2016

Contributions to cryptography with restricted conditions

Weiwei Liu
University of Wollongong

Contributions to Cryptography with Restricted Conditions

Weiwei Liu

This thesis is presented as part of the requirements for the conferral of the degree:

Doctor of Philosophy (Integrated)

The University of Wollongong
School of Computing and Information Technology

May 9, 2016

Declaration

I, Weiwei Liu, declare that this thesis submitted in partial fulfilment of the requirements for the conferral of the degree Doctor of Philosophy (Integrated), from the University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Weiwei Liu
May 9, 2016

Abstract

Cryptography with restricted conditions refers to cryptographic primitives with special requirements or conditions. For example, a proxy signature scheme only allows a proxy signer with valid delegation to issue signatures on behalf of the original signer, while for k-time anonymous authentication, a service provider can be ensured that a user can only have anonymous access to the services for up to k-times. Due to the different requirements in various types of security systems, more and more cryptographic primitives with new features are emerging. In this thesis, we study several cryptographic primitives with restricted conditions and their applications, including proxy signature and its variant in the attribute-based setting, oblivious transfer, k-time anonymous authentication and their applications in the e-coupon systems.

To address the problem that a proxy signer might abuse the delegated signing right from the original signer, in this thesis, we present a k-time proxy signature scheme that only permits a designated proxy signer to generate a pre-determined number of proxy signatures. In the subsequent work, we implement proxy signature in attribute-based setting, that is, an original signer with a set of attributes can delegate his signing right to a proxy signer with a normal public and private key pair. One interesting feature of the proposed scheme is that by verifying a proxy signature, the public can be convinced the signature is generated by the proxy signer with valid deletion from the original signer whose attributes satisfy a pre-claimed predicate. Then we identify one attack that has been neglected in many existing delegation-by-warrant proxy signature schemes. We present the details of this attack and propose a general solution that can efficiently thwart the attack.

In this thesis, we also construct several e-coupon systems with new properties. In the first e-coupon system, the user identity privacy would be revealed if a dishonest user requests more than pre-determined number of services specified in the coupon. Different from other e-cash and k-time anonymous authentication schemes, we achieve traceability without involvement of a trusted third party. Besides, for the first time, we formalize the concept of privacy of purchase, that is, the choices of the users when redeeming a coupon with the server is hidden. Moreover, we propose a new oblivious transfer (OT) scheme with retrievable receiver's privacy and design another e-coupon system based on our new OT scheme. If a user remains honest, the user anonymity and privacy of purchase are both well protected. Otherwise, the identity and purchase privacy of the user can be revealed by the service provider.

Acknowledgments

I would like to express my sincere appreciation to my supervisors Prof. Yi Mu and Dr. Guomin Yang, for their excellent patience and guidance. They have been providing invaluable encouragement and suggestions to me. Without their continuous support, the completion of this thesis would be impossible. Besides my supervisors, I would like to thank Prof. Willy Susilo, Prof. Minjie Zhang, Dr. Man Ho Au, Dr. Yong Yu, Dr. Fuchun Guo, Dr. Jinguang Han and Dr. Xinyi Huang, for their insightful comments and valuable suggestions to my study and research.

I would like to thank all my colleagues and friends in Australia, a non-exhaustive list of whom includes: Vu Duc Nguyen, Yanguang Tian, Rongmao Chen, Nan Li, Jianchang Lai, Zhongyuan Yao, Shiwei Zhang, Shengmin Xu, Jiannan Wei and Xiaoyu Yu, for their accompany during my PhD study.

It is an honour for me to be a research student in the Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong. I also take this opportunity to thank the China Scholarship Council and University of Wollongong for their financial support to my study.

Last but not least, I am sincerely grateful to my parents, for everything.

Publications

This thesis is related to the following publications/manuscripts.

1. Weiwei Liu, Guomin Yang, Yi Mu, and Jiannan Wei. k-time Proxy Signature: Formal Definition and Efficient Construction. ProvSec, volume 8209 of Lecture Notes in Computer Science, page 154-164. Springer, (2013).

2. Weiwei Liu, Yi Mu, and Guomin Yang. Attribute-Based Signing Right Delegation. NSS, volume 8792 of Lecture Notes in Computer Science, page 323-334. Springer, (2014).

3. Weiwei Liu, Yi Mu, and Guomin Yang. An Efficient Privacy-Preserving E-coupon System. Inscrypt, volume 8957 of Lecture Notes in Computer Science, page 3-15. Springer, (2014).

4. Weiwei Liu, Yi Mu, Guomin Yang and Xinyi Huang. Improved Security of Delegation-by-Warrant Proxy Signatures Schemes (submitted).

5. Weiwei Liu, Yi Mu, Guomin Yang and Jingguang Han. Efficient Oblivious Transfer with Retrievable Receiver's Privacy (submitted).

6. Weiwei Liu, Yi Mu, Guomin Yang and Yong Yu. Efficient E-Coupon Systems with Strong User Privacy (submitted).

List of Notations

The following abbreviations are used throughout this thesis.

κ	A security parameter
1^κ	A string of length κ
$\mathbb{G}$	A group
$\mathbb{G}_q$	A group of order q
$\mathbb{N}$	The nature number set
$\mathbb{Z}$	The set of integers
$\mathbb{Z}_p$	The set of integers modulo p
$a \in_R \mathcal{A}$	a is randomly chosen from set $\mathcal{A}$
$a \notin \mathcal{A}$	a is not in set $\mathcal{A}$
$a\|\|b$	The concatenation of strings a and b
$\mathbb{A}$	An attribute universe
Υ	A predicate
$\Pr[A]$	The probability that event A happens

List of Abbreviations

The following abbreviations are used throughout this thesis. Some special abbreviations will be defined when they are first used.

ABS	Attribute-Based Signature
ABPS	Attribute-Based Proxy Signature
CMA	Chosen Message Attack
CPA	Chosen Plaintext Attack
CCA1	Non-Adaptive Chosen Ciphertext Attack
CCA2	Adaptive Chosen Ciphertext Attack
DS	Digital Signature
EUF	Existential Unforgery
IND	Indistinguishability
OT	Oblivious Transfer
OTRRP	Oblivious Transfer with Retrievable Receiver's Privacy
PKC	Public Key Crytography
PKE	Public Key Encryption
PPT	Probabilistic Polynomial Time
PKI	Public Key Infrastructure
TTP	Trusted Third Party

Contents

List of Tables

Chapter 1

Introduction

Cryptography plays a central role in ensuring the security of data in storage and transmission. Roughly speaking, cryptography is an inclusive field covering a wide range of topics from encryption, signature, authentication to zero-knowledge proof, which provide the vital security properties like confidentiality, integrity, non-reputation and authenticity. In many real-world applications such as e-cash [CFN88], e-coupon [CES+05], e-voting [Buc04] and trial browsing [TFS04], besides the security properties mentioned above, there are some special requirements and conditions such as restricting the number of times that a user can access to the services. In this thesis, we investigate some cryptography primitives with restricted conditions.

