D_{ID} to the user via a secure channel.

Sign: Provided a message m and the private key of the signer D_{ID}, the signer generates an undeniable signature σ on m.

Confirmation: An interactive (or non-interactive) protocol between the signer and the verifier (possibly designated) that takes as input a valid message-signature pair (m, σ), and the private key of the signer D_{ID} and outputs a non-transferable proof on the validity of the message-signature pair (m, σ).

Disavowal: Similar to the Confirmation protocol, except that an invalid signature is provided and the output is a proof on the invalidity of the message-signature pair (m, σ).

2.5.1 (a) Existing Identity-Based Undeniable Signature Schemes

In the following, we provide a background on the existing identity-based undeniable signature schemes and discuss about their features and similarities.

The Han et al. Scheme (Han et al., 2003)

After the proposal of identity-based cryptography (Boneh & Franklin, 2001), the first identity-based undeniable signature scheme was proposed by Han et al. (2003). The authors proposed their scheme as an identity-based confirmer signature scheme. However, based on its structure, their scheme is an undeniable signature scheme. This was first pointed out in (Zhang et al., 2005). As depicted in Table 2.1, Han et al.'s scheme has the shortest signature length among the identity-based undeniable signature schemes in the literature.

Zhang et al. (2005) pointed out two weaknesses in Han et al.'s scheme and mounted two attacks by exploiting the weaknesses. The first attack is the forgery attack, whereby, the adversary exploits the flaw in the Confirmation protocol and generates a valid Confirmation proof for a forgery signature. The second attack is the denial attack, where the weakness in the Disavowal protocol allows a malicious signer to generate Disavowal proof transcripts for valid signatures that she generated honestly.

The Chow Scheme (Chow, 2005)

In 2005, Chow (2005) introduced the concept of verifiable pairing which allows the user to prove the validity/invalidity of a Diffie-Hellman tuple in a non-transferable manner. Employing the new technique in signature schemes enables the signer to prove the existence of the link between her public key and the signature without leaking any information about her private key. With the aim of overcoming the weaknesses and addressing the attacks mounted by Zhang et al. (2005), Chow employed the concept of verifiable pairing in the Confirmation and Disavowal protocols of Han et al.'s (2003) scheme.

The Extract and Sign algorithms of Chow's revised scheme were identical to the original scheme of Han et al. (2003). However, in the Setup algorithm, the PKG generates and publishes one additional public key $P_{inv} = s^{-1}P$, where in Han et al.'s scheme, the PKG generates only one public key P_{Pub}. The inclusion of P_{inv} in the public parameters is necessary in order to incorporate the concept of verifiable pairing. Chow illustrated that both of the attacks mounted by Zhang et al. (2005) could be prevented by employing the concept of verifiable pairing in the Confirmation and Disavowal protocols. The revised scheme with the same Sign algorithm and signature size is considered as the most efficient identity-based undeniable signature scheme in the literature.

The Libert and Quisquater Scheme (Libert & Quisquater, 2004)

Inspired by the work of Galbraith and Mao (2003), Libert and Quisquater (2004) proposed the first provably secure identity-based undeniable signature scheme. As the first provable secure identity-based undeniable signature scheme, their scheme structure as well as the proposed security models was later used as a model for developing other schemes, such as the convertible identity-based undeniable signature scheme of Wu et al. (2008) and Duan's certificateless undeniable signature scheme (Duan, 2008). By taking advantage of the reduction technique proposed by Goh and Jarecki (2003), Libert and Quisquater avoided the security degradation carried by the forking lemma (Pointcheval & Stern, 2000) and relied the unforgeability and invisibility of their scheme on the hardness of the BDH and the DBDH problems respectively.

The Wu et al. Scheme (Wu et al., 2008)

Following the work of Libert and Quisquater (2004), Wu et al. (2008) proposed the first convertible identity-based undeniable signature scheme to the literature. The feature of convertibility, as proposed by Boyar et al. (1991) enables the signer of an undeniable signature to convert her signatures to ordinary digital signatures. This property is attractive in situations where the signed data lose their sensitivity over time (e.g. an agreement signed by two companies on increasing the price of a good in future). Two types of conversion were introduced: selective conversion and universal conversion. The former is the ability of the signer to convert a single signature, and the latter enables the signer to convert all her undeniable signatures to ordinary digital signatures.

the authors formulated the security models of convertible undeniable signatures in an identity-based setting for the first time and relied the unforgeability and invisibility of their scheme on the hardness of the CDH and the DBDH problems respectively.

2.5.2 Certificateless Undeniable Signature Schemes

Certificateless cryptography bridges between traditional public key cryptography and identity-based cryptography. In the following, we first define the notion of certificateless undeniable signature scheme and then, continue to review the existing certificateless undeniable signature schemes in the literature.

Typically, a certificateless undeniable signature scheme is consisted of the following algorithms and protocols.

Setup: Upon inputting a security parameter k, it produces the KGC's key pair (s, P_{Pub}). Where s is the master secret key and P_{Pub} is the corresponding public key. The KGC also generates and publishes the system public parameters *params* in the system.

Set-user-key: Using this algorithm, the user with identity ID picks her secret value $x_{ID} \in X$ (where X denotes the set of valid secret values) and computes the corresponding public key P_{ID}.

Partial-private-key-extract: Upon submitting the user's identity ID (and possibly her public key P_{ID}), the KGC uses the master secret key s to compute the user's partial private key d_{ID}.

Set-private-key: After the user computes her secret value x_{ID} and receives her partial private key d_{ID}, she uses this algorithm to form her private key S_{ID}.

Sign: Provided a message m and the private key of the signer S_{ID}, the signer issues a signature σ on the message m.

Confirmation: An interactive (or non-interactive) protocol between the signer and the verifier (possibly designated) that takes as input a valid message-signature pair (m, σ), and the private key of the signer S_{ID} and outputs a non-transferable proof on the validity of the message-signature pair (m, σ).

Disavowal: Similar to the Confirmation protocol, except that an invalid signature is provided and the output is a proof on the invalidity of the message-signature pair (m, σ).

2.5.2 (a) Existing Certificateless Undeniable Signature Schemes

In the following, we provide a background on the existing certificateless undeniable signature schemes and discuss about their features and similarities.

The Duan Scheme (Duan, 2008)

The first certificateless undeniable signature scheme was put forth by Duan (2008). The author formulated the security model of undeniable signature schemes in a certificateless setting for the first time. Due to the similarities, the proposed scheme along with the security models can be considered as the certificateless version of the work by Libert and Quisquater (2004). In addition, Duan provided a rigorous security proof to relate the unforgeability and invisibility of the proposed scheme to the hardness of the BDH and the DBDH problems respectively.

The Zhao and Ye Scheme (Zhao & Ye, 2012)

Duan's scheme, as the only certificateless undeniable signature scheme in the literature, requires two expensive pairing computations in its Sign algorithm which leads to a longer signature length. With the aim of proposing a more efficient scheme, Zhao and Ye (2012) proposed a new provable secure certificateless undeniable signature scheme. While the new scheme provides a weaker security assurance as compared to Duan's scheme since it is only secure in a weaker security model, it is significantly efficient as it does not need any pairing evaluations in its Sign algorithm and has much shorter signature length.

The new scheme employs Chaum's (1991) Zero-Knowledge Interactive Proofs (ZKIP) in its Confirmation and Disavowal protocols. The authors relied the unforgeability and invisibility of their scheme on the hardness of the CDH and the 3-DDH problems respectively.

2.6 Summary

We started this chapter by providing a brief overview on the fundamentals of provable security and presented an introductory on the mathematical primitives, definitions and some cryptographic primitives that are going to be used throughout this thesis. In addition, we recalled the definition of undeniable signature scheme and some of its main security notions.

In the second part, we provided a brief survey on the development course of certificate-free undeniable signature schemes and highlighted the additional features of the existing schemes in the literature and discussed about their security. Comparing to other variations of certificate-free privacy preserving signatures (e.g. designated verifier signatures, designated confirmer signatures, etc.), the research on certificate-free undeniable signatures has been quite slow as there are only three secure identity-based and two certificateless undeniable signature schemes in the literature.

CHAPTER 3

CRYPTANALYSIS AND DESIGN OF IDENTITY-BASED UNDENIABLE SIGNATURE SCHEMES

3.1 Introduction

The idea of identity-based cryptography was first envisioned by Shamir (1985) with the aim of overcoming the well-documented issues in traditional public key cryptography. In such systems, the public key of the user is derived from her publicly available information (e.g. an IP address of a network host, an email address associated with a user, etc.) and thus, its authenticity can be easily verified without the need of certificates.

The intention of ordinary digital signatures is to be universally verifiable, where any user who has access to the signer's public key can verify the validity of her signatures. Although the self-authenticating aspect of ordinary digital signatures may become a security or privacy concern for signers. Chaum and van Antwerpen (1989) proposed the notion of undeniable signature schemes to limit the self-authenticating property of ordinary digital signatures. Undeniable signature schemes provide the signer with a special ability to decide who can be convinced from the validity/invalidity of her signature. Since the introduction of undeniable signature schemes, there has been a wide range of research covering a variety of different features and security levels for such schemes (Boyar et al., 1991; Chaum, 1995; Duan, 2008; Galbraith & Mao, 2003; Kurosawa & Heng, 2005; Kurosawa & Takagi, 2006; Libert & Quisquater, 2004; Monnerat & Vaudenay, 2004, 2006; Qiong & Wong, 2009).

Incontestably, the efficiency of cryptographic schemes is one of the key criteria when such schemes are to be implemented in real world scenario. Table 3.1 below compares the efficiency of the proposed identity-based undeniable signature schemes in the literature. It compares the computations needed in the Sign algorithm and the signature size (in bits) of the schemes.

Table 3.1: Efficiency of Identity-Based Undeniable Signature Schemes

Schemes	Underlying Assumption		Signature Generation	Signature Length (in bits)
	Unforgeability	Invisibility		
Han et al. (2003)	Broken (Zhang et al., 2005)		$pm + pa$	$2\|\mathbb{G}_1\| \approx 320$
Libert and Quisquater (2004)	BDH	DBDH	pe	$\|\mathbb{G}_2\| + \|r\| \approx 1124$
Wu et al. (2008)	CDH	DBDH	$1pe + 1pa + 1pm$	$\|\mathbb{G}_2\| + 2\|\mathbb{G}_1\| \approx 1344$

In Table 3.1 above, *pe* denotes pairing evaluation and *pm* and *pa* denote point multiplication and point addition (in group $\mathbb{G}_1$), respectively.

Chow (2005) proposed the concept of verifiable pairing which allows the user to prove the validity/invalidity of a Diffie-Hellman tuple in a non-transferable manner. Moreover, by modifying the Confirmation and Disavowal protocols and incorporating the new technique (i.e. verifiable pairing) in the Han et al. (2003) scheme, Chow addressed both of the attacks mounted by Zhang et al. (2005). Hence, the new scheme with the same Sign algorithm and signature size is considered as the most efficient identity-based undeniable signature scheme in the literature.

Contributions

In this chapter, we first point out two weaknesses in Chow's (2005) scheme and mount two attacks by exploiting them. In our first attack, we target the unforgeability of the scheme and show that by exploiting the flaw in the signature structure of Chow's scheme, the adversary is able to mount a universal forgery with a known message attack. In our second attack, by exploiting the flaw in the structure of the Confirmation and Disavowal protocols of Chow's scheme, we show that a malicious verifier is able to violate the notion of non-transferability of the scheme by transferring his knowledge on the validity/invalidity of a message-signature pair to any third party.

Proposing short and efficient signature schemes has been a promising line of research (Boneh, Lynn, & Shacham, 2001; Zhang, Safavi-Naini, & Susilo, 2004; Katz & Wang, 2003). Efficient signature schemes with short signature length are required in devices with low computation power which are operating in low bandwidth communication environments. In the existing secure identity-based undeniable signature schemes (Libert & Quisquater, 2004; Wu et al., 2008), the Sign algorithm needs at least one pairing evaluation which results in generating relatively longer signature length. In this chapter, we propose a provably secure short and efficient identity-based undeniable signature scheme. The signature generation in our scheme does not need any pairing evaluation, and the signature size of our scheme is significantly smaller than the ones in the existing secure schemes (Libert & Quisquater, 2004; Wu et al., 2008). Moreover, we prove the security of our scheme in the random oracle model by relying its unforgeability and invisibility on the hardness of the CDH problem and 3-DDH problem respectively.

The rest of the chapter is organised as follows. In Section 3.2, we define the security models of identity-based undeniable signature schemes. In Section 3.3, we recall Chow's scheme in detail and propose our attacks on the proposed scheme. In Section 3.4, we propose our concrete scheme, provide a formal security analysis for the proposed scheme and discuss about its efficiency and extensions. Finally, we conclude this chapter in Section 3.5.

3.2 Security Models of Identity-Based Undeniable Signature Schemes

Following the existing works on provably secure undeniable signature schemes (Chaum & van Antwerpen, 1989; Chaum, 1991; Libert & Quisquater, 2004; Kurosawa & Heng, 2005), in this section, we formulate the security models of identity-based undeniable signature schemes.

The existential unforgeability of an identity-based undeniable signature scheme is defined as follows.

Definition 3.1. *An identity-based undeniable signature scheme is existentially un-*

forgeable under adaptive chosen message and identity attacks if no PPT adversary $\mathcal{A}$ has a non-negligible advantage in the following game:

1. The challenger $\mathcal{C}$ initiates the Setup algorithm and sends the system public parameters *params* to $\mathcal{A}$.
2. $\mathcal{A}$ performs a series of queries:

- **Extract query:** Given an identity ID, $\mathcal{A}$ receives the private key D_{ID} associated with ID.
- **Sign query:** $\mathcal{A}$ generates a message m and an identity ID and queries the Sign oracle for a signature σ on the pair (m, ID).
- **Confirmation/Disavowal query:** $\mathcal{A}$ creates a tuple (m, σ, ID) and receives a non-transferable proof on the validity/invalidity of the produced message-signature pair (m, σ) for the identity ID.

At the end of the game, $\mathcal{A}$ outputs the forgery tuple (ID^*, m^*, σ^*). $\mathcal{A}$ wins the game if the identity ID^* was never queried to the Extract oracle, and the pair (ID^*, m^*) was never queried to the Sign oracle.

Provided a message-signature pair (m, σ) and the identity ID of the signer, the notion of invisibility for an identity-based undeniable signature scheme implies the inability of a dishonest verifier to decide on the validity or invalidity of the signature without the help of the signer. The following definition formally defines the notion of invisibility for an identity-based undeniable signature scheme.

Definition 3.2. *An identity-based undeniable signature scheme is considered to fulfil the notion of invisibility under adaptive chosen message and identity attacks if no PPT distinguisher $\mathcal{D}$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ initiates the Setup algorithm and sends the system public parameters *params* to $\mathcal{D}$.
2. $\mathcal{D}$ is allowed to perform queries (polynomially bounded) as in Definition 3.1.

3. After the first round of queries, $\mathcal{D}$ outputs a message-identity pair (m^*, ID^*), wherein ID^* was never queried to the Extract oracle and requests a challenge signature on (m^*, ID^*). The challenge signature σ^* is issued by $\mathcal{C}$ based on the outcome of a random coin toss $b \in \{0,1\}$. If $b = 1$, σ^* is generated naturally by initiating the Sign oracle. Otherwise, σ^* is chosen randomly from the signature space S.
4. $\mathcal{D}$ performs the second round of queries with the following restrictions:
   - No Sign query is allowed on the pair (m^*, ID^*).
   - No Extract query on ID^*.
   - No Confirmation/Disavowal query on the tuple (m^*, σ^*, ID^*).

Finally, $\mathcal{D}$ outputs a guess b'.

The distinguisher $\mathcal{D}$ wins the game if $b' = b$.

The notion of anonymity for an identity-based undeniable signature schemes implies that provided a valid message-signature pair (m, σ) and the identities of two possible signers ID_0 and ID_1, it should be infeasible for a dishonest verifier to decide who generated the signature. The following definition formally defines the notion of anonymity for an identity-based undeniable signature scheme.

Definition 3.3. *An identity-based undeniable signature scheme is considered to fulfil the notion of anonymity under adaptive chosen message and identity attacks if no PPT distinguisher $\mathcal{D}$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ initiates the Setup algorithm and sends the system public parameters *params* to the distinguisher $\mathcal{D}$.
2. $\mathcal{D}$ is allowed to perform queries (polynomially bounded) as in Definition 3.1.
3. After the first round of queries, $\mathcal{D}$ outputs a tuple (m^*, ID_0, ID_1), wherein ID_0 or ID_1 were never queried to the Extract oracle and requests a challenge signature. The challenger then flips a hidden and random coin $b \in \{0,1\}$ and computes the challenge signature σ^* using the private key associated with the identity ID_b.

4. $\mathcal{D}$ performs the second round of queries with the following restrictions:
   - No Sign query is allowed on message m^* for identities ID_0 and ID_1.
   - No Extract query on identities ID_0 or ID_1.
   - No Confirmation/Disavowal query on (m^*, σ^*, ID_0) or (m^*, σ^*, ID_1).

Finally, $\mathcal{D}$ outputs a guess b'.

The distinguisher $\mathcal{D}$ wins the game if $b' = b$.

Based on the work of Galbraith and Mao (2003), the notion of anonymity is equivalent to the notion of invisibility (in the sense of the security model in Definition 3.2). Consequently, we can use the same technique as proposed by Galbraith and Mao (2003) to prove the anonymity of our scheme under the hardness of the 3-DDH problem.

3.3 Cryptanalysis on Chow's Identity-Based Undeniable Signature Scheme

First, we recall the construction of Chow's scheme and then show our attacks by exploiting the weaknesses in its structure.

3.3.1 The Chow Scheme (Chow, 2005)

Chow's scheme consists of the following algorithms and protocols.

Setup: Upon inputting the security parameter k, this algorithm generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q \geq 2^k$, picks $P \in \mathbb{G}_1$ randomly as a generator of $\mathbb{G}_1$ and selects an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. It also chooses two cryptographic hash functions: $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and sets A to be a large number (about 10^{20}) and $[A] = \{1,2,3,\ldots,A\}$. Next, the PKG picks $s \in \mathbb{Z}_q$ at random as its secret key and calculates $P_{Pub} = sP$ and $P_{inv} = s^{-1}P$ as the corresponding public keys. Lastly, the PKG publishes the system public parameters as $params : (q, \mathbb{G}_1, \mathbb{G}_2, e(.,.), P, P_{Pub}, P_{inv}, H, H_1)$.

Extract: Upon submitting the user's identity ID, the PKG calculates the user's private key pair as (D_{ID}, L_{ID}) where $D_{ID} = sQ_{ID} = sH_1(ID)$ and $L_{ID}= s^{-1}\ Q_{ID}= s^{-1}H_1(ID)$.

Sign: In order for the signer with identity ID_S to issue a signature on message $m \in \{0,1\}^*$, she picks $k \in \mathbb{Z}_q$ randomly and computes the signature as $\sigma = \{R = kP, \mathcal{S} = k^{-1}D_S + H(m)L_S\}$.

Confirmation: Upon receiving a valid message-signature pair $(m, \sigma = (R, \mathcal{S}))$, the signer with identity ID_S uses her private key L_S to verify the validity of the signature as follows.

1. The verifier initiates the protocol and starts by choosing $y \in \mathbb{Z}_q$ and $x \in [A]$ at random and sends $C_1 = xyR$ and $C_2 = xyP$ to the signer.
2. Upon receiving C_1 and C_2, the signer chooses $z \in \mathbb{Z}_q$ at random and computes $X = e(R + P_{Pub}, P - L_S), T = z^{-1}C_1, U = zP_{inv}$ and $V = zL_S$ and sends (X, T, U, V) to the verifier.
3. The verifier first checks if the equalities $e(P,V) = e(Q_S, U)$ and $e(T,U) = e(P_{inv}, C_1)$ hold, he computes $W = e(V,T)$, and accepts the proof if and only if $e(R, \mathcal{S})^x = e(P_{Pub}, Q_S)W^{H(m)y^{-1}}$ and $W^{y^{-1}}X^x e(P, Q_S)^x = e(R + P_{Pub}, P)^x$ hold.

Disavowal: Upon receiving an invalid message-signature pair $(m, \sigma = (R, \mathcal{S}))$, the signer with identity ID_S uses her private key L_S to disavow the validity of the signature as follows.

1. The verifier initiates the protocol and starts by choosing $y \in \mathbb{Z}_q$ and $x \in [A]$ at random and sends $C_1 = xyR$ and $C_2 = xyP$ to the signer.
2. Upon receiving C_1 and C_2, the signer chooses $z \in \mathbb{Z}_q$ at random and computes $T = z^{-1}C_1, U = zP_{inv}$ and $V = zL_S$ and sends (T, U, V) to the verifier.
3. The verifier first checks if the equalities $e(P,V) = e(Q_S, U)$ and $e(T,U) = e(P_{inv}, C_1)$ hold, he computes $B = \frac{e(C_1, \mathcal{S})}{e(xyP_{Pub}, Q_S)W^{H(m)}}$ and sends $C = B^{y^{-1}}$ to the signer.
4. The signer calculates $C' = \frac{e(R, \mathcal{S})}{e(P_{Pub}, Q_S)e(R, L_S)^{H(m)}}$ and by using C', she computes x' from C and sends x' to the verifier.
5. The verifier will only accept the proof if $x' = x$ holds.

3.3.2 Universal Forgery with a Known Message Attack

Given the adversary has obtained a message-signature pair $(m, \sigma = (R, \mathcal{S}))$ valid for the signer with identity ID_S, he can perform the following steps to mount a universal forgery with a known message attack and forge signatures on any arbitrary message m^* of his choice.

- Compute $\alpha = \frac{H(m)}{H(m^*)} \in \mathbb{Z}_q$ and α^{-1}.
- Compute the forgery signature as $\sigma^* = (R^*, S^*) = (\alpha R, \alpha^{-1} S)$.

Lemma 3.1. *Given the message-signature pair $(m, \sigma = (R, \mathcal{S}))$ is valid for the signer with identity ID_S, the forgery message-signature pair $(m^*, \sigma^* = (R^*, \mathcal{S}^*))$ is also valid. Therefore, the forgery pair (m^*, σ^*) would pass the Confirmation protocol and the signer has no way to disavow the validity of the forgery signature σ^*.*

Proof. Based on the Sign algorithm of Chow's scheme, $R = kP$ for random $k \in \mathbb{Z}_q$ and $\mathcal{S} = k^{-1} D_{ID} + H(m) L_{ID}$ (where D_S and L_S are the private keys of the signer, and $H(m) \in \mathbb{Z}_q$). From the obtained valid message-signature pair $(m, \sigma = (R, \mathcal{S}))$, we have the following.

$$R^* = \alpha R = \alpha k P \tag{3.1}$$

$$\mathcal{S}^* = \alpha^{-1} \mathcal{S} = (\alpha k)^{-1} D_S + \alpha^{-1} H(m) L_S \tag{3.2}$$

Note that $\alpha = \frac{H(m)}{H(m^*)}$ and $\alpha^{-1} = \frac{H(m^*)}{H(m)}$ and therefore, we have:

$$\begin{aligned} \mathcal{S}^* &= (\alpha k)^{-1} D_S + \frac{H(m^*)}{H(m)} H(m) L_S && (3.3) \\ &= (\alpha k)^{-1} D_S + H(m^*) L_S && (3.4) \end{aligned}$$

Let $k^* = \alpha k$ then,

$$\begin{aligned} \sigma^* &= (k^* P, ({k^*}^{-1} D_S + H(m^*) L_S) && (3.5) \\ &= (R^*, \mathcal{S}^*) && (3.6) \end{aligned}$$

Therefore, the forgery signature σ^* is valid on message m^*. It can be easily shown that the forgery message-signature pair $(m^*, \sigma^* = (R^*, \mathcal{S}^*))$ will pass the Confirmation protocol while the signer ID_S has a negligible chance (roughly around 10^{-20}) to disavow its validity. □

3.3.3 Attack on the Non-Transferability

In the following, we show that the notion of the non-transferability of the revised scheme by Chow (2005) is overlooked. More precisely, we show that the employment of verifiable pairing in the Confirmation and Disavowal protocols of the revised scheme enables a dishonest verifier to transfer his knowledge of the validity/invalidity of an undeniable signature to a third party. This violates the vital property of non-transferability of the Confirmation and Disavowal protocols of undeniable signature schemes. The dishonest verifier works as follows in order to mount the attack to transfer his knowledge of the validity of a message-signature pair $(m, \sigma = (R, \mathcal{S}))$ to a third party, after engaging with the signer in the Confirmation protocol. Here, we only show the attack on the Confirmation protocol of the scheme, the same technique can be applied for the Disavowal protocol.

1. The verifier picks at random $x' \in [A]$ and sets the value of y'=$H(m,R,\mathcal{S},ID_S,x') \in \mathbb{Z}_q$, where ID_S is the identity of the signer. Next, he sets the values of $C_1^* = x'y'R$ and $C_2^* = x'y'P$ and sends C_1^* and C_2^* to the signer.
2. The signer chooses a random $z' \in \mathbb{Z}_q$ and computes $T' = z'^{-1}C_1^*$, $U' = z'P_{inv}$, $V' = z'L_S$ and $X' = e(R+P_{Pub}, P-L_S)$ and sends (T', U', V', X') to the verifier.

The dishonest verifier transfers his knowledge of the proof by sending $(C_1^*, C_2^*, T', U', V', X', x')$ to a third party Carol. In order to verify the proof, Carol woks as follows.

1. She first checks if $C_1^* = x'H(m,R,\mathcal{S},ID_S,x')R$ and $C_2^* = x'H(m,R,\mathcal{S},ID_S,x')P$ hold, she verifies the validity of the tuple (T', U', V') by checking $e(V', P) = e(U', Q_S)$ and $e(U', T') = e(P_{inv}, C_1^*)$. If both of the equalities hold, then she knows that no one except the signer with identity ID_S could have generated such tuple.
2. Next, she computes $W' = e(V', P)$ and accepts the validity of the message-signature pair $(m, \sigma = (R, \mathcal{S}))$ if and only if $e(R, \mathcal{S})^{x'} = e(P_{pub}, Q_S)^{x'} W'^{H(m)y'^{-1}}$

and $W'^{y'^{-1}} X'^{x'} e(P, Q_S)^{x'} = e(R + P_{Pub}, P)^{x'}$ hold.

In Zero-Knowledge Interactive Proof (ZKIP) (Chaum, 1991; Ogata, Kurosawa, & Heng, 2006), the verifier selects two random values (similar to first step in the Confirmation protocol) and forms the challenge value. However, before the last move, the verifier sends the random values for the signer and the signer decommits if and only if the challenge is formed correctly. In the above protocol, on the other hand, there is no way for the signer to check the correctness of C_1 and C_2 which enables the dishonest verifier to form them dishonestly and mount the attack.

3.4 The Proposed Scheme

In this section, we propose our efficient identity-based undeniable scheme, provide a formal security proof to rely its security on the hardness of some well-known problems and lastly, discuss about its efficiency and extensions.

Setup: By taking as input a security parameter k, the PKG generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^k$, and an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Next, it picks an arbitrary generator $P \in \mathbb{G}_1$, a random secret $s \in \mathbb{Z}_q$, and sets the system public key as $P_{Pub} = sP$. Finally, the PKG chooses four cryptographic hash functions where $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_1 \times \{0,1\}^* \rightarrow \mathbb{Z}_q$, and $H_3, H_4 : \mathbb{G}_1 \times \ldots \times \{0,1\}^* \rightarrow \mathbb{Z}_q$, and publishes the system public parameters as $params = (q, \mathbb{G}_1, \mathbb{G}_2, e(.,.), P, P_{Pub}, H_1, H_2, H_3, H_4)$.

Extract: Provided the user's identity ID, the system public parameters $params$ and the master secret key s, the PKG computes $Q_{ID} = H_1(ID)$ and outputs the user's private key as $D_{ID} = sQ_{ID}$.

Sign: Given a message m to be signed, the signer picks $r_1, r_2 \in \mathbb{Z}_q$ at random to compute $\mathcal{S}_1 = r_1 P$, $\mathcal{S}_2 = r_2 P$, $\mu = H_2(\mathcal{S}_1, \mathcal{S}_2, m)$ and $\mathcal{S}_3 = (\mu r_1 + r_2) D_S$ and forms the signature as $\sigma = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$.

Confirmation: Given a valid message-signature pair $(m, \sigma = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3))$, the alleged signer (with identity ID_S) takes the following steps in order to create a non-transferable proof for the designated verifier with identity ID_V.

1. Compute $Q_V = H_1(ID_V)$ and choose $v \in \mathbb{Z}_q$ and $U \in \mathbb{G}_1$ at random to compute $W = e(P,U)e(P_{Pub},Q_V)^v$.
2. Choose a random $R \in \mathbb{G}_1$ to compute:
$$Z_1 = e(P,R) \tag{3.7}$$
$$Z_2 = e(\mu\mathcal{S}_1 + \mathcal{S}_2, R) \tag{3.8}$$
3. Compute $h_C = H_3(W, Z_1, Z_2, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, m)$ and $T = R - (h_C + v)D_S$ and form the proof as (U, v, h_C, T).

After receiving the proof (U, v, h_C, T), the verifier (with identity ID_V) computes $\mu = H_2(\mathcal{S}_1, \mathcal{S}_2, m)$ and $Q_S = H_1(ID_S)$ to form $W' = e(P,U)e(P_{Pub},Q_V)^v$, $Z_1' = e(P,T)e(P_{Pub}, Q_S)^{(h_C+v)}$, and $Z_2' = e(\mu\mathcal{S}_1 + \mathcal{S}_2, T)e(P,\mathcal{S}_3)^{(h_C+v)}$. The verifier accepts the proof only if $h_C = H_3(W', Z_1', Z_2', \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, m)$.

Disavowal: Given an invalid message-signature pair $(m, \sigma = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3))$, the alleged signer (with identity ID_S) takes the following steps in order to generate a non-transferable proof for the designated verifier with identity ID_V.

1. Compute $Q_V = H_1(ID_V)$ and choose $v \in \mathbb{Z}_q$ and $U \in \mathbb{G}_1$ at random to compute $W = e(P,U)e(P_{Pub},Q_V)^v$.
2. Choose a random $\tau \in \mathbb{Z}_q$ and compute $\mu = H(\mathcal{S}_1, \mathcal{S}_2, m)$ to calculate $C = (\frac{e(\mu\mathcal{S}_1+\mathcal{S}_2,D_S)}{e(P,\mathcal{S}_3)})^\tau$.
3. Next, the signer has to prove her knowledge of the tuple $(\omega, X) \in \mathbb{Z}_q \times \mathbb{G}_1$, whereas, $C = \frac{e(\mu\mathcal{S}_1+\mathcal{S}_2,X)}{e(P,\mathcal{S}_3)^\omega}$ and $1 = \frac{e(P,X)}{e(P_{Pub},Q_S)^\omega}$. In order to do so, she computes as follows.
   a) Pick at random $j \in \mathbb{Z}_q$ and $Y \in \mathbb{G}_1$ to compute:
   $$N_1 = \frac{e(P,Y)}{e(P_{Pub},Q_S)^j} \tag{3.9}$$
   $$N_2 = \frac{e(\mu\mathcal{S}_1 + \mathcal{S}_2, Y)}{e(P,\mathcal{S}_3)^j} \tag{3.10}$$
   b) Calculate $h_D = H_4(C, W, N_1, N_2, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, m)$ and set $K = Y - (h_D + v)X$ and $a = j - (h_D + v)\omega$ to form the proof as (C, U, v, h_D, K, a).

The verifier (with identity ID_V) verifies the proof (C,U,v,h_D,K,a) as follows. He first checks if $C=1$, he will reject the proof. Otherwise, he computes $W' = e(P,U)e(P_{Pub},Q_V)^{v}$, $N_1' = \frac{e(P,K)}{e(P_{Pub},Q_S)^{a}}$, $N_2 = \frac{e(\mu\mathcal{S}_1+\mathcal{S}_2,K)}{e(P,\mathcal{S}_3)^{a}}C^{(h_D+v)}$ and accepts the proof only if $h_D = H_4(C,W'$, $N_1', N_2', \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, m)$.

3.4.1 Security Analysis

The method used in the Confirmation and Disavowal protocols of our scheme is the pairing-based version of the non-interactive designated verifier proofs proposed by Jakobsson, Sako, and Impagliazzo (1996). It is quite easy to show the completeness of the proofs. In the following, we show that both the Confirmation and Disavowal protocols of our scheme possess the property of non-transferability and soundness.

3.4.1 (a) Soundness

In order to prove the soundness of the Confirmation and Disavowal protocols, we need to prove that no one without the knowledge of the private key of the signer D_S is able to generate such proofs on the validity/invalidity of any message-signature pair (m,σ). In the following, we prove this property for the Confirmation protocol of our scheme. We remark that the same method could be applied to prove the soundness of the Disavowal protocol.