1.1 Background

A valid digital signature convinces a recipient that the message was sent by a claimed sender and the message was not altered in transit, while the sender cannot deny having sent the message later. Digital signature has been applied widely in software distribution, financial transactions, and in other cases where it is important to ensure the authenticity, integrity and non-reputation. Proxy signature is a special type of digital signature, where an original signer (or delegator) can delegate his/her signing right to a proxy signer. Thereafter, the proxy signer can sign documents on behalf of the original signer. Roughly speaking, a secure proxy signature scheme should satisfy the following requirements.

- **Verifiability**: Given a proxy signature, a verifier can be convinced that the proxy signature is indeed a valid signature generated by the proxy signer with proper delegation from an original signer on the signed message.
- **Identifiability**: Given a proxy signature, a verifier is able to determine the identities of the corresponding original signer and proxy signer.
- **Unforgeablity**: No one, except the designated proxy signer, can create a valid proxy signature.
- **Undenability**: A proxy signer can not deny at a later time on a proxy signature that he has created before.
- **Prevention of misuse**: It is required in the first type of proxy signature schemes that the proxy signing key can not be used for purposes other than creating proxy

signatures. Once misused, the identity of the misbehaving proxy signer can be determined explicitly.

Proxy signature and its extended variants have been found very useful in many practical applications, such as distributed systems [Neu93], grid computing [FKTT98], and mobile agent applications [LKK01b].

Attribute-based signature (ABS) is another special type of digital signature that has been proposed recently. It can be treated as an extension of identity-based signature (IBS) but has better fine-grained control over the signer's identification information. In an ABS, a signer with attribute set $\mathcal{A}$ will first obtain a secret key from the central authority (or key generation center), and then can use the obtained secret key to sign any messages. The signature can be verified with regards to an attribute predicate Υ and the verification will be successful if and only if the signer's attribute set $\mathcal{A}$ satisfies Υ. However, the verifier cannot gain any information about the signer's attributes except the fact that they satisfy the pre-claimed predicate.

ABS has been found useful in the circumstances where the capabilities of the users depend on certain combinations of their attributes. Attribute-based proxy signature (ABPS) is a nature extension of ABS. Compared with normal proxy signature (PKI-based setting), the users in ABPS are identified by attributes. ABPS has many potential applications, for example, attribute-based authentication [MPR11]. Consider a database whose access control is described in a policy such that only users who hold authorised attribute keys can access it. An authorised user can delegate his/her signing rights to another user so that the latter can also access the database and collect information when the former is not available.

Oblivious Transfer (OT) is one of the fundamental cryptographic primitives that has been used widely in various security applications such as exchange of secrets [Rab81], contract signing [EGL85], secure multiparty computation [Yao82] and private information retrieval [CKGS98]. An oblivious transfer scheme is an interactive protocol running between a sender with a set of messages $\{m_1, m_2, \ldots, m_n\}$ and a receiver with a set of choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. After running the protocol, the receiver learns the intended messages $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ but cannot learn anything about m_i for $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. Meanwhile, the receiver's choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ are completely hidden from the sender.

An electronic coupon (or e-coupon) can be used by a user to obtain an electronic good or service from a service provider, which is usually a coupon issuer. E-coupon systems are similar but different from electronic cash (or e-cash) systems [CFN88, NMV97, Bra93, AWSM07]. One major difference is that an e-coupon system only involves two parties: the user and the coupon issuer, while in an e-cash system there is a third party which is the bank. An e-coupon system has less algorithms/protocols compared with e-cash. The coupon issuer can issue a coupon to a user through a

coupon issuing algorithm; then the user can redeem the coupon at a later time to obtain the good/service specified in the coupon. Similar to e-cash systems, e-coupon systems are very useful in e-commerce, especially when the shops don't want to have the bank involved in the transactions.

1.2 Motivations and Our Results

In this thesis, we focus on certain cryptographic primitives with constrained conditions including proxy signature, oblivious transfer, k-time anonymous authentication and their applications. The main motivations and our results are summarized as follows.

1. **k-Time proxy signature**. In proxy signature, one practical issue is how to prevent the proxy signers from misusing the signing ability delegated by the original signer. One solution in existing certificate-based proxy signature schemes [LKK01b, XZF05, Wan05, HSMW06, YMS+12] is to specify the validity period of the delegation in the warrant (which is essentially a signature of the original signer). In this thesis, we propose a new k-time proxy signature scheme, in which a proxy signer is only able to generate a constant number of proxy signatures on behalf of the original signer. We provide a formal and complete treatment for multi-time (or k-time) proxy signature schemes. We first provide a formal security model for such schemes. Then we propose a new k-time proxy signature scheme based on the Schnorr signature scheme and verifiable secret sharing. In our scheme, the original signer can specify in the warrant the number of proxy signatures a proxy signer can produce. If the proxy signer produces more than predetermined number of proxy signatures, his/her private key can be computed by the public. We prove the security of the proposed scheme under the random oracle model.

2. **Attribute-based signing right delegation**. Attribute-based signature is a new primitive that has been found useful in many real-world applications [MPR11]. Attribute-based proxy signature is a natural extension of ABS. Consider the aforementioned attribute-based access control for a database where only users who hold certain attributes can access the data. An authorised user can delegate his/her signing right to another user so that the latter can also access the database and collect information when the former is not available. The delegated signer is called a proxy of the original signer. In our proposed scheme, the verifier can be convinced that a valid proxy signer holds the right delegation from an authorised original signer and therefore can access the

database. We define the security models for ABPS and propose a threshold ABPS scheme that is proven secure under the proposed security models.

3. **Improving the security of delegation-by-warrant proxy signature schemes**. Among all the constructions of proxy signature, the delegation-by-warrant method has gained the most popularity. A large number of delegation-by-warrant proxy signature schemes [Zha97, LKK01b, Wan05, HSMW06] have been proposed. These delegation-by-warrant proxy signature schemes can be further classified into two categories according to whether the proxy signature is generated by the proxy signer using his own private key or not. In the first type, the proxy signer generates a new proxy signing key using the delegation information and his own private key. The proxy signatures are generated under the new proxy signing key. The proxy signature schemes in [Zha97, LKK01b, Wan05, LYMW13] fall into the first type. In the second type, the proxy signer issues a proxy signature using his own private key. The proxy signatures are essentially combinations of the original signer's signature on the warrant and the proxy signer's signature on the message. Such proxy signature schemes could be found in [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a]. We show that an attack has been neglected by the second type of proxy signature schemes proposed in the literature. Our attack is based on a realistic assumption that an adversary has access to the original signer and the proxy signer's standard signatures. We show that under such a circumstance, many proxy signature schemes [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a] that have been proven secure are in fact insecure.

   We demonstrate the attack by launching it against an identity-based proxy signature scheme [WMS$^+$07] that has been proven secure. We show that a malicious adversary can create a proxy signature on a message, if he has access to the standard signatures of the original signer and the proxy signer. Thus, these proxy signature schemes [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a], which we believe is not a complete list, are in fact insecure under the circumstance we consider. We then propose an efficient solution to modify the identity-based proxy signature scheme [WMS$^+$07] in order to thwart this attack. It is worth noting that the same method can also be applied to other proxy signatures [HSMW06, LKZC07, SXYM11, LMY14a] to resist the attack.

4. **An efficient privacy-preserving e-coupon system**. There have been a number of e-coupon systems [CES$^+$05, Ngu06b, CGH06] proposed in the literature. Although user anonymity has been considered in all the above e-coupon systems, none of them has considered traceability against dishonest users. That is, if a dishonest user redeems a multi-coupon more than the pre-

determined number of times, it is desirable to allow the coupon issuer to trace the identity of the user. On the other hand, for honest users, we should ensure that their identities will remain anonymous to the coupon issuer.

Another desirable feature of an e-coupon system is user privacy (privacy of purchase). Different from user anonymity, here we are concerning the privacy of the goods/services chosen by the users during the redemption process. In general, an e-coupon can contain a number of options that have the same value and a user can choose any one of them. If the shop can know the good/service chosen by the user in the redemption process, then it is possible for the shop to link two redemptions performed by the same customer (e.g., a customer may prefer to redeem the same item among several transactions). Therefore, it is also desirable to keep the buying behavior of a user secret from the coupon issuer. It is worth noting that user privacy is possible in the electronic world since we are considering electronic goods, so there is no physical reduction of the goods from the shop's 'warehouse', while in the physical world, the shop can always trace the number of each good to find out the item redeemed by the user.

We propose a new e-coupon system, which can achieve all the desirable properties, namely unforgeability, anonymity for honest users, double redemption detection, traceability against dishonest users, and user privacy. It is worth noting that different from a fair e-cash system [SPC95], the traceability in our e-coupon system is performed by the merchant (i.e., coupon issuer) rather than the bank.

5. **Oblivious transfer with retrievable receiver's privacy**. Oblivious transfer (OT) has served as a useful primitive in designing privacy-preserving systems in which the choices of the users should be hidden. All the previous research on OT aimed to design OT schemes with unconditional receiver and sender privacy. However, in real-world applications [AIR01, LMY14b], it is desirable for the sender to trace the choices of the receiver if they misbehave. Thus, the previous OT schemes are not suitable in these scenarios. We propose a new OT scheme with retrievable receiver's privacy such that the privacy of an honest receiver is protected unconditionally while all the previous choices of a misbehaving receiver can be revealed by the sender if the receiver makes more than the pre-determined number of choices in the OT protocol. We prove the security of the proposed OT scheme under the half-simulation model [NP05].

6. **Efficient e-coupon systems with strong user privacy**. We propose two novel e-coupon systems supporting multiple usage of an electronic coupon. Besides the security requirements mentioned above, our e-coupon systems can

achieve two new properties. First, the proposed e-coupon systems allows an honest user to redeem a valid coupon for up to k times, where k is a predetermined number set by the coupon issuer. Besides, if a malicious user attempts redeeming a coupon for more than k times, both the identity privacy and redemption privacy could be traced by the coupon issuer. We also define the formal security models for these new security requirements, and show that our new e-coupon systems are proven secure in the proposed models.

1.3 Thesis Organization

The rest of this thesis is organized as follows.

In Chapter 2, we review some previous research on proxy signature, attribute-based signature, oblivious transfer and e-coupon systems.

In Chapter 3, we review some background knowledge in cryptography. We introduce the concept of cyclic group, bilinear pairing, Lagrange interpolation and present the complexity assumptions used in this thesis. Besides, we introduce some basic cryptographic primitives such as digital signature, proxy signature, public key encryption, and so on.

In Chapter 4, we propose a k-time proxy signature scheme which can restrict the number of proxy signature generated by a proxy signer in the name of an original signer. We present the formal definition and security model for k-time proxy signature. We prove the security of the proposed scheme under the proposed security model using random oracle.

In Chapter 5, we introduce an attribute-based proxy signature scheme exploiting public key-based proxy signature and attribute-based signature. We first present the formal definition and security model of ABPS. Then we analyse the security of the proposed scheme under the proposed security model.

In Chapter 6, we present an attack to one type of delegation-by-warrant proxy signature schemes. We take a concrete ID-based proxy signature scheme as an example to explain how the attack works. Then we propose an improved scheme, in which we propose a solution that can also be applied to other proxy signature schemes to prevent the attack. We prove the security of the improved scheme to demonstrate that the improved scheme is not only secure under the previous adversarial model by also can resist the new attack.

In Chapter 7, we design a privacy-preserving e-coupon system with some new properties. We first present the formal definition and security model of such an e-coupon system, in which we formalize two new properties about user privacy (privacy of purchase) and traceability. We then construct an e-coupon system that can achieve all the security properties. We analyze the security of the proposed e-coupon

system and show that it is secure under the defined security model.

In Chapter 8, we present two new oblivious transfer schemes with retrievable receiver's privacy (OTRRP). We first introduce the formal definition and security model for OTRRP. Then we present two different constructions of OTRRP. We analyse the security of the proposed OTRRP schemes and prove that they are secure under the half-simulation model.

In Chapter 9, we design another two e-coupon systems that support multiple-usage of an electronic coupon. We first present the formal definition and security model for the new e-coupon system. Then we construct two new e-coupon systems achieving different security properties. We prove that the proposed e-coupon systems are secure under the defined security model.

In Chapter 10, we conclude this thesis and present some future work.

Chapter 2

Literature Review

In this chapter, we review prior research on some cryptography primitives with restricted conditions, including proxy signature, attribute-based signature, oblivious transfer and e-coupon systems.