Let us assume that the signer is able to generate two values $T_1$and T_2 for a valid Confirmation proof transcript by forming W, Z_1, and Z_2, using two hash values h_{C_1} and h_{C_2}. Then, we have $e(P,(T_1-T_2))^{(h_{C_1}-h_{C_2})^{-1}} = e(P_{Pub},Q_S)$ and $e(\mu\mathcal{S}_1+\mathcal{S}_1,(T_1 - T_2))^{(h_{C_1}-h_{C_2})^{-1}} = e(P,\mathcal{S}_2)$ which directly implies the link between the signer's private key and the signature. Only with the knowledge of the signer's private key one can compute $T_1$and T_2 in such a way that the inverses of $e(P_{Pub},Q_S)$ with respect to P and $e(P,\mathcal{S}_2)$ with respect to $\mu\mathcal{S}_1+\mathcal{S}_1$ are equal.

3.4.1 (b) Non-Transferability

To prove the non-transferability of the Confirmation and Disavowal protocols of our scheme, we need to show that the designated verifier can use his private key to generate proofs on the validity/invalidity of signatures which are indistinguishable from the ones generated by the alleged signer.

Provided a message-signature pair $(m,\sigma=(\mathcal{S}_1,\mathcal{S}_2))$ (for the alleged signer with identity ID_S) along with a Confirmation proof transcript (U,v,h_C,T) for the designated verifier (with identity ID_V), he would use his private key D_V to exploit the trapdoor function $W=(P,U)e(P_{Pub},Q_V)^v$ in order to form a new proof. The designated verifier starts by picking $i\in\mathbb{Z}_q$ and $T,N\in\mathbb{G}_1$ at random so as to compute $W=e(P,N)$, $Z_1=e(P,T)e(P_{Pub},Q_S)^i$ and $Z_2=e(\mu\mathcal{S}_1+\mathcal{S}_1,T)e(P,\mathcal{S}_2)^i$ to form $h'_C=H_3(W,Z_1,Z_2,\mathcal{S}_1,\mathcal{S}_2,m)$. Next, he computes $v=i-h'_C$ and $U=N-vD_V$ to form a new proof as (U,v,h'_C,T). It is easy to show that the new proof (U,v,h'_C,T) can be verified to be identical to the original proof, issued using the signer's private key. Using the same trapdoor function $W=(P,U)e(P_{Pub},Q_V)^v$, we can show that the Disavowal protocol of our scheme possesses the property of non-transferability.

3.4.1 (c) Non-Impersonation

The notion of non-impersonation, as discussed in the previous chapter, prevents the adversary from initiating the confirmation/disavowal protocol on behalf of the signer without having knowledge on her private key. As it is explicitly highlighted in the same paper that this security notion is introduced (Kurosawa & Heng, 2005), undeniable schemes that make use of non-interactive designated verifier proofs of Jakobsson et al. (1996) in the body of their confirmation/disavowal protocol are secure against impersonation attacks. Therefore, since we are using the same method in the Confirmation and Disavowal protocols of our scheme, our scheme is secure against such attacks as well.

Theorem 3.1. *If there exists an adversary $\mathcal{A}$ that can submit q_E extract queries, q_S sign queries, and q_{H_i} queries to the random oracle H_i for $i\in\{1,2,3,4\}$ and be able* *to succeed in an existential forgery (win the game defined in Definition 3.1) against our proposed scheme with non-negligible success probability $\varepsilon_{\mathcal{A}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{A}$ to solve a random instance (P, aP, bP) of the CDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geq (1 - \frac{1}{q_{H_1}})^{q_E}(1 - \frac{1}{q_{H_1}})^{q_S}\frac{1}{q_{H_2}}\varepsilon_{\mathcal{A}}$$

Proof. We show that if there exists an adversary $\mathcal{A}$ which is able to win the unforgeability game as defined in Definition 3.1 with probability $\varepsilon_{\mathcal{A}}$, then there exists another PPT algorithm $\mathcal{C}$ which can use $\mathcal{A}$ to solve a random instance (P, aP, bP) of the CDH problem with probability $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ acts as $\mathcal{A}$'s simulator, it first provides $\mathcal{A}$ with the system wide public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, e(.,.), P, P_{Pub}, H_1, H_2, H_3, H_4)$ where $P_{Pub} = aP$ and a is unknown to $\mathcal{C}$. Next, $\mathcal{C}$ chooses a random number $\pi \in \{1, 2, \ldots, q_{H_1}\}$.

$\mathcal{C}$ answers to $\mathcal{A}$'s queries by keeping lists κ_i for $i \in \{1, 2, 3, 4\}$ as follows:

Query on $H_1(ID_i)$: In order to handle queries on H_1 for an identity ID_i, $\mathcal{C}$ first checks if $i = \pi$, it returns $H_1(ID_i) = bP$ and records $(ID_\pi, bP, \perp)$ in κ_1. Otherwise, it chooses a random $\alpha_i \in \mathbb{Z}_q$ and returns $H_1(ID_i) = \alpha_i P$ and stores $(ID_i, \alpha_i P, \alpha_i)$ in κ_1.

Query on $H_2(\mathcal{S}_{1_i}, \mathcal{S}_{2_i}, m_i)$: $\mathcal{C}$ responds to H_2 queries completely at random by returning a random value $\mu_i \in \mathbb{Z}_q$ and inserting $(\mathcal{S}_{1_i}, \mathcal{S}_{2_i}, m_i, \mu_i)$ in κ_2.

Query on H_3, H_4 : $\mathcal{C}$ responds to H_3 and H_4 oracles randomly and stores the responses in κ_3 and κ_4 respectively.

Extract query: Upon receiving an extract query on an identity ID_i, $\mathcal{C}$ checks if $i = \pi$, it terminates and outputs $\perp$. Otherwise, if $i \neq \pi$, it scans κ_1 to find $(ID_i, \alpha_i P, \alpha_i)$ and outputs the private key as $D_{ID_i} = \alpha_i P_{Pub}$.

Sign query: $\mathcal{A}$ can query for a signature on any tuple (m_i, ID_i), where m_i is the message to be signed and ID_i is the identity of the alleged signer. Upon receiving such query, $\mathcal{C}$ checks if $i = \pi$, it terminates and outputs $\perp$. Otherwise, if $i \neq \pi$, $\mathcal{C}$ scans κ_1 to find the tuple $(ID_i, \alpha_i P, \alpha_i)$, picks $r_{1_i}, r_{2_i} \in \mathbb{Z}_q$ at random, and computes $\mathcal{S}_{1_i} = r_{1_i}P, \mathcal{S}_{2_i} = r_{2_i}P$, $\mu_i = H_2(\mathcal{S}_{1_i}, \mathcal{S}_{2_i}, m_i)$, and $\mathcal{S}_{3_i} = (\mu_i r_{1_i} + r_{2_i})(\alpha_i P_{Pub})$ to output the signature as $\sigma_i = (\mathcal{S}_{1_i}, \mathcal{S}_{2_i}, \mathcal{S}_{3_i})$.

Confirmation/Disavowal query: $\mathcal{A}$ is allowed to query for the confirmation/disavowal proof on any message-signature pair $(m_i, \sigma_i^{'} = (\mathcal{S}_{1_i}^{'}, \mathcal{S}_{2_i}^{'}, \mathcal{S}_{3_i}^{'}))$ for an alleged signer with identity ID_i. Upon receiving such query, $\mathcal{C}$ scans κ_1 to find the tuple $(ID_i, \alpha_i P, \alpha_i)$ and computes $\mu_i^{'} = H_2(m_i, \mathcal{S}_{1_i}^{'}, \mathcal{S}_{2_i}^{'})$. Next, $\mathcal{C}$ checks if $e(\mathcal{S}_{2_i}^{'}, P) = e(\mu_i^{'}\mathcal{S}_{1_i}^{'} + \mathcal{S}_{2_i}^{'}, \alpha_i P_{Pub})$, it simulates the Confirmation protocol. Otherwise, it simulates the Disavowal protocol for $\mathcal{A}$.

In order to simulate the Confirmation or Disavowal protocols, $\mathcal{C}$ uses its control over random oracles to form the proof as follows. Here, we only demonstrate the proof for Confirmation protocol, the proof of Disavowal protocol can be generated similarly. To generate a confirmation proof on a tuple $(m_i, ID_i, \sigma_i = (\mathcal{S}_{1_i}, \mathcal{S}_{2_i}, \mathcal{S}_{3_i}))$ for a designated verifier with identity ID_V, $\mathcal{C}$ picks $U, T \in \mathbb{G}_1$ and $v, h_C \in \mathbb{Z}_q$ randomly and computes $W = e(P, U)e(P_{Pub}, Q_V)^v$, $Z_1 = e(P, T)e(P_{Pub}, Q_S)^{(h_{C_i}+v)}$, $\mu_i = H_2(m_i, \mathcal{S}_{1_i}, \mathcal{S}_{2_i})$, and $Z_2 = e(\mu_i \mathcal{S}_{1_i} + \mathcal{S}_{2_i}, T)e(P, \mathcal{S}_{3_i})^{(h_{C_i}+v)}$. It then sets the value of $H_3(W, Z_1, Z_2, \mathcal{S}_{1_i}, \mathcal{S}_{2_i}, \mathcal{S}_{3_i}, m_i)$ as h_{C_i} and sends the proof as (U, v, h_{C_i}, T). $\mathcal{C}$ may fail in simulating the Confirmation/Disavowal protocol if the same tuple $(W, Z_1, Z_2, \mathcal{S}_{1_i}, \mathcal{S}_{2_i}, \mathcal{S}_{3_i}, m_i)$ was queried to H_3 before, however, this may only happen with a negligible probability.

At the end of the game, $\mathcal{A}$ outputs a tuple $(m^*, ID^*, \sigma^* = (\mathcal{S}_1^*, \mathcal{S}_2^*, \mathcal{S}_3^*))$, where σ^* is a valid signature on message m^* for the alleged signer with identity ID^*. $\mathcal{A}$ would successfully win the unforgeability game (as defined in Definition 3.1) if σ^* was never outputted from the Sign oracle and ID^* was never queried to the Extract oracle. After $\mathcal{A}$ outputs the forgery tuple (m^*, ID^*, σ^*), $\mathcal{C}$ checks if $ID^* \neq ID_\pi$, it terminates and outputs $\perp$. Otherwise, if $ID^* = ID_\pi$, it runs $\mathcal{A}$ again with same random tape only different choice of H_2 to get another forgery tuple $(m^*, \sigma^{*'} = (\mathcal{S}_1^*, \mathcal{S}_2^*, \mathcal{S}_2^{*'}), ID^*)$. Based on the forking lemma (Pointcheval & Stern, 2000), the second forgery can be obtained with the overwhelming probability, and since both signatures , σ^* and $\sigma^{*'}$ are valid with the respect to the hash function H_2, we have $\mathcal{S}_3^* = \mathcal{S}_3^{*'}$. Therefore, $\mathcal{C}$ can obtain $\mathcal{S}_3^* - \mathcal{S}_3^{*'} = (hr_1^* + r_2^*)D_{ID^*} - (h^{'}r_1^* + r_2^*)D_{ID^*}$ where $D_{ID^*} = abP$ and compute the solution of a random instance (P, aP, bP) of the CDH problem by computing $abP = (h-h^{'})^{-1}(r_1^{*^{-1}}(\mathcal{S}_3^{*} - \mathcal{S}_3^{*^{'}}))$.

If $\mathcal{C}$ does not fail during the simulation process, it will be able to solve the CDH problem with the probability $\frac{1}{q_{H_2}}$. $\mathcal{C}$ may fail in the simulation process if $\mathcal{A}$ queries the Extract oracle, or the Sign oracle with queries that are associated with identity ID_{π}. The probability that none of the failure cases occur for $\mathcal{C}$ is $(1-\frac{1}{q_{H_1}})^{q_E}(1-\frac{1}{q_{H_1}})^{q_S}$. Therefore, given $\mathcal{A}$ is able to successfully forge signature in our scheme with probability $\varepsilon_{\mathcal{A}}$, then $\mathcal{C}$ can solve a random instance (P, aP, bP) of the CDH problem with probability $\varepsilon_{\mathcal{C}} \geq (1-\frac{1}{q_{H_1}})^{q_E}(1-\frac{1}{q_{H_1}})^{q_S}\frac{1}{q_{H_2}}\varepsilon_{\mathcal{A}}$. □

Theorem 3.2. *If there exists a distinguisher $\mathcal{D}$ that can submit q_E extract queries, q_S sign queries, and q_{H_i} queries to the random oracle H_i for $i \in \{1,2,3,4\}$ and be able to breach the invisibility property (win the game defined in Definition 3.2) of our proposed scheme with non-negligible success probability $\varepsilon_{\mathcal{D}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{D}$ to solve a random instance (P, aP, bP, cP, Z) of the 3-DDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geq (1-\frac{1}{q_{H_1}})^{q_E}(1-\frac{1}{q_{H_1}})^{q_S}\varepsilon_{\mathcal{D}}$$

Proof. We show that if there exists a distinguisher $\mathcal{D}$ which is able to win the invisibility game as defined in Definition 3.2 with probability $\varepsilon_{\mathcal{D}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{D}$ to solve a random instance (P, aP, bP, cP, Z) of the 3-DDH problem with probability $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ acts as $\mathcal{D}$'s simulator, it first provides $\mathcal{D}$ with the system public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, e(.,.), P, P_{Pub}, H_1, H_2, H_3, H_4)$, where $P_{Pub} = aP$ and a is unknown to $\mathcal{C}$.

$\mathcal{C}$ responds to $\mathcal{D}$'s queries by keeping lists κ_i for $i \in \{1,2,3,4\}$. Before $\mathcal{D}$ starts his queries, $\mathcal{C}$ chooses a random number $\pi \in \{1, 2, \ldots, q_{H_2}\}$ and treats all the queries identical to those in the proof of of the Theorem 3.1.

After the first round of quires, $\mathcal{D}$ requests a challenge signature on (ID^*, m^*). Upon $\mathcal{D}$'s request, $\mathcal{C}$ checks if $ID^* \neq ID_\pi$, $\mathcal{C}$ terminates and outputs $\perp$. Otherwise, it picks $\tau, \theta \in \mathbb{Z}_q$ and acomputes $\mathcal{S}_1^* = \tau cP$,$\mathcal{S}_2^* = \theta cP$, $\mu^* = H_2(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*)$, and $\mathcal{S}_3^* = \tau Z + \theta \mu^* Z$ and send the signature as $\sigma^* = (\mathcal{S}_1^*, \mathcal{S}_2^*, \mathcal{S}_3^*)$.

$\mathcal{D}$ starts the second round of queries with the restrictions defined in Definition 3.2. At the end of the game, $\mathcal{D}$ outputs its decision bit $\gamma \in \{0,1\}$. If $\gamma = 1$, it indicates that the signature σ^* is valid, and consequently, $\mathcal{C}$ outputs 1 to declare that (P, aP, bP, cP, Z) is a valid 3-Diffie-Hellman tuple. Otherwise, if $\gamma = 0$, indicates that the signature σ^* is invalid, and consequently, $\mathcal{C}$ outputs 1 to declare that (P, aP, bP, cP, Z) is an invalid 3-Diffie-Hellman tuple, where $Z \neq abcP$.

In order to assess the probability that $\mathcal{C}$ does not fail in the simulation process, we first consider the cases that $\mathcal{C}$ would fail. $\mathcal{C}$ may fail in the simulation process if $\mathcal{D}$ queries the Extract oracle on an identity ID_i where $i = \pi$. $\mathcal{C}$ may also fail if $\mathcal{D}$ queries for a signature associated with the identity ID_π. It is easy to see that the probability to avoid all the failure states is $(1 - \frac{1}{q_{H_1}})^{q_E}(1 - \frac{1}{q_{H_1}})^{q_S}$. Therefore the probability for $\mathcal{C}$ to solve the 3-DDH problem given $\mathcal{D}$ is able to win the invisibility game of our signature scheme is $\varepsilon_{\mathcal{C}} \geq (1 - \frac{1}{q_{H_1}})^{q_E}(1 - \frac{1}{q_{H_1}})^{q_S}\varepsilon_{\mathcal{D}}$. □

3.4.2 Efficiency and Extensions

Efficiency: Developing efficient identity-based digital signatures has sparked a significant amount of research and many efficient schemes have been introduced to the literature (Hess, 2003; Paterson & Schuldt, 2006). However, in the case of identity-based undeniable signature schemes, all the existing schemes (Libert & Quisquater, 2004; Wu et al., 2008) require at least one pairing evaluation to be carried out in their sign algorithm which leads to signatures that are members of the cyclic group $\mathbb{G}_2$. The cost of performing pairing evaluation lies very high above other common computations (e.g. multi-exponentiation, exponentiation, etc.), which limits the use of such schemes.

Comparing to the existing identity-based undeniable signature schemes, our scheme does not need any pairing evaluation in its Sign algorithm and its signature size is considerably smaller than the ones in the existing schemes. Table 3.2 below provides a quick efficiency and signature size comparison between our proposed scheme and the existing ones.

Table 3.2: Efficiency Comparison of Our Proposed Scheme and the Existing Identity-Based Undeniable Signature Schemes

Schemes	**Signature Generation**	**Signature Length**
Our Proposed Scheme	$2pm$	$3\lvert\mathbb{G}_1\rvert$
Libert and Quisquater (2004)	pe	$\lvert\mathbb{G}_2\rvert + \lvert r\rvert$
Wu et al. (2008)	$1pe+1pa+1pm$	$\lvert\mathbb{G}_2\rvert + 2\lvert\mathbb{G}_1\rvert$

In Table 3.2, *pe* denotes pairing evaluation and *pm* and *pa* denote point multiplication and point addition (in group $\mathbb{G}_1$), respectively. As aforementioned, the cost of point addition and multiplication is insignificant comparing to the cost of pairing evaluation. As depicted in Table 3.2, the signature size in our proposed scheme is cogently smaller than the schemes proposed by Libert and Quisquater (2004) and Wu et al. (2008). For example, for 128 bit security, Libert and Quisquater's and Wu et al.'s signature length are 1124 (for $|r| = 100$) and 1344 bits, respectively, while the signature length in our scheme is approximately 480 bits.

Convertibility: The notion of convertibility was first introduced to the index of undeniable signature schemes by Boyar et al. (1991). Convertibility provides the signer with the option to convert her undeniable signatures to ordinary digital signatures (i.e. publicly verifiable). The conversion takes place in two forms, namely, selective and universal. The former enables the signer to generate a token (selective token) which converts one of her signatures and the latter enables the signer to generate a token which makes all her undeniable signatures publicly verifiable.

By eliminating the trapdoor commitments from the proofs of the Confirmation/Disavowal protocols of our scheme, selective token can be generated to enable any participants in the system to verify the validity/invalidity of the signature. Here we only show the conversion for a valid message-signature pair, the same method can be applied to generate a token for an invalid message-signature pair. The signer with identity ID_S works as follows to generate a selective token on the validity of a valid message-signature pair $(m, \sigma = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3))$.

1. Choose a random $R \in \mathbb{G}_1$ to compute $Z_1 = e(P, R)$ and $Z_2 = e(\mu\mathcal{S}_1 + \mathcal{S}_2, R)$.
2. Compute $h_C = H_3(Z_1, Z_2, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, m)$ and $T = R - h_C D_S$ and form the proof as (h_C, T).

Upon receiving the selective token (h_C, T), any user in the system can verify the validity of the message-signature pair (m, σ) by computing $Z_1' = e(P, T)e(P_{Pub}, Q_S)^{h_C}$, $Z_2' = e(\mu\mathcal{S}_1 + \mathcal{S}_2, T)e(P, \mathcal{S}_3)^{h_C}$, and $h_C' = H_3(Z_1', Z_2', \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, m)$ and checking if $h_C = h_C'$ holds.

3.5 Summary

In this chapter, we analysed the security of Chow's (2005) scheme as the most efficient identity-based undeniable signature scheme and pointed out two weaknesses in the construction of Chow's scheme. We then showed that the revised scheme by Chow (2005) is not secure by mounting two attacks on the unforgeability and non-transferability of the scheme.

We then put forth a new efficient identity-based undeniable signature scheme. Our scheme is better than the existing identity-based undeniable signature scheme in terms of efficiency as it does not require any pairing evaluations in its Sign algorithm and has the shortest signature length among all the identity-based undeniable signature schemes in the literature. Our scheme incorporates the pairing-based version of the non-interactive designated verifier proofs in its Confirmation and Disavowal protocols to prevent blackmailing (Desmedt & Yung, 1991; Jakobsson, 1995) and man-in-the-middle (Desmedt, Goutier, & Bengio, 1987) attack and provides the signer with the option to selectively convert her undeniable signatures to publicly verifiable ones. Lastly, we proved the security of our scheme by relying its unforgeability and invisibility on the hardness of the CDH and the 3-DDH problems respectively.

CHAPTER 4

CRYPTANALYSIS AND DESIGN OF CERTIFICATELESS UNDENIABLE SIGNATURE SCHEMES

4.1 Introduction

In contemplation of bridging the gap between traditional public key cryptography and identity-based cryptography, Al-Riyami and Paterson (2003) introduced the concept of certificateless cryptography. In certificateless paradigms, the TTP (i.e. KGC) only supplies one half of the user's private key (i.e. partial private key) which is computed from her publicly available information. The other half of the user's private key (i.e. secret value) is computed and kept secure by the user herself. In a certificateless system, users are in charge of computing and publishing (e.g. on a public bulletin) their public keys. Due to the lack of the infrastructure to authenticate users' public keys in certificateless systems, it is vital to consider an adversary who is able to replace the user's public key with any public key of his choice (Al-Riyami & Paterson, 2003). Therefore, in the security models of certificateless systems, we always consider two types of adversaries as follows.

- **Type I Adversary $\mathcal{A}_I$:** This type of adversary simulates a third party adversary that has no possible knowledge on the master secret key. However, due to the aforementioned characteristic of certificateless systems, $\mathcal{A}_I$ is allowed to replace the public key of any user with a public key of his choice.
- **Type II Adversary $\mathcal{A}_{II}$:** This type of adversary simulates a malicious KGC. Therefore, $\mathcal{A}_{II}$ is assumed to have knowledge over the master secret key s which enables him to compute the partial private key of any user in the system. Nonetheless, $\mathcal{A}_{II}$ is not permitted to replace the user's public key.

As depicted in the security models of many certificateless schemes, a Type I adversary can gain knowledge on the secret value of users by either querying the secret value Extract oracle (Au et al., 2007; Choi, Park, & Lee, 2011; Duan, 2008; Hu, Wong, Zhang, & Deng, 2007; Huang, Mu, Susilo, Wong, & Wu, 2007) or by replacing the public key of the users with public keys of his choice (where he may know the corresponding secret value) (Al-Riyami & Paterson, 2003; Duan, 2008; Huang et al., 2007; Tso, Huang, & Susilo, 2012). Moreover, we know that a Type II adversary can easily compute the users' partial private keys since he has complete knowledge over the master secret key. Therefore, the security models of certificateless systems should be formulated in a way to prevent the adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}\}$ to use his knowledge on a portion of the user's private key in initiating cryptographic operations on behalf of the user. For instance, in the case of certificateless undeniable signature schemes, the security model should be formulated in such a way to prevent the adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}\}$ from violating any of the security notions of such schemes (e.g. unforgeability and invisibility).

Duan (2008) proffered the first provably secure certificateless undeniable signature scheme to the literature. The proposed scheme along with its security models could be considered as a certificateless version of the work proposed by Libert and Quisquater (2004). However, Duan's scheme requires two expensive pairing evaluations in its Sign algorithm which leads to a longer signature length. With the aim of proposing a more efficient scheme, Zhao and Ye (2012) proposed a new provable secure certificateless undeniable signature scheme which does not require any pairing evaluations in its Sign algorithm and has a considerably smaller signature size. However, the scheme provides less security assurance than the scheme proposed by Duan (2008) as it is secure in a weaker security model. Zhao and Ye's scheme employs Chaum's (1991) ZKIPs in its Confirmation and Disavowal protocols. The proposed scheme is shown to be unforgeable under the CDH assumption. The authors stated that their scheme is invisible under the DDH assumption. However, the assumption they used is actually called the 3-DDH assumption which is also called the DDH problem in $\mathbb{G}_1$ by Chabanne, Phan, and Pointcheval (2005).

Contributions

In this chapter, we first point out two weaknesses in the structure of the efficient scheme proposed by Zhao and Ye (2012). Moreover, we exploit these weaknesses in order to mount two attacks on the proposed scheme. In the first attack, we target the invisibility of the scheme and show how a Type I adversary can verify the validity/invalidity of a message-signature pair without the help of its signer. In the second attack, we show that the Confirmation and Disavowal protocols of the proposed scheme do not satisfy the property of non-impersonation and a Type I adversary is able to impersonate the signer by initiating these protocols with any third party on her behalf. Next, we put forth a revised scheme which overcomes both of the attacks, while enjoys from an equally efficient Sign algorithm. We employ the pairing-based version of non-interactive designated verifier proofs of Jakobsson et al. (1996) to provide more secure and efficient Confirmation and Disavowal protocols. Similar to the original scheme, the revised scheme is only secure in a weaker security model.

Next, we formalise a strong security model for certificateless undeniable signature schemes. We then propose an efficient certificateless undeniable signature scheme which is secure in the strong security model. Comparing to the only certificateless undeniable signature scheme which is secure in the strong security model (Duan, 2008), our scheme is much more efficient in signature generation, proof generation and proof verification steps. Lastly, we prove the unforgeability, invisibility and anonymity of our scheme in a strong security model based on the hardness of some well-known pairing-based problems in the random oracle model.

The rest of this chapter is organised as follows. In Section 4.2, we recall the structure of Zhao and Ye's scheme and mount our attacks on the invisibility and non-impersonation of their proposed scheme. In Section 4.3, we propose a revised scheme and discuss about its security and features. We provide the strong security models for certificateless undeniable signature schemes in Section 4.4. In Section 4.5, we proffer our efficient certificateless undeniable signature scheme, provide a formal security analysis and discuss about its efficiency and extensions. Finally, we conclude this chapter in Section 4.6.

4.2 Cryptanalysis on Zhao and Ye's Certificateless Undeniable Signature Scheme

In this section, we review the efficient certificateless undeniable signature scheme of Zhao and Ye (2012) and then show our attacks by exploiting the weaknesses in its structure.

4.2.1 The Zhao and Ye Scheme (Zhao & Ye, 2012)

Setup: By choosing $k \in \mathbb{Z}_q$ as the security parameter, the KGC runs this algorithm by generating an additive cyclic group $\mathbb{G}_1$ of prime order $q \geqslant 2^k$ and a multiplicative cyclic group $\mathbb{G}_2$ of the same order. The KGC continues by selecting an arbitrary generator $P \in \mathbb{G}_1$ and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Next, it chooses two cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{G}_1$, and generates its key pair by selecting $s \in \mathbb{Z}_q$ as the master secret key and computing $P_{Pub} = sP$ as the corresponding public key. Lastly, it publishes the system public parameters as $params = (\mathbb{G}_1, \mathbb{G}_2, q, P, e(.,.), H_1, H_2, P_{Pub})$.

Partial-Private-Key-Extract: Provided the user's identity ID and the system public parameters $params$, the KGC computes $Q_{ID} = H_1(ID)$ and sets the partial private key of the user as $D_{ID} = sQ_{ID}$ and transmits it to the user in a secure manner.

Set-Secret-Value: The user with identity ID chooses $s_{ID} \in \mathbb{Z}_q$ at random as her secret value.

Set-Private-Key: After the user received her partial private key D_{ID} and selected her secret value s_{ID}, she can form her private key as $SK_{ID} = (D_{ID}, s_{ID})$.

Set-Public-Key: The user with identity ID uses her secret value s_{ID} to compute her public key as $PK_{ID} = (PK_1, PK_2) = (s_{ID}P, s_{ID}Q_{ID})$.

Sign: Provided the system public key parameters $params$ and a message $m \in \{0,1\}^*$ to be signed, the user with identity ID chooses $r \in \mathbb{Z}_q$ at random, computes $U = rP$ and $V = rs_{ID}H_2(m, ID, PK_{ID}, U) + D_{ID}$ and forms the signature as $\sigma = (U, V)$.

Verify: Provided a message-signature pair $(m, \sigma = (U, V))$, the alleged signer (with identity ID) first verifies the validity of public keys by checking $e(PK_2, P) = e(PK_1, Q_{ID})$. If the equality holds, it continues to check $e(V,P) = e(H_2(m, ID, PK_{ID}, U), U)^{s_{ID}} e($ and outputs *valid* if it holds, and *invalid* otherwise.

Confirmation: Given a valid message-signature pair (m, σ), the alleged signer (with identity ID) uses Chaum's (1991) ZKIPs in order to prove that $(e(P,P), e(P, PK_1), e(H_2(m, ID, PK_{ID}, U), U), {}^{e(V,P)}/{}_{e(Q_{ID}, P_{Pub})})$ is a valid Diffie-Hellman (DH) tuple as follows:

1. The verifier chooses $a, b \in \mathbb{Z}_q$ at random, computes $c = e(P,P)^a e(H_2(m, ID, PK_{ID}, U), U)^b$, and sends c to the signer.
2. The signer chooses $r \in \mathbb{Z}_q$ at random, computes $z_1 = ce(P,P)^r$ and $z_2 = z_1^{s_{ID}}$, and sends (z_1, z_2) to the verifier.
3. The verifier sends (a, b) to the signer.
4. The signer checks if $c = e(P,P)^a e(H_2(m, ID, PK_{ID}, U), U)^b$ holds, then, she sends r to the verifier.
5. The verifier will only accept the proof if $z_1 = e(P,P)^{a+r} e(H_2(m, ID, PK_{ID}, U), U)^b$ and $z_2 = e(P, PK_1)^{a+r} ({}^{e(V,P)}/{}_{e(Q_{ID}, P_{Pub})})^b$ hold.

Disavowal: Given an invalid message-signature pair $(m, \sigma = (U, V))$, the alleged signer (with identity ID) uses Chaum's (1991) ZKIPs in order to prove that $(e(P,P), e(P, PK_1), e(H_2(m, ID, PK_{ID}, U), U), {}^{e(V,P)}/{}_{e(Q_{ID}, P_{Pub})})$ is a none DH-tuple. For the details of the protocol, we refer the reader to the papers by Chaum (1991) and Zhao and Ye (2012).

4.2.2 Attack on the Invisibility

As aforementioned, the security models of certificateless schemes have to be formulated in such a way to prevent both adversary types (i.e. Type I and Type II) to violate any of the security notions related to such schemes. In order to ensure security in such systems, the security models of certificateless schemes enable a Type I adversary $\mathcal{A}_I$ to get access to the users' secret values by either querying the secret value extract oracle (Au et al., 2007; Choi et al., 2011; Duan, 2008; Hu et al., 2007; Huang et al., 2007) or the public key replacement oracle (Al-Riyami & Paterson, 2003; Duan, 2008; Huang et al., 2007; Tso et al., 2012) which enables the adversary to replace the public key of the users with any public key of his choice (that he may know the corresponding secret value).

As clearly stated in the security models of Zhao and Ye's (2012) scheme, in addition to having access to the secret value extract oracle, $\mathcal{A}_I$ has access to a Sign oracle which is able to return valid signatures under the replaced public keys. In the following attack, we consider the latter approach where the adversary replaces the target signer's public key and requests for a valid signature (note that the same attack could be mounted if the secret value of the signer is queried from the secret value extract oracle defined in the security model of Zhao and Ye (2012)). The details of the attack are as follows.

1. The adversary $\mathcal{A}_I$ picks $s'_{ID} \in \mathbb{Z}_p$ and computes the corresponding public key $PK'_{ID} = (PK'_1, PK'_2) = (s'_{ID}P, s'_{ID}Q_{ID})$.
2. Then, it requests for a valid signature under the replaced public key PK'_{ID} (note that the replaced public key PK'_{ID} is valid since $e(PK'_2, P) = e(Q_{ID}, PK'_1)$).
3. Upon receiving such a request, the signer picks $r \in \mathbb{Z}_q$ at random and forms $U = rP$ and $V = rs'_{ID}H_2(m, ID, PK'_{ID}, U) + D_{ID}$ to output the signature as $\sigma = (U, V)$.

It can be easily observed that $\mathcal{A}_I$ can verify the validity or invalidity of the signature on its own by checking if $e(V, P) = e(H_2(m, ID, PK'_{ID}, U), U)^{s'_{ID}} e(Q_{ID}, P_{Pub})$ holds, and therefore, violating the invisibility property of the proposed scheme.

This attack is due to a flaw in the signature structure of the proposed scheme. As shown above, the adversary $\mathcal{A}_I$ with the knowledge of only the secret value of the signer is able to verify the validity or invalidity of any of the signer's signatures without her help. This results in violating one of the vital security notions of undeniable signature schemes (i.e. invisibility) which distinguishes such schemes from ordinary digital signatures.

4.2.3 Attack on the Non-Impersonation

Following the above attack, we show how the same adversary $\mathcal{A}_I$ can impersonate the signer by only having knowledge on the secret value of the target signer. More specifically, the attack enables the adversary to initiate the Confirmation or Disavowal protocol with any third party on behalf of the signer.

Here, we only demonstrate the attack on the Confirmation protocol (the same attack can also be mounted on the Disavowal protocol) and show that it is exactly identical with the original protocol initiated by the signer.

For a valid message-signature pair $(m, \sigma = (U,V))$, the adversary $\mathcal{A}_I$ has to prove to the third party that $(e(P,P), e(P,PK_1^{'}), e(H_2(m,ID,PK_{ID}^{'},U),U), {}^{e(V,P)}/_{e(Q_{ID},P_{Pub})})$ is a valid DH-tuple using the ZKIP as follows:

1. The verifier chooses $a,b \in \mathbb{Z}_q$ at random, computes $c = e(P,P)^a e(H_2(m,ID,PK_{ID}^{'},U),U)^b$ and sends c to $\mathcal{A}_I$.
2. $\mathcal{A}_I$ chooses $r \in \mathbb{Z}_q$ at random, computes $z_1 = ce(P,P)^r$ and $z_2 = z_1^{s_{ID}^{'}}$, and sends (z_1,z_2) to the verifier.
3. The verifier sends (a,b) to $\mathcal{A}_I$.
4. $\mathcal{A}_I$ checks if $c = e(P,P)^a e(H_2(m,ID,PK_{ID}^{'},U),U)^b$ holds, he sends r to the verifier.
5. The verifier will only accept the proof if $z_1 = e(P,P)^{a+r} e(H_2(m,ID,PK_{ID}^{'},U),U)^b$ and $z_2 = e(P,PK_1^{'})^{a+r}({}^{e(V,P)}/_{e(Q_{ID},P_{Pub})})^b$ hold.

The second attack is resulted from the poorly structured Confirmation and Disavowal protocols of the proposed scheme. The authors made use of the ZKIP in the Confirmation and Disavowal protocols. However, the only private input of the signer to these protocols is her secret value. This is against the fundamental security requirements of certificateless schemes where the users are required to provide their whole private key (consisting of the secret value and the partial private key) in order to prevent the adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}\}$ to initiate any cryptographic operations on their behalf.

4.3 The Revised Scheme

In this section, we propose a revised scheme that addresses both of the above weaknesses and prevents the attacks in the previous section. From the efficiency point of view, the Sign algorithm of the revised scheme is as efficient as the original scheme with the same signature length. In order to overcome the second attack and provide more efficiency and additional security, we employ the non-interactive designated verifier proofs of Jakobsson et al. (1996) in the Confirmation and Disavowal protocols of the revised scheme. The primary objective of introducing such proofs was to address the man-in-the-middle (Desmedt et al., 1987) and blackmailing attacks (Desmedt & Yung, 1991; Jakobsson, 1995) on undeniable signature schemes. The non-interactive designated verifier proofs are also more efficient since they reduce the number of interactions between the signer and the verifier to only one move. The Setup and Partial-private-key-extract algorithms of the revised scheme is identical to the one in the original scheme, except that in the Setup algorithm, we define two new hash functions $H_3, H_4 : \mathbb{G}_2 \times \ldots \times \mathbb{G}_1 \to \mathbb{Z}_q$ and included them in the public parameters.

Set-User-Key: The user with identity ID chooses $s_{ID} \in \mathbb{Z}_q$ at random as her secret value and computes her public key as $PK_{ID} = s_{ID}P$.

Set-Private-Key: After the user received her partial private key D_{ID} and selected her secret value s_{ID}, she can form her private key as $SK_{ID} = (D_{ID}, s_{ID})$.

Sign: Provided the system public key parameters *params* and a message $m \in \{0,1\}^*$ to be signed, the signer with identity ID_S chooses $r \in \mathbb{Z}_q$ at random, computes $U = rP$ and $V = r(s_S H_2(m, ID_S, PK_S, U) + D_S)$ and forms the signature as $\sigma = (U,V)$.

Verify: Provided a message-signature pair $(m, \sigma = (U,V))$, the alleged signer (with identity ID_S) uses her private key SK_S and checks if $e(V,P) = e(H_2(m, ID_S, PK_S, U), U)^{s_S} e(D_S, U)$ holds, it outputs *valid*. Otherwise, she outputs *invalid*.

Confirmation: Given a valid message-signature (m, σ) pair to be confirmed, the signer (with identity ID_S and public key PK_S) works as follows in order to generate a non-interactive Confirmation proof transcript for the designated verifier (with identity ID_V and public key PK_V).

1. Compute $Q_V = H_1(ID_V)$ and pick at random $J, W \in \mathbb{G}_1$ and $\beta, \tau, \nu \in \mathbb{Z}_q$ to compute the following:

$$n_1 = e(P,J)e(P_{Pub}, Q_V)^{\nu} \tag{4.1}$$

$$n_2 = \nu PK_V + \tau P \tag{4.2}$$

$$p_1 = e(P,W) \tag{4.3}$$

$$p_2 = e(P,P)^{\beta} \tag{4.4}$$

$$p_3 = e(H_2(m, ID_S, PK_S, U), U)^{\beta} e(W,U) \tag{4.5}$$

2. Set the values of $h_C = H_3(n_1, n_2, p_1, p_2, p_3, \sigma)$, $I = W - (h_C + \nu)D_S$ and $u = \beta - (h_C + \nu)s_S$ to form the Confirmation proof transcript as $(n_1, n_2, J, \nu, \tau, I, u, h_C)$.

In order to verify the veracity of the Confirmation proof transcript $(n_1, n_2, J, \nu, \tau, I, u, h_C)$, the designated verifier computes the following:

$$n_1^{'} = e(P,J)e(P_{Pub}, Q_V)^{\nu} \tag{4.6}$$

$$n_2^{'} = \nu PK_V + \tau P \tag{4.7}$$

$$p_1^{'} = e(P,I)e(P_{Pub}, Q_S)^{(h_C+\nu)} \tag{4.8}$$

$$p_2^{'} = e(P,P)^{u} e(P, PK_S)^{(h_C+\nu)} \tag{4.9}$$

$$p_3^{'} = e(H_2(m, ID_S, PK_S, U), U)^{u} e(U,I) e(P,V)^{(h_C+\nu)} \tag{4.10}$$

and will only accept the proof if $h_C = H_3(n_1^{'}, n_2^{'}, p_1^{'}, p_2^{'}, p_3^{'}, \sigma)$.

Disavowal: Given an invalid message-signature pair (m, σ), the signer (with identity ID_S and public key PK_S) works as follows in order to generate a non-interactive Confirmation proof transcript for the designated verifier (with identity ID_V and public key PK_V).

1. Parse σ into (U,V), compute $Q_V = H_1(ID_V)$ and pick $J \in \mathbb{G}_1$ and $\tau, v, \gamma \in \mathbb{Z}_q$ at random in order to compute the values of $n_1 = e(P,J)e(P_{Pub}, Q_V)^{\nu}$, $n_2 = vPK_V + \tau P$ and $C = (\frac{e(H_2(m,ID_S,PK_S,U),U)^{s_S} e(D_S,U)}{e(P,V)})^{\gamma}$.
2. The signer has to prove her knowledge of a tuple $(T, \gamma, \omega) \in \mathbb{G}_1 \times \mathbb{Z}_q \times \mathbb{Z}_q$ where $C = \frac{e(H_2(m,ID_S,PK_S,U),U)^{\omega} e(T,U)}{e(P,V)^{\gamma}}$, $\frac{e(T,P)}{e(Q_S,P_{Pub})^{\gamma}} = 1$ and $\frac{\omega P}{\gamma PK_S} = 1$. In order to do so, the signer works as follows.
   a) Pick $X \in \mathbb{G}_1$ and $a, i \in \mathbb{Z}_q$ at random to compute $j_1 = \frac{e(P,X)}{e(Q_S,P_{Pub})^a}$, $j_2 = \frac{e(P,P)^i}{e(P,PK_S)^a}$, and $j_3 = \frac{e(H_2(m,ID_S,PK_S,U),U)^i e(U,X)}{e(P,V)^a}$.
   b) Set the values of $h_D = H_4(C, n_1, n_2, j_1, j_2, j_3, \sigma)$, $w_1 = i - (h_D + \nu)\omega$, $w_2 = a - (h_D + \nu)\gamma$, and $Y = X - (h_D + v)T$ to form the proof as $(C, J, \tau, v, h_D, Y, w_1, w_2)$.

Upon receiving the Disavowal proof transcript $(C, J, \tau, v, h_D, Y, w_1, w_2)$, the designated verifier checks if $C \neq 1$, he verifies the proof by computing the following:

$$n_1^{'} = e(P,J)e(P_{Pub}, Q_V)^{\nu} \tag{4.11}$$

$$n_2^{'} = vPK_V + \tau P \tag{4.12}$$

$$j_1^{'} = \frac{e(P,Y)}{e(Q_S, P_{Pub})^{w_2}} \tag{4.13}$$

$$j_2^{'} = \frac{e(P,P)^{w_1}}{e(P, PK_S)^{w_2}} \tag{4.14}$$

$$j_3^{'} = \frac{e(H_2(m, ID_S, PK_S, U), U)^{w_1} e(U,Y)}{e(V,P)^{w_2}} C^{(h_D+\nu)} \tag{4.15}$$

and will only accept the proof if $h_D = H_4(C, n_1', n_2', j_1', j_2', j_3', \sigma)$.

4.3.1 Security Analysis

We ensure the security of the new scheme against the aforementioned attacks by enforcing the signer to use her whole private key to verify signatures in the Verify algorithm and also when generating proofs in the Confirmation and Disavowal protocols. Using the same reduction technique as Zhao and Ye's (2012) scheme, we can also relate the unforgeability and invisibility of our scheme to the hardness of the CDH problem and the 3-DDH problem respectively.

It is trivial to show the completeness of both the Confirmation and Disavowal protocols of the new scheme. We can use the same technique as in the previous chapter (see Section 3.4.2) in order to prove the soundness, non-transferability and non-impersonation of the new scheme.

4.3.2 Efficiency and Extensions

Efficiency: As mentioned above, the Sign algorithm of our proposed scheme is as efficient as the one in Zhao and Ye's scheme. While the Confirmation and Disavowal protocols of our scheme are more efficient in communication (since they are non-interactive) and provide more flexibility for the signer, they require more pairing evaluations. The structure of our scheme is more compact and less complex as we combined the Set-secret-value and Set-public-key algorithms into a single algorithm (i.e. Set-user-key) and the public key of users in our scheme is consisted of only a single point in $\mathbb{G}_1$. Therefore, the Verify algorithm of our scheme is more efficient (saves two pairing evaluations) as the signer is not required to run the validity check on the public key.

Convertibility: Our scheme provides the signer with the option to selectively convert her undeniable signatures to ordinary digital signatures by omitting the trapdoor commitments from the the proof of the Confirmation and Disavowal protocols. To generate a selective token on a valid message-signature pair $(m, \sigma = (U, V))$, the signer with identity ID_S (and public key PK_S) chooses $T \in \mathbb{G}_1$ and $y \in \mathbb{Z}_q$ at random to form $c_1 = e(P,T) \in \mathbb{G}_1, \quad c_2 = e(P,P)^y \in \mathbb{G}_1$, and $c_3 = e(H_2(m,ID_S,PK_S,U),U)^y e(T,U) \in \mathbb{G}_1$ and sets $h_{SC} = H_2(c_1,c_2,c_3,\sigma)$, $I = T - h_{SC}(D_A)$ and $i = y - s_S^{h_{SC}}$ and outputs the proof as (I,i,h_{SC}). Upon receiving the selective token (I,i,h_{SC}), any user in the system can check the validity of the message-signature pair $(m,\sigma = (U,V))$ by computing $c_1^{'} = e(P,I)e(P,P_{Pub})$, $c_2^{'} = e(P,P)^i e(P,PK_S)$, and $c_3^{'} = e(H_2(m,ID_S,PK_S,U),U)^i e(I,U)e(P,V)$ and checking if $h_{SC} = H_4(c_1^{'},c_2^{'},c_3^{'},\sigma)$ holds. Note that the same method can be applied in the Disavowal protocol of the revised scheme.

4.4 Security Models of Certificateless Undeniable Signature Schemes

In some of the proposed security models (Al-Riyami & Paterson, 2003; Huang et al., 2007), a Type II adversary (malicious KGC) is assumed to generate its key pair honestly and initiate attacks only after the Setup step (i.e. assume that the malicious KGC is benign at the beginning). Au et al. (2007) defined a security model against a malicious-but-passive KGC, where a malicious KGC is assumed to generate its key pair dishonestly (i.e. KGC is malicious from the beginning) and compute the private key of the target user from her public key. Although this type of adversary was never captured in the security models of Al-Riyami and Paterson (2003), Li, Chen, and Sun (2005) or Huang et al. (2007), the authors showed that these schemes and any other scheme which has the same key generation as Al-Riyami and Paterson's (2003) scheme, is vulnerable against this type of attack.

We make use of the binding method (Al-Riyami & Paterson, 2003) in our scheme in order to ensure the security against malicious-but-passive KGC attacks (we discuss on how we address the attack in Section 4.5.1). Moreover, using the binding technique, we can lift the trust level of KGC in our scheme to level 3 of Girault's trust level hierarchy (1991) (this was the main incentive in (Al-Riyami & Paterson, 2003)).

There are seven different oracles which can be queried by an adversary $\mathcal{A} \in \{\mathcal{A}_I, \mathcal{A}_{II}\}$ based on the games' specifications which are to be discussed a bit later.

Hash oracle $(\mathcal{O}^{hash})$**:** $\mathcal{A}$ can query different hash functions H in the system with any inputs of his choice.

Public key request $(\mathcal{O}^{pub\_key\_req})$**:** $\mathcal{A}$ can query for the public keys of any user in the system.

Partial private key extract $(\mathcal{O}^{part\_key\_extract})$**:** By providing a user's identity ID and the corresponding public keys (TV_{ID}, TS_{ID}), $\mathcal{A}$ can query for the partial private key d_{ID} of the user.

Secret value extract $(\mathcal{O}^{sec\_val\_extract})$**:** By providing a user's identity ID and the corresponding public keys (TV_{ID}, TS_{ID}), $\mathcal{A}$ can query for the user's secret value x_{ID}.

Private key extract $(\mathcal{O}^{priv\_key\_extract})$**:** By providing a user's identity ID and the corresponding public keys (TV_{ID}, TS_{ID}), $\mathcal{A}$ can query for the private key S_{ID} of the user.

Public key replacement $(\mathcal{O}^{pub\_key\_replace})$**:** $\mathcal{A}$ is allowed to replace public keys $(TV_{ID}$ $, TS_{ID})$ of any user ID with public keys of its choice $(TV^{'}_{ID}, TS^{'}_{ID})$.

Sign oracle $(\mathcal{O}^{sign})$**:** By providing a message m, and the identity ID of the alleged signer (with public keys (TV_{ID}, TS_{ID})), the oracle returns a valid undeniable signature σ. Note that the public keys could have been replaced prior to this query.

Confirmation/Disavowal oracle $(\mathcal{O}^{conf/disav})$**:** By providing a valid (invalid) message-signature pair (m, σ), the identity ID of the claimed signer (with public keys (TV_{ID}, TS_{ID})), and possibly the identity and public keys of the designated verifier, this oracle returns the Confirmation (Disavowal) proof transcript in order to prove the validity (invalidity) of the signature σ.

In the following, we propose our security models for certificateless undeniable signature schemes. Our securiy models are inspired by the works on certificateless signatures (Al-Riyami & Paterson, 2003; Huang et al., 2007; Li et al., 2005) and the only certificateless undeniable siganture scheme that is secure in the strong security model (Duan, 2008). We first define our security models against a Type I adversary (through Definition 4.1 to 4.3) and then against a Type II adversary (through Definition 4.4 to

4.6).

Definition 4.1. *We consider a certificateless undeniable signature scheme to be existentially unforgeable under adaptive chosen message, identity, and public key attacks if no PPT Type I adversary $\mathcal{F}_I$ has a non-negligible advantage in the following game:*

Setup phase. The challenger $\mathcal{C}$ runs the Setup algorithm and provides $\mathcal{F}_I$ with the system public parameters *params*.

Query phase. The adversary $\mathcal{F}_I$ can adaptively query $\mathcal{O}^{hash}$, $\mathcal{O}^{pub\_key\_req}$, $\mathcal{O}^{part\_key\_extract}$, $\mathcal{O}^{priv\_key\_extract}$, $\mathcal{O}^{pub\_key\_replace}$, $\mathcal{O}^{sign}$ and $\mathcal{O}^{conf/disav}$. $\mathcal{C}$ will respond to all the queries accordingly (as stated in the above definition).

At the end of the game, $\mathcal{F}_I$ will output a valid message-signature pair (m^*, σ^*) for a signer with identity ID^* and public keys (TV_{ID^*}, TS_{ID^*}). $\mathcal{F}_I$ wins the above game if $(ID^*, TV_{ID^*}, TS_{ID^*})$ was never queried to $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{priv\_key\_extract}$ and σ^* was never outputted by $\mathcal{O}^{sign}$ on the input of m^* and $(ID^*, TV_{ID^*}, TS_{ID^*})$.

Definition 4.2. *A certificateless undeniable signature scheme is considered to fulfil the notion of invisibility under adaptive chosen message, identity, and public key attacks if no PPT Type I adversary $\mathcal{D}_I$ has a non-negligible advantage in the following game:*

Setup phase. This phase takes place identical to the game of Definition 4.1.

Query phase (before challenge). The adversary $\mathcal{D}_I$ can initiate polynomially bounded number of queries as defined in the game of Definition 4.1.

Challenge phase. After the first round of queries, $\mathcal{D}_I$ requests a challenge signature on a message m^* for a signer with identity ID^* and public keys (TV_{ID^*}, TS_{ID^*}). Where the tuple $(ID^*, TV_{ID^*}, TS_{ID^*})$ was never queried to $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{priv\_key\_extract}$. Then, the challenger $\mathcal{C}$ generates the challenge signature σ^* based on the outcome of a random coin toss $b \in \{0, 1\}$. If $b = 0$, $\mathcal{C}$ will select a random $\sigma^* \in \mathcal{S}$, where $\mathcal{S}$ is the signature space and sends σ^* to $\mathcal{D}_I$. Otherwise, if $b = 1$, the challenger will generate a valid signature σ^* and sends it back to $\mathcal{D}_I$.

Query phase (after challenge). $\mathcal{D}_I$ initiates the second round of queries, this time, $\mathcal{D}_I$ is not allowed to query $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{priv\_key\_extract}$ on the identity ID^* with public keys (TV_{ID^*}, TS_{ID^*}), nor the Confirmation/Disavowal oracle on $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$.

At the end of the game, $\mathcal{D}_I$ will output its decision bit $b^{'} \in \{0,1\}$ and wins the game if $b^{'} = b$.

Definition 4.3. *A certificateless undeniable signature scheme is considered to fulfil the notion of anonymity under adaptive chosen message, identity, and public key attacks if no PPT Type I adversary $\mathcal{D}_I$ has a non-negligible advantage in the following game:*

Setup phase. This phase takes place identical to the game of Definition 4.1.

Query phase (before challenge). The adversary $\mathcal{D}_I$ can initiate polynomially bounded number of queries as defined in the game of Definition 4.1.

Challenge phase. After the first round of queries, $\mathcal{D}_I$ produces a message m^*, and two tuples (ID_0, TV_0, TS_0) and (ID_1, TV_1, TS_1) containing the identities and the public keys of two possible signers with the limitation that they were never queried to $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{priv\_key\_extract}$. The challenger $\mathcal{C}$ responds based on the outcome of a random coin toss $b \in \{0,1\}$ and generates the challenge signature σ^* on the message m^* for a signer with identity ID_b and public keys (TV_b, TS_b).

Query phase (after challenge). $\mathcal{D}_I$ initiates the second round of queries, this time, $\mathcal{D}_I$ is not allowed to query $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{priv\_key\_extract}$ on (ID_0, TV_0, TS_0) or (ID_1, TV_1, TS_1), nor the Confirmation/Disavowal oracle on tuples $(m^*, \sigma^*, ID_0, TV_0, TS_0)$ or $(m^*, \sigma^*, ID_1, TV_1, TS_1)$.

At the end of the game, $\mathcal{D}_I$ will output its decision bit $b^{'} \in \{0,1\}$ and wins the game if $b^{'} = b$.

Definition 4.4. *We consider a certificateless undeniable signature scheme to be existentially unforgeable under adaptive chosen message, identity, and public key attacks if no PPT Type II adversary $\mathcal{F}_{II}$ has a non-negligible advantage in the following game:*

Setup phase. The challenger $\mathcal{C}$ runs the Setup algorithm and provides $\mathcal{F}_{II}$ with the master secret key s and the system public parameters $params$.

Query phase. The adversary $\mathcal{F}_{II}$ can adaptively query $\mathcal{O}^{hash}$, $\mathcal{O}^{pub\_key\_req}$, $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{priv\_key\_extract}$, $\mathcal{O}^{pub\_key\_replace}$, $\mathcal{O}^{sign}$ and $\mathcal{O}^{conf/disav}$. $\mathcal{C}$ will respond to all the queries accordingly (as stated in the above definition).

At the end of the game, $\mathcal{F}_{II}$ will output a valid message-signature pair (m^*, σ^*) for a signer with identity ID^* and public keys (TV_{ID^*}, TS_{ID^*}). $\mathcal{F}_{II}$ wins the above game if $(ID^*, TV_{ID^*}, TS_{ID^*})$ was never queried to $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{priv\_key\_extract}$ or $\mathcal{O}^{pub\_key\_replace}$, and σ^* was never outputted by $\mathcal{O}^{sign}$ on the input of m^* and $(ID^*, TV_{ID^*}, TS_{ID^*})$.

Definition 4.5. *A certificateless undeniable signature scheme is considered to fulfil the notion of invisibility under adaptive chosen message, identity, and public key attacks if no PPT Type II adversary $\mathcal{D}_{II}$ has a non-negligible advantage in the following game:*

Setup phase. This phase takes place identical to the game of Definition 4.4.

Query phase (before challenge). The adversary $\mathcal{D}_{II}$ can initiate polynomially bounded number of queries as defined in the game of Definition 4.4.

Challenge phase. After the first round of queries, $\mathcal{D}_{II}$ requests a challenge signature on a message m^* for a signer with identity ID^* and public keys (TV_{ID^*}, TS_{ID^*}). Where the tuple $(ID^*, TV_{ID^*}, TS_{ID^*})$ was never queried to $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{priv\_key\_extract}$ or $\mathcal{O}^{pub\_key\_replace}$. The challenger $\mathcal{C}$ then generates the challenge signature σ^* based on the outcome of a random coin toss $b \in \{0, 1\}$. If $b = 0$, $\mathcal{C}$ will select a random $\sigma^* \in \mathcal{S}$, where $\mathcal{S}$ is the signature space and sends σ^* to $\mathcal{D}_{II}$. Otherwise, if $b = 1$, the challenger will generate a valid signature σ^* and sends it back to $\mathcal{D}_{II}$.

Query phase (after challenge). $\mathcal{D}_{II}$ initiates the second round of queries, this time, $\mathcal{D}_{II}$ is not allowed to query $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{priv\_key\_extract}$ or $\mathcal{O}^{pub\_key\_replace}$ on the identity ID^* with public keys (TV_{ID^*}, TS_{ID^*}), nor the Confirmation/Disavowal oracle on $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$.

At the end of the game, $\mathcal{D}_{II}$ will output its decision bit $b \in \{0,1\}$ and wins the game if $b^{'} = b$.

Definition 4.6. *A certificateless undeniable signature scheme is considered to fulfil the notion of anonymity under adaptive chosen message, identity, and public key attacks if no PPT Type II adversary $\mathcal{D}_{II}$ has a non-negligible advantage in the following game:*

Setup phase. This phase takes place identical to the game of Definition 4.4.

Query phase (before challenge). The adversary $\mathcal{D}_{II}$ can initiate polynomially bounded number of queries as defined in the game of Definition 4.4.

Challenge phase. After the first round of queries, $\mathcal{D}_{II}$ produces a message m^*, and two tuples (ID_0, TV_0, TS_0) and (ID_1, TV_1, TS_1) containing the identities and public keys of two possible signers with the limitation that they were never queried to $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{priv\_key\_extract}$ or $\mathcal{O}^{pub\_key\_replace}$. The challenger $\mathcal{C}$ responds based on the outcome of a random coin toss $b \in \{0,1\}$ and generates the challenge signature σ^* on the message m^* for a signer with identity ID_b and public keys (TV_b, TS_b).

Query phase (after challenge). $\mathcal{D}_{II}$ initiates the second round of queries, this time, $\mathcal{D}_{II}$ is not allowed to query $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{priv\_key\_extract}$ or $\mathcal{O}^{pub\_key\_replace}$ on (ID_0, TV_0, TS_0) or (ID_1, TV_1, TS_1), nor the Confirmation/Disavowal oracle on tuples $(m^*, \sigma^*, ID_0, TV_0, TS_0)$ or $(m^*, \sigma^*, ID_1, TV_1, TS_1)$.

At the end of the game, $\mathcal{D}_{II}$ will output its decision bit $b \in \{0,1\}$ and wins the game if $b^{'} = b$.

Based on the work of Galbraith and Mao (2003), the notion of anonymity is equivalent to the notion of invisibility in the sense we stated in our security models. Consequently, we can use the same technique as proposed by Galbraith and Mao (2003) to prove the anonymity of our scheme against Type I and Type II adversaries under the hardness of the DBDH problem.

4.5 The Proposed Scheme

In this section, we propose our efficient certificateless undeniable signature scheme, provide a formal security proof to rely its security to the hardness of some well-known mathematical problems and discuss about its efficiency and extensions.

Setup: The Setup algorithm is initiated by the KGC. It takes two security parameters k and l, and generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q \geq 2^k$, a generator P of $\mathbb{G}_1$, and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. It also chooses 4 cryptographic hash functions: $H_1 : \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \times \{0,1\}^l \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_1$, and $H_3, H_4 : \mathbb{G}_2 \times \ldots \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q$. Next, it picks $s \in \mathbb{Z}_q$ randomly as the KGC's secret key and calculates $P_{Pub} = sP$ as the corresponding public key. The KGC's public key P_{Pub} and system's public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, P_{Pub}, H_1, H_2, H_3, H_4)$ will be made available to all system users.

Set-user-keys: The user with identity ID chooses $x_{ID} \in \mathbb{Z}_q$ randomly as her secret value and computes $TV_{ID} = x_{ID}P$ and $TS_{ID} = x_{ID}P_{Pub}$ as the corresponding public keys.

Partial-private-key-extraction: Provided the user's identity ID and public keys (TV_{ID}, TS_{ID}), the KGC computes her partial private key as $d_{ID} = sQ_{ID} = sH_1(ID, TV_{ID}, TS_{ID})$, and delivers it to the user in a secure manner.

Set-private-key: After the user with identity ID and public keys (TV_{ID}, TS_{ID}) received her partial private key d_{ID}, she will form her private key as $S_{ID} = x_{ID}d_{ID}$.

Sign: To issue a signature on message $m \in \{0,1\}^*$, the signer with identity ID_S and public keys (TV_S, TS_S) chooses a random string $r \in \{0,1\}^l$ and computes $h_S = H_2(m, r, ID_S, TV_S, TS_S)$. She then uses her private key S_S to calculate $\lambda = e(h_S, S_S)$, and forms the signature $\sigma = (r, \lambda)$.

Confirmation: Given a valid message-signature pair $(m, \sigma = (r, \lambda))$, the alleged signer with identity ID_S and public keys (TV_S, TS_S) generates a non-interactive Confirmation proof for the designated verifier (with identity ID_V and public keys (TV_V, TS_V)) as follows.

1. Choose $\nu \in \mathbb{Z}_q$ and $U, Y \in \mathbb{G}_1$ at random to compute the following:

$$W = e(P,U)e(TS_V, Q_V)^{\nu} \tag{4.16}$$

$$N = e(P,Y) \tag{4.17}$$

$$O = e(H_2(m,r,ID_S,TV_S,TS_S),Y) \tag{4.18}$$

2. Set the values of $h_C = H_3(W,N,O,m,\sigma)$ and $B = Y - (h_C + \nu)S_S$ to form the Confirmation proof transcript as (U, ν, h_C, B).

Upon receiving the Confirmation proof transcript (U, ν, h_C, B), the designated verifier checks if $e(TV_S, P_{Pub}) = e(TS_S, P)$ holds, he computes the following in order to confirm the validity of the message-signature pair $(m, \sigma = (r, \lambda))$ for the alleged signer.

$$W^{'} = e(P,U)e(TS_V, Q_V)^{\nu} \tag{4.19}$$

$$N^{'} = e(P,B)e(TS_S, Q_S)^{(h_C+\nu)} \tag{4.20}$$

$$O^{'} = e(H_2(m,r,ID_S,TV_S,TS_S),B)\lambda^{(h_C+\nu)} \tag{4.21}$$

At the end, the designated verifier will only accept the proof if $h_C = H_3(W^{'}, N^{'}, O^{'}, m, \sigma)$ holds.

Disavowal: Given an invalid message-signature pair $(m, \sigma = (r, \lambda))$, the claimed signer with identity ID_S and public keys (TV_S, TS_S) generates a non-interactive Disavowal proof for the designated verifier (with identity ID_V and public keys (TV_V, TS_V)) as follows.

1. Parse σ into (r, λ) and choose $\nu, \alpha \in \mathbb{Z}_q$ and $U \in \mathbb{G}_1$ at random in order to compute the following:

$$W = e(P,U)e(TS_V, Q_V)^{\nu} \tag{4.22}$$

$$C = \left(e(H_2(m,r,ID_S,TV_S,TS_S),S_S)/\lambda\right)^{\alpha} \tag{4.23}$$

2. The signer has to prove her knowledge of a pair $(J,\alpha) \in \mathbb{G}_1 \times \mathbb{Z}_q$ where $C = \left(e(H_2(m,r,ID_S,TV_S,TS_S),J)/\lambda^{\alpha}\right)$ and $e(P,J) = e(TS_S,Q_S)^{\alpha}$ hold. In order to do so, she works as follows.
   a) Choose $y \in \mathbb{Z}_q$ and $I \in \mathbb{G}_1$ randomly to compute $K = e(P,I)e(TS_S,Q_S)^{-y}$, and $L = e(H_2(m,r,ID_S,TV_S,TS_S),I)\lambda^{-y}$
   b) Set the values of $h_D = H_4(C,W,K,L,m,\sigma)$, $R = I - (h_D + \nu)J$ and $\mu = y - (h_D + \nu)\alpha$ to form the Disavowal proof transcript as (C,U,ν,h_D,R,μ).

Given the Disavowal proof transcript (C,U,ν,h_D,R,μ), the designated verifier first checks if $e(TV_S,P_{Pub}) = e(TS_S,P)$ and $C \neq 1$ hold, he computes the following in order to verify the validity of the Disavowal proof transcript.

$$W^{'} = e(P,U)e(TS_V,Q_V)^{\nu} \tag{4.24}$$

$$K^{'} = e(P,R)e(TS_S,Q_S)^{-\mu} \tag{4.25}$$

$$L^{'} = e(H_2(m,r,ID_S,TV_S,TS_S),R)\lambda^{-\mu}C^{(h_D+\nu)} \tag{4.26}$$

At the end, the designated verifier will only accept the proof if $h_D = H_4(C,W^{'},K^{'},L^{'},m,\sigma)$ holds.

4.5.1 Security Analysis

As mentioned previously, we employed the binding method as was introduced by Al-Riyami and Paterson (2003). The binding method helps to elevate the trust level on the KGC to trust level 3 in Girault's (1991) hierarchy and more importantly, it addresses the attack against a malicious-but-passive KGC (Au et al., 2007). This is due to the fact that when using the binding method, the KGC would need the knowledge of the user's (with identity ID) public keys (TV_{ID},TS_{ID}) when forming $Q_{ID} = H_1(ID,TV_{ID},TS_{ID})$. Evidently, in the Setup stage, the malicious-but-passive KGC does not have any possible information on the target user's public keys, and hence, it would not be able to compute Q_{ID} and set its key pair maliciously in order to mount the malicious-but-passive KGC attack to extract the private key of the target user from her public key.

We use the pairing-based version of the Jakobsson et al.'s (1996) method in the body of our Confirmation protocol and the method of Camenisch and Shoup (2003) in the Disavowal protocol of our scheme to prove the inequality of two discrete logarithms. As shown in Chapter 3 (see Section 3.4.1), we can show that both the Confirmation and Disavowal protocols of our scheme are sound and complete while enjoying from the properties of non-transferability and non-impersonation.

We prove that our scheme is existentially unforgeable and has the property of invisibility against both Type I and Type II adversaries in the random oracle model. We prove the security of our scheme against a Type I adversary in Theorems 4.1 and 4.2, and against a Type II adversary in Theorems 4.3 and 4.4.

We use the same approach as proposed by Goh and Jarecki (2003) to avoid using the forking lemma (Pointcheval & Stern, 2000) and obtain a tighter security reduction in our security proof.

Theorem 4.1. *If there exists a Type I adversary $\mathcal{F}_I$ that can submit q_E private key and partial private key extract queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, and q_{H_i} queries to random oracle H_i for $i \in \{1,2,3,4\}$ and be able to succeed in an existential forgery (win the game defined in Definition 4.1) against our proposed scheme with a non-negligible success probability $\varepsilon_{\mathcal{F}_\mathcal{I}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{F}_I$ to solve a random instance (P, aP, bP, cP) of the BDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{F}_I} - (2q_{H_3} + q_{CD} + 1)2^{-k}}{\mathbf{e}(q_E + 1)(q_{CD} + 1)}$$

Proof. We prove that if there exists a Type I adversary $\mathcal{F}_I$ which can win the game defined in Definition 4.1, then one can construct a PPT algorithm $\mathcal{C}$ that can run $\mathcal{F}_I$ as its subroutine to solve a random instance (P, aP, bP, cP) of the BDH problem with probability at least $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ works as $\mathcal{F}_I$'s challenger. It starts by initiating the Setup algorithm, and provides $\mathcal{F}_I$ with system public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, P_{Pub}, H_1, H_2, H_3, H_4)$, where $P_{Pub} = aP$ and a is unknown to $\mathcal{C}$.