2.1 Proxy Signature

Proxy signature was first proposed by Mambo, Usuda and Okamoto in 1996 [MUO96]. In their work they presented three different methods in constructing proxy signature schemes, namely full delegation, partial delegation, and delegation by warrant. It has been shown impractical to construct proxy signature schemes by means of full delegation since an original signer has to handle his own secret to the proxy signer. Partial delegation proxy signature schemes can be further divided into proxy-protected and proxy-unprotected schemes according to whether a verifier can decide the proxy signature is generated by a proxy signer or the original signer. In a subsequent work, Kim et al. [KPW97] proposed a new type of proxy signature combining partial delegation and warrant. They further showed that such a combination can provide a higher level of security. Since then many proxy signature schemes based on partial delegation and warrant have been proposed (e.g., [LKK01b, XZF05, Wan05, HSMW06, YMS+12]).

Lee et al. [LKK01b] presented several attacks against previous proxy signature schemes and constructed a novel strong non-designated proxy signature scheme, which could be used in multi-proxy signature. Besides, the authors provides new classifications of proxy signatures, namely strong and weak proxy signatures, designated and non-designated proxy signature and self-proxy signature. Xu et al. [XZF05] extended proxy signature into identity-based setting, where a user is identified by some unique information about the identity of the user (e.g. user's email address). They also proposed the sound security models for proxy signature in identity-based setting and the first identity-based proxy signature scheme proven secure using random oracles. Wang proposed a designated-verifier proxy signature scheme from two-party Schnorr signature and analyzed its security in [Wan05]. In addition, the author discussed weaker designated-verifier proxy signature and strong designated-verifier proxy signature as an extension of [LKK01b], with the difference that the proxy signature could only be verified by the designated verifier instead of the public. Huang et al. [HSMW06] proposed the first proxy signature scheme that

the security is not relied on random oracle. However, the proposed proxy signature scheme do not support strong unforgeability, which means any one could produce a new proxy signature after seeing a proxy signature and the corresponding message. Yong et al. [YMS+12] proposed a proxy signature scheme whose security is based on the integer factorization problem in the random oracle, which are different from proxy signature schemes mentioned above whose security relying on discrete logarithm problem and its variants.

Besides proxy signature schemes mentioned above, many extensions on proxy signature have also been proposed according to different application needs, such as threshold proxy signature [Zha97], blind proxy signature [ZSNL], one-time proxy signature [MH05], and so on. Threshold proxy signature, also known as multi-proxy signature, enables an original signer to delegate his signing right to multiple proxy signers. The proxy signers need to work together in order to produce a valid proxy signature on behalf of the original signer. One-time proxy signature puts strict restrictions on the signing capability of a proxy signer, who is only allowed to generate one valid proxy signature on behalf of the original signer. Blind proxy signature allows a user to obtain a valid signature on a message in a way that the proxy signer learns neither the message nor the resulting signature.

In proxy signature, one important problem is to prevent the proxy signers from abusing the signing ability from the original signer. The conventional solution is to specify the valid time period of the delegation in the warrant. When a verifier examines a proxy signature, he first checks the warrant to verifiy if the proxy signature falls into the valid time period. However, this is not sufficient since a malicious proxy signer can still produce a large number of proxy signatures in a short period if his computation power is strong enough. It is worth noticing proxy signature with revocation [LHH05, DSP07] has been proposed as an independent work to allow an original signer to revoke the signing rights from a proxy signer whenever it is necessary. However, the proposed proxy signature schemes are still suffering to the problem mentioned above.

2.2 Attribute-Based Signature

Attribute-based signature is treated as an extension of identity-based signature [Sha84] by allowing the identity of a user to be a set of descriptive attributes rather than a single unique string. Several ABS schemes supporting different types of predicates have been proposed after the concept of attribute-based cryptography was proposed in [SW05]. Li and Kim [LK08] proposed two attribute-based ring signature schemes. In their construction, the ring is formed by the users with a set of same attributes. In this way, the identity of the signer could be hidden in the

ring. However, as one of the basic properties, ABS has already enforced anonymity. Essentially, ring and group are comparable to special cases of ABS. In a subsequent work, Li et al. [LAS+10] proposed two ABS constructions supporting flexible threshold predicates. In their schemes, the predicate is a set of n attributes, and the signer must possess at least k ($k \leq n$) of them in order to generate a valid signature. The verifier can be convinced that the signer is really holding k out of n attributes, but cannot find out which k attributes are possessed by the signer. Later, Maji et al. [MPR11] proposed another ABS scheme where the attribute predicates can be expressed as monotone-span programs. Then in [OT14], Okamoto and Takashima proposed the first ABS scheme that can support more general non-monotone predicates.

ABS has been found useful in the circumstances where the capabilities of the users depend on certain combinations of their attributes. Attribute-based proxy signature (ABPS) is a nature extension of ABS. Compared with normal proxy signature (PKI-based setting), the users in ABPS are identified by attributes. ABPS has many potential applications, for example, attribute-based authentication [MPR11]. Consider a database whose access control is described in a policy such that only users who hold authorised attribute keys can access it. An authorised user can delegate his/her signing rights to another user so that the latter can also access the database and collect information when the former is not available.

2.3 Oblivious Transfer

An oblivious transfer scheme is an interactive protocol running between a sender with messages $\{m_1, m_2, \ldots, m_n\}$ and a receiver with choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. After running the protocol, the receiver learns the intended messages $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ but cannot learn anything about m_i for $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. Meanwhile, the receiver's choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ are completely hidden from the sender.

The concept of oblivious transfer was introduced by Rabin in 1981 [Rab81]. In their original construction, the sender sends a single bit 0 or 1 to the receiver in such a way that with 1/2 probability the receiver will receive the same bit and with 1/2 probability that the receiver will receive nothing. At the same time, the sender has no idea whether the receiver receives the message or not. Since then, oblivious transfer has attracted a lot of attentions, and a number of works [EGL85, BCR86, NP99, CT05, CNS07] have been done to improve the original OT scheme in different aspects.

Even et al. [EGL85] proposed a 1-out-of-2 OT (OT_2^1) scheme, in which the sender obliviously sends a message m_i, $i \in \{0, 1\}$, to the receiver. Shortly after that, Brassard et al. [BCR86] extended the OT_2^1 [EGL85] to a more general k-out-of-n

(OT_n^k) setting, where the receiver is able to make multiple choices $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ ($\sigma_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq k$) from a set of n messages $\{m_1, m_2, \ldots, m_n\}$ held by the sender, meanwhile the receiver's choices remain oblivious to the sender. Since then, many subsequent works [MZV02, CT05] aimed to design more efficient OT_n^k schemes. Different from normal OT_n^k, another important research direction on OT is adaptive OT_n^k [NP99]. In adaptive OT_n^k, the receiver can choose the messages adaptively, namely, the i-th value chosen by the receiver depends on the first $i-1$ values.

In the early OT schemes reviewed above, there is no condition on restricting the receiver's ability. Any user in the system can act as a receiver and run the OT protocol to choose messages held by the sender obliviously. To address this problem, Coull et al. [CGH09] proposed an OT scheme supporting access control using state graphs, where for every transaction, the state of the receiver shifts from one to another. The receiver can access the protected services only if some of his states are not used. Camenisch et al. [CDN09] proposed another approach to enforce access control. In their system, the receiver first authenticates himself to a trusted third party to obtain some credentials. Later, the receiver proves to the sender that he possesses a valid credential from the third party using zero- knowledge proof. However, in this construction, the access policy is publicly known. To address this problem, Camenisch et al. [CDNZ11] proposed another oblivious transfer with access control in which only the receivers whose attributes satisfy a predicate can access the services. In order to reduce the computation and communication cost, Han et al. [HSMY12] proposed two efficient oblivious transfer schemes without zero-knowledge proof. In addition, different form previous schemes, the receivers could obtain credentials from a trusted third party but do not have to authenticate themselves. Thus, the communication and computation cost is lower than previous schemes supporting access control.

There have been a lot of research works [NP05, CNS07, KN09] on defining OT security, which can be classified into honest-but-curious model, half-simulation model [NP05] and full-simulation model [CNS07, KN09], according to whether the OT scheme can provide simulatable security for the sender and/or receiver. In the honest-but-curious model, all participants in the protocol are assumed to be honest, which makes this model too idealistic for practical use. Naor and Pinkas [NP05] introduced the half-simulation model that allows malicious senders and receivers. However, in this model, the security of the sender and receiver are considered separately. Half-simulation model achieves simulatable security for sender privacy and computationally indistinguishability for receiver privacy. In the full-simulation model [CNS07, KN09], it achieves simulatable security for both the receiver and sender.

However, previous research on oblivious transfer mainly focused on designing OT with unconditional sender and receiver's privacy. In real-world applications, for example in digital content browsing [TFS04], OT can be applied to hide what contents have been browsed by a user. However, if a user is identified to be malicious, it is desirable to determine what contents (especially if the contents are sensitive) have been read by the user. Traditional OT schemes cannot meet the requirement in this situation.

2.4 Privacy-Preserving E-Coupon Systems

There are a number of e-coupon systems proposed in the literature. Chen et al. [CES+05] presented a privacy-preserving e-coupon system, in which the users are allowed to redeem a single e-coupon for a pre-determined number of times. In order to reduce the cost for issuing and redeeming coupons, Nguyen [Ngu06b] later presented another more efficient e-coupon system which has constant communication and computation cost. In [CGH06], Canard et al. also proposed another interesting multi-coupon system which allows a user to transfer some value of a multi-coupon to another user.

Although user anonymity has been considered in all the above e-coupon systems, none of them has considered traceability against dishonest users. That is, if a dishonest user redeems a multi-coupon more than the pre-determined number of times, it is desirable to allow the coupon issuer to trace the identity of the user. On the other hand, for honest users, we should ensure that their identities will remain anonymous to the coupon issuer.

Another desirable feature of an e-coupon system is user privacy. Different from user anonymity, here we are concerning the privacy of the goods/services chosen by the users during the redemption process. In general, an e-coupon can contain a number of options that have the same value and a user can choose any one of them. If the shop can know the good/service chosen by the user in the redemption process, then it is possible for the shop to link two redemptions performed by the same customer (e.g., a customer may prefer to redeem the same item among several transactions). Therefore, it is also desirable to keep the buying behavior of a user secret from the coupon issuer. It is worth noting that user privacy is possible in the electronic world since we are considering electronic goods, so there is no physical reduction of the goods from the shop's 'warehouse', while in the physical world, the shop can always trace the number of each good to find out the item redeemed by the user.

It is worth noticing that k-time anonymous authentication schemes [TFS04, Ngu06a] have been proposed independently for applications that need to restrict

the number of times that users can access to a service. The difference is that a trusted group manger is involved in these k-time anonymous schemes to achieve traceability against malicious users. In our e-coupon system, we only require the system parameters to be generated from a trusted source. The proposed e-coupon system can trace the dishonest user without involvement of a trusted third party.

Chapter 3

Preliminaries

We introduce some mathematical background about cryptography and some cryptographic primitives that will be used throughout this thesis.

3.1 Mathematical Background

3.1.1 Finite Field

Definition 3.1. *A field with finitely many elements is called a finite field (Galois field). We denote a finite field with n elements by $\mathbb{F}_n$.*

3.1.2 Group

Definition 3.2. *Let G be a set and suppose that $\circ$ is a binary operation on G. We say the pair $(G, \circ)$ is a group if it has the following properties.*

- ***Associativity***. *The operation $\circ$ is associative; That is, $(g \circ h) \circ k = g \circ (h \circ k)$ for all $g, h, k \in G$.*
- ***identity***. *There exists an identity for $\circ$. That is, there exists $e \in G$ such that $g \circ e = e \circ g$ for all $g \in G$.*
- ***Invertibility***. *There exist inverses for $\circ$. That is, for each $g \in G$, there exists $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = e$.*
- ***Closure***. *We say that law of closure holds for $\circ$. That is, when $\circ$ acts on two elements of G the results is also in G.*

If G is finite and has n elements, then we call n the order of G and we write $o(G) = n$. If G is infinite we say that G has infinite order and we write $o(G) = \infty$.

3.1.3 Abelian Group

Definition 3.3. *We say a group $(G, \circ)$ is an Abelian group if it has the following property.*

- ***Commutativity***. *The operation $\circ$ is commutative if $g \circ h = h \circ g$ for all $g, h \in G$.*

3.1.4 Cyclic group

Definition 3.4. *An Abelian group $(G, \circ)$ is cyclic if there exists an element $g \in G$, for any element $h \in G$, there exists $n \in \mathbb{N}$ such that $h = g^n$. We call g a generator of G.*

When we mention a group $(G, \circ)$, we usually omit the binary operation $\circ$ for simplicity. If the order of a group is a prime number, we call it a prime order group. It is worth noticing that all groups of prime order are cyclic groups. All group elements could be generated by a non-identity element in the group. In other words, all the elements except the identity are generators of the prime order group.