$\mathcal{F}_I$ can make queries to random oracles H_i for $i = \{1,2,3,4\}$ and other oracles as defined in the game of Definition 4.1. $\mathcal{C}$ responds to these queries by keeping lists κ_i for $i = \{1,2,3,4\}$ and a list κ_0 in order to keep track of the values of identity, public keys and the corresponding secret value. We assume $\mathcal{F}_I$ always makes a public key request before a H_1 query, and a H_1 query before it requests for the partial private key of the user.

Query on $H_1(ID, TV_{ID}, TS_{ID})$: In order to handle queries on H_1 for an identity ID with public keys (TV_{ID}, TS_{ID}), $\mathcal{C}$ first chooses a random $\alpha \in \mathbb{Z}_q$ and flips a random coin X that is taking the value of 0 with probability φ_1 and the value of 1 with probability $1 - \varphi_1$ (the value of φ_1 will be calculated in our proof later). Lastly, $\mathcal{C}$ inserts $(ID, TV_{ID}, TS_{ID}, \alpha, X)$ into κ_1 and returns $H_1(ID, TV_{ID}, TS_{ID}) = \alpha(bP)$ if $X = 1$, and $H_1(ID, TV_{ID}, TS_{ID}) = \alpha P$ if $X = 0$.

Query on $H_2(m, r, ID, TV_{ID}, TS_{ID})$: To answer queries on H_2, $\mathcal{C}$ first chooses $\beta \in \mathbb{Z}_q$ randomly and flips a random coin Y that takes the value of 0 with probability φ_2 and the value of 1 with probability $1 - \varphi_2$ (the value of φ_2 will be calculated in our proof later). Lastly, $\mathcal{C}$ records $(m, r, ID, TV_{ID}, TS_{ID}, \beta, Y)$ in κ_2 and returns $H_2(m, r, ID, TV_{ID}, TS_{ID}) = \beta(cP)$ if $Y = 1$, and $H_2(m, r, ID, TV_{ID}, TS_{ID}) = \beta P$ if $Y = 0$.

Query on H_3 and H_4: Queries on H_3 and H_4 will be handled by $\mathcal{C}$ in a random manner, and the outputs will be stored in κ_3 and κ_4 respectively.

Public key request: To handle a public key request on an identity ID, $\mathcal{C}$ checks if $(ID, x_{ID}, TV_{ID}, TS_{ID})$ already exists in κ_0, then $\mathcal{C}$ returns (TV_{ID}, TS_{ID}). Otherwise, it picks a random $x_{ID} \in \mathbb{Z}_q$, computes $TV_{ID} = x_{ID}P$ and $TS_{ID} = x_{ID}P_{Pub}$, returns (TV_{ID}, TS_{ID}) to $\mathcal{F}_I$, and lastly, records $(ID, x_{ID}, TV_{ID}, TS_{ID})$ in κ_0.

Partial private key extract: Upon receiving an identity ID with public keys (TV_{ID}, TS_{ID}), $\mathcal{C}$ scans κ_1 for a tuple $(ID, TV_{ID}, TS_{ID}, \alpha, X)$, if $X = 1$, $\mathcal{C}$ reports *failure* and aborts the simulation. Otherwise, it outputs the partial private key as $d_{ID} = \alpha P_{Pub}$.

Private key extract: To handle a private key extraction query on an identity ID with public keys (TV_{ID}, TS_{ID}), $\mathcal{C}$ scans κ_1 for $(ID, TV_{ID}, TS_{ID}, \alpha, X)$. If $X = 1$, $\mathcal{B}$ reports *failure* and aborts the simulation. Otherwise, it searches κ_0 to find $(ID, x_{ID}, TV_{ID}, TS_{ID})$ and returns the private key of the user ID as $S_{ID} = \alpha TS_{ID}$.

Public key replacement: If $\mathcal{F}_I$ wishes to replace the public keys (TV_{ID}, TS_{ID}) for identity ID with public keys of its choice $(TV_{ID}^{'}, TS_{ID}^{'})$, $\mathcal{C}$ checks κ_0 to find $(ID, x_{ID}, TV_{ID}, TS_{ID})$, if such tuple exists, it will replace it with $(ID, -1, TV_{ID}^{'}, TS_{ID}^{'})$, where -1 means that the public keys have been replaced. Otherwise, $\mathcal{C}$ simply adds a tuple $(ID, -1, TV_{ID}^{'}, TS_{ID}^{'})$ to κ_0.

Sign query: $\mathcal{F}_I$ is allowed to query the Sign oracle in order to receive valid signatures on any tuple $(m, ID, TV_{ID}, TS_{ID})$, where m is a message to be signed by a signer with identity ID and public keys (TV_{ID}, TS_{ID}). This oracle is able to produce valid signatures even for identities where the public keys of the user have been replaced. $\mathcal{C}$ starts by picking a random $r \in \{0,1\}^l$, and scans κ_2 for a tuple $(m, r, ID, TV_{ID}, TS_{ID}, \ldots)$. If such tuple already exists in κ_2, $\mathcal{C}$ picks another r until no tuple $(m, r, ID, TV_{ID}, TS_{ID}, \ldots)$ is found in κ_2. When a proper r is found, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$ and inserts $(m, r, ID, TV_{ID}, TS_{ID}, \beta, 0)$ (this implies that a H_2 query on $(m, r, ID, TV_{ID}, TS_{ID})$ will be replied by βP). Lastly, $\mathcal{C}$ computes $\lambda = e(\beta TS_{ID}, Q_{ID})$ and forms the signature as $\sigma = (\lambda, r)$.

Confirmation/Disavowal query: Upon $\mathcal{F}_I^{'}$s request for a confirmation/disavowal proof transcript on any tuple $(m, \sigma^{'} = (\lambda^{'}, r^{'}), TV_S, TS_S, ID_S, ID_V)$, where ID_S is the identity of a signer with public keys (TV_S, TS_S) and ID_V is the identity of a designated verifier. $\mathcal{C}$ responds in one of the following ways:

If the tuple $(m,r^{'},ID_S,TV_S,TS_S,\ldots)$ was never queried to H_2 oracle, $\mathcal{C}$ proceeds as in Sign oracle to generate a valid sub-signature λ. $\mathcal{C}$ then checks if $\lambda = \lambda^{'}$, it will return the Confirmation protocol transcript. On the other hand, if $\lambda \neq \lambda^{'}$, $\mathcal{C}$ will return the Disavowal protocol transcript.

If $(m,r^{'},ID_S,TV_S,TS_S,\beta,Y)$ exists in κ_2, and $Y = 0$, $\mathcal{C}$ will compute the valid sub-signature as $\lambda = e(\beta TS_S,Q_S)$. Similar to above, it will output the Confirmation protocol transcript if $\lambda = \lambda^{'}$, and the Disavowal protocol transcript otherwise.

If $(m,r^{'},ID_S,TV_S,TS_S,\beta,Y)$ exists in κ_2 and $Y = 1$, $\mathcal{C}$ scans κ_1 in order to find a tuple $(ID_S,TV_S,TS_S,\alpha,X)$. If $X = 1$, $\mathcal{C}$ outputs *failure* and aborts. On the other hand, if $X = 0$, it will form the valid signature as $\lambda = e(\beta(cP),TS_S)^{\alpha}$ and output the Confirmation protocol transcript if $\lambda = \lambda^{'}$. Otherwise, if $\lambda \neq \lambda^{'}$, $\mathcal{C}$ will return the Disavowal protocol transcript.

$\mathcal{C}$ may fail in the simulation of the non-interactive designated verifier proofs of Confirmation/Disavowal protocol if a collision occurs in simulating H_3 or H_4 oracle. The probability for the occurrence of such collision is at most $(q_{H_3}+q_{CD})2^{-k}$, considering $q_{H_3} \approx q_{H_4}$.

At the end of the game, $\mathcal{F}_I$ outputs a tuple $(m^*,\sigma^*,ID^*,TV_{ID^*},TS_{ID^*})$ where $\sigma^* = (r^*,\lambda^*)$ is a valid signature on message m^* for identity ID^* with public keys TV_{ID^*} and TS_{ID^*}. In order for $\mathcal{F}_I$ to win, $(ID^*,TV_{ID^*},TS_{ID^*})$ should have never been queried to the partial private key or the private key extraction oracles. Upon $\mathcal{F}_I$'s success, $\mathcal{C}$ searches κ_1 and κ_2 to find $(ID^*,\alpha^*,TV_{ID^*},TS_{ID^*},X)$ (due to the assumption made above, existence of such tuple is certain in κ_1) and $(m^*,r^*,ID^*,TV_{ID^*},TS_{ID^*},\beta^*,Y)$; if $X = 0$ or $Y = 0$, $\mathcal{C}$ reports *failure* and aborts. Again, if $(m^*,r^*,ID^*,TV_{ID^*},TS_{ID^*},\beta^*,Y)$ does not exist in κ_2, $\mathcal{C}$ reports *failure* and aborts. In the case that the public keys (TV_{ID^*},TS_{ID^*}) were never replaced before, $\mathcal{C}$ can simply extract the user's secret value from κ_0 and compute $(\lambda^*)^{\frac{1}{x_{ID^*}\alpha^*\beta^*}}$ as the solution of the random instance (P,aP,bP,cP) of the BDH assumption. On the other hand, if the public keys (TV_{ID^*},TS_{ID^*}) have been replaced before, then for σ^* to be valid we know that $e(TV_{ID^*}, P_{Pub}) = e(TS_{ID^*}, P)$. Therefore, based on the knowledge of exponent assumption as introduced by Damgård (1992) and Hada and Tanaka (1998), $\mathcal{F}_I$ can extract x since $e(xP, aP) = e(x(aP), P)$, and consequently $\mathcal{C}$ outputs $(\lambda^*)^{\frac{1}{x\alpha^*\beta^*}}$ as the solution of the random instance (P, aP, bP, cP) of the BDH problem.

In order to compute the success probability of $\mathcal{C}$, we have to consider the cases that $\mathcal{C}$ may fail. $\mathcal{C}$ can fail in either the simulation process or in solving the BDH problem after $\mathcal{F}_I$ outputted the forgery signature. $\mathcal{C}$ will fail in the simulation process if $\mathcal{F}_I$ queries a partial private key extraction on an identity ID and public keys (TV_{ID}, TS_{ID}) where $H_1(ID, TV_{ID}, TS_{ID}) = \alpha(bP)$. $\mathcal{C}$ will also fail in simulating the Confirmation/Disavowal protocol when $\mathcal{F}_I$ queries a tuple $(m, \sigma, TV_S, TS_S, ID_S, ID_V)$, where $H_2(m, r, ID_S, TV_S, TS_S) = \beta(cP)$ and $H_1(ID_S, TS_S, TV_S) = \alpha(bP)$. Moreover, $\mathcal{C}$ can fail in solving the BDH problem if the forgery tuple $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$ is such that $H_1(ID^*, TV_{ID^*}, TS_{ID^*})$ was defined to be αP or $H_2(m^*, r^*, ID^*, TV_{ID^*}, TS_{ID^*})$ was defined to be βP. Therefore, following Coron's (2000) method, the probability for $\mathcal{C}$ to avoid all the failure states is $\varphi_1^{q_E}(1-\varphi_1)\varphi_2^{q_{CD}}(1-\varphi_2)$ where q_E is the number of partial private key and private key extract queries and q_{CD} is the number of confirmation/disavowal queries. By optimising the probabilities φ_1 and φ_2, the probability for $\mathcal{C}$ to avoid all the failure states is equal to $1/\mathbf{e}(q_E+1)(q_{CD}+1)$ (where $\mathbf{e}$ is the base of natural logarithm). There is also the probability that $\mathcal{F}_I$ never queried $H_2(m^*, r^*, ID^*, TV_{ID^*}, TS_{ID^*})$, this may occur with probability 2^{-k}. It is possible that $\mathcal{F}_I$ produced a forgery signature σ^* and proved its validity where it did not use the valid private key S_{ID^*}, this case may only happen if $H_3(W, N, O, m^*, \sigma^*)$ is set as a particular value with probability $q_{H_3}2^{-k}$. As mentioned above, $\mathcal{C}$ may also fail in simulating the Confirmation and Disavowal protocol if a collision occurs in the domain of q_{H_3}. This incident may happen with probability $(q_{H_3} + q_{CD})2^{-k}$. Following the proof, the success probability of $\mathcal{C}$ is at least $\frac{\varepsilon_{\mathcal{F}_I} - (2q_{H_3}+q_{CD}+1)2^{-k}}{\mathbf{e}(q_E+1)(q_{CD}+1)}$. □

Theorem 4.2. *If there exists a Type I adversary $\mathcal{D}_I$ that can submit q_E private key and partial private key extract queries, q_{US} sign queries, q_{CD} confirmation and dis-*

avowal queries, and q_{H_i} queries to random oracle H_i for $i \in \{1,2,3,4\}$ and be able to breach the invisibility property (win the game defined in Definition 4.2) of our proposed scheme with non-negligible success probability $\varepsilon_{\mathcal{D}_I}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{D}_I$ to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability:

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{D}_I} - (q_{H_3} + q_{CD})2^{-k}}{\mathbf{e}(q_E + 1)}$$

Proof. We prove that if there exists a Type I adversary $\mathcal{D}_I$ which is able to win the game defined in Definition 4.2 with probability $\varepsilon_{\mathcal{D}_I}$, then one can build another algorithm $\mathcal{C}$ which is able to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ acts as $\mathcal{D}_I$'s challenger, it starts by initiating the Setup algorithm as in the proof of Theorem 4.1 wherein, $P_{Pub} = aP$ and a is unknown to $\mathcal{C}$. $\mathcal{D}_I$ starts by querying different oracles as explained in Definition 4.2. We assume $\mathcal{D}_I$ always makes a public key request before a H_1 query, and a H_1 query before it requests for the partial private key of the user. Similar to the proof of Theorem 4.1, $\mathcal{C}$ answers to $\mathcal{D}_I$ queries by using lists κ_i for $i = \{1,2,3,4\}$ and a list κ_0 in order to keep track of the values of identity, public keys and the corresponding secret value.

Query on $H_1(ID, TV_{ID}, TS_{ID})$: Queries to H_1 are handled identical to the proof of Theorem 4.1.

Query on $H_2(m, r, ID, TV_{ID}, TS_{ID})$: To answer queries on H_2, $\mathcal{C}$ scans κ_2 to find $(m, r, ID, TV_{ID}, TS_{ID}, \beta, Y)$. If such tuple exists, $\mathcal{C}$ outputs βP when $Y = 0$ and $\beta(cP)$ when $Y = 1$. Otherwise, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$, returns βP to $\mathcal{D}_I$, and inserts $(m, r, ID, TV_{ID}, TS_{ID}, \beta, 0)$ into κ_2.

Query on H_3 and H_4: Queries to H_3 and H_4 are handled identical to the proof of Theorem 4.1.

Queries on public key, partial private key extract, private key, public key replacement, and sign oracles are handled identical to the proof of Theorem 4.1.

Confirmation/Disavowal query: Due to the behaviour of H_2, $\mathcal{C}$ is able to calculate a valid signature σ in order to compare it with any signature $\sigma^{'}$ queried to the Confirmation/Disavowal oracle and generate confirmation/disavowal proofs consistent with validity/invalidity of $\sigma^{'}$.

Similar to the proof of Theorem 4.1, a collision may occur in the domain of H_3 or H_4 oracles when simulating Confirmation/Disavowal protocol.

After the first round of queries, $\mathcal{D}_I$ outputs a challenge tuple $(m^*, ID^*, TV_{ID^*}, TS_{ID^*})$, where m^* is a message to be signed, ID^* is the identity of a signer, and TV_{ID^*} and TS_{ID^*} are the original public keys. Note that $(ID^*, TV_{ID^*}, TS_{ID^*})$ was never queried to the partial private key or the private key extraction oracles. $\mathcal{C}$ scans κ_1 to find $(ID^*, TV_{ID^*}, TS_{ID^*}, \alpha, X)$ (due to the assumption made above, we know that such tuple exists in κ_1). If $X = 0$, $\mathcal{C}$ aborts and outputs failure. Otherwise, if $X = 1$, $\mathcal{C}$ proceeds by picking a random $r \in \{0,1\}^l$ and checking if $(m^*, r, ID^*, TV_{ID^*}, TS_{ID^*}, \ldots)$ exists in κ_2. If it does, $\mathcal{C}$ picks another r until it finds an appropriate r whereby no such tuple $(m^*, r, ID^*, TV_{ID^*}, TS_{ID^*}, \ldots)$ exists in κ_2. Thereupon, $\mathcal{C}$ defines $H_2(m^*, r, ID^*, TV_{ID^*}, TS_{ID^*})$ as $\beta(cP)$ and records $(m^*, r, ID^*, TV_{ID^*}, TS_{ID^*}, \beta, 1)$ in κ_2. Lastly, $\mathcal{C}$ computes $\lambda^* = h^{x_{ID^*}\alpha\beta}$ and sets the challenge signature as $\sigma^* = (r, \lambda^*)$.

$\mathcal{D}_I$ starts the second round of queries, however, this time $\mathcal{D}_I$ is withheld from a partial private key or private key extraction query on $(ID^*, TV_{ID^*}, TS_{ID^*})$, sign query on $(m^*, r, ID^*, TV_{ID^*}, TS_{ID^*})$, and confirmation/disavowal query on $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$.

After the second round of queries, $\mathcal{D}_I$ outputs its decision bit $b \in \{0,1\}$. If $b = 0$, it indicates that σ^* is an invalid signature, consequently, $\mathcal{C}$ outputs 0 to declare that (P, aP, bP, cP, h) is an invalid DBDH tuple. If $b = 1$, it indicates that σ^* is a valid signature and consequently, $\mathcal{C}$ outputs 1 to declare that (P, aP, bP, cP, h) is a valid DBDH tuple.

In order to compute the success probability of $\mathcal{C}$, we first consider the situations that $\mathcal{C}$ might fail. $\mathcal{C}$ may fail in partial private key or private key extraction queries where $H_1(ID, TV_{ID}, TS_{ID})$ was defined to be $\alpha(bP)$. $\mathcal{C}$ may also fail in the challenge phase where the challenge identity ID^* is such that $H_1(ID^*, TV_{ID^*}, TS_{ID^*})$ was defined as αP. Consequently, the probability for $\mathcal{C}$ not to fail is $\varphi_1^{q_E}(1-\varphi_1)$ which is maximised at ${}^1/_{\mathbf{e}(q_E+1)}$, when the optimal value of φ_1 is used. Similar to the proof of Theorem 4.1, $\mathcal{C}$ may also fail in simulation of the Confirmation and Disavowal protocol (in a case of collision) with probability $(q_{H_3}+q_{CD})2^{-k}$. Following the proof, given $\varepsilon_{\mathcal{D}_I}$ as the success probability of $\mathcal{D}_I$, the success probability of $\mathcal{C}$ is at least $\frac{\varepsilon_{\mathcal{D}_I}-(q_{H_3}+q_{CD})2^{-k}}{\mathbf{e}(q_E+1)}$. □

Theorem 4.3. *If there exists a Type II adversary $\mathcal{F}_{II}$ that can submit q_E secret value extract, private key extract and public key replacement queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, and q_{H_i} queries to random oracle H_i for $i \in \{1,2,3,4\}$ and be able to succeed in an existential forgery (win the game defined in Definition 4.4) against our proposed scheme with non-negligible success probability $\varepsilon_{\mathcal{F}_{II}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{F}_{II}$ to solve a random instance (P, aP, bP, cP) of the BDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{F}_{II}}-(2q_{H_3}+q_{CD}+1)2^{-k}}{\mathbf{e}(q_E+1)(q_{CD}+1)}$$

Proof. We prove that if there exists a Type II adversary $\mathcal{F}_{II}$ that is able to win the game defined in Definition 4.4 with probability $\varepsilon_{\mathcal{F}_{II}}$, then a PPT algorithm $\mathcal{C}$ can be built that runs $\mathcal{F}_{II}$ as its subroutine and is able to solve a random instance (P, aP, bP, cP) of the BDH problem with probability $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ plays as $\mathcal{F}_{II}$'s challenger, and starts by initiating the Setup algorithm and providing $\mathcal{F}_{II}$ with the master secret key s and the system's public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, H_1, H_2, H_3, H_4)$. Evidently, P_{Pub} is not included in the public parameters as it can be easily computed by $\mathcal{F}_{II}$.

$\mathcal{F}_{II}$ performs polynomially bounded number of queries as defined in Definition 4.4. As in Theorem 4.1, $\mathcal{C}$ handles these queries by keeping lists κ_i for $i \in \{1,2,3,4\}$, and a list κ_0 in order to keep track of the values of identity, secret value, and the corresponding public keys of users in the system. We assume $\mathcal{F}_{II}$ makes a public key request before a H_1 query.

Query on $H_1(ID, TV_{ID}, TS_{ID})$: To answer such queries on H_1 (for an identity ID with public keys (TV_{ID}, TS_{ID})), $\mathcal{C}$ picks a random $\alpha \in \mathbb{Z}_q$, inserts $(ID, TV_{ID}, TS_{ID}, \alpha)$ in κ_1, and returns $Q_{ID} = \alpha(bP)$ to $\mathcal{F}_{II}$.

Query on $H_2(m, r, ID, TV_{ID}, TS_{ID})$: In order to answer queries on H_2, $\mathcal{C}$ first picks a random $\beta \in \mathbb{Z}_q$ and flips a coin Y that is truly random taking the value of 0 with probability φ_2 and the value of 1 with probability $1 - \varphi_2$ (the value of φ_2 will be computed later in our proof). Next, $\mathcal{C}$ inserts $(m, r, ID, TV_{ID}, TS_{ID}, \beta, Y)$ into κ_2 and returns $H_2(m, r, ID, TV_{ID}, TS_{ID}) = \beta(cP)$ if $Y = 1$ and $H_2(m, r, ID, TV_{ID}, TS_{ID}) = \beta P$ if $Y = 0$.

Query on H_3 and H_4: Queries on H_3 and H_4 will be handled by $\mathcal{C}$ in a random manner and the outputs will be stored in κ_3 and κ_4 respectively.

Public key request: Upon submitting an identity ID, $\mathcal{C}$ picks a random $\delta \in \mathbb{Z}_q$ and flips a coin X that is truly random taking the value of 0 with probability φ_1 and the value of 1 with probability $1 - \varphi_1$ (the value of φ_1 will be computed later in our proof). If $X = 0$, $\mathcal{C}$ sets the public keys as $TV_{ID} = \delta P$ and $TS_{ID} = \delta P_{Pub}$. Otherwise, if $X = 1$, $\mathcal{C}$ sets the public keys as $TV_{ID} = \delta(aP)$ and $TS_{ID} = s\delta(aP)$. In both cases, $\mathcal{C}$ inserts the tuple $(ID, \delta, TV_{ID}, TS_{ID}, X)$ in κ_0.

Secret value extract: In order to respond to a secret key extraction query on an identity ID with public keys (TV_{ID}, TS_{ID}), $\mathcal{C}$ scans κ_0 for $(ID, \delta, TV_{ID}, TS_{ID}, X)$. If $X = 1$, $\mathcal{C}$ reports *failure* and aborts the simulation. Otherwise, if $X = 0$, it returns δ as the secret value of the user.

Private key extract: In order to respond to a private key extraction query on identity ID with public keys (TV_{ID}, TS_{ID}), $\mathcal{C}$ scans κ_0 for $(ID, \delta, TV_{ID}, TS_{ID}, X)$. If $X = 1$, $\mathcal{C}$ reports *failure* and aborts the simulation. Otherwise, it searches κ_1 to find $(ID, TV_{ID}, TS_{ID}, \alpha)$ and returns the private key of the user as $S_{ID} = s\delta Q_{ID}$.

Public key replacement: If $\mathcal{F}_{II}$ wishes to replace the public keys (TV_{ID}, TS_{ID}) for identity ID with public keys of its choice (TV'_{ID}, TS'_{ID}), $\mathcal{C}$ checks κ_0 to find $(ID, x_{ID}, TV_{ID}, TS_{ID}, \ldots)$, if such tuple exists, it will replace it with $(ID, -1, TV'_{ID}, TS'_{ID}, \ldots)$, where -1 means that the public keys have been replaced. Otherwise, $\mathcal{C}$ adds a tuple $(ID, -1, TV'_{ID}, TS'_{ID}, \ldots)$ to κ_0.

Sign query: In order to respond to a sign query on any tuple $(m, ID, TV_{ID}, TS_{ID})$, $\mathcal{C}$ picks a random $r \in \{0,1\}^l$ and checks if κ_2 already contains $(m, r, ID, TV_{ID}, TS_{ID}, \ldots)$, if yes, it proceeds until it finds an appropriate r where no such tuple $(m, r, ID, TV_{ID}, TS_{ID}, \ldots)$ exists in κ_2. When such an acceptable r is found, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$ and inserts $(m, r, ID, TV_{ID}, TS_{ID}, \beta, 0)$ in κ_2, implying that the value of $H_2(m, r, ID, TV_{ID}, TS_{ID})$ is set as βP. Lastly, $\mathcal{C}$ computes $\lambda = e(\beta TS_{ID}, Q_{ID})$ and forms the signature $\sigma = (r, \lambda)$.

Confirmation/Disavowal query: Upon $\mathcal{F}_{II}$'s request for a Confirmation/Disavowal proof transcript on any tuple $(m, \sigma' = (r, \lambda'), TV_S, TS_S, ID_S, ID_V)$, where ID_S is the identity of a signer with public keys TV_S and TS_S and ID_V is the identity of a designated verifier. $\mathcal{C}$ responds in one of the following ways:

If the tuple $(m, r, ID_S, TV_S, TS_S, \ldots)$ was never queried to H_2 oracle, $\mathcal{C}$ proceeds as in Sign oracle to generate a valid sub-signature λ, and checks if $\lambda = \lambda'$, it will return the Confirmation protocol transcript, and the Disavowal protocol transcript otherwise.

If $(m, r, ID_S, TV_S, TS_S, \beta, Y)$ exists in κ_2 and $Y = 0$, $\mathcal{C}$ will compute the valid sub-signature as $\lambda = e(\beta TS_S, Q_S)$. Similar to above, it will output the Confirmation protocol transcript if $\lambda = \lambda'$, and the Disavowal protocol transcript otherwise.

If $(m, r, ID_S, TV_S, TS_S, \beta, Y)$ exists in κ_2 and $Y = 1$, $\mathcal{C}$ scans κ_0 in order to find a tuple $(ID_S, \delta, TV_S, TS_S, X)$. If $X = 1$, $\mathcal{C}$ outputs failure and aborts. On the other hand, if $X = 0$, $\mathcal{C}$ will form the valid sub-signature as $\lambda = e(\beta(cP), sQ_S)^{\delta}$ and output the Confirmation protocol transcript if $\lambda = \lambda'$. Otherwise, if $\lambda \neq \lambda'$, $\mathcal{C}$ will return the Disavowal protocol transcript.

$\mathcal{C}$ may fail in the simulation of the non-interactive designated verifier proofs of Confirmation/Disavowal protocols if a collision occurs in simulating H_3 or H_4 oracle. The probability for the occurrence of such collision is at most $(q_{H_3} + q_{CD})2^{-k}$, considering $q_{H_3} \approx q_{H_4}$.

At the end of the game, $\mathcal{F}_{II}$ outputs a tuple $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$, where $\sigma^* = (r^*, \lambda^*)$ is a valid signature on message m^* for a signer with identity ID^* and public keys TV_{ID^*} and TS_{ID^*}. In order for $\mathcal{F}_{II}$ to win, $(ID^*, TV_{ID^*}, TS_{ID^*})$ should have not been queried to secret value extract, public key replacement or private key extraction oracles. Upon $\mathcal{F}_{II}$'s success, $\mathcal{C}$ scans κ_0 and κ_2 in order to find tuples $(ID^*, \delta^*, TV_{ID^*}, TS_{ID^*}, X)$ and $(m^*, ID^*, TV_{ID^*}, TS_{ID^*}, \beta^*, Y)$, respectively. Then if $X = 0$ or $Y = 0$, $\mathcal{C}$ outputs failure and aborts. Also if no such tuple $(m^*, ID^*, TV_{ID^*}, TS_{ID^*}, \beta^*, Y)$ exists in κ_2, $\mathcal{C}$ will output failure and aborts the simulation. Otherwise, if σ^* is a valid signature, $\mathcal{C}$ outputs $(\lambda^*)^{\frac{1}{s\alpha\beta\delta}}$ as the solution of the random instance (P, aP, bP, cP) of the BDH problem.

In order to compute the success probability of $\mathcal{C}$, we have to consider the cases that $\mathcal{C}$ may fail. $\mathcal{C}$ can fail in either the simulation process or in solving the BDH problem after $\mathcal{F}_{II}$ outputted the forgery signature. $\mathcal{B}$ will fail in the simulation process if $\mathcal{F}_{II}$ initiates a secret value or private key extract query on an identity ID where $TV_{ID} = \delta(aP)$ and $TS_{ID} = s\delta(aP)$. $\mathcal{C}$ will also fail in simulating the Confirmation/Disavowal protocol when $\mathcal{F}_{II}$ queries a tuple $(m, \sigma, TV_S, TS_S, ID_S, ID_V)$, where $H_2(m, r, ID, TV_S, TS_S) = \beta(cP)$ and $TV_S = \delta(aP)$ and $TS_S = s\delta(aP)$. $\mathcal{C}$ will also fail if the forgery tuple $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$ is such that TV_{ID^*} and TS_{ID^*} were defined to be δP and δP_{Pub} respectively. Besides, $\mathcal{C}$ will again fail, if the forgery tuple is such that $H_2(m^*, r^*, ID^*, TV_{ID^*}, TS_{ID^*})$ was defined as βP. The probability for $\mathcal{C}$ to avoid all the failure states is $\varphi_1^{q_E}(1-\varphi_1)\varphi_2^{q_{CD}}(1-\varphi_2)$, where q_E is the number of secret value and private key extract queries and q_{CD} is the number of confirmation/disavowal queries. By maximising the probabilities φ_1 and φ_2, the success probability of $\mathcal{C}$ would be $^1/_{\mathbf{e}(q_E+1)(q_{CD}+1)}$. Similar to the proof of Theorem 4.1, considering the probability that $(m^*, r^*, ID^*, TV_{ID^*}, TS_{ID^*})$ was never queried to H_2 oracle, the probability that $\mathcal{F}_{II}$ did not use the valid private key S_{ID^*} when generating the forgery signature σ^*, and the probability of failure in simulating the Confirmation/Disavowal protocol; the success probability of $\mathcal{C}$ to a solve random instance (P, aP, bP, cP) of the BDH problem is at least $\frac{\varepsilon_{\mathcal{F}_{II}}-(2q_{H_3}+q_{CD}+1)2^{-k}}{\mathbf{e}(q_E+1)(q_{CD}+1)}$. □

Theorem 4.4. *If there exists a Type II adversary $\mathcal{D}_{II}$ that can submit q_E secret value extract, private key extract and public key replacement queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, and q_{H_i} queries to random oracle H_i for $i \in \{1,2,3,4\}$ and be able to breach the invisibility property (win the game defined in Definition 4.5) of our proposed scheme non-negligible success probability $\varepsilon_{\mathcal{D}_{II}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{D}_{II}$ to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{D}_{II}} - (q_{H_3} + q_{CD})2^{-k}}{\mathbf{e}(q_E+1)}$$

Proof. We prove that if there exists a Type II adversary $\mathcal{D}_{II}$ which is able to win the game defined in Definition 4.5 with probability $\varepsilon_{\mathcal{D}_{II}}$, then one can build another PPT algorithm $\mathcal{C}$ which is able to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ acts as $\mathcal{D}_{II}$'s challenger, it starts by initiating the Setup algorithm as in the proof of Theorem 4.3. $\mathcal{D}_{II}$ starts by querying different oracles as explained in the Definition 4.5. We assume $\mathcal{D}_{II}$ makes a public key request before a H_1 query. Similar to the proof of Theorem 4.3, $\mathcal{C}$ answers to $\mathcal{D}_{II}$ queries by using lists κ_i for $i = \{1,2,3,4\}$ and a list κ_0 in order to keep track of the values of identity, secret value, and the corresponding public keys of users in the system.

Query on $H_1(ID, TV_{ID}, TS_{ID})$: Queries to H_1 are handled identical to the proof of Theorem 4.3.

Query on $H_2(m,r,ID,TV_{ID},TS_{ID})$: In order to answer queries on H_2, $\mathcal{C}$ scans κ_2 to find $(m,r,ID,TV_{ID},TS_{ID},\beta,Y)$. If such tuples exists, $\mathcal{C}$ outputs βP when $Y=0$, and $\beta(cP)$ when $Y=1$. Otherwise, if no such tuple exists in κ_2, $\mathcal{C}$ first picks a random $\beta \in \mathbb{Z}_q$, returns βP to $\mathcal{D}_{II}$, and inserts $(m,r,ID,TV_{ID},TS_{ID},\beta,0)$ into κ_2.

Query on H_3 and H_4: Queries to H_3 and H_4 are handled identical to the proof of Theorem 4.3.

Queries on public key, secret value extract, private key extract, public key replacement, and sign oracles are handled identical to the proof of Theorem 4.3.

Confirmation/Disavowal query: Due to the behaviour of H_2 oracle, $\mathcal{C}$ is able to calculate valid signature σ in order to compare with any signature $\sigma^{'}$ queried to the Confirmation/Disavowal oracle and generate Confirmation (Disavowal) proofs consistent with validity (invalidity) of $\sigma^{'}$.

Similar to the proof of Theorem 4.3, a collision may occur in the domain of H_3 or H_4 oracle when simulating Confirmation/Disavowal protocol.

After the first round of queries, $\mathcal{D}_{II}$ outputs a challenge tuple $(m^*,ID^*,TV_{ID^*},TS_{ID^*})$, where m^* is a message to be signed, ID^* is the identity of a signer, and TV_{ID^*} and TS_{ID^*} are the corresponding public keys. Note that $(ID^*,TV_{ID^*},TS_{ID^*})$ should have never been queried to the secret value extract, public key replacement or the private key extract oracles. $\mathcal{C}$ scans κ_0 to find $(ID^*,\delta,TV_{ID^*},TS_{ID^*},X)$. If $X=0$, $\mathcal{C}$ aborts and outputs failure. Otherwise, if $X=1$, $\mathcal{C}$ proceeds by picking a random $r \in \{0,1\}^l$ and checking if $(m^*,r,ID^*,TV_{ID^*},TS_{ID^*},\ldots)$ exists κ_2. If it does, $\mathcal{C}$ picks another r until it finds an appropriate r where no such tuple $(m^*,r,ID^*,TV_{ID^*},TS_{ID^*},\ldots)$ exists in κ_2. Thereupon, $\mathcal{C}$ defines $H_2(m^*,r,ID^*,TV_{ID^*},TS_{ID^*})$ as $\beta(cP)$ and records $(m^*,r,ID^*,TV_{ID^*},TS_{ID^*},\beta,1)$ in κ_2. Lastly, $\mathcal{C}$ computes $\lambda^*=h^{s\delta\alpha\beta}$ and sets the challenge signature as $\sigma^*=(r,\lambda^*)$.

$\mathcal{D}_{II}$ starts the second round of queries, however, this time $\mathcal{D}_{II}$ is withheld from a secret value extract, public key replacement or private key extract query on $(ID^*, TV_{ID^*}, TS_{ID^*})$, sign query on $(m^*, r, ID^*, TV_{ID^*}, TS_{ID^*})$, and confirmation/disavowal query on $(m^*, \sigma^*, ID^*, TV_{ID^*}, TS_{ID^*})$.

After the second round of queries, $\mathcal{D}_{II}$ outputs its decision bit $b \in \{0,1\}$. If $b = 0$, it indicates that σ^* is an invalid signature, consequently, $\mathcal{C}$ outputs 0 to declare that (P, aP, bP, cP, h) is an invalid DBDH tuple. If $b = 1$, it indicates that σ^* is a valid signature and consequently, $\mathcal{C}$ outputs 1 to declare that (P, aP, bP, cP, h) is a valid DBDH tuple.

In order to assess the success probability of $\mathcal{C}$, we first consider the situations that $\mathcal{C}$ might fail. $\mathcal{C}$ may fail if $\mathcal{D}_{II}$ initiate a secret value or private key extract query on an identity ID where $TV_{ID} = \delta(aP)$ and $TS_{ID} = s\delta(aP)$. $\mathcal{C}$ might also fail if the challenge identity ID^* is such that the public keys TV_{ID^*} and TS_{ID^*} are defined as βP and βP_{Pub} respectively. Consequently, the probability for $\mathcal{C}$ not to fail is $\varphi_1^{q_E}(1-\varphi_1)$ which is maximised at $^1/_{\mathbf{e}(q_E+1)}$ when the optimal value of φ_1 is used. Similar to the proof of Theorem 4.3, $\mathcal{C}$ may also fail in simulation of the Confirmation and Disavowal protocols with probability $(q_{H_3} + q_{CD})2^{-k}$. Following the proof, given $\varepsilon_{\mathcal{D}_{II}}$ as the success probability of $\mathcal{D}_{II}$, $\mathcal{C}$'s success probability is at least $\frac{\varepsilon_{\mathcal{D}_{II}} - (q_{H_3}+q_{CD})2^{-k}}{\mathbf{e}(q_E+1)}$. □

4.5.2 Efficiency and Extensions

Efficiency: Efficiency is of the major concerns when designing cryptographic schemes. The efficiency of pairing-based cryptographic scheme is usually evaluated based on the number of the pairing evaluations, exponentiation and scalar multiplications. Pairing evaluation is far more expensive comparing to exponentiation and scalar multiplications and is the main benchmark when evaluating the efficiency of a pairing-based cryptographic scheme. For instance, if we are using the Java library proposed by (Tan, Heng, & Goi, 2010) in order to implement a pairing-based scheme, a scalar multiplication is approximately 1000 times and exponentiation is about 100 times faster than pairing evaluation.

Table 4.1 below illustrates the efficiency comparison between our proposed scheme and the only certificateless undeniable signature scheme which is secure in the strong security model (Duan, 2008). It depicts the number of pairing evaluations (pe), exponentiation in $\mathbb{G}_2$ (ex), and scalar multiplications in $\mathbb{G}_1$ (sm) in both schemes.

Table 4.1: Efficiency Comparison of Certificateless Undeniable Signature Schemes

		Duan's Scheme (2008)	Our Proposed Scheme
Sign	-	$2pe+1ex$	$1pe$
Confirmation	Signer	$6pe+3ex+3sm$	$4pe+1ex+1sm$
	Verifier	$8pe+6ex+1sm$	$7pe+3ex$
Disavowal	Signer	$10pe+8ex+4sm$	$6pe+4ex+2sm$
	Verifier	$8pe+7ex+1sm$	$7pe+4ex$
Signature Length	-	$\|\mathbb{G}_2\|+\|r\|$	$\|\mathbb{G}_2\|+\|r\|$

As it is shown in Table 4.1, while the signature length of our scheme is equal to the one proposed by Duan (2008), our scheme is much more efficient in signature generation, proof generation, and proof verification. Besides, the length of our Confirmation and Disavowal proofs are each $2q$-bit shorter comparing to the ones in the Duan scheme.

Furthermore, we can reduce our signature size by using Katz and Wang's (2003) technique by replacing the l-length random value r with a single bit while maintaining the same security level.

Convertibility: A signer in our scheme is able to selectively convert her signatures to universally verifiable ones by omitting the trapdoor functions in the Confirmation and Disavowal protocols. In order to generate a universally verifiable proof on a valid tuple $(m,\sigma,ID_S,TV_S,TS_S)$, signed by the signer (with identity ID_S and public keys (TV_S,TS_S)), she computes $N=e(P,Y)$, and $O=e(H_2(m,r,ID_S,TV_S,TS_S),Y)$, forms $h_{SC}=H_3(N,O,m,\sigma)$, and calculates $B=Y-h_{SC}S_A$ to publish the proof as (h_{SC},B). It can be easily shown that any user in the system is able to verify the validity of the signature σ using the proof (h_{SC}, B). This technique can be directly applied to generate universally verifiable Disavowal proofs for an invalid signature.

4.6 Summary

In this chapter, we first analysed the security of Zhao and Ye's (2012) efficient certificateless undeniable signature scheme and mounted two attacks by targeting the invisibility and non-impersonation of the their scheme. In addition, we proffered a revised scheme which prevents both of the attacks and provides the signer with the option to selectively convert her undeniable signatures to publicly verifiable ones. However, the revised scheme is only secure in the weak security model. We then proposed a new provably secure certificateless undeniable signature scheme. The new scheme is secure in the strong security model and is better than Duan's scheme (2008) (which is the only secure scheme in the strong security model) in terms of efficiency as it requires less pairing evaluations in its signature generation, proof generation and proof verification. We then proved the unforgeability of our scheme under the hardness of the BDH problem and related its invisibility to the hardness of the DBDH problem in the random oracle model.

CHAPTER 5

DESIGN OF A CONVERTIBLE CERTIFICATELESS UNDENIABLE SIGNATURE SCHEMES

5.1 Introduction

The notion of undeniable signature schemes (Chaum & van Antwerpen, 1989) was proposed to suit the signer's need in situations when privacy is of the main concern. Basically, undeniable signature schemes provide authentication while preserving the privacy of the signer. The validity or invalidity of an undeniable signature can only be verified with the cooperation of its signer. In order to address non-repudiation, undeniable signature schemes are equipped with an additional protocol (i.e. Disavowal protocol) which enables the signer to deny the validity of invalid signatures in court.

Boyar et al. (1991) proffered the notion of convertible undeniable signatures. The new notion enables the signer of an undeniable signature to convert her signatures to ordinary digital signatures. This feature becomes favourable in situations where the signed data lose their sensitivity and the signer decides to make them publicly verifiable. The conversion can take place in two forms: selective conversion which allows the signer to convert a single signature, and universal conversion which enables the signer to convert all her undeniable signatures to publicly verifiable ones. Since its introduction, many variations of convertible undeniable signature schemes have been proposed to the literature (Kurosawa & Takagi, 2006; Laguillaumie & Vergnaud, 2005).

Contributions

In this chapter, we define the security models of convertible certificateless undeniable signature schemes for the first time. More precisely, we formally define the notions of existential unforgeability, invisibility, and anonymity of convertible undeniable signature schemes in a certificateless setting. Following the work of Huang et al. (2007), we consider the strongest type of adversary (super Type I/II adversary) in defining our proposed security models.

We then propose the first concrete convertible certificateless undeniable signature scheme. Initially, signatures in our scheme can only be verified with the cooperation of their signer via the Confirmation and Disavowal protocols. Howbeit, the signer has the ability to convert her undeniable signatures to universally verifiable ones via the selective or universal convert algorithm. We employ the pairing-based version of Jakobsson et al.'s (1996) method to provide non-interactive designated verifier proofs in the Confirmation and Disavowal protocols of our scheme to protect our scheme from blackmailing (Desmedt & Yung, 1991; Jakobsson, 1995) and man-in-the-middle (Desmedt et al., 1987) attacks. Furthermore, we make use of the binding method (Al-Riyami & Paterson, 2003) to lift the trust level on KGC to trust level 3 in Girault's hierarchy (1991). Lastly, we prove the security of our scheme based on the hardness of some hard well-known problems in the random oracle model.

The organisation of the rest of this chapter is as follows. We first define the notion of convertible certificateless undeniable signature schemes in Section 5.2. We formalise the security models of convertible undeniable signature schemes in a certificateless setting in Section 5.3. In Section 5.4, we put forth our concrete scheme, provide a formal security analysis and discuss about its efficiency and extensions. We conclude this chapter in Section 5.5.

5.2 Convertible Certificateless Undeniable Signature Scheme

A convertible certificateless undeniable signature scheme consists of the following algorithms and protocols:

Setup: A probabilistic algorithm that is run by the KGC and takes as input security parameter(s), and returns the KGC's key pair (s, P_{Pub}). Where s is the master secret key and P_{Pub} is the corresponding public key. Moreover, it outputs the system's public parameters *params* which is shared in the system. For the sake of brevity, we omit *params* as the input of the rest of the algorithms/protocols.

Set-secret-value and public-key: This algorithm is run by the user, it picks at random $x_{ID}, z_{ID} \in X$ as her secret values and computes the corresponding public key as $PK_{ID} = (PV_{ID}, PS_{ID})$.

Partial-private-key-extract: This algorithm is run by the KGC and takes as input the master secret key s, a user's identity ID and public key PK_{ID}, it returns the partial private keys of the user as DV_{ID} and DS_{ID}.

Set-private-key: This algorithm is run by the user and takes as input a user's identity ID, secret values (x_{ID}, z_{ID}), and partial private keys (DV_{ID}, DS_{ID}) and outputs the user's private key as a signing key pair (z_{ID}, DS_{ID}) and a verifying key pair (x_{ID}, DV_{ID}).

Sign: This algorithm is run by the signer and takes as input a message m to be signed, the signer's identity ID_S, and private key (a signing key pair (z_S, DS_S) and a verifying key pair (x_S, DV_S)), it outputs a convertible certificateless undeniable signature σ.

Verify: This algorithm takes as input a message-signature pair (m, σ), the signer's identity ID_S (and public key $PK_S = (PS_S, PV_S)$) and the verifying key pair (x_S, DV_S). It outputs a decision bit $b \in \{valid, invalid\}$.

Confirmation protocol: A protocol (conceivably non-interactive) that takes as input a valid message-signature pair (m, σ), the alleged signer's identity ID_S and her verifying key pair (x_S, DV_S), and possibly the identity ID_V and public key PK_V of a designated verifier. It outputs a non-transferable (possibly non-interactive and designated verifier) proof which can convince the verifier ID_V about the validity of the signature σ.

Disavowal protocol: Similar to the Confirmation protocol, where an invalid message-signature pair (m, σ) is provided and the claimed signer generates a proof in order to prove the invalidity of the signature.

Selective-conv: This algorithm takes as input a message-signature pair (m, σ), the alleged signer's identity ID_S and the verifying key pair (x_S, DV_S). It outputs a selective token $tk_{(m,\sigma)}^{(ID_S, PK_S)}$ on the validity/invalidity of (m, σ).

Selective-vfy: This algorithm takes as input a message-signature pair (m,σ), the alleged signer's identity ID_S (with public key PK_S) and a selective token $tk_{(m,\sigma)}^{(ID_S,PK_S)}$, it outputs a decision bit $d \in \{valid, invalid\}$.

Universal-conv: This algorithm takes as input the signer's identity ID_S and her verifying key pair (x_S, DV_S), it outputs a universal token $tk_*^{(ID_S,PK_S)}$.

Universal-vfy: This algorithm takes as input a message-signature pair (m,σ), the signer's identity ID_S (with public key PK_S) and a universal token $tk_*^{(ID_S,PK_S)}$, it outputs a decision bit $d \in \{valid, invalid\}$.

5.3 Security Models of Convertible Certificateless Undeniable Signature Schemes

Since there is no certificate to deliver the authentication of the users' public keys in certificateless systems, we always consider two types of adversaries when defining the security model of schemes in certificateless systems (Al-Riyami & Paterson, 2003). A Type I adversary who can replace the public key of any user with public key of his choice, but has no access to the master secret key, and a Type II adversary who has complete knowledge on the master secret key, but is not allowed to replace the public key of the target user. More precisely, a Type II adversary is assumed to have knowledge on all the users' partial private keys. The security models which are defined in this section are inspired by the works on certificateless signatures (Al-Riyami & Paterson, 2003; Huang et al., 2007; Li et al., 2005; Duan, 2008) and convertible undeniable signature schemes (Boyar et al., 1991; Huang, Mu, Susilo, & Wu, 2007; Kurosawa & Takagi, 2006; Laguillaumie & Vergnaud, 2005) in the literature.

Following the work of Huang et al. (2007), we assume that the Sign oracle in our security model is able to generate valid signatures even for public keys that were replaced by the adversary. This strong assumption results in a more powerful security model.

Note that in our security model, we allow the Type I adversary to query for the partial private keys of any user in the system but the target user.

Definition 5.1. *A convertible certificateless undeniable signature scheme is said to* *be existentially unforgeable under adaptive chosen message, identity and public key attacks if no PPT Type I adversary $\mathcal{F}_I$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ chooses a security parameter k, runs the Setup algorithm, and sends the system public parameters *params* to $\mathcal{F}_I$.
2. $\mathcal{F}_I$ initiates polynomially bounded number of queries to the following oracles:

Hash oracle $(\mathcal{O}^{hash})$**:** $\mathcal{F}_I$ can query for the hash value of any input of its choice to all hash oracles H.

Public ke request $(\mathcal{O}^{pub\_key\_req})$**:** Upon receiving a public key query on an identity ID, the challenger picks the secret values (x_{ID}, z_{ID}) of the user and computes the corresponding public key $PK_{ID} = (PS_{ID}, PV_{ID})$ and sends it to $\mathcal{F}_I$.

Partial private key extract $(\mathcal{O}^{part\_key\_extract})$**:** $\mathcal{F}_I$ is able to query for the partial private keys of any user with identity ID and public key PK_{ID}. Upon receiving such query, $\mathcal{C}$ computes the corresponding partial private keys (DV_{ID}, DS_{ID}) and delivers them to $\mathcal{F}_I$.

Secret value extract $(\mathcal{O}^{sec\_val\_extract})$**:** $\mathcal{F}_I$ is able to query for the secret values of any user with identity ID and public key PK_{ID}. In order to respond to such queries, $\mathcal{C}$ returns the corresponding secret values (x_{ID}, z_{ID}) to $\mathcal{F}_I$. Note that $\mathcal{F}_I$ is not allowed to query for the secret values of public keys that it had replaced prior to this query.

Public key replacement $(\mathcal{O}^{pub\_key\_replace})$**:** $\mathcal{F}_I$ is allowed to replace the public key $PK_{ID} = (PS_{ID}, PV_{ID})$ of any user with public key of his choice $PK'_{ID} = (PS'_{ID}, PV'_{ID})$ (which he may know the corresponding secret values).

Sign oracle $(\mathcal{O}^{sign})$**:** $\mathcal{F}_I$ chooses a message m, and requests a signature for a signer with identity ID and public key PK_{ID}. $\mathcal{C}$ then computes a valid signature σ and returns it to $\mathcal{F}_I$. Note that the public key PK_{ID} could have been replaced prior to this query.

Verify oracle $(\mathcal{O}^{verify})$**:** $\mathcal{F}_I$ generates a message-signature pair (m, σ) for a signer with identity ID and public key PK_{ID} and queries $\mathcal{O}^{verify}$. Upon receiving such query, $\mathcal{C}$ checks the validity of the signature and returns a decision bit $b \in \{valid, invalid\}$ to $\mathcal{F}_I$.

Confirmation/Disavowal oracle $(\mathcal{O}^{conf/disav})$**:** $\mathcal{F}_I$ forms a tuple (m, ID, PK_{ID}, σ), and queries for a non-transferable (possibly designated verifier) proof on validity/invalidity of (m, ID, PK_{ID}, σ). $\mathcal{C}$ starts by initiating $\mathcal{O}^{verify}$ on (m, ID, PK_{ID}, σ) and generates either the Confirmation or Disavowal proof based on the output of $\mathcal{O}^{verify}$.

Selective-conv oracle $(\mathcal{O}^{sel-conv})$**:** $\mathcal{F}_I$ forms a tuple (m, ID, PK_{ID}, σ), and queries for a selective token. $\mathcal{C}$ starts by initiating $\mathcal{O}^{verify}$ on (m, ID, PK_{ID}, σ) and generates a selective token $tk_{(m,\sigma)}^{(ID,PK_{ID})}$ on validity/invalidity of the tuple (m, ID, PK_{ID}, σ) based on the output of $\mathcal{O}^{verify}$.

Universal-conv oracle $(\mathcal{O}^{univ-conv})$**:** $\mathcal{F}_I$ is allowed to query for the universal token of any user (with identity ID and public key PK_{ID}) in the system. Upon receiving such query, $\mathcal{C}$ retrieves the verifying key pair (x_{ID}, DV_{ID}) of the user, and sends the universal token $tk_*^{(ID,PK_{ID})}$ to $\mathcal{F}_I$. Note that $\mathcal{F}_I$ is prohibited to query for the universal token of the identities for which he had replaced the public keys.

3. At the end of the game, $\mathcal{F}_I$ outputs a tuple $(m^*, \sigma^*, ID^*, PK_{ID^*})$. The adversary $\mathcal{F}_I$ wins the game if it never queried (ID^*, PK_{ID^*}) to $\mathcal{O}^{part\_key\_extract}$ and the tuple (m^*, ID^*, PK_{ID^*}) was never queried to $\mathcal{O}^{sign}$.

Definition 5.2. *A convertible certificateless undeniable signature scheme is said to have the property of invisibility under adaptive chosen message, identity and public key attacks if no PPT Type I adversary $\mathcal{D}_I$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ chooses a security parameter k, runs the Setup algorithm, and sends the system public parameters $params$ to $\mathcal{D}_I$.
2. $\mathcal{D}_I$ is able to perform polynomially bounded number of queries as in the above game.
3. After the first round of queries, $\mathcal{D}_I$ outputs a tuple (m^*, ID^*, PK_{ID^*}) which it would want to be challenged on. Note that (ID^*, PK_{ID^*}) should have never been

submitted to $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{univ-conv}$. The challenger $\mathcal{C}$ then computes the signature based on the outcome of a hidden coin toss $b \in \{0,1\}$. If $b = 0$, it chooses a random σ^* from the signature space $\mathcal{S}$. Alternatively, if $b = 1$, $\mathcal{C}$ computes the signature σ^* by running the $\mathcal{O}^{sign}$ as normal. Lastly, $\mathcal{C}$ returns σ^* to $\mathcal{D}_I$.

4. $\mathcal{D}_I$ initiates the second round of queries. Nevertheless, $\mathcal{D}_I$ is prohibited from initiating the following queries:
   a) Querying (m^*, ID^*, PK_{ID^*}) to $\mathcal{O}^{sign}$.
   b) Querying (ID^*, PK_{ID^*}) to either $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{univ-conv}$.
   c) Finally, querying $(m^*, ID^*, PK_{ID^*}, \sigma^*)$ to either $\mathcal{O}^{conf/disav}$ or $\mathcal{O}^{sel-conv}$.
5. At the end of the game, $\mathcal{D}_I$ outputs a bit $b^{'}$. $\mathcal{D}_I$ wins the game if $b^{'} = b$.

Definition 5.3. *A convertible certificateless undeniable signature scheme is said to have the property of anonymity under adaptive chosen message, identity and public key attacks if no PPT Type I adversary $\mathcal{D}_I$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ chooses a security parameter k, runs the Setup algorithm, and sends the system public parameters *params* to $\mathcal{D}_I$.
2. $\mathcal{D}_I$ is able to perform polynomially bounded number of queries as in the game of Definition 5.1.
3. After the first round of queries, $\mathcal{D}_I$ outputs two challenge tuples (m^*, ID_0, PK_{ID_0}) and (m^*, ID_1, PK_{ID_1}). Note that (ID_0, PK_{ID_0}) or (ID_1, PK_{ID_1}) should have never been queried to $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{univ-conv}$. $\mathcal{C}$ then tosses a hidden coin $b \in \{0,1\}$, and computes the challenge signature σ_b which is valid for the signer with identity ID_b and public key PK_{ID_b}.
4. $\mathcal{D}_I$ initiates the second round of queries. Nevertheless, $\mathcal{D}_I$ is prohibited from initiating the following queries:
   a) Querying either (m^*, ID_0, PK_{ID_0}) or (m^*, ID_1, PK_{ID_1}) to $\mathcal{O}^{sign}$.
   b) Querying (ID_0, PK_{ID_0}) or (ID_1, PK_{ID_1}) to either $\mathcal{O}^{part\_key\_extract}$ or $\mathcal{O}^{univ-conv}$.
   c) Finally, querying $(m^*, ID_0, PK_{ID_0}, \sigma_b)$ or $(m^*, ID_1, PK_{ID_1}, \sigma_b)$ to either $\mathcal{O}^{conf/disav}$ or $\mathcal{O}^{sel-conv}$.

5. At the end of the game, $\mathcal{D}_I$ outputs a bit b'. $\mathcal{D}_I$ wins the game if $b' = b$.

A Type II adversary is assumed to have knowledge over the master secret key s, and therefore, it can easily compute the partial private key of any user in the system. In order to establish a more powerful security model, we allow the Type II adversary to replace the public key of any user except the target user.

Definition 5.4. *A convertible certificateless undeniable signature scheme is said to be existentially unforgeable under adaptive chosen message, identity and public key attacks if no PPT Type II adversary $\mathcal{F}_{II}$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ chooses a security parameter k, runs the Setup algorithm, and sends the master secret key s and the system public parameters *params* to $\mathcal{F}_{II}$.
2. $\mathcal{F}_{II}$ can initiate queries (polynomially bounded) to all those oracles (excluding $\mathcal{O}^{part\_key\_extract}$) defined in the game of Definition 5.1.

3. At the end of the game, $\mathcal{F}_{II}$ outputs a tuple $(m^*, \sigma^*, ID^*, PK_{ID^*})$. The adversary $\mathcal{F}_{II}$ wins the game if it never queried (ID^*, PK_{ID^*}) to either $\mathcal{O}^{sec\_val\_extract}$ or $\mathcal{O}^{pub\_key\_replace}$, and (m^*, ID^*, PK_{ID^*}) was never queried to $\mathcal{O}^{sign}$.

Definition 5.5. *A convertible certificateless undeniable signature scheme is said to have the property of invisibility under adaptive chosen message, identity and public key attacks if no PPT Type II adversary $\mathcal{D}_{II}$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ chooses a security parameter k, runs the Setup algorithm, and sends the master secret key s and the system public parameters *params* to $\mathcal{D}_{II}$.
2. $\mathcal{D}_{II}$ is able to perform polynomially bounded number of queries as in the above game.
3. After the first round of queries, $\mathcal{D}_{II}$ outputs a tuple (m^*, ID^*, PK_{ID^*}) which it would want to be challenged on. Note that (ID^*, PK_{ID^*}) should have never been queried to either $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{pub\_key\_replace}$ or $\mathcal{O}^{univ-conv}$. The challenger

$\mathcal{C}$ then computes the signature based on the outcome of a hidden coin toss $b \in \{0,1\}$. If $b = 0$, it chooses a random signature σ^* from the signature space $\mathcal{S}$. Alternatively, if $b = 1$, $\mathcal{C}$ computes the signature σ^* by running $\mathcal{O}^{sign}$ as normal. Lastly, $\mathcal{C}$ returns σ^* to $\mathcal{D}_{II}$.

4. $\mathcal{D}_{II}$ initiates the second round of queries. Nevertheless, $\mathcal{D}_{II}$ is prohibited from initiating the following queries:
   a) Querying (m^*, ID^*, PK_{ID^*}) to $\mathcal{O}^{sign}$.
   b) Querying (ID^*, PK_{ID^*}) to either $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{pub\_key\_replace}$ or $\mathcal{O}^{univ-conv}$.
   c) Finally, querying $(m^*, ID^*, PK_{ID^*}, \sigma^*)$ to either $\mathcal{O}^{conf/disav}$ or $\mathcal{O}^{sel-conv}$.
5. At the end of the game, $\mathcal{D}_{II}$ outputs a bit b'. $\mathcal{D}_{II}$ wins the game if $b' = b$.

Definition 5.6. *A convertible certificateless undeniable signature scheme is said to have the property of anonymity under adaptive chosen message, identity and public key attacks if no PPT Type II adversary $\mathcal{D}_{II}$ has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ chooses a security parameter k, runs the Setup algorithm and sends the master secret key s and the system public parameters *params* to $\mathcal{D}_{II}$.
2. $\mathcal{D}_{II}$ is able to perform polynomially bounded number of queries as in the game of Definition 5.4.
3. After the first round of queries, $\mathcal{D}_{II}$ outputs two challenge tuples (m^*, ID_0, PK_{ID_0}) and (m^*, ID_1, PK_{ID_1}). Note that (ID_0, PK_{ID_0}) or (ID_1, PK_{ID_1}) should have never been queried to $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{pub\_key\_replace}$ or $\mathcal{O}^{univ-conv}$. $\mathcal{C}$ then tosses a hidden coin $b \in \{0,1\}$, and computes the challenge signature σ_b which is valid for the signer with identity ID_b and public key PK_{ID_b}.
4. $\mathcal{D}_{II}$ initiates the second round of queries. Nevertheless, $\mathcal{D}_{II}$ is prohibited from initiating the following queries:
   a) Querying either (m^*, ID_0, PK_{ID_0}) or (m^*, ID_1, PK_{ID_1}) to $\mathcal{O}^{sign}$.
   b) Querying (ID_0, PK_{ID_0}) or (ID_1, PK_{ID_1}) to either $\mathcal{O}^{sec\_val\_extract}$, $\mathcal{O}^{pub\_key\_replace}$ or $\mathcal{O}^{univ-conv}$.
   c) Finally, querying $(m^*, ID_0, PK_{ID_0}, \sigma_b)$ or $(m^*, ID_1, PK_{ID_1}, \sigma_b)$ to either $\mathcal{O}^{conf/disav}$ or $\mathcal{O}^{sel-conv}$.

5. At the end of the game, $\mathcal{D}_{II}$ outputs a bit $b^{'}$. $\mathcal{D}_{II}$ wins the game if $b^{'} = b$.

Following the work of Galbraith and Mao (2003), Huang et al. (2007) proved that the notions of anonymity and invisibility are equivalent in the context of convertible undeniable signature schemes. Therefore, using the same approach (Huang et al., 2007), we can rely the anonymity of our scheme on the DBDH assumption.

5.4 The Proposed Scheme

In this section, we first propose our convertible certificateless undeniable scheme, provide a security analysis and discuss about its efficiency and extensions.

Setup: By taking as input security parameters k and l, the KGC generates groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q > 2^k$, an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and an arbitrary generator $P \in \mathbb{G}_1$. It then sets its key pair by picking $s \in \mathbb{Z}_q$ randomly as the master secret key and computing $P_{Pub} = sP$ as the corresponding public key. Lastly, it chooses cryptographic hash functions H_i for $i \in \{1,\ldots,8\}$ and publishes the system public parameters as $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, P_{Pub}, H_i\ for\ i \in \{1,\ldots,8\})$. We remark that H_i will be viewed as random oracle in our security proof.

Set-user-secret-value and public-key: A user with identity ID starts by picking z_{ID} ,$x_{ID} \in \mathbb{Z}_q$ randomly as her secret values, computing $PS_{ID} = z_{ID}P$ and $PV_{ID} = x_{ID}P$, and forming her public key as $PK_{ID} = (PS_{ID}, PV_{ID})$.

Set-partial-private-key: Given the master secret key s, and the user's identity ID with public key PK_{ID}, the KGC computes $Q_{k_{ID}} = H_1(ID, PK_{ID})$ and $Q_{v_{ID}} = H_2(ID, PK_{ID}, \text{“}verify\text{”})$, and outputs the partial private keys as $DS_{ID} = sQ_{k_{ID}}$ and $DV_{ID} = sQ_{v_{ID}}$.

Set-private-key: The private key of the user ID with public key $PK_{ID} = (PV_{ID}, PS_{ID})$ will be set as two pairs: the signing key pair (z_{ID}, DS_{ID}) and the verifying key pair (x_{ID}, DV_{ID}).

Sign: In order to sign a message $m \in \{0,1\}^*$, the signer (with identity ID_S and public key PK_S) works as follows:

- Pick a random string $r \in \{0,1\}^l$ to compute $h_3 = H_3(m, r, ID_S)$, and $h_4 = H_4(m, r, ID_S, PK_S)$.
- Choose $t \in \mathbb{Z}_q$ randomly and use her private key pairs (z_S, DS_S) and (x_S, DV_S) to compute the values of $\mathcal{S}_1 = e(h_3, x_S Q_{v_S}) e(h_4, DV_S)$, $\mathcal{S}_2 = tP$, and $\mathcal{S}_3 = DS_S + tH_5(\mathcal{S}_1, \mathcal{S}_2, m, ID_S) + z_S H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID_S, PK_S)$.

The signer then forms the signature as $\sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$.

Verify: In order to verify the validity of a tuple $(m, ID_S, PK_S, \sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3))$, the signer works as follows.

- Check if $e(P, \mathcal{S}_3) = e(P_{Pub}, Q_{k_S}) e(\mathcal{S}_2, H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID_S)) e(PS_S, H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID_S, PK_S))$ does not hold, output $reject$.
- Else, check if $\mathcal{S}_1 = e(H_3(m, r, ID_S), x_S Q_{v_S}) e(H_4(m, r, ID_S, PK_S), DV_S)$ holds, output $valid$. Otherwise, it outputs $invalid$.

Confirmation protocol: Given a designated verifier's identity ID_V with public key $PK_V = (PV_V, PS_V)$ and a valid message-signature (m, σ) pair to be confirmed, the signer (with identity ID_S and public key PK_S) works as follows in order to generate a non-transferable Confirmation proof transcript for the designated verifier.

1. Compute $Q_{v_V} = H_2(ID_V, PK_V, \text{“}verify\text{”}), h_3 = H_3(m, r, ID_S)$, and $h_4 = H_4(m, r, ID_S, PK_S)$.
2. Pick $K, W \in \mathbb{G}_1$ and $\beta, \tau, \nu \in \mathbb{Z}_q$ at random to calculate the following:

$$n_1 = e(P, K) e(P_{Pub}, Q_{v_V})^{\nu} \tag{5.1}$$

$$n_2 = \tau P + \nu PV_V \tag{5.2}$$

$$g_1 = e(P, W) \tag{5.3}$$

$$g_2 = e(P,P)^{\beta} \tag{5.4}$$

$$g_3 = e(h_3, Q_{v_S})^{\beta} e(h_4, W) \tag{5.5}$$

3. Set the values of $h_C = H_7(n_1, n_2, g_1, g_2, g_3, \sigma)$, $b = \beta - (h_C + \nu)x_S$ and $B = W - (h_C + \nu)DV_S$ and output the Confirmation proof transcript as $(K, \nu, \tau, b, B, h_C)$.