3.1.5 Elliptic Curve

Elliptic curve was suggested by Miller [Mil85] for constructing public-key cryptographic systems.

Definition 3.5. *An elliptic curve is a plane curve over a finite field which consists of the points satisfying the equation $y^2 = x^3 + ax + b$, along with a distinguished point at infinity, denoted by ∞.*

- *Constants $a, b \in \mathbb{R}$ satisfying the discriminant $\triangle = -4a^3 - 27b^2 \neq 0$.*
- *A non-singular elliptic curve is the set E of solutions $(x, y) \in \mathbb{R} \times \mathbb{R}$ to the equation $y^2 = x^3 + ax + b$ along with a point O, referred to as the point at infinity.*

3.1.6 Bilinear Pairing

Definition 3.6. *Let G_1, G_2 be additive groups of prime order p and let G_3 be a multiplicative group of prime order p. $e : G_1 \times G_2 \to G_3$ is a bilinear mapping having the following properties.*

- ***Bilinearity**: $e(aP, bR) = e(P, R)^{ab} = e(bP, aR)$ for any $P \in G_1, R \in G_2$ and $a, b \in \mathbb{R}$.*
- ***Non-degeneracy**: There exist $P \in G_1, R \in G_2$ such that $e(P, R) \neq 1_{G_3}$.*
- ***Efficiency**: $e(P, R)$ can be efficiently calculated for all $P \in G_1, R \in G_2$.*

3.1.7 Lagrange Interpolation

Given t points $q(1), q(2), ..., q(t)$ on a $t - 1$ polynomial q, one could use Lagrange interpolation [Sha79] to compute $q(i)$ for any $i \in \mathbb{Z}_p$ through

$$q(i) \quad = \quad \sum_{j=1}^{t} q(j)\Delta_{j,s}(i).$$

The Lagrange coefficient $\Delta_{j,S}(i)$ of $q(j)$ in the computation of $q(i)$ can be computed as

$$\Delta_{j,S}(i) = \prod_{1 \leq \pi \leq t, \pi \neq j} \frac{i - \pi}{j - \pi}.$$

3.2 Complexity Assumptions

We present the complexity assumptions used in this thesis.

3.2.1 Discrete Logarithm Assumption

Definition 3.7. ***Discrete Logarithm Assumption*** *[McC90]: Let $\mathbb{G}$ be a cyclic group of order p with a generator g, the discrete logarithm problem (DL) states that given $h \in \mathbb{G}$, compute $r \in \mathbb{Z}_p$ such that $h = g^r$.*

Define the success probability of a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ in solving the DL problem as:

$$Adv^{DL}_{\mathcal{A},\mathbb{G}}(\kappa) = \Pr(\mathsf{log}_g h \leftarrow g, h \in \mathbb{G}).$$

Where κ is the security parameter. The discrete logarithm assumption states that for all PPT algorithm $\mathcal{A}$, $Adv^{DL}_{\mathcal{A},\mathbb{G}}(\mathsf{log}_g h)(\kappa)$ is negligible in κ.

3.2.2 Computational Diffie-Hellman Assumption

Definition 3.8. ***Computational Diffie-Hellman (CDH) Problem*** *[DH76]: Let $\mathbb{G}$ be a cyclic group of prime order q with a generator g. Given $g, g^a, g^b \in \mathbb{G}$ for some random $a, b \in \mathbb{Z}_q$, compute $g^{ab} \in \mathbb{G}$. Define the success probability of a polynomial algorithm $\mathcal{A}$ in solving the CDH problem as:*

$$Succ^{CDH}_{\mathcal{A},\mathbb{G}}(\kappa) = \Pr[\mathcal{A}(g, g^a, g^b) = g^{ab} : a, b \in_R \mathbb{Z}_q].$$

Where $\kappa = \mathsf{log}(q)$ is the security parameter. The CDH assumption states that for any polynomial algorithm adversary $\mathcal{A}$, $Succ^{CDH}_{\mathcal{A},\mathbb{G}}(\kappa)$ is negligible in κ.

3.2.3 Decisional Diffie-Hellman Assumption

Definition 3.9. ***Decisional Diffie-Hellman (DDH) Problem*** *[Bon98]: Given a cyclic group $\mathbb{G}$ of prime order q with a generator g, the DDH problem states that, given $g, g^a, g^b, Z \in G_q$ for some unknown $a, b \in \mathbb{Z}_q$ and a random generator g, decide whether $Z = g^{ab}$. Define the success probability of a polynomial algorithm $\mathcal{A}$ in*

solving the DDH problem as:

$$Succ^{DDH}_{\mathcal{A},G_q}(\kappa) = |\Pr[\mathcal{A}(G_q, g, g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}(G_q, g, g^a, g^b, Z) = 1]|.$$

where $\kappa = \mathsf{log}(q)$ is the security parameter, the DDH assumption states that, for any polynomial algorithm $\mathcal{A}$, $Succ^{DDH}_{\mathcal{A},G_q}(\kappa)$ is negligible in κ.

3.2.4 One-More Discrete Logarithm Assumption

Definition 3.10. ***One More Discrete Logarithm Problem (OMDL)*** *[BNPS03]: Given a cyclic group G_q of order q and g is a generator of G_q, Let $DLog_g(\cdot)$ be the discrete logarithm oracle that takes an element in G_q and returns the discrete logarithm in base g. Let $C(\cdot)$ be a challenge oracle which takes no input and returns a random element in G_q. Let $W_1, W_2, \ldots, W_t$ denote the challenges returned by $C(\cdot)$, we say an OMDL adversary $\mathcal{A}$ wins if $\mathcal{A}$ can output a sequence of $w_1, w_2, \ldots, w_t$ $\in \mathcal{Z}_q$ satisfying $g^{w_i} = W_i$ and the number of queries q_C make by $\mathcal{A}$ to the discrete logarithm oracle $DLog(\cdot)$ is less than t.*
Define the success probability of a polynomial algorithm $\mathcal{A}$ in solving the OMDL problem as:

$$Succ^{OMDL}_{\mathcal{A},G_q}(\kappa) = \Pr[w_1, w_2, \ldots, w_t \leftarrow \mathcal{A}_{DLog(\cdot), q_C < t}(g, (W_1, W_2, \ldots, W_t \leftarrow C(\cdot)))]$$

the OMDL assumption is that, for any polynomial algorithm $\mathcal{A}$, $Succ^{OMDL}_{\mathcal{A},G_q}(\kappa)$ is negligible in κ.