In order to verify the veracity of the Confirmation proof transcript $(K, \nu, \tau, b, B, h_C)$, the designated verifier checks if $e(P, \mathcal{S}_3) = e(P_{Pub}, Q_{k_S})e(\mathcal{S}_2, H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID_S))e(PS_S, H_6 (\mathcal{S}_1\mathcal{S}_2, m, ID_S, PK_S))$ holds, then it forms $Q_{v_S} = H_2(ID_V, PK_V, \text{“}verify\text{”}), h_3 = H_3(m, r, ID_S)$ and $h_4 = H_4(m, r, ID_S, PK_S)$ and computes the following:

$$n_1^{'} = e(P, K)e(P_{Pub}, Q_{v_V})^{\nu} \tag{5.6}$$

$$n_2^{'} = \tau P + \nu PV_V \tag{5.7}$$

$$g_1^{'} = e(P, B)e(P_{Pub}, Q_{v_S})^{(h_C+\nu)} \tag{5.8}$$

$$g_2^{'} = e(P, P)^{b} e(P, PV_S)^{(h_C+\nu)} \tag{5.9}$$

$$g_3^{'} = e(h_3, Q_{v_S})^{b} e(h_4, B)\mathcal{S}_1^{(h_C+\nu)} \tag{5.10}$$

The verifier ID_V will only accept the proof if $h_C = H_7(n_1^{'}, n_2^{'}, g_1^{'}, g_2^{'}, g_3^{'}, \sigma)$.

Disavowal protocol: Given a designated verifier's identity ID_V with public key $PK_V = (PV_V, PS_V)$ and an invalid message-signature (m, σ) pair to be disavowed, the signer (with identity ID_S and public key PK_S) works as follows in order to generate a non-transferable Disavowal proof transcript for the designated verifier.

1. Parse σ into $(r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ and compute $Q_{v_V} = H_2(ID_V, PK_V, \text{“}verify\text{”}), h_3 = H_3(m, r, ID_S)$, and $h_4 = H_4(m, r, ID_S, PK_S)$.
2. Pick $K \in \mathbb{G}_1$ and $\tau, \nu \in \mathbb{Z}_q$ at random in order to compute the values of $n_1 = e(P, K)e(P_{Pub}, Q_{v_V})^{\nu}$ and $n_2 = \tau P + \nu PV_V$.

3. Pick $\omega \in \mathbb{Z}_q$ and compute $C = (\frac{e(h_3,Q_{v_S})^{x_S} e(h_4,DV_S)}{\mathcal{S}_1})^{\omega}$.

4. The signer has to prove her knowledge of a tuple $(T,\mu,\alpha) \in \mathbb{G}_1 \times \mathbb{Z}_q \times \mathbb{Z}_q$ where $C = \frac{e(h_3,Q_{v_S})^{\mu} e(h_4,T)}{\mathcal{S}_1^{\alpha}}$, $e(P,T) = e(Q_{v_S},P_{Pub})^{\alpha}$ and $\mu P = \alpha PV_S$. In order to do so, she works as follows.

   a) Pick $U \in \mathbb{G}_1$ and $a,i \in \mathbb{Z}_q$ at random and compute the following:

$$j_1 = \frac{e(P,U)}{e(Q_{v_S},P_{Pub})^a} \tag{5.11}$$

$$j_2 = \frac{e(P,P)^i}{e(P,PV_S)^a} \tag{5.12}$$

$$j_3 = \frac{e(h_3,Q_{v_S})^i e(h_4,U)}{\mathcal{S}_1^a} \tag{5.13}$$

   b) Set the values of $h_D = H_8(C,n_1,n_2,j_1,j_2,j_3,\sigma)$, $Y = U-(h_D+\nu)T$, $w_1 = i-(h_D+\nu)\mu$ and $w_2 = a-(h_D+\nu)\alpha$ in order to form the proof as $(C,K,\tau,\nu,h_D,Y,w_1,w_2)$.

Upon receiving the Disavowal proof transcript, the designated verifier checks if $e(P,\mathcal{S}_3) = e(P_{Pub},Q_{k_S})e(\mathcal{S}_2,H_5(\mathcal{S}_1,\mathcal{S}_2,m,ID_S))e(PS_S,H_6(\mathcal{S}_1,\mathcal{S}_2,m,ID_S,PK_S))$ and $C \neq 1$ hold, he forms $h_3 = H_3(m,r,ID_S)$ and $h_4 = H_4(m,r,ID_S,PK_S)$ and computes as follows.

$$n_1^{'} = e(P,K)e(P_{Pub},Q_{v_V})^{\nu} \tag{5.14}$$

$$n_2^{'} = \tau P + \nu PV_V \tag{5.15}$$

$$j_1^{'} = \frac{e(P,Y)}{e(Q_{v_S},P_{Pub})^{w_2}} \tag{5.16}$$

$$j_2^{'} = \frac{e(P,P)^{w_1}}{e(P,PV_S)^{w_2}} \tag{5.17}$$

$$j_3^{'} = \frac{e(h_3,Q_{v_S})^{w_1} e(h_4,Y)}{\mathcal{S}_1^{w_2}} C^{(h_D+\nu)} \tag{5.18}$$

The verifier ID_V will only accept the proof if $h_D = H_8(C,n_1^{'},n_2^{'},j_1^{'},j_2^{'},j_3^{'},\sigma)$.

Selective-conv: Given a message-signature pair (m,σ), the signer (with identity ID_S and public key PK_S) generates a selective token $tk_{(m,\sigma)}^{(ID_S,PK_S)}$ on validity/invalidity of the provided message-signature pair (m,σ) as follows.

In order to generate a selective token $tk_{(m,\sigma)}^{(ID_S,PK_S)}$, given (m,σ) is valid, the signer works as follows.

1. Pick $W \in \mathbb{G}_1$ and $\beta \in \mathbb{Z}_q$ at random, and compute $h_3 = H_3(m,r,ID_S)$, and $h_4 = H_4(m,r,ID_S,PK_S)$ in order to compute the following:
$$g_1 = e(P,W) \tag{5.19}$$
$$g_2 = e(P,P)^{\beta} \tag{5.20}$$
$$g_3 = e(h_3,Q_{v_S})^{\beta} e(h_4,W) \tag{5.21}$$
2. Set the values of $h_C = H_7(g_1,g_2,g_3,\sigma)$, $b = \beta - h_C x_S$ and $B = W - h_C DV_S$ and output the selective token as $tk_{(m,\sigma)}^{(ID_S,PK_S)} = (b,B,h_C)$.

Given (m,σ) is invalid, the signer works as follow so as to compute a selective token $tk_{(m,\sigma)}^{(ID_S,PK_S)}$ on the invalidity of (m,σ).

1. Pick $\omega \in \mathbb{Z}_q$ at random to compute $C_1 = (\frac{e(h_3,Q_{v_S})^{x_S} e(h_4,DV_S)}{S_1})^{\omega}$.
2. Same as in the Disavowal protocol, the signer has to prove her knowledge of a tuple $(T,\mu,\alpha) \in \mathbb{G}_1 \times \mathbb{Z}_q \times \mathbb{Z}_q$ where $C = \frac{e(h_3,Q_{v_S})^{\mu} e(h_4,T)}{S_1^{\alpha}}$, $e(P,T) = e(Q_{v_S},P_{Pub})^{\alpha}$ and $\mu P = \alpha PV_S$. In order to do so, she computes as follows.
   a) Pick $U \in \mathbb{G}_1$ and $a,i \in \mathbb{Z}_q$ at random to compute the following:
$$j_1 = \frac{e(P,U)}{e(Q_{v_S},P_{Pub})^{a}} \tag{5.22}$$
$$j_2 = \frac{e(P,P)^{i}}{e(P,PV_S)^{a}} \tag{5.23}$$
$$j_3 = \frac{e(h_3,Q_{v_S})^{i} e(h_4,U)}{S_1^{a}} \tag{5.24}$$
   b) Set the values of $h_D = H_8(C,j_1,j_2,j_3,\sigma)$, $Y = U - h_D T$, $w_1 = i - h_D\mu$ and $w_2 = a - h_D\alpha$ and output the selective token as $tk_{(m,\sigma)}^{(ID_S,PK_S)} = (C,h_D,Y,w_1,w_2)$.

Selective-vfy: Given a message-signature pair (m,σ), the identity of the alleged signer ID_S with public key PK_S, and a selective token $tk_{(m,\sigma)}^{(ID_S,PK_S)}$, the verifier computes $h_3 = H_3(m, r, ID_S)$ and $h_4 = H_4(m, r, ID_S, PK_S)$ and works as follow in order to verify the validity/invalidity of the provided message-signature pair $(m, \sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3))$.

- Provided (m, σ) is valid for the signer. The verifier works as follow so as to verify the validity of the signature σ using the knowledge of the selective token $tk_{(m,\sigma)}^{(ID_S, PK_S)} = (b, B, h_C)$. The verification starts by checking if $e(P, \mathcal{S}_3) = e(P_{Pub}, Q_{k_S}) e(\mathcal{S}_2, H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID_S)) e(PS_S, H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID_S, PK_S))$ holds. The verifier then computes the values of $g_1^{'} = e(P, B) e(P_{Pub}, Q_{v_S})^{h_C}$, $g_2^{'} = e(P, P)^b e(P, PV_S)^{h_C}$ and $g_3^{'} = e(h_3, Q_{v_S})^b e(h_4, B) \mathcal{S}_1^{h_C}$. He will accept the proof if and only if $h_C = H_7(g_1^{'}, g_2^{'}, g_3^{'}, \sigma)$.
- Upon receiving the selective token $tk_{(m,\sigma)}^{(ID_S, PK_S)} = (C, h_D, Y, w_1, w_2)$, for an invalid message-signature pair (m, σ), the verifier first checks if $e(P, \mathcal{S}_3) = e(P_{Pub}, Q_{k_S}) e(\mathcal{S}_2, H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID_S)) e(PS_S, H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID_S, PK_S))$ and $C \neq 1$ hold. He computes the values of $j_1^{'} = \frac{e(P,Y)}{e(Q_{v_S} P_{Pub})^{w_2}}$, $j_2^{'} = \frac{e(P,P)^{w_1}}{e(P, PV_S)^{w_2}}$, and $j_3^{'} = \frac{e(h_3, Q_{v_S})^{w_1} e(h_4, Y)}{\mathcal{S}_1^{w_2} C^{h_D}}$. He will accept the proof if and only if $h_D = H_8(C, j_1^{'}, j_2^{'}, j_3^{'}, \sigma)$.

Universal-conv: The signer (with identity ID_S and public key PK_S) is able to convert all his undeniable signatures to conventional digital signatures by publishing the universal token as $tk_*^{(ID_S, PK_S)} = (DV_S, x_{ID_S})$.

Universal-vfy: Given the universal token $tk_*^{(ID_S, PK_S)} = (DV_S, x_{ID_S})$, anyone in the system can verify the validity/invalidity of any message-signature pair $(m, \sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3))$ issued by the signer. The verifier first validates the veracity of the token $tk_*^{(ID_S, PK_S)} = (DV_S, x_{ID_S})$ by checking if $e(P, DV_S) = e(P_{Pub}, Q_{v_S})$ and $x_{ID_S} P = PV_V$ hold. If the token is valid, then he can check the correctness of the message-signature pair by checking if $e(P, \mathcal{S}_3) = e(P_{Pub}, Q_{k_S}) e(\mathcal{S}_2, H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID_S)) e(PS_S, H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID_S, PK_S))$ holds and decide if the validity or invalidity of the signature by checking if the equation $\mathcal{S}_1 = e(H_3(m, r, ID_S), Q_{v_S})^{x_S} e(H_4(m, r, ID_S, PK_S), DV_S)$ holds or not.

5.4.1 Security Analysis

In the Confirmation and Disavowal protocols of our scheme, we employed the pairing-based version of the non-interactive designated verifier proofs of Jakobsson et al. (1996). Therefore, we can use the same approach as in Chapter 3 (see Section 3.4.1) in order to prove the soundness, completeness, non-impersonation, and non-transferability of the Confirmation and Disavowal protocols of our proposed scheme.

Theorem 5.1. *If there exists a Type I adversary $\mathcal{F}_I$ that can submit q_E partial private key extract queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, q_{CV} selective and universal conversion queries, and q_{H_i} queries to random oracle H_i for $i \in \{1,\ldots,8\}$ and be able to succeed in an existential forgery (win the game defined in Definition 5.1) against our proposed scheme with a non-negligible success probability $\varepsilon_{\mathcal{F_I}}$, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{F}_I$ to solve a random instance (P, aP, bP) of the CDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{F_I}} - (q_{CD} + q_{H_7} + 1)2^{-k}}{\mathbf{e}(q_E + 1)}$$

Proof. We show that if there exists a Type I adversary $\mathcal{F}_I$ which can win the game defined in Definition 5.1, then one can construct a PPT algorithm $\mathcal{C}$ that makes use of $\mathcal{F}_I$ to solve a random instance (P, aP, bP) of the CDH problem with probability at least $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ works as $\mathcal{F}_I$'s challenger, it starts by running the Setup algorithm and providing $\mathcal{F}_I$ with the system public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, P_{Pub}, H_i$ where $i \in \{1,\ldots,8\})$. Where $P_{Pub} = aP$ and a is unknown to $\mathcal{C}$.

$\mathcal{F}_I$ is allowed to query different random oracles H_i for $i = \{1,\ldots,8\}$ and other oracles (e.g. partial private key extract, public key replacement, etc.) as defined in the game of Definition 5.1. $\mathcal{C}$ handles these queries by keeping lists κ_i for $i = \{1,\ldots,8\}$ and a list κ_0 in order to keep track of the values of identities, public keys and the corresponding secret values. Without loss of generality, we assume $\mathcal{F}_I$ behaves well, i.e. $\mathcal{F}_I$ always makes a public key request before it queries on H_1 or H_2 oracles and always makes a H_1 and H_2 query before it requests for the partial private key of the user.

Query on $H_1(ID, PK_{ID})$: To answer such queries on an identity $ID \in \{0,1\}^*$ and public key PK_{ID}, $\mathcal{C}$ picks a random $\alpha \in \mathbb{Z}_q$ and flips a coin X that is truly random taking the value of 0 with probability φ_1 and the value of 1 with probability $1-\varphi_1$ (the value of φ_1 will be computed later in our proof). Next, $\mathcal{C}$ inserts (ID, PK_{ID}, α, X) into κ_1 and returns $H_1(ID, PK_{ID}) = \alpha(bP)$ if $X = 1$ and $H_1(ID, PK_{ID}) = \alpha P$ if $X = 0$.

Query on $H_2(ID, PK_{ID}, \text{“}verify\text{”})$: In order to answer queries on H_2, $\mathcal{C}$ first checks if κ_2 already contained a tuple $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$, it returns βP to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$, adds $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$ to κ_2 and returns βP to $\mathcal{F}_I$.

Query on $H_3(m, r, ID)$: In order to answer queries on H_3, $\mathcal{C}$ first checks if κ_3 already contained a tuple (m, r, ID, γ), then it returns γP to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ picks a random $\gamma \in \mathbb{Z}_q$, adds (m, r, ID, γ) to κ_3 and returns γP to $\mathcal{F}_I$.

Query on $H_4(m, r, ID, PK_{ID})$: In order to answer queries on H_4, $\mathcal{C}$ first checks if κ_4 already contained $(m, r, ID, PK_{ID}, \eta)$, then it returns ηP to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ picks a random $\eta \in \mathbb{Z}_q$, adds $(m, r, ID, PK_{ID}, \eta)$ to κ_3 and returns ηP to $\mathcal{F}_I$.

Query on $H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID)$: In order to answer queries on H_5, $\mathcal{C}$ first checks if κ_5 already contained $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$, then it returns $\lambda_1 P$ to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ picks a random $\lambda_1 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$ to κ_5 and returns $\lambda_1 P$ to $\mathcal{F}_I$.

Query on $H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID})$: In order to answer queries on H_6, $\mathcal{C}$ first checks if κ_6 already contained $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$, then it returns $\lambda_2 P$ to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ picks a random $\lambda_2 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$ to κ_6 and returns $\lambda_2 P$ to $\mathcal{F}_I$.

Query on H_7 and H_8: Queries on H_7 and H_8 oracles will be handled randomly, and the response will be stored in κ_7 and κ_8 respectively.

Public key request: Upon submitting an identity ID, $\mathcal{C}$ first checks if $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ already exists in κ_0, then it returns $PK_{ID} = (PV_{ID}, PS_{ID})$. Otherwise, $\mathcal{C}$ picks $x_{ID}, z_{ID} \in \mathbb{Z}_q$ randomly and computes $PV_{ID} = x_{ID}P$ and $PS_{ID} = z_{ID}P$, returns (PV_{ID}, PS_{ID}) to $\mathcal{F}_I$, and records $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ in κ_0.

Public key replacement: When $\mathcal{F}_I$ wishes to replace the public key $PK_{ID} = (PV_{ID}, PS_{ID})$ for identity ID with public key of his choice $PK_{ID}^{'} = (PV_{ID}^{'}, PS_{ID}^{'})$, $\mathcal{C}$ first checks κ_0 to find $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$, if such tuple exists, it will replace it with $(ID, -1, -1, PV_{ID}^{'}, PS_{ID}^{'})$, where -1 means that the public key has been replaced. Otherwise, $\mathcal{C}$ adds a tuple $(ID, -1, -1, PV_{ID}^{'}, PS_{ID}^{'})$ to κ_0. In this case if κ_1 and κ_2 contain tuples $(ID, PK_{ID}, \alpha, \dots)$ and $(ID, PK_{ID}, \text{"}verify\text{"}, \beta)$, $\mathcal{C}$ simulates H_1 and H_2 oracles and updates κ_1 and κ_2 respectively.

Secret value extract: In order to respond to a secret value extract query on identity ID with public key $PK_{ID} = (PV_{ID}, PS_{ID})$, $\mathcal{C}$ scans κ_0 for a tuple $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ and returns the secret values of the user ID as pair (x_{ID}, z_{ID}). Where $PK_{ID} = (PV_{ID}, PS_{ID})$ is the original public key of the user.

Partial private key extract: Upon receiving an identity ID, $\mathcal{C}$ scans κ_1 for a tuple (ID, PK_{ID}, α, X), if $X = 1$, $\mathcal{C}$ reports *failure* and terminates the simulation. Otherwise, it searches κ_2 to find $(ID, PK_{ID}, \text{"}verify\text{"}, \beta)$ and returns $(\alpha P_{Pub}, \beta P_{Pub})$ as the partial private keys of ID.

Sign query: $\mathcal{F}_I$ is allowed to query the Sign oracle in order to receive valid signatures on any tuple $(m, ID, PK_{ID} = (PV_{ID}, PS_{ID}))$. Upon receiving such a query, $\mathcal{C}$ works as follows.

1. $\mathcal{C}$ first picks $r \in \{0,1\}^l$ at random and queries $H_3(m, r, ID)$ and $H_4(m, r, ID, PK_{ID})$ in order to retrieve the values of $h_3 = \gamma P$ and $h_4 = \eta P$ respectively. $\mathcal{C}$ then continues to compute $\mathcal{S}_1 = e(h_3, \beta PV_{ID})e(h_4, \beta P_{Pub})$.
2. Thereupon, $\mathcal{C}$ picks a random $v \in \mathbb{Z}_q$ to set $\mathcal{S}_2 = vP_{pub}$. Next, it scans κ_5 in order to find a tuple $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \dots)$ (if such tuple exists, $\mathcal{C}$ picks another v and forms the value of $\mathcal{S}_2 = vP_{pub}$ until no such tuple $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \dots)$ exists

in κ_5) and sets $H_5(\mathcal{S}_1,\mathcal{S}_2,m,ID) = v^{-1}(qP - H_1(ID,PK_{ID}))$.

3. Lastly, $\mathcal{C}$ picks $q \in \mathbb{Z}_q$ at random, and simulate $H_6(\mathcal{S}_1,\mathcal{S}_2,m,ID,PK_{ID})$ to form $\mathcal{S}_3 = qP_{pub} + \lambda_2 PS_{ID}$ and outputs the signature as $\sigma = (r,\mathcal{S}_1,\mathcal{S}_2,\mathcal{S}_3)$.

Verify: $\mathcal{F}_I$ forms a tuple $(m,\sigma^{'},ID_S,PK_S = (PV_S,PS_S))$, where $\sigma^{'}$ is a signature on message m, and PK_{ID} is the public key of the signer with identity ID_S. $\mathcal{F}_I$ is allowed to request for the validity of any such tuple. Upon receiving such query, $\mathcal{C}$ parses $\sigma^{'}$ into $(r^{'},\mathcal{S}_1^{'},\mathcal{S}_2^{'},\mathcal{S}_3^{'})$ and checks if $e(P,\mathcal{S}_3^{'}) = e(P_{Pub},Q_{k_S})e(\mathcal{S}_2^{'},H_5(\mathcal{S}_1^{'},\mathcal{S}_2^{'},m,ID_S))e(PS_S,H_6(\mathcal{S}_1^{'},\mathcal{S}_2^{'},m,ID_S,PK_S)$ holds, it computes $\mathcal{S}_1 = e(H_3(m,r,ID_S),\beta PV_S)e(H_4(m,r,ID_S,PK_S),\beta P_{Pub})$ and checks if $\mathcal{S}_1 = \mathcal{S}_1^{'}$, then it outputs $valid$. Otherwise, if $\mathcal{S}_1 \neq \mathcal{S}_1^{'}$, $\mathcal{C}$ outputs $invalid$.

Confirmation/Disavowal query: $\mathcal{F}_I$ forms a tuple $(m,\sigma^{'},ID_S,PK_S = (PV_S,PS_S))$, where $\sigma^{'}$ is a signature on message m, PK_S is the public key of the signer with identity ID_S. $\mathcal{F}_I$ is allowed to request for the transcript of Confirmation/Disavowal protocol on any such tuple for a designated verifier with identity ID_V and public key $PK_V = (PV_V,PS_V)$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle and generates either the Confirmation or Disavowal proof transcript based on the output of the Verify oracle.

Simulating the non-interactive designated verifier proofs of the Confirmation and Disavowal protocols are quite easy, therefore, we do not provide the details here. However, $\mathcal{C}$ can fail in the proof simulation process if the value provided to random oracles H_7 or H_8 had been queried before, such case of collision will occur with a probability smaller than $q_{H_7}2^{-k}$ assuming $q_{H_7} \approx q_{H_8}$.

Selective-conv query: $\mathcal{F}_I$ is permitted to query for selective token on any tuple $(m,\sigma^{'},ID,PK_{ID})$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle to check the validity of the signature $\sigma^{'}$ and outputs the selective token $tk^{(ID,PK_{ID})}_{(m,\sigma)}$ on validity/invalidity of $(m,\sigma^{'},ID,PK_{ID})$. The process of generating the token is very similar to the Confirmation/Disavowal proof simulation process so we do not show it here.

Universal-conv query: $\mathcal{F}_I$ is permitted to query for universal token of any user (with identity ID and public key $PK_{ID} = (PV_{ID}, PS_{ID})$) in the system. Upon receiving such query, $\mathcal{C}$ scans κ_0 and κ_2 in order to find $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ and $(ID, PK_{ID},$ "$verify$"$, \beta)$ and outputs the universal token as $tk_*^{(ID,PK_{ID})} = (x_{ID}, \beta P_{Pub})$. However, if the public key of the user $PK_{ID} = (PV_{ID}, PS_{ID})$ had been changed prior to this query, $\mathcal{C}$ outputs the universal token as $tk_*^{(ID,PK_{ID})} = (-1, \beta P_{Pub})$, where -1 implies that the user secret value is not available due to prior public key replacement query on ID.

Finally, $\mathcal{F}_I$ outputs a tuple $(m^*, \sigma^*, ID^*, PK_{ID^*} = (PV_{ID^*}, PS_{ID^*}))$ where $\sigma^* = (r^*, \mathcal{S}_1^*, \mathcal{S}_2^*, \mathcal{S}_3^*)$ is a valid signature on message m^* for identity ID^* with public key PK_{ID^*}. $\mathcal{F}_I$ wins the game if ID^* was never queried to the partial private key extract oracle. Upon $\mathcal{F}_I$'s success, $\mathcal{C}$ scans κ_1 to find $(ID^*, PK_{ID^*}, \alpha^*, X)$; if $X = 0$, $\mathcal{C}$ reports *failure* and terminates. Otherwise, $\mathcal{C}$ knows that $e(P, \mathcal{S}_3^*) = e(P_{Pub}, Q_{k_{ID^*}})e(\mathcal{S}_2^*, H_5(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*, ID^*))e(PS_{ID^*}, H_6(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*, ID^*, PK_{ID^*})$, therefore, $\mathcal{C}$ recovers the values of λ_1^* and λ_2^* from κ_5 and κ_6 respectively and computes $e(P, \mathcal{S}_3^* - \lambda_1^*\mathcal{S}_2^* - \lambda_2^*PS_{ID^*}) = e(P_{Pub}, Q_{k_{ID^*}}) = e(P, DS_{ID^*})$. Eventually, $\mathcal{C}$ outputs $\alpha^{*^{-1}}(\mathcal{S}_3^* - \lambda_1^*\mathcal{S}_2^* - \lambda_2^*PS_{ID^*})$ as the solution of the random instance (P, aP, bP) of the CDH problem.

In order to compute the success probability of $\mathcal{C}$, we have to consider the cases that $\mathcal{C}$ may fail. $\mathcal{C}$ can fail in either the simulation process or in solving the CDH problem after $\mathcal{F}_I$ outputted the forgery signature. $\mathcal{C}$ will fail in the simulation process if $\mathcal{F}_I$ queries the partial private key of an identity ID (with public key PK_{ID}) where $H_1(ID, PK_{ID}) = \alpha(bP)$. $\mathcal{C}$ would also fail in computing the CDH problem after $\mathcal{F}_I$ outputted the successful forgery when $H_1(ID^*, PK_{ID^*}) = \alpha P$. Following Coron's (2000) technique the probability that $\mathcal{C}$ avoids all the failure cases is at least $\varphi_1^{q_E}(1 - \varphi_1)$, where q_E is the number of partial private key extract queries. Consequently, if $\mathcal{C}$ uses the optimal value $\varphi_{1,max} = {}^{q_E}/_{(q_E+1)}$, his success probability would be greater than ${}^1/_{e(q_E+1)}$. This is due to the fact that $({}^{q_E}/_{(q_E+1)})^{q_E}$ approaches ${}^1/_{\mathbf{e}}$ (where $\mathbf{e}$ is the base of natural logarithm) for large q_E. Moreover, it is possible that $\mathcal{F}_I$ never queried (ID^*, PK_{ID^*}) to H_1, this case may happen with probability less than ${}^1/_{2^k}$. As we mentioned above, $\mathcal{C}$ may also fail in simulation of the Confirmation and Disavowal protocols, this case will happen with probability less than $(q_{CD}+q_{H_7})/2^k$. Following the proof, $\mathcal{C}$'s advantage $\varepsilon_{\mathcal{C}}$ in solving a random instance (P, aP, bP) of the CDH problem is at least $\frac{\varepsilon_{\mathcal{F}_{\mathcal{I}}}-(q_{CD}+q_{H_7}+1)2^{-k}}{\mathbf{e}(q_E+1)}$. □

Theorem 5.2. *If there exists a Type I adversary $\mathcal{D}_I$ that can submit q_E partial private key extract and universal conversion queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, q_{CV} selective conversion queries, and q_{H_i} queries to random oracle H_i for $i \in \{1, \ldots, 8\}$ and be able to breach the invisibility property (win the game defined in Definition 5.2) of our proposed scheme with non-negligible success probability ε_{D_I}, then there exists a PPT algorithm $\mathcal{C}$ which can use $\mathcal{D}_I$ to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{D}_I} - (q_{CD} + q_{H_7})2^{-k}}{\mathbf{e}(q_E + 1)}$$

Proof. We assume that if there exists a Type I adversary $\mathcal{D}_I$ who has no knowledge over the master secret key s and is able to win the game defined in Definition 5.2, then we show that a PPT algorithm $\mathcal{C}$ can be built which uses $\mathcal{D}_I$ as its subroutine to solve a random instance (P, aP, bP, cP, h) of the DBDH problem.

$\mathcal{C}$ plays the role of $\mathcal{D}_I$'s challenger. It starts by running the Setup algorithm and provides $\mathcal{D}_I$ with the system public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, P_{Pub}, H_i$ where $i \in \{1, \ldots, 8\})$. Where $P_{Pub} = aP$ and a is unknown to $\mathcal{C}$. $\mathcal{D}_I$ is allowed to make queries to different oracles as defined in the game of Definition 5.2. $\mathcal{C}$ answers these queries by keeping lists κ_i for $i = \{1, \ldots, 8\}$ and a list κ_0 in order to keep track of the values of identities, public keys and the corresponding secret values. Without loss of generality, we assume $\mathcal{D}_I$ behaves well, i.e. $\mathcal{D}_I$ always makes a public key request before it queries on H_1 or H_2 oracles and always makes a H_1 and H_2 query before it requests for the partial private key of the user.

Query on $H_1(ID, PK_{ID})$: To answer a query on $H_1(ID, PK_{ID})$, $\mathcal{C}$ first checks if κ_1 already contained a tuple (ID, PK_{ID}, β), then $\mathcal{C}$ returns βP. Otherwise, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$, adds (ID, PK_{ID}, β) to κ_1 and returns βP to $\mathcal{D}_I$.

Query on $H_2(ID, PK_{ID}, “verify”)$: When $\mathcal{D}_I$ queries H_2 on an identity ID and public key PK_{ID}, $\mathcal{C}$ picks a random $\alpha \in \mathbb{Z}_q$ and flips a coin X that is truly random taking the value of 0 with probability φ_1 and the value of 1 with probability $1 - \varphi_1$ (the value of φ_1 will be computed later in our proof). $\mathcal{C}$ inserts $(ID, PK_{ID}, “verify”, \alpha, X)$ into κ_2 and returns $H_2(ID, PK_{ID}, “verify”) = \alpha(bP)$ if $X = 1$ and $H_2(ID, PK_{ID}, “verify”) = \alpha P$ if $X = 0$.

Query on $H_3(m, r, ID)$: In order to answer queries on H_3, $\mathcal{C}$ first checks if κ_3 already contained a tuple (m, r, ID, η), then it returns ηP to $\mathcal{D}_I$. Otherwise, $\mathcal{C}$ picks a random $\eta \in \mathbb{Z}_q$, adds (m, r, ID, η) to κ_3 and returns ηP to $\mathcal{D}_I$.

Query on $H_4(m, r, ID, PK_{ID})$: Upon receiving queries on $H_4(m, r, ID, PK_{ID})$, $\mathcal{C}$ first checks if κ_4 already contained a tuple $(m, r, ID, PK_{ID}, \gamma, Y)$ and $Y = 0$, it returns γP to $\mathcal{D}_I$. Otherwise, if $Y = 1$, $\mathcal{C}$ returns $\gamma(cP)$. On the other hand, if no such tuple exists in κ_4, $\mathcal{C}$ picks $\gamma \in \mathbb{Z}_q$ and inserts $(m, r, ID, PK_{ID}, \gamma, 0)$ into κ_4 and returns $H_4(m, r, ID, PK_{ID}) = \gamma P$.

Query on $H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID)$: To handle such queries on H_5, $\mathcal{C}$ first checks if κ_5 already contained a tuple $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$, returns $\lambda_1 P$ to $\mathcal{D}_I$. Otherwise, $\mathcal{C}$ picks a random $\lambda_1 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$ to κ_5 and returns $\lambda_1 P$ to $\mathcal{D}_I$.

Query on $H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID})$: In order to answer queries on H_6, $\mathcal{C}$ first checks if κ_6 already contained a tuple $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$, returns $\lambda_2 P$ to $\mathcal{D}_I$. Otherwise, $\mathcal{C}$ picks a random $\lambda_2 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$ to κ_6 and returns $\lambda_2 P$ to $\mathcal{D}_I$.

Query on H_7 and H_8: Queries on H_7 and H_8 oracles will be handled randomly, and the response will be stored in κ_7 and κ_8 respectively.

Public key request: Upon a public key query for identity ID, $\mathcal{C}$ first checks if $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ exists in κ_0, then it returns $PK_{ID} = (PV_{ID}, PS_{ID})$. Otherwise if no such tuple exists, $\mathcal{C}$ picks $x_{ID}, z_{ID} \in \mathbb{Z}_q$ randomly, computes $PV_{ID} = x_{ID}P$ and $PS_{ID} = z_{ID}P$

and returns $PK_{ID} = (PV_{ID}, PS_{ID})$ to $\mathcal{D}_I$. It also records $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ in κ_0.

Public key replacement: When $\mathcal{D}_I$ wishes to replace the public key $PK_{ID} = (PV_{ID}, PS_{ID})$ of identity ID with $PK_{ID}^{'} = (PV_{ID}^{'}, PS_{ID}^{'})$, $\mathcal{C}$ scans κ_0 to find $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$, if such tuple exists, it will replace it with $(ID, -1, -1, PV_{ID}^{'}, PS_{ID}^{'})$ where -1 means that the public key of the user has been replaced. Otherwise, $\mathcal{C}$ adds $(ID, -1, -1, PV_{ID}^{'}, PS_{ID}^{'})$ to κ_0. In this case if κ_1 and κ_2 contain tuples (ID, PK_{ID}, β) and (ID, PK_{ID}, α, X), $\mathcal{C}$ simulates H_1 and H_2 and updates κ_1 and κ_2 respectively.

Secret value extract: In order to respond to a secret key extract query on identity ID with public key $PK_{ID} = (PV_{ID}, PS_{ID})$, $\mathcal{C}$ scans κ_0 for a tuple $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ and returns the secret values of the user ID as pair (x_{ID}, z_{ID}). Where PK_{ID} is the original public key of the user.

Partial private key extract: Upon receiving an identity ID, $\mathcal{C}$ scans κ_2 for a tuple $(ID, PK_{ID}, \text{“}verify\text{”}, \alpha, X)$; if $X = 1$, $\mathcal{C}$ reports *failure* and terminates the simulation. Otherwise, it searches κ_1 to find (ID, PK_{ID}, β) and returns $(\alpha P_{Pub}, \beta P_{Pub})$ as the partial private keys of ID.

Sign query: $\mathcal{D}_I$ is allowed to query the Sign oracle in order to receive valid signatures on any tuple $(m, ID, PK_{ID} = (PV_{ID}, PS_{ID}))$, upon receiving such query, $\mathcal{C}$ works as follows.

1. Thereupon, $\mathcal{C}$ first picks a random string $r \in \{0,1\}^l$ and checks if κ_4 contains $(m, r, ID, PK_{ID}, \ldots)$, if yes, $\mathcal{C}$ will continue until it finds an admissible r which no tuple $(m, r, ID, PK_{ID}, \ldots)$ exists in κ_4. $\mathcal{C}$ then scans κ_3 to find (m, r, ID, η) and forms $\mathcal{S}_1 = e(\eta Q_{v_{ID}}, PV_{ID}) e(\gamma P_{Pub}, Q_{v_{ID}})$.
2. $\mathcal{C}$ picks a random $v \in \mathbb{Z}_q$ to compute $\mathcal{S}_2 = vP$.
3. Lastly, $\mathcal{C}$ scans κ_5 and κ_6 to compute $\mathcal{S}_3 = \beta P_{Pub} + \lambda_1 \mathcal{S}_2 + \lambda_2 PS_{ID}$ and outputs the signature as $\sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$.

Verify: $\mathcal{D}_I$ forms a tuple $(m,\sigma^{'},ID_S,PK_S=(PV_S,PS_S))$, where $\sigma^{'}$ is a signature on message m, and PK_S is the public key of the signer with identity ID_S. $\mathcal{D}_I$ is allowed to query for the validity of any such tuple. Upon receiving such query, $\mathcal{C}$ is able to reconstruct the signature on any tuple (m,ID_S,PK_S) for a random r (this is due to the random behaviour of H_4). $\mathcal{C}$ starts by parsing $\sigma^{'}$ into $(r^{'},\mathcal{S}_1^{'},\mathcal{S}_2^{'},\mathcal{S}_3^{'})$ and checking if $e(P,\mathcal{S}_3^{'}) = e(P_{Pub},Q_{k_S})e(\mathcal{S}_2^{'},H_5(\mathcal{S}_1^{'},\mathcal{S}_2^{'},m,ID_S))e(PS_S,H_6(\mathcal{S}_1^{'},\mathcal{S}_2^{'},m,ID_S,PK_S)$ hold, then $\mathcal{C}$ retrieves the value of γ from κ_4, forms $\mathcal{S}_1 = e(\eta Q_{v_S},PV_S)e(\gamma P_{Pub},Q_{v_S})$ and outputs $valid$ if $\mathcal{S}_1 = \mathcal{S}_1^{'}$. Otherwise, if $\mathcal{S}_1 \neq \mathcal{S}_1^{'}$, it returns $invalid$.

Confirmation/Disavowal query: $\mathcal{D}_I$ forms a tuple $(m,\sigma^{'},ID_S,PK_S=(PV_S,PS_S))$, where $\sigma^{'}$ is a signature on message m, PK_S is the public key of the signer with identity ID_S. $\mathcal{D}_I$ is allowed to request for the transcript of Confirmation/Disavowal protocol on any such tuple for a designated verifier with identity ID_V and public key $PK_V = (PV_V,PS_V)$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle and generates either the Confirmation or Disavowal proof transcript based on the output of the Verify oracle.

Simulating the non-interactive designated verifier proofs of the Confirmation and Disavowal protocols are quite easy, therefore, we do not provide the details here. However, $\mathcal{C}$ can fail in the proof simulation process if the value provided to random oracle H_7 or H_8 had been queried before, such case of collision will occur with a probability smaller than $q_{H_7}2^{-k}$ assuming that $q_{H_7} \approx q_{H_8}$.

Selective-conv query: $\mathcal{D}_I$ is permitted to query for selective token on any tuple $(m,\sigma^{'}, ID,PK_{ID})$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle to check the validity of the message-signature pair and outputs a selective token $tk_{(m,\sigma)}^{(ID,PK_{ID})}$ on validity/invalidity of $(m,\sigma^{'},ID,PK_{ID})$. The process of generating the token is very similar to the Confirmation/Disavowal proof simulation process so we do not show it here.

Universal-conv query: $\mathcal{D}_I$ is permitted to query for universal token of any user (with identity ID and public key $PK_{ID} = (PV_{ID},PS_{ID})$) in the system. Upon receiving such query, $\mathcal{C}$ scans κ_2 for the tuple $(ID, PK_{ID}, \text{“}verify\text{”}, \alpha, X)$, if $X = 1$, $\mathcal{C}$ reports *failure* and terminates the simulation. Otherwise, it scans κ_0 so as to find $(ID, x_{ID}, z_{ID}, PV_{ID}, PS_{ID})$ and outputs the universal token as $tk_*^{(ID,PK_{ID})} = (x_{ID}, \alpha P_{Pub})$. However, if the public key of the user $PK_{ID} = (PV_{ID}, PS_{ID})$ had been changed prior to this query, $\mathcal{C}$ outputs the universal token as $tk_*^{(ID,PK_{ID})} = (-1, \beta P_{Pub})$, where -1 implies that the user secret value is not available due to prior public key replacement query on ID.

After the first cycle of queries, $\mathcal{D}_I$ produces a message m^*, an identity ID^* and public key $PK_{ID^*} = (PV_{ID^*}, PS_{ID^*})$ where (ID^*, PK_{ID^*}) was never queried to partial private key, or universal-conv oracles. It then requests a signature on the challenge tuple (m^*, ID^*, PK_{ID^*}). In order to respond to $\mathcal{D}_I$'s request, $\mathcal{C}$ starts by scanning κ_2 to find $(ID^*, PK_{ID^*}, \text{“}verify\text{”}, \alpha, X)$, if $X = 0$, $\mathcal{C}$ reports *failure* and terminates. Otherwise, $\mathcal{C}$ picks a random string $r \in \{0,1\}^l$ and checks if κ_4 contains $(m^*, r, ID^*, PK_{ID^*}, \ldots)$, if yes, $\mathcal{C}$ will continue until it finds an admissible r which no tuple $(m^*, r, ID^*, PK_{ID^*}, \ldots)$ exists in κ_4. When such r is found, $\mathcal{C}$ defines the value of $H_4(m^*, r, ID^*, PK_{ID^*}) = \gamma(bP)$ and adds the tuple $(m^*, r, ID^*, PK_{ID^*}, \gamma, 1)$ in κ_4. $\mathcal{C}$ then computes the value of $\mathcal{S}_1^* = e(Q_{v_{ID^*}}, \eta PV_{ID^*})h^{\gamma\alpha}$ and forms the values of $\mathcal{S}_2^*$ and $\mathcal{S}_3^*$ identical to the Sign oracle and outputs the signature as $\sigma^* = (r, \mathcal{S}_1^*, \mathcal{S}_2^*, \mathcal{S}_3^*)$.

At the second cycle of queries, $\mathcal{D}_I$ can query different oracles similar to those in the first cycle. However, it is disallowed to request the following queries:

1. Partial private key, or universal conversion query of user ID^* with public key PK_{ID^*}.
2. Sign query on (m^*, ID^*, PK_{ID^*}).
3. Confirmation/disavowal or selective conversion query on $(m^*, \sigma^*, ID^*, PK_{ID^*})$.

At the end of the second cycle, $\mathcal{D}_I$ outputs a bit $b' \in \{0,1\}$. If $b' = 1$, it means that $\mathcal{D}_I$ considers the challenged signature σ^* to be valid, then $\mathcal{C}$ will also output 1 to indicate that $h = e(P,P)^{abc}$. Otherwise, $\mathcal{D}_I$ considers σ^* to be a random string and outputs $b' = 0$, consequently, $\mathcal{C}$ will also output 0 to indicate that $h \neq e(P,P)^{abc}$.

In order to compute the success probability of $\mathcal{C}$, we have to consider the cases that $\mathcal{C}$ may fail. $\mathcal{C}$ can fail in simulation process if it receives a partial private key extract, or universal conversion query where $H_2(ID, PK_{ID}, \text{“}verify\text{”}) = \alpha(bP)$. $\mathcal{C}$ also fails if the challenge identity is such that $H_2(ID^*, PK_{ID^*}, \text{“}verify\text{”}) = \alpha P$. The probability for $\mathcal{C}$ to avoid all failure states is $\varphi_1^{q_E}(1-\varphi_1)$, where q_E is the number of partial private key and universal conversion queries. Following Coron's method (2000) the optimal value of φ_1 is $\varphi_{1,max} = q_E/(q_E+1)$. Substituting the optimal probability $\varphi_{1,max}$, the probability that $\mathcal{C}$ does not fail is $\frac{1}{\mathbf{e}(q_E+1)}$. As mentioned above, $\mathcal{C}$ may also fail in simulation the Confirmation and Disavowal protocols with probability less than $(q_{CD}+q_{H_7})/2^k$. Following these analysis, we can easily see that $\mathcal{C}$'s advantage $\varepsilon_{\mathcal{C}}$ in solving the DBDH problem is at least $\frac{\varepsilon_{\mathcal{D}_I}-(q_{CD}+q_{H_7})2^{-k}}{\mathbf{e}(q_E+1)}$. □

Theorem 5.3. *If there exists a Type II adversary $\mathcal{F}_{II}$ which can submit q_E secret value extract and public key replacement queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, q_{CV} selective and universal conversion queries, and q_{H_i} queries to random oracle H_i for $i \in \{1,\ldots,8\}$ and be able to succeed in an existential forgery (win the game defined in Definition 5.4) against our proposed scheme with non-negligible success probability $\varepsilon_{\mathcal{F}_{II}}$, then there exists an algorithm $\mathcal{C}$ which can use $\mathcal{F}_{II}$ to solve a random instance (P, aP, bP) of the CDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{F}_{II}} - (q_{CD}+q_{H_7}+1)2^{-k}}{\mathbf{e}(q_E+1)}$$

Proof. We show that if there exists a Type II adversary $\mathcal{F}_{II}$ which can win the game defined in Definition 5.4, then one can construct a PPT algorithm $\mathcal{C}$ that makes use of $\mathcal{F}_{II}$ to solve the CDH problem with probability at least $\varepsilon_{\mathcal{C}}$.

$\mathcal{C}$ works as $\mathcal{F}_{II}$'s challenger and starts by running the Setup algorithm. It provides $\mathcal{F}_{II}$ with the master secret key s (we assume that $\mathcal{F}_{II}$ is a malicious KGC therefore, he has complete knowledge over the master secret key) along with the public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, H_i$ where $i \in \{1,\ldots,8\})$. Note that P_{Pub} is not included in $params$ since it could be easily computed by $\mathcal{F}_{II}$.

$\mathcal{F}_{II}$ is allowed to query different random oracles H_i for $i = \{1, \ldots, 8\}$ and other oracles (e.g. secret value extract, public key replacement, etc.) as defined in the game of Definition 5.4. $\mathcal{C}$ handles these queries by keeping lists κ_i for $i = \{1, \ldots, 8\}$ and a list κ_0 in order to keep track of the values of identities, public keys and the corresponding secret values. Without loss of generality, we assume $\mathcal{F}_{II}$ behaves well, i.e. $\mathcal{F}_{II}$ always makes a public key request before it queries on H_1 or H_2 oracles.

Query on $H_1(ID, PK_{ID})$**:** In order for $\mathcal{C}$ to answer such queries on an identity $ID \in \{0,1\}^*$ and public key PK_{ID}, it picks a random $\alpha \in \mathbb{Z}_q$ and inserts (ID, PK_{ID}, α) in κ_1 and returns $H_1(ID, PK_{ID}) = \alpha P$.

Query on $H_2(ID, PK_{ID}, \text{“}verify\text{”})$**:** In order to answer queries on H_2, $\mathcal{C}$ first checks if κ_2 already contained $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$, then $\mathcal{C}$ returns βP to $\mathcal{F}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$, adds $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$ to κ_2 and returns βP to $\mathcal{F}_{II}$.

Query on $H_3(m, r, ID)$**:** In order to answer queries on H_3, $\mathcal{C}$ first checks if κ_3 already contained (m, r, ID, γ), then it returns γP to $\mathcal{F}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\gamma \in \mathbb{Z}_q$, adds (m, r, ID, γ) to κ_3 and returns γP to $\mathcal{F}_{II}$.

Query on $H_4(m, r, ID, PK_{ID})$**:** In order to answer queries on H_4, $\mathcal{C}$ first checks if κ_4 already contained a tuple $(m, r, ID, PK_{ID}, \eta)$, then $\mathcal{C}$ returns ηP to $\mathcal{F}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\eta \in \mathbb{Z}_q$, adds $(m, r, ID, PK_{ID}, \eta)$ to κ_4 and returns ηP to $\mathcal{F}_{II}$.

Query on $H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID)$ **:** In order to answer queries on H_5, $\mathcal{C}$ first checks if κ_5 already contained $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$, then $\mathcal{C}$ returns $\lambda_1 P$ to $\mathcal{F}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\lambda_1 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$ to κ_5 and returns $\lambda_1 P$ to $\mathcal{F}_{II}$.

Query on $H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID})$**:** In order to answer queries on H_6, $\mathcal{C}$ picks a random

$\lambda_2 \in \mathbb{Z}_q$ and inserts $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$ in κ_6 and returns $H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}) = \lambda_2(bP)$.

Query on H_7 and H_8: Queries on H_7 and H_8 oracles will be handled randomly, and the response will be stored in κ_7 and κ_8 respectively.

Public key request: Upon receiving a public key request on identity ID, $\mathcal{C}$ first chooses $x_{ID} \in \mathbb{Z}_q$ and forms $PV_{ID} = x_{ID}P$. It then picks a random $\delta \in \mathbb{Z}_q$ and flips a coin X that is truly random taking the value of 0 with probability φ_1 and value of 1 with probability $1 - \varphi_1$. If $X = 1$, $\mathcal{C}$ sets $PS_{ID} = \delta(aP)$. Alternatively, if $X = 0$, $\mathcal{C}$ sets $PS_{ID} = \delta P$. Finally, $\mathcal{C}$ inserts $(ID, x_{ID}, \delta, PV_{ID}, PS_{ID}, X)$ into κ_0 and returns $PK_{ID} = (PV_{ID}, PS_{ID})$ to $\mathcal{F}_{II}$.

Public key replacement: $\mathcal{F}_{II}$ is able to replace the public key $PK_{ID} = (PV_{ID}, PS_{ID})$ of identity ID with $PK^{'}_{ID} = (PV^{'}_{ID}, PS^{'}_{ID})$. To respond to such queries, $\mathcal{C}$ checks κ_0 to find $(ID, x_{ID}, \delta, PV_{ID}, PS_{ID}, \ldots)$, if such tuple exists, it will replace it with $(ID, -1, -1, PV^{'}_{ID}$ $, PS^{'}_{ID}, \ldots)$ where -1 means that the public key of the user has been replaced. Otherwise, $\mathcal{C}$ adds a tuple $(ID, -1, -1, PV^{'}_{ID}, PS^{'}_{ID}, \ldots)$ to κ_0, in this case, if κ_1 and κ_2 contain tuples (ID, PK_{ID}, α) and $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$, $\mathcal{C}$ simulates H_1 and H_2 and updates κ_1 and κ_2 respectively.

Secret value extract: In order to respond to a secret key extract query on identity ID with public key $PK_{ID} = (PV_{ID}, PS_{ID})$, $\mathcal{C}$ scans κ_0 for a tuple $(ID, x_{ID}, \delta, PV_{ID}, PS_{ID}, X)$. If $X = 1$, $\mathcal{C}$ reports *failure* and terminates the simulation. Otherwise, $\mathcal{C}$ returns the secret values of the user ID as pair (x_{ID}, δ), where $PK_{ID} = (PV_{ID}, PS_{ID})$ is the original public key of the user.

Sign query: $\mathcal{F}_{II}$ is allowed to query the Sign oracle in order to receive valid signatures on any tuple $(m, ID, PK_{ID} = (PV_{ID}, PS_{ID}))$. In order to respond to such queries, $\mathcal{C}$ works as follows.

1. Picks $r \in \{0,1\}^l$ at random and queries $H_3(m,r,ID)$ and $H_4(m,r,ID,PK_{ID})$ in order to retrieve the values of $h_3 = \gamma P$ and $h_4 = \eta P$ respectively. $\mathcal{C}$ then computes the value of $\mathcal{S}_1 = e(h_3, \beta PV_{ID})e(h_4, \beta P_{Pub})$.
2. Next, $\mathcal{C}$ picks a random $v \in \mathbb{Z}_q$ and forms the value of $\mathcal{S}_2 = vPS_{ID}$. Next, it scans κ_5 in order to find a tuple $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \ldots)$ (if such tuple exists, $\mathcal{C}$ picks another v and forms the value of $\mathcal{S}_2 = vPS_{ID}$ until no such tuple $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \ldots)$ exists in κ_5) and sets $H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID) = v^{-1}(qP - H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}))$.
3. Lastly, $\mathcal{C}$ picks $q \in \mathbb{Z}_q$ at random, forms $\mathcal{S}_3 = qPS_{ID} + sH_1(ID, PK_{ID})$ and outputs the signature as $\sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$.

Verify: $\mathcal{F}_{II}$ forms a tuple $(m, \sigma', ID_S, PK_S = (PV_S, PS_S))$, where σ' is a signature on message m, and PK_S is the public key of the signer with identity ID_S. $\mathcal{F}_{II}$ is allowed to request for the validity of any such tuple. Upon receiving such query, $\mathcal{C}$ parses σ' into $(r, \mathcal{S}_1', \mathcal{S}_2', \mathcal{S}_3')$ and checks if $e(P, \mathcal{S}_3') = e(P_{Pub}, Q_{k_S})e(\mathcal{S}_2', H_5(\mathcal{S}_1', \mathcal{S}_2', m, ID_S))e(PS_S, H_6(\mathcal{S}_1', \mathcal{S}_2', m, ID_S, PK_S)$ hold, it computes $\mathcal{S}_1 = e(H_3(m,r,ID), \beta PV_S)e(H_4(m,r,ID,PK_S), \beta P_{Pub})$ and checks if $\mathcal{S}_1 = \mathcal{S}_1'$, it outputs $valid$. Otherwise, if $\mathcal{S}_1 \neq \mathcal{S}_1'$ it outputs $invalid$.

Confirmation/Disavowal query: $\mathcal{F}_{II}$ forms a tuple $(m, \sigma', ID_S, PK_S = (PV_S, PS_S))$, where σ' is a signature on message m, PK_S is the public key of the signer with identity ID_S. $\mathcal{F}_{II}$ is allowed to request for the transcript of Confirmation/Disavowal protocol on any such tuple for a designated verifier with identity ID_V and public key $PK_V = (PV_V, PS_V)$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle and generates either the Confirmation or Disavowal proof transcript based on the output of the Verify oracle.

Simulating the non-interactive designated verifier proofs of the Confirmation and Disavowal protocols are quite easy, therefore, we do not provide the details here. However, $\mathcal{C}$ can fail in the proof simulation process if the value provided to random oracle H_7 or H_8 had been queried before, such case of collision will occur with a probability smaller than $q_{H_7}2^{-k}$ assuming $q_{H_7} \approx q_{H_8}$.

Selective-conv query: $\mathcal{F}_{II}$ is permitted to query for selective token on any tuple $(m, \sigma', ID, PK_{ID})$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle to check the validity of the message-signature pair and outputs a selective token $tk_{(m,\sigma)}^{(ID,PK_{ID})}$ on validity/invalidity of $(m, \sigma', ID, PK_{ID})$. The process of generating the token is very similar to the Confirmation/Disavowal proof simulation process so we do not show it here.

Universal-conv query: $\mathcal{F}_{II}$ is permitted to query for universal token of any user (with identity ID and public key $PK_{ID} = (PV_{ID}, PS_{ID})$) in the system. Upon receiving such query, $\mathcal{C}$ scans κ_0 and κ_2 in order to find $(ID, x_{ID}, \delta, PV_{ID}, PS_{ID}, X)$ and $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$ and outputs the universal token as $tk_{*}^{(ID,PK_{ID})} = (x_{ID}, \beta P_{Pub})$. However, if the public key of the user $PK_{ID} = (PV_{ID}, PS_{ID})$ had been changed prior to this query, $\mathcal{C}$ outputs the universal token as $tk_{*}^{(ID,PK_{ID})} = (-1, \beta P_{Pub})$, where -1 implies that the user secret value is not available due to prior public key replacement query on ID.

Finally, $\mathcal{F}_{II}$ outputs a tuple $(m^*, \sigma^*, ID^*, PK_{ID^*} = (PV_{ID^*}, PS_{ID^*}))$ where $\sigma^* = (r^*, \mathcal{S}_1^*, \mathcal{S}_2^*, \mathcal{S}_3^*)$ is a valid signature on message m^* for identity ID^* with public key PK_{ID^*}. $\mathcal{F}_{II}$ wins the game if ID^* was never queried to secret value extract oracle, or public key replacement oracle. Upon $\mathcal{F}_{II}$'s success, $\mathcal{C}$ scans κ_0 to find the tuple $(ID^*, x_{ID^*}, \delta^*, PV_{ID^*}, PS_{ID^*}, X)$; if $X = 0$, $\mathcal{C}$ reports *failure* and terminates. Otherwise, $\mathcal{C}$ knows that $e(P, \mathcal{S}_3^*) = e(P_{Pub}, Q_{k_{ID^*}}) e(\mathcal{S}_2^*, H_5(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*, ID^*)) e(PS_{ID^*}, H_6(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*, ID^*, PK_{ID^*}))$, therefore, $\mathcal{C}$ recovers the values of λ_1^* and α^* from κ_5 and κ_1 respectively and computes $e(P, \mathcal{S}_3^* - \lambda_1^* \mathcal{S}_2^* - sH_1(ID^*, PK_{ID^*})) = e(PS_{ID^*}, H_6(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*, ID^*, PK_{ID^*}))$. Finally, $\mathcal{C}$ outputs $(\lambda_2^* \delta^*)^{-1}(\mathcal{S}_3^* - \lambda_1^* \mathcal{S}_2^* - sH_1(ID^*, PK_{ID^*}))$ as the solution of the random instance (P, aP, bP) of the CDH problem.

In order to compute the success probability of $\mathcal{C}$, we have to consider the cases that $\mathcal{C}$ may fail. $\mathcal{C}$ can fail in either the simulation process or in solving the CDH problem after $\mathcal{F}_{II}$ outputted the forgery signature. $\mathcal{C}$ may fail in the simulation process if $\mathcal{F}_{II}$ queries for the secret value of an identity ID where $PS_{ID} = \delta(aP)$. $\mathcal{C}$ can also fail in computing the CDH problem after $\mathcal{F}_{II}$ outputted a successful forgery when the public key of the user ID^* is set as $PS_{ID^*} = \delta P$. Therefore, the probability that $\mathcal{C}$ avoids all the failure cases is at least $\varphi_1^{q_E}(1-\varphi_1)$, where q_E is the number of secret value and public key replacement queries. Then, if $\mathcal{C}$ uses the optimal value $\varphi_{1,max} = q_E/(q_E+1)$, his success probability would be greater than $\frac{1}{e(q_E+1)}$. Moreover, it is possible that $\mathcal{F}_{II}$ never queried $H_6(\mathcal{S}_1^*, \mathcal{S}_2^*, m^*, ID^*, PK_{ID^*})$ for the forgery tuple, this case may happen with probability less that $1/2^k$. As we mentioned above, $\mathcal{C}$ may also fail in simulation of the Confirmation and Disavowal protocols, this case will happen with probability less than $(q_{CD}+q_{H_7})/2^k$. Following the proof, $\mathcal{C}$'s advantage $\varepsilon_{\mathcal{C}}$ in solving a random instance (P, aP, bP) of the CDH problem is at least $\frac{\varepsilon_{\mathcal{F}_{II}}-(q_{CD}+q_{H_7}+1)2^{-k}}{\mathbf{e}(q_E+1)}$. □

Theorem 5.4. *If there exists a Type II adversary $\mathcal{D}_{II}$ that can submit q_E secret value extract, public key replacement, and universal conversion queries, q_{US} sign queries, q_{CD} confirmation and disavowal queries, q_{CV} selective conversion queries, and q_{H_i} queries to random oracle H_i for $i \in \{1, \dots, 8\}$ and be able to breach the invisibility property (win the game defined in Definition 5.5) of our proposed scheme with non-negligible success probability $\varepsilon_{D_{II}}$, then there exists an PPT algorithm $\mathcal{C}$ which can use $\mathcal{D}_{II}$ to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability:*

$$\varepsilon_{\mathcal{C}} \geqslant \frac{\varepsilon_{\mathcal{D}_{II}} - (q_{CD} + q_{H_7})2^{-k}}{\mathbf{e}(q_E + 1)}$$

Proof. We show that if there exists a Type II adversary $\mathcal{D}_{II}$ which can win the game defined in Definition 5.5, then one can construct a PPT algorithm $\mathcal{C}$ that makes use of $\mathcal{D}_{II}$ to solve a random instance (P, aP, bP, cP, h) of the DBDH problem with probability at least $\varepsilon_{\mathcal{C}}$. $\mathcal{C}$ works as $\mathcal{D}_{II}$'s challenger and starts by running the Setup algorithm. It provides $\mathcal{D}_{II}$ with the master secret key s (we assume that $\mathcal{D}_{II}$ is a malicious KGC therefore, he has complete knowledge over the master secret key) along with the public parameters $params = (q, \mathbb{G}_1, \mathbb{G}_2, P, H_i$ where $i \in \{1, \dots, 8\})$.

$\mathcal{D}_{II}$ is allowed to query different random oracles H_i for $i = \{1, \dots, 8\}$ and other oracles (e.g. secret value extract, public key replacement, etc.) as defined in the game of Definition 5.5. $\mathcal{C}$ handles these queries by keeping lists κ_i for $i = \{1, \dots, 8\}$ and a list

κ_0 in order to keep track of the values of identities, public keys and the corresponding secret values. Without loss of generality, we assume $\mathcal{D}_{II}$ behaves well, i.e. $\mathcal{D}_{II}$ always makes a public key request before it queries on H_1 or H_2 oracles.

Query on $H_1(ID, PK_{ID})$**:** In order to answer queries on H_1, $\mathcal{C}$ first checks if κ_1 already contained (ID, PK_{ID}, β), then it returns βP. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\beta \in \mathbb{Z}_q$, adds (ID, PK_{ID}, β) to κ_1 and returns βP to $\mathcal{D}_{II}$.

Query on $H_2(ID, PK_{ID}, \text{“}verify\text{”})$**:** In order for $\mathcal{C}$ to answer H_2 queries on an identity ID and public key PK_{ID}, it picks a random $\alpha \in \mathbb{Z}_q$, inserts $(ID, PK_{ID}, \text{“}verify\text{”}, \alpha)$ into κ_2 and returns $H_1(ID, PK_{ID}, \text{“}verify\text{”}) = \alpha(bP)$.

Query on $H_3(m, r, ID)$**:** In order to answer queries on H_3, $\mathcal{C}$ first checks if κ_3 already contained a tuple (m, r, ID, γ, Y) and $Y = 0$, $\mathcal{C}$ returns γP to $\mathcal{D}_{II}$. Otherwise, if $Y = 1$, $\mathcal{C}$ returns $\gamma(cP)$. On the other hand, if no such tuple exists in κ_3, $\mathcal{C}$ picks $\gamma \in \mathbb{Z}_q$, inserts $(m, r, ID, \gamma, 0)$ into κ_3 and returns $H_3(m, r, ID) = \gamma P$.

Query on $H_4(m, r, ID, PK_{ID})$**:** In order to answer queries on H_4, $\mathcal{C}$ first checks if κ_4 already contained $(m, r, ID, PK_{ID}, \eta)$, it returns ηP to $\mathcal{D}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\eta \in \mathbb{Z}_q$, adds $(m, r, ID, PK_{ID}, \eta)$ to κ_4 and returns ηP to $\mathcal{D}_{II}$.

Query on $H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID)$**:** In order to answer queries on H_5, $\mathcal{C}$ first checks if κ_5 already contained $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$, it returns $\lambda_1 P$ to $\mathcal{D}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\lambda_1 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, \lambda_1)$ to κ_5 and returns $\lambda_1 P$ to $\mathcal{D}_{II}$.

Query on $H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID})$**:** In order to answer queries on H_6, $\mathcal{C}$ first checks if κ_6 already contained $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$, it returns $\lambda_2 P$ to $\mathcal{D}_{II}$. Otherwise, if no such tuple exists, $\mathcal{C}$ picks a random $\lambda_2 \in \mathbb{Z}_q$, adds $(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID}, \lambda_2)$ to κ_6 and returns $\lambda_2 P$ to $\mathcal{D}_{II}$.

Query on H_7 **and** H_8**:** Queries on H_7 and H_8 oracles will be handled randomly, and the response will be stored in κ_7 and κ_8 respectively.

Public key request: Upon a public key request on an identity ID, $\mathcal{C}$ first chooses $z_{ID} \in \mathbb{Z}_q$ and forms $PS_{ID} = z_{ID}P$. Then, it picks a random $\delta \in \mathbb{Z}_q$ and flips a coin X that is truly random taking the value of 0 with probability φ_1 and the value of 1 with probability $1-\varphi_1$. If $X = 1$, $\mathcal{C}$ sets $PV_{ID} = \delta(aP)$ and inserts $(ID, \delta, z_{ID}, PV_{ID}, PS_{ID}, X)$ into κ_0. Otherwise, if $X = 0$, $\mathcal{C}$ sets $PV_{ID} = \delta P$ and inserts $(ID, \delta, z_{ID}, PV_{ID}, PS_{ID}, X)$ into κ_0.

Public key replacement: When $\mathcal{D}_{II}$ wishes to replace the public key $PK_{ID} = (PV_{ID}, PS_{ID})$ of identity ID with $PK'_{ID} = (PV'_{ID}, PS'_{ID})$, $\mathcal{C}$ checks κ_0 to find $(ID, \delta, z_{ID}, PV_{ID}, PS_{ID}, \ldots)$, if such tuple exists, it will replace it with $(ID, -1, -1, PV'_{ID}, PS'_{ID}, \ldots)$ where -1 means that the public keys has been replaced. Otherwise, $\mathcal{C}$ adds a tuple $(ID, -1, -1, PV'_{ID}, PS'_{ID}, \ldots)$ to κ_0. In this case, if κ_1 and κ_2 contain (ID, PK_{ID}, α) and $(ID, PK_{ID}, \text{“}verify\text{”}, \beta)$, $\mathcal{C}$ simulates H_1 and H_2 and updates κ_1 and κ_2.

Secret value extract: In order to respond to a secret key extract query on identity ID, $\mathcal{C}$ scans κ_0 for a tuple $(ID, x_{ID}, \delta, PV_{ID}, PS_{ID}, X)$. If $X = 1$, $\mathcal{C}$ reports *failure* and terminates the simulation. Otherwise, it returns the secret values of the user ID as pair a (x_{ID}, δ), where $PK_{ID} = (PV_{ID}, PS_{ID})$ is the original public key of the user.

Sign query: $\mathcal{D}_{II}$ is allowed to query the Sign oracle in order to receive valid signatures on any tuple $(m, ID, PK_{ID} = (PV_{ID}, PS_{ID}))$, upon receiving such query $\mathcal{C}$ works as follows.

1. $\mathcal{C}$ first picks a random string $r \in \{0,1\}^l$ and checks if κ_3 contains a tuple $(m, r, ID, \ldots)$, if yes, $\mathcal{C}$ will continue until it finds an admissible r which no tuple $(m, r, ID, \ldots)$ exists in κ_3. $\mathcal{C}$ then scans κ_4 to find the tuple $(m, r, ID, PK_{ID}, \eta)$ and forms $\mathcal{S}_1 = e(\eta Q_{v_{ID}}, PV_{ID})e(\gamma P_{Pub}, Q_{v_{ID}})$.
2. $\mathcal{C}$ picks a random $v \in \mathbb{Z}_q$ and forms the value of $\mathcal{S}_2 = vP$.
3. Lastly, $\mathcal{C}$ simulates $H_5(\mathcal{S}_1, \mathcal{S}_2, m, ID)$ and $H_6(\mathcal{S}_1, \mathcal{S}_2, m, ID, PK_{ID})$ and computes

the values of $\mathcal{S}_3 = \beta P_{Pub} + \lambda_1 \mathcal{S}_2 + \lambda_2 PS_{ID}$ and outputs the signature as $\sigma = (r, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$.