3.2.5 One More Diffie-Hellman Assumption

Definition 3.11. ***One More Diffie-Hellman (OMDH) Assumption*** *[NP05]: Given a cyclic group G_q of prime order q and g is a generator of G, let $DH(\cdot)$ be the Diffie-Hellman oracle that takes $X = g^x, Y = g^y \in G_q$ for some $x, y \in \mathbb{Z}_q$ and returns the Diffie-Hellman value $Z = g^{xy}$. Let $C(\cdot)$ be a challenge oracle that takes no input and returns a random element in G_q. Let $Y_1, Y_2, \ldots, Y_t$ denote the challenges returned by $C(\cdot)$, we say an OMDH adversary $\mathcal{A}$ wins if $\mathcal{A}$ can output the sequence of Diffie-Hellman values $Z_1, Z_2, \ldots, Z_t$ of all DHP instances with input X, Y_i, $i = 1, 2, \ldots, t$ and the number of queries q_{dh} made by $\mathcal{A}$ to the Diffie-Hellman oracle $DH(\cdot)$ is less than t. Define the success probability of a polynomial algorithm $\mathcal{A}$ in solving the OMDH problem as:*

$$Succ^{OMDH}_{\mathcal{A},G_q}(\kappa) = \Pr[Z_1, Z_2, \ldots, Z_t \leftarrow \mathcal{A}_{DH(\cdot), q_{dh} < t}(X, (Y_1, Y_2, \ldots, Y_t \leftarrow C(\cdot)))]$$

the OMDH assumption states that, for any polynomial algorithm $\mathcal{A}$, $Succ^{OMDH}_{\mathcal{A},G_q}(\kappa)$ is negligible in κ.

3.2.6 Knowledge of Exponent Assumption

Definition 3.12. ***Knowledge of Exponent Assumption*** *[BP04]: Given a cyclic group G_q of order q, for any adversary $\mathcal{A}$ that takes input q, g, g^a and returns (C, Y) with $Y = C^a$, there exists an "extractor" $\bar{\mathcal{A}}$, which given the same inputs as $\mathcal{A}$ returns c such that $g^c = C$.*

3.3 Cryptographic Primitives

We introduce several basic cryptographic primitives in this section.

3.3.1 Public Key Encryption

The goal of public key encryption (PKE) is to ensure the stored or transmitting data be inaccessible to unauthorised parties. In PKE, all the users possess unique public and private key pairs which are binding to their identities through a trusted public key infrastructure (PKI). To send a message, the sender encrypts the message with the recipient's public key. In this way, only the user with knowledge of the corresponding private key could recover the message.

A public key encryption scheme consists a tuple of PPT algorithms as follows:

- **KeyGen**: On input of a security parameter κ, the key generation algorithm outputs a private and public key pair (sk, pk) for a user.
- **Encryption**: On input of a message m and the recipient's public key pk, the encryption algorithm outputs a ciphertext c of m. The encryption algorithm could either be a probabilistic or deterministic algorithm.
- **Decryption**: On input of the recipient's private key and a ciphertext c, the deterministic decryption algorithm outputs the plaintext m of c.

The security goals of encryption schemes is to achieve indistinguishability [GM84] and non-malleability [DDN91]. The notation of indistinguishability (IND) is formalized by Goldwasser and Micali [GM84] to capture the adversary's inability in obtaining any information about the plaintext given knowledge of the corresponding ciphertext. While non-malleability (NM) is proposed by Dwork and Naor [DDN91] in modelling the adversary's inability when given a challenge ciphertext, to output another ciphertext such that the corresponding plaintexts are meaningfully related.

In cryptanalysis of cryptographic schemes, the capabilities of different adversaries are modelled by their abilities in querying different oracles. In public key encryption, three different types of adversaries are considered, namely, chosen-plaintext attack (CPA), non-adaptive chosen-ciphertext attack (CCA1) and chosen-ciphertext attack (CCA2). A CPA adversary has access to the encryption oracle and can query any plaintext of his choice to obtain the corresponding ciphertext. It is worth noticing that the encryption oracle could be modelled by giving the public key suffices to the adversary in public-key setting. Besides the public keys, a CCA1 adversary [NY90] has access to a decryption oracle, to which he could query any ciphertext of his choice and obtain the corresponding plaintext. In CCA1, the adversary has access to the decryption oracle before his being given the challenge ciphertext. In contrast, a CCA2 adversary [RS91] can query the decryption oracle after he obtains the challenge ciphertext. The only restriction on a CCA2 adversary is that he should not send the challenge ciphertext to the decryption oracle.Therefore, the security of public key encryption schemes could be evaluated by six notions of security, namely, IND-CPA, IND-CCA1, IND-CCA2, NM-CPA, NM-CCA1, NM-CCA2 [BDPR98].

3.3.2 Digital Signature

Digital signature schemes provide vital secure properties like integrity, authentication and non-reputation, where integrity ensures that the data has not been modified during transmission, authentication convinces that the data is indeed from a claimed user while non-reputation implies the user can not deny at a later time his operation on the data.

A digital signature scheme consists of a tuple of PPT algorithms as follows.

- **KeyGen**: On input of a security parameter κ, the key generation algorithm outputs a private and public key pair (sk, pk) for a user.
- **Sign**: On input of a message m and the user's private key sk, the sign algorithm outputs a signature σ of m. The sign algorithm could either be a probabilistic or deterministic algorithm.
- **Verity**: On input of the user's public key, a signature σ and m, the verify algorithm outputs '1' (accept) or '0' (reject).

We introduce several attacks in digital signature schemes.

- Key-only attack. In this attack, the attacker only has access to the signer's public key.
- Known message attack. The adversary is given access to signatures for a list of messages but these messages are not chosen by him.

- Chosen message attack. In this attack, the adversary can choose a set of messages and observe the corresponding signatures. However, the whole message list is constructed before the adversary sees any signature.

- Adaptive chosen message attack. The adversary can observe the signature of any message of his choice. Besides, he can requests signatures of messages which depend on previous obtained signatures [GMR88].

We describe some security notions for digital signatures:

- Existential forgery. An adversary could forge a signature for at least one message, while this message does not have to be of his choice (The adversary has no control over the messages that he could obtain the signatures).

- Selective forgery. An adversary could forge a signature on a pre-chosen message.

- Universal forgery. The adversary can generate an acceptable signature on any message without having the secret key.

- Total break. The adversary could calculate the signing key and totally break the system.

In practice, EUF-CMA is the most common security notion for DS schemes. We say a DS scheme is EUF-CMA secure if we can prove the DS scheme is existential unforgeable under the adaptive chosen-message attack.