Verify: $\mathcal{D}_{II}$ forms a tuple $(m, \sigma^{'}, ID_S, PK_S = (PV_S, PS_S))$, where $\sigma^{'}$ is a signature on message m, and PK_S is the public key of the signer with identity ID_S. $\mathcal{D}_{II}$ is allowed to request for the transcript of Confirmation/Disavowal protocol on any such tuple. Upon receiving such query, $\mathcal{C}$ is able to reconstruct the signature on the tuple (m, ID_S, PK_S) with the random r (this is due to the random behaviour of H_3). $\mathcal{C}$ starts by parsing $\sigma^{'}$ into $(r^{'}, \mathcal{S}_1^{'}, \mathcal{S}_2^{'}, \mathcal{S}_3^{'})$ and checks if $e(P, \mathcal{S}_3^{'}) = e(P_{Pub}, Q_{k_S}) e(\mathcal{S}_2^{'}, H_5(\mathcal{S}_1^{'}, \mathcal{S}_2^{'}, m, ID_S)) e(PS_S,$ $H_6(\mathcal{S}_1^{'}, \mathcal{S}_2^{'}, m, ID_S, PK_S)$ holds. $\mathcal{C}$ retrieves the value of γ from κ_3 and forms $\mathcal{S}_1 = e(\gamma Q_{v_{ID_S}},$
$PV_{ID_S}) e(\eta P_{Pub}, Q_{v_{ID_S}})$. Then, if $\mathcal{S}_1 = \mathcal{S}_1^{'}$, $\mathcal{C}$ outputs *valid*. Otherwise, if $\mathcal{S}_1 \neq \mathcal{S}_1^{'}$ $\mathcal{S}_1 = \mathcal{S}_1^{'}$, $\mathcal{C}$ outputs *invalid*.

Confirmation/Disavowal query: $\mathcal{D}_{II}$ forms a tuple $(m, \sigma^{'}, ID_S, PK_S = (PV_S, PS_S))$, where $\sigma^{'}$ is a signature on message m, PK_S is the public key of the signer with identity ID_S. $\mathcal{D}_{II}$ is allowed to request for the transcript of Confirmation/Disavowal protocol on any such tuple for a designated verifier with identity ID_V and public key $PK_V = (PV_V, PS_V)$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle and generates either the Confirmation or Disavowal proof transcript based on the output of the Verify oracle.

Simulating the non-interactive designated verifier proofs of the Confirmation and Disavowal protocols are quite easy, therefore, we do not provide the details here. However, $\mathcal{C}$ would fail in the proof simulation process if the value provided to random oracle H_7 or H_8 had been queried before, such case of collision will occur with a probability smaller than $q_{H_7} 2^{-k}$ assuming that $q_{H_7} \approx q_{H_8}$.

Selective-conv query: $\mathcal{D}_{II}$ is permitted to query for selective token on any tuple $(m, \sigma^{'}, ID, PK_{ID})$. Upon receiving such query, $\mathcal{C}$ simulates the Verify oracle to check the validity of the message-signature pair and outputs a selective token $tk_{(m,\sigma)}^{(ID,PK_{ID})}$ on validity/invalidity of $(m,\sigma^{'},ID,PK_{ID})$. The process of generating the token is very similar to the Confirmation/Disavowal proof simulation process so we do not show it here.

Universal-conv query: D_{II} is permitted to query for universal token of any user ID (with public key PK_{ID}) in the system. Upon receiving such query, $\mathcal{C}$ simulates the partial private key extract on ID, if the query is successful, it scans κ_2 so as to find $(ID,PK_{ID},\text{``}verify\text{''},\alpha)$ and outputs the universal token as $tk_*^{(ID,PK_{ID})} = (x_{ID},\alpha P_{Pub})$. However, if the public of the user $PK_{ID} = (PV_{ID},PS_{ID})$ had been changed prior to this query, $\mathcal{C}$ outputs the universal token as $tk_*^{(ID,PK_{ID})} = (-1,\beta P_{Pub})$, where -1 implies that the user secret value is not available due to prior public key replacement query on ID.

After the first cycle of queries, $\mathcal{D}_{II}$ produces a message m^*, an identity ID^* and public key $PK_{ID^*} = (PV_{ID^*},PS_{ID^*})$ where no secret value extract query, public key replacement query or universal conversion query was made on the identity ID^* with public key PK_{ID^*}. It then requests a signature on the challenge tuple (m^*,ID^*,PK_{ID^*}). In order to respond to $\mathcal{D}_{II}$'s request, $\mathcal{C}$ starts by scanning κ_0 to find a tuple $(ID,\delta,z_{ID^*}$, $PV_{ID^*},PS_{ID^*},X)$. If $X=0$, $\mathcal{C}$ reports *failure* and terminates. Otherwise, $\mathcal{C}$ picks a random string $r \in \{0,1\}^l$ and checks if κ_3 contains a tuple $(m^*,r,ID^*,\ldots)$, if yes, $\mathcal{C}$ will continue until it finds an admissible r which no tuple $(m^*,r,ID^*,\ldots)$ exists in κ_3. When such r is found, $\mathcal{C}$ defines the value of $H_3(m^*,r,ID^*) = \gamma(cP)$ and adds $(m^*,r,ID^*,\gamma,1)$ into κ_3. $\mathcal{C}$ then computes the value of $\mathcal{S}_1^* = h^{\gamma\alpha}e(Q_{v_{ID^*}},\eta P_{Pub})$ and forms the values of $\mathcal{S}_2^*$ and $\mathcal{S}_3^*$ identical to the Sign oracle and outputs the signature as $\sigma^* = (r,\mathcal{S}_1^*,\mathcal{S}_2^*,\mathcal{S}_3^*)$.

At the second cycle of queries, $\mathcal{D}_{II}$ queries the above oracles similar to the first cycle. However, it is disallowed to request the following:

1. Secret value extract, public key replacement, or universal conversion query for ID^*.
2. Sign query on (m^*,ID^*,PK_{ID^*}).

3. Confirmation/disavowal or selective conversion query on (m^*, σ^*, ID^*).

At the end of the second cycle, $\mathcal{D}_{II}$ outputs a bit $b^{'}$. If $b^{'} = 1$, means that $\mathcal{D}_{II}$ considers the challenge signature σ^* to be valid, then $\mathcal{C}$ will also output 1 to indicate that $h = e(P,P)^{abc}$. Otherwise, $\mathcal{D}_{II}$ considers σ^* to be a random string and outputs $b^{'} = 0$, consequently, $\mathcal{C}$ will also outputs 0 to indicate that $h \neq e(P,P)^{abc}$.

In order to compute the success probability of $\mathcal{C}$, we have to consider the cases that $\mathcal{C}$ may terminate unexpectedly. $\mathcal{C}$ can fail in simulation process if it receives a secret value extract query where $PV_{ID} = \delta(aP)$. $\mathcal{C}$ will also fail if the challenge identity is such that $PV_{ID^*} = \delta P$. The probability for $\mathcal{C}$ to avoid all failure states is $\varphi_1^{q_E}(1-\varphi_1)$, where q_E is the number of secret value extract and public key replacement queries. Following Coron's (2000) method, the optimal value of φ_1 is $\varphi_{1,max} = {}^{q_E}/_{(q_E+1)}$. Substituting the optimal probability$\varphi_{1,max}$, the probability that $\mathcal{C}$ does not fail is $\frac{1}{\mathbf{e}(q_E+1)}$. As mentioned above, $\mathcal{C}$ may also fail in simulation the Confirmation/Disavowal transcript with probability less than ${}^{(q_{CD}+q_{H_7})}/_{2^k}$. Following these analyse, we can see that $\mathcal{C}$'s advantage $\varepsilon_{\mathcal{C}}$ in solving the DBDH problem is at least $\frac{\varepsilon_{\mathcal{D}_{II}}-(q_{CD}+q_{H_7})2^{-k}}{\mathbf{e}(q_E+1)}$. □

5.4.2 Efficiency and Extensions

Efficiency: Since most of our signature components (2 out of 3) are points in $\mathbb{G}_1$, we can use the standard point compression techniques to reduce the signature size. Moreover, the value of $e(P_{Pub}, Q_{k_S})$ does not depend on the message and is fixed when verifying signatures (in the Confirmation/Disavowal, Selective-vfy and Universal-vfy protocol) for a particular signer and therefore, can be precomputed. We also point that by using the same approach which was proposed by Katz and Wang in (2003), we can simply replace the l-bit length r with a single bit while achieving the same security. This will result in a shorter signature by the factor of $l-1$.

Verification delegation: In some situations, it may be favourable to enable a semi-trusted third party to generate proofs of the Confirmation and Disavowal protocols for designated verifiers on behalf of the signer. In our scheme, the secret input of the signer to the Confirmation and Disavowal protocols is her verifying key pair (x_{ID}, DV_{ID}). Therefore, the signer is able to pass the verifying key pair (as the universal token) to a semi-trusted third party in order to enable him to generate non-interactive designated verifier proofs on the validity/invalidity of her signatures. Note that the proofs generated by the semi-trusted third party and the signer are indistinguishable.

This feature is similar to the notion of designated confirmer signatures which was introduced by Chaum (1995). Designated confirmer signature scheme is an extension of undeniable signature schemes, the signer has the ability to generate signatures in a manner that a designated third party (i.e. designated confirmer) is able to engage in the Confirmation/Disavowal protocols (without the help of the signer) to prove the validity/invalidity of a signature to the verifier.

Using the same method, the signer is enabled to designate a third party in order to selectively convert her undeniable signatures to universally verifiable ones. This is due to the fact that the secret input of the signer to Selective-conv protocol is her verifying key pair (x_{ID}, DV_{ID}), which could be passed to a semi-trusted third party as the signer's universal token without enabling him to generate signatures on behalf of the signer.

5.5 Summary

In this chapter, we formalised the security model of convertible undeniable signature schemes in a certificateless setting for the first time and presented the first convertible certificateless undeniable signature scheme to the literature. We also highlighted some interesting features that our scheme can provide for the signer by enabling him to delegate a semi-trusted third party to initiate the Confirmation or Disavowal protocol or even issue selective tokens on her behalf. Lastly, we proved the security of our scheme by relying its unforgeability, invisibility and anonymity against Type I/II adversary based on the hardness of some well-known mathematical problems in the random oracle model.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The primary focus of this research was to study certificate-free undeniable signature schemes. More specifically, our main objectives were to study and analyse the security of the existing schemes and design schemes which are either more efficient, more secure, or provide more added features.

We undertook a stringent analysis on the security of the existing schemes in the literature to identity the flaws in their structure which could lead to possible attacks on their security. The accurate analysis resulted in attacks on the two efficient certificate-free undeniable signature schemes. In our first attempt, we mounted two attacks on Chow's (2005) identity-based undeniable signature scheme, targeting its unforgeability and non-transferability. In our second attempt, we explored two breaches in the structure of the efficient certificateless scheme of Zhao and Ye (2012) which led to two attacks on the invisibility and non-impersonation of their scheme. We then put forth a revised scheme which has the same efficient Sign algorithm as Zhao and Ye's scheme and is secure against both of the aforementioned attacks.

In our endeavour to propose new schemes, we proposed a provably secure efficient identity-based undeniable signature scheme. Unlike the existing provably secure identity-based undeniable signature schemes (Libert & Quisquater, 2004; Wu et al., 2008), the new scheme does not need any pairing evaluations in its Sign algorithm and has shorter signature length. This is a substantial improvement since the efficiency of the new scheme enables its implementation on mobile devices with lower computation and communication capability.

Duan's (2008) scheme was the only certificateless undeniable signature scheme which is secure in the strong security model. We proffered a new certificateless undeniable signature scheme which is more efficient than Duan's scheme both for the signer and the verifier. Considering the cost of pairing evaluations, the fewer number of such expensive computations in the Sign algorithm and proof generation/verification algorithm of our proposed scheme, makes our scheme considerably cheaper to implement. In addition, a formal security proof was provided to relate the security of the new scheme to the hardness of some hard mathematical problems.

Our last contribution was to introduce the first convertible certificateless undeniable signature scheme. We formalised the security of convertible undeniable signature schemes in a certificateless setting for the first time and put forth the first instantiation of such schemes in the literature. We also proved that our scheme is secure in the random oracle model, given the CDH and the DBDH problems are intractable.

6.2 Future Work

The future work on certificate-free undeniable signature schemes can be on developing pairing-free schemes which could be implemented on mobile devices with low computational power. We have proposed three new certificate-free undeniable signature schemes. Therefore, our future focus would be to implement these schemes on mobile devices and strive to improve their efficiency particularly in the Confirmation and Disavowal protocols.

Certainly, more research needs to be done to develop certificate-free undeniable signature schemes with additional features. There are many interesting variations of undeniable signature schemes such as time-selective convertible undeniable signatures (Laguillaumie & Vergnaud, 2005), convertible undeniable partially blind signatures (Koide, Tso, & Okamoto, 2008), etc., which are only available in traditional public key setting and are not yet available in either identity-based or certificateless settings.

All of the certificate-free undeniable signature schemes that are proposed to this day are only secure in the random oracle model. Nonetheless, random oracles enable the construction of efficient schemes, but, it is almost impossible to instantiate such oracles in the real world. While devising certificate-free undeniable signature schemes in the standard model provides more security assurance, it greatly minimises the efficiency of the scheme. Therefore, proposing efficient certificate-free undeniable signature schemes in the standard model could also be a possible direction of the future research.

REFERENCES

[1] Al-Riyami, S., & Paterson, K. (2003). Certificateless public key cryptography. In C.-S. Laih (Ed.), *Lecture Notes in Computer Science: Vol. 2894. Advances in Cryptology - ASIACRYPT* (pp. 452–473). Springer-Verlag.

[2] Au, M. H., Mu, Y., Chen, J., Wong, D. S., Liu, J. K., & Yang, G. (2007). Malicious KGC attacks in certificateless cryptography. In *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security* (pp. 302–311). ACM.

[3] Bellare, M., Boldyreva, A., & Palacio, A. (2004). An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In C. Cachin & J. Camenisch (Eds.), *Lecture Notes in Computer Science: Vol. 3027. Advances in Cryptology - EUROCRYPT* (pp. 171–188). Springer-Verlag.

[4] Bellare, M., & Rogaway, P. (1993). Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (pp. 62–73). ACM.

[5] Boneh, D., & Franklin, M. (2001). Identity-based encryption from the Weil pairing. In J. Kilian (Ed.), *Lecture Notes in Computer Science: Vol. 2139. Advances in Cryptology - CRYPTO* (pp. 213–229). Springer-Verlag.

[6] Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the Weil pairing. In C. Boyd (Ed.), *Lecture Notes in Computer Science: Vol. 2248. Advances in Cryptology - ASIACRYPT* (pp. 514–532). Springer-Verlag.

[7] Boyar, J., Chaum, D., Damgård, I., & Pedersen, T. (1991). Convertible undeniable signatures. In A. Menezes & S. Vanstone (Eds.), *Lecture Notes in Computer Science: Vol. 537. Advances in Cryptology - CRYPTO* (pp. 189–205). Springer-Verlag.

[8] Boyd, C., & Foo, E. (1998). Off-line fair payment protocols using convertible signatures. In K. Ohta & D. Pei (Eds.), *Lecture Notes in Computer Science: Vol. 1514. Advances in Cryptology - ASIACRYPT* (pp. 271–285). Springer-Verlag.

[9] Camenisch, J., & Shoup, V. (2003). Practical verifiable encryption and decryption of discrete logarithms. In D. Boneh (Ed.), *Lecture Notes in Computer Science: Vol. 2729. Advances in Cryptology - CRYPTO* (pp. 126–144). Springer-Verlag.

[10] Canetti, R. (1997). Towards realizing random oracles: Hash functions that hide all partial informa-

tion. In B. Kaliski (Ed.), *Lecture Notes in Computer Science: Vol. 1294. Advances in Cryptology - CRYPTO* (pp. 455–469). Springer-Verlag.

[11] Canetti, R., Goldreich, O., & Halevi, S. (2004). The random oracle methodology, revisited. *J. ACM*, *51*(4), 557–594.

[12] Chabanne, H., Phan, D., & Pointcheval, D. (2005). Public traceability in traitor tracing schemes. In R. Cramer (Ed.), *Lecture Notes in Computer Science: Vol. 3494. Advances in Cryptology - EUROCRYPT* (pp. 542–558). Springer-Verlag.

[13] Chaum, D. (1991). Zero-knowledge undeniable signatures. In I. Damgård (Ed.), *Lecture Notes in Computer Science: Vol. 473. Advances in Cryptology - EUROCRYPT* (pp. 458–464). Springer-Verlag.

[14] Chaum, D. (1995). Designated confirmer signatures. In A. De Santis (Ed.), *Lecture Notes in Computer Science: Vol. 950. Advances in Cryptology - EUROCRYPT* (pp. 86–91). Springer-Verlag.

[15] Chaum, D., & van Antwerpen, H. (1989). Undeniable signatures. In G. Brassard (Ed.), *Lecture Notes in Computer Science: Vol. 435. Advances in Cryptology - CRYPTO* (pp. 212–216). Springer-Verlag.

[16] Chaum, D., van Heijst, E., & Pfitzmann, B. (1992). Cryptographically strong undeniable signatures, unconditionally secure for the signer. In J. Feigenbaum (Ed.), *Lecture Notes in Computer Science: Vol. 576. Advances in Cryptology - CRYPTO* (pp. 470–484). Springer-Verlag.

[17] Cheon, J. H., & Lee, D. H. (2002). *Diffie-Hellman problems and bilinear maps.* Cryptology ePrint Archive, Report 2002/117 [Online]. Available: http://eprint.iacr.org/2002/117.

[18] Choi, K. Y., Park, J. H., & Lee, D. H. (2011). A new provably secure certificateless short signature scheme. *Computers & Mathematics with Applications*, *61*(7), 1760–1768.

[19] Chow, S. (2005). Verifiable pairing and its applications. In C. Lim & M. Yung (Eds.), *Lecture Notes in Computer Science: Vol. 3325. Information Security Applications* (pp. 170–187). Springer-Verlag.

[20] Coron, J.-S. (2000). On the exact security of full domain hash. In M. Bellare (Ed.), *Lecture Notes in Computer Science: Vol. 1880. Advances in Cryptology - CRYPTO* (pp. 229–235). Springer-Verlag.

[21] Damgård, I. (1992). Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum (Ed.), *Lecture Notes in Computer Science: Vol. 576. Advances in Cryptology - CRYPTO* (pp. 445–456). Springer-Verlag.

[22] Desmedt, Y., Goutier, C., & Bengio, S. (1987). Special uses and abuses of the fiat-shamir passport protocol. In C. Pomerance (Ed.), *Lecture Notes in Computer Science: Vol. 293. Advances in Cryptology - CRYPTO* (pp. 21–39). Springer-Verlag.

[23] Desmedt, Y., & Yung, M. (1991). Weaknesses of undeniable signature schemes. In D. Davies (Ed.), *Lecture Notes in Computer Science: Vol. 547. Advances in Cryptology - EUROCRYPT* (pp. 205–220). Springer-Verlag.

[24] Diffie, W., & Hellman, M. E. (1976a). Multiuser cryptographic techniques. In *Proceedings of the National Computer Conference and Exposition* (pp. 109–112). ACM.

[25] Diffie, W., & Hellman, M. E. (1976b). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*, 644–654.

[26] Duan, S. (2008). Certificateless undeniable signature scheme. *Information Sciences*, *178*(3), 742–755.

[27] Galbraith, S. D., & Mao, W. (2003). Invisibility and anonymity of undeniable and confirmer signatures. In *Lecture Notes in Computer Science: Vol. 2612. Topics in Cryptology - CT-RSA* (pp. 80–97). Springer-Verlag.

[28] Gao, C., Yao, Z.-a., Xie, D., & Wei, B. (2011). Electronic Sealed-Bid Auctions with Incoercibility. In X. Wan (Ed.), *Lecture Notes in Electrical Engineering: Vol. 99. Electrical Power Systems and Computers* (pp. 47–54). Springer-Verlag.

[29] Girault, M. (1991). Self-certified public keys. In *Lecture Notes in Computer Science: Vol. 547. Advances in Cryptology - EUROCRYPT* (pp. 490–497). Springer-Verlag.

[30] Goh, E.-J., & Jarecki, S. (2003). A signature scheme as secure as the Diffie-Hellman problem. In E. Biham (Ed.), *Lecture Notes in Computer Science: Vol. 2656. Advances in Cryptology - EUROCRYPT* (pp. 647–647). Springer-Verlag.

[31] Goldwasser, S., & Kalai, Y. T. (2003). On the (in)security of the Fiat-Shamir paradigm. In *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science* (pp. 102–113).

[32] Goldwasser, S., & Micali, S. (1984). Probabilistic encryption. *Journal of Computer and System Sciences*, *28*(2), 270–299.

[33] Goldwasser, S., Micali, S., & Rivest, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, *17*, 281–308.

[34] Hada, S., & Tanaka, T. (1998). On the existence of 3-round zero-knowledge protocols. In H. Krawczyk (Ed.), *Lecture Notes in Computer Science: Vol. 1462. Advances in Cryptology - CRYPTO* (pp. 408–423). Springer-Verlag.

[35] Han, S., Yeung, W. K., & Wang, J. (2003). Identity-based confirmer signatures from pairings over elliptic curves. In *Proceedings of the 4th ACM Conference on Electronic Commerce* (pp. 262–263). ACM.

[36] Hess, F. (2003). Efficient identity based signature schemes based on pairings. In K. Nyberg & H. Heys (Eds.), *Lecture Notes in Computer Science: Vol. 2595. Selected Areas in Cryptography* (pp. 310–324). Springer-Verlag.

[37] Horwitz, J., & Lynn, B. (2002). Toward hierarchical identity-based encryption. In L. Knudsen (Ed.), *Lecture Notes in Computer Science: Vol. 2332. Advances in Cryptology - EUROCRYPT* (pp. 466–481). Springer-Verlag.

[38] Hu, B., Wong, D., Zhang, Z., & Deng, X. (2007). Certificateless signature: a new security model and an improved generic construction. *Designs, Codes and Cryptography*, *42*, 109–126.

[39] Huang, X., Mu, Y., Susilo, W., Wong, D. S., & Wu, W. (2007). Certificateless signature revisited. In J. Pieprzyk, H. Ghodosi, & E. Dawson (Eds.), *Lecture Notes in Computer Science: Vol. 4586. Information Security and Privacy* (pp. 308–322). Springer-Verlag.

[40] Huang, X., Mu, Y., Susilo, W., & Wu, W. (2007). Provably secure pairing-based convertible undeniable signature with short signature length. In T. Takagi, T. Okamoto, E. Okamoto, & T. Okamoto (Eds.), *Lecture Notes in Computer Science: Vol. 4575. Pairing-Based Cryptography* (pp. 367–391). Springer-Verlag.

[41] Jakobsson, M. (1995). Blackmailing using undeniable signatures. In A. De Santis (Ed.), *Lecture Notes in Computer Science: Vol. 950. Advances in Cryptology - EUROCRYPT* (pp. 425–427). Springer-Verlag.

[42] Jakobsson, M., Sako, K., & Impagliazzo, R. (1996). Designated verifier proofs and their applications. In U. Maurer (Ed.), *Lecture Notes in Computer Science: Vol. 1070. Advances in Cryptology*

- *EUROCRYPT* (pp. 143–154). Springer-Verlag.

[43] Katz, J., & Wang, N. (2003). Efficiency improvements for signature schemes with tight security reductions. In *Proceedings of the 10th ACM Conference on Computer and Communications Security* (pp. 155–164). ACM.

[44] Koide, A., Tso, R., & Okamoto, E. (2008). Convertible undeniable partially blind signature from bilinear pairings. In *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing.* (pp. 77–82).

[45] Kurosawa, K., & Heng, S.-H. (2005). 3-move undeniable signature scheme. In R. Cramer (Ed.), *Lecture Notes in Computer Science: Vol. 3494. Advances in Cryptology - EUROCRYPT* (pp. 181–197). Springer-Verlag.

[46] Kurosawa, K., & Takagi, T. (2006). New approach for selectively convertible undeniable signature schemes. In X. Lai & K. Chen (Eds.), *Lecture Notes in Computer Science: Vol. 4284. Advances in Cryptology - ASIACRYPT* (pp. 428–443). Springer-Verlag.

[47] Laguillaumie, F., Paillier, P., & Vergnaud, D. (2005). Universally convertible directed signatures. In B. Roy (Ed.), *Lecture Notes in Computer Science: Vol. 3788. Advances in Cryptology - ASIACRYPT* (pp. 682–701). Springer-Verlag.

[48] Laguillaumie, F., & Vergnaud, D. (2005). Time-selective convertible undeniable signatures. In A. Menezes (Ed.), *Lecture Notes in Computer Science: Vol. 3376. Topics in Cryptology - CT-RSA* (pp. 154–171). Springer-Verlag.

[49] Li, X., Chen, K., & Sun, L. (2005). Certificateless signature and proxy signature schemes from bilinear pairings. *Lithuanian Mathematical Journal*, *45*, 76–83.

[50] Libert, B., & Quisquater, J.-J. (2004). Identity-based undeniable signatures. In T. Okamoto (Ed.), *Lecture Notes in Computer Science: Vol. 2964. Topics in Cryptology - CT-RSA* (pp. 112–125). Springer-Verlag.

[51] Maurer, U., Renner, R., & Holenstein, C. (2004). Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In M. Naor (Ed.), *Lecture Notes in Computer Science: Vol. 2951. Theory of Cryptography* (pp. 21–39). Springer-Verlag.

[52] Monnerat, J., & Vaudenay, S. (2004). Generic homomorphic undeniable signatures. In P. Lee (Ed.), *Lecture Notes in Computer Science: Vol. 3329. Advances in Cryptology - ASIACRYPT* (pp. 1–6). Springer-Verlag.

[53] Monnerat, J., & Vaudenay, S. (2006). Short 2-move undeniable signatures. In P. Nguyen (Ed.), *Lecture Notes in Computer Science: Vol. 4341. Progress in Cryptology - VIETCRYPT* (pp. 19–36). Springer-Verlag.

[54] Neven, G. (2004). *Provably secure identity-based identification schemes and transitive signatures*. Unpublished doctoral dissertation, Katholieke Universiteit Leuven.

[55] Nielsen, J. (2002). Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In M. Yung (Ed.), *Lecture Notes in Computer Science: Vol. 2442. Advances in Cryptology - CRYPTO* (pp. 191–214). Springer-Verlag.

[56] Ogata, W., Kurosawa, K., & Heng, S.-H. (2006). The security of the FDH variant of Chaum's undeniable signature scheme. *IEEE Transactions on Information Theory*, *52*(5), 2006–2017.

[57] Paterson, K., & Schuldt, J. (2006). Efficient identity-based signatures secure in the standard model. In L. Batten & R. Safavi-Naini (Eds.), *Lecture Notes in Computer Science: Vol. 4058. Information Security and Privacy* (Vol. 4058, pp. 207–222). Springer-Verlag.

[58] Phong, L., Kurosawa, K., & Ogata, W. (2010). Provably secure convertible undeniable signatures with unambiguity. In J. Garay & R. De Prisco (Eds.), *Lecture Notes in Computer Science: Vol. 6280. Security and Cryptography for Networks* (pp. 291–308). Springer-Verlag.

[59] Pointcheval, D., & Stern, J. (2000). Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, *13*, 361–396.

[60] Qiong, H., & Wong, D. S. (2009). *New constructions of convertible undeniable signature schemes without random oracles.* Cryptology ePrint Archive, Report 2009/517 [Online]. Available: http://eprint.iacr.org/2009/517.

[61] Sakurai, K., & Miyazaki, S. (2000). An anonymous electronic bidding protocol based on a new convertible group signature scheme. In E. Dawson, A. Clark, & C. Boyd (Eds.), *Lecture Notes in Computer Science: Vol. 1841. Information Security and Privacy* (pp. 385–399). Springer-Verlag.

[62] Shamir, A. (1985). Identity-based cryptosystems and signature schemes. In G. Blakley & D. Chaum (Eds.), *Lecture Notes in Computer Science: Vol. 196. Advances in Cryptology - CRYPTO* (pp. 47–53). Springer-Verlag.

[63] Shoup, V. (1997). Lower bounds for discrete logarithms and related problems. In *Lecture Notes in Computer Science: Vol. 1233. Advances in Cryptology - EUROCRYPT* (pp. 256–266). Springer-Verlag.

[64] Tan, S.-Y., Heng, S.-H., & Goi, B.-M. (2010). Java implementation for pairing-based cryptosystems.

[65] Tso, R., Huang, X., & Susilo, W. (2012). Strongly secure certificateless short signatures. *Journal of Systems and Software*, *85*(6), 1409–1417.

[66] Wu, W., Mu, Y., Susilo, W., & Huang, X. (2008). Provably secure identity-based undeniable signatures with selective and universal convertibility. In D. Pei, M. Yung, D. Lin, & C. Wu (Eds.), *Lecture Notes in Computer Science: Vol. 4990. Information Security and Cryptology* (pp. 25–39). Springer-Verlag.

[67] Yap, W.-S., Heng, S.-H., & Goi, B.-M. (2006). Certificateless encryption schemes revisited. In *Proceedings of the mmu international symposium on information and communications technologies 2006 - m2usic 2006* (pp. 471–474).

[68] Zhang, F., Safavi-Naini, R., & Susilo, W. (2004). An efficient signature scheme from bilinear pairings and its applications. In F. Bao, R. Deng, & J. Zhou (Eds.), *Lecture Notes in Computer Science: Vol. 2947. Public Key Cryptography - PKC* (pp. 277–290). Springer-Verlag.

[69] Zhang, F., Safavi-Naini, R., & Susilo, W. (2005). Attack on Han et al.'s id-based confirmer (undeniable) signature at ACM-EC'03. *Applied Mathematics and Computation*, *170*(2), 1166–1169.

[70] Zhao, W., & Ye, D. (2012). Certificateless undeniable signatures from bilinear maps. *Information Sciences*, *199*, 204–215.

ABBREVIATION

3-DDH 3-Decisional Diffie-Hellman.

BDH Bilinear Diffie-Hellman.

CA Certificate Authority.

CDH Computational Diffie-Hellman.

CLC Certificateless Cryptography.

DBDH Decisional Bilinear Diffie-Hellman.

DDH Decisional Diffie-Hellman.

DL Diffie-Hellman.

IDC Identity-Based Cryptography.

KGC Key Generation Centre.

PKG Private Key Generator.

TPKC Traditional Public Key Cryptography.

TTP Trusted Third Party.

ZKIP Zero-Knowledge Interactive Proof.

PUBLICATION LIST

Journal Articles

[1] Behnia, R., Heng, S.-H., & Gan, C.-S. (2011a). The applications of DH-tuple witness indistinguishable protocols. *International Journal of Cryptology Research*, *3(1)*, 34–47.

[2] Behnia, R., Heng, S.-H., & Gan, C.-S. (2011b). Development of undeniable signature schemes without certificates. *International Journal of Cryptology Research*, *3(1)*, 48–67.

[3] Behnia, R., Heng, S.-H., & Gan, C.-S. (2012). Short and efficient identity-based undeniable signature scheme. In S. Fischer-Hübner, S. Katsikas, & G. Quirchmayr (Eds.), *Lecture Notes in Computer Science: Vol. 7449. Trust, Privacy and Security in Digital Business* (pp. 143–148). Springer-Verlag.

[4] Behnia, R., Heng, S.-H., & Gan, C.-S. (2013a). An efficient certificateless undeniable signature scheme. *Manuscript submitted for publication*.

[5] Behnia, R., Heng, S.-H., & Gan, C.-S. (2013b). Notes on two flawed attacks on undeniable signature schemes. *Manuscript submitted for publication*.

[6] Behnia, R., Heng, S.-H., & Gan, C.-S. (2013c). On the security of an efficient certificateless undeniable signature scheme. *To appear in INFORMATION - International Information Institute Journal*.

[7] Behnia, R., Heng, S.-H., & Gan, C.-S. (2013d). Provably secure certificateless convertible undeniable signature scheme. *Manuscript submitted for publication.*

[8] Behnia, R., Heng, S.-H., & Gan, C.-S. (2013e). Security of NIDV proof systems for certificate-free undeniable signature schemes. *To appear in International Journal of Cryptology Research.*

[9] Behnia, R., Heng, S.-H., & Gan, C.-S. (2013f). Weaknesses of an efficient identity-based undeniable signature scheme. *To appear in INFORMATION - International Information Institute Journal*.

Conference Proceedings

[1] Behnia, R., Heng, S.-H., & Gan, C.-S. (2012a). Attacks on Chow's identity-based undeniable signature scheme. Paper presented at ACSA-Summer 2012, Vancouver, Canada.

[2] Behnia, R., Heng, S.-H., & Gan, C.-S. (2012b). On the security of pairing-based non-interactive designated verifier proofs of undeniable signature schemes. In *IEEE Conference on Sustainable Utilization and Development in Engineering and Technology* (pp. 207–212).

[3] Behnia, R., Heng, S.-H., & Gan, C.-S. (2012c). Remarks on a certificateless undeniable signature scheme. Paper presented at WCC-ACSA 2012, Jeju, Korea. [The best paper award].