3.3.3 Proxy Signature

Proxy signature is a variant of digital signature that enables one signer named proxy signer to generate signatures on behalf of another signer named original signer, while the public could be convinced that a proxy signature is generated by a proxy signer with proper delegation form an original signer.

A proxy signature scheme consists of a tuple of algorithms as follows.

- **Setup**: This algorithm takes the security parameter κ as input and returns the public parameters $params$.

- **KeyGen**: The Key Generation algorithm takes the system parameters $params$ as input and outputs a user key pair (pk, sk). Let (sk_p, pk_p) and (sk_o, pk_o) denote the key pairs of the proxy signer and original signer generated through key generation algorithm.

- **DskGen**: The delegation signing key generation algorithm takes the private key sk_o of the original signer, public key pk_p of the proxy signer and a warrant w

including certain delegation information as input and outputs a delegation signing key dsk for the proxy signer.

- **PskGen**: This algorithm takes the delegation signing key dsk from the original signer and the private key sk_p of the proxy signer and outputs a proxy signing key psk.

- **ProSig**: The proxy signing algorithm takes the proxy signing key psk and a message m as input, and outputs a proxy signature σ.

- **ProVer**: The proxy signature verification algorithm takes the public keys pk_o and pk_p, a warrant w, a message m, and a proxy signature σ as input, and outputs either '1' or '0'.

There have been some work on defining security models of proxy signature [MOY04, HSMW06, SMP08, BPW12]. Duo to [HSMW06], three different types of adversaries should be taken into consideration in proxy signature.

- Outsider adversary. An outsider adversary only has access to the pubic keys of the original signer and proxy signer.

- Malicious original signer. A malicious original signer has access to the public keys of the proxy signers in addition to the private key of the original signer.

- Malicious proxy signer. A malicious proxy signer has access to the public keys of the original signers and the private key of the proxy signer.

The security goal of proxy signature is the same as digital signature. The security proof of a proxy signature scheme is to show the proxy signature scheme is unforgeable with the above three type of adversaries.

3.3.4 Oblivious Transfer

Oblivious transfer protocol is an interactive protocol running between a sender and a receiver such that the sender sends some information to the receiver while the result remains oblivious to the sender. There are two participants in an OT system, namely, a sender S and a receiver R. S possesses a set of messages $\{m_1, m_2, \ldots, m_n\}$ and R makes a set of choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ such that $\sigma_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq k$. To be specific, an OT scheme can achieve the following properties:

1. The receiver can only obtain a fix number of messages $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ from the message set $\{m_1, m_2, \ldots, m_n\}$ held by the sender where $\sigma_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq k$. The receiver's choice is hidden from the sender.

2. The receiver cannot learn anything on message m_i such that $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ for $1 \leq i \leq n$.

An OT scheme is essentially an interactive protocol consisting of a tuple of PPT algorithms as follows.

- **Setup**: Taking as input of a security parameter κ, the setup algorithm outputs the system public parameters $params$.
- **KeyGen**: Taking as input of the public parameter $params$, the key generation algorithm outputs a retrievable key pair (pk, sk) for a user.
- **Commitment** (optional): Taking as input of the system parameters $params$, the messages $m_1, m_2, \ldots, m_n$, the commitment algorithm outputs a set of ciphertext $c_1, c_2, \ldots, c_n$.
- **Request**: Taking as input of the intended indexes σ, this algorithms outputs the commitment of the user's choice A_σ.
- **Response**: Taking as input of the commitment A_σ from the receiver, the secret of the sender, the output of the algorithm is response of the sender.
- **Extract**: Taking as input of the response D_σ from the sender, the cipertext c_α and the system parameters $params$, output the message m_σ of the receiver's choice.

There have been some work [NP05, CNS07, KN09] on modelling security of OT schemes. The half-simulation model, introduced by Naor and Pinkas has bee applied widely in evaluating the security of OT schemes. In this model, the security of the sender and receiver is considered separately. The receiver's privacy achieves computational indistinguishability which means one choice σ of the receiver is indistinguishable from another choice σ' from the view of the sender. While the sender security achieves a strong notion named simulatable security, which means any malicious receiver R in the real world, we could build an ideal game with the help of a trusted third party in which there is receiver R' such that the outputs of R and R' are indistinguishable.

3.3.5 Proof of Knowledge

A proof of knowledge system is an interactive proof system that allows one party, usually called prover P, to convince another party, called verifier V, that the prover knows some facts. Let x be an instance of a language L in NP and $W(x)$ be the set of witnesses for x that will be accepted in the proof. Define the relation $R = \{(x, w) : x \in L, w \in W(x)\}$. Let $\phi : \{0,1\}^* \to [0,1]$ be the knowledge error

function. A proof of knowledge system for the relation R should satisfy the following properties [BG92].

1. **Completeness**: The prover P who knows a witness w for x can convince the verifier V of knowledge x with overwhelming probability.

$$\Pr(P(x, w) \leftrightarrow V(x) \rightarrow 1) = 1.$$

2. **Proof of knowledge**: If there exists a prover $\hat{P}$ that has probability ϵ in convincing V, then there exists a knowledge extractor E which given oracle access to $\hat{P}$ can extract a witness of x with probability at least $\epsilon - \phi(x)$, i.e.,

$$\Pr(w \leftarrow E^{\hat{P}(x)}(x)) \geq \Pr(\hat{P}(x) \leftrightarrow V(x) \rightarrow 1) - \phi(x).$$

3.3.6 Hash Functions

Hash functions have been applied widely in constructing digital signature, message authentication codes and key agreement/distribution protocols. A hash function provides a short 'fingerprint' as a function of the data given to it. To be specific, a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$ has the following properties.

- Given $x \in \mathbb{X}$, it is efficient to compute $H(x)$.
- It is infeasible to find two inputs $x, x' \in \mathbb{X}$ such that $H(x) = H(x')$.

In order to the primary properties of hash functions, cryptographic hash functions have the following additional properties.

- **Pre-image resistance**: Given a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$ and $y \in \mathbb{Y}$, there is no efficient mechanism to find $x \in \mathbb{X}$ such that $H(x) = y$.
- **Second pre-image resistance**:Given a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$ and $x \in \mathbb{X}$, there is no efficient mechanism to find $x' \in \mathbb{X}$ such that $x \neq x'$ and $H(x) = H(x')$.
- **Collision resistance**:Given a hash function $H : \mathbb{X} \rightarrow \mathbb{Y}$, there is no efficient mechanism to find $x, x' \in \mathbb{X}$ such that $x \neq x'$ and $H(x) = H(x')$.

Hash functions with pre-image resistance and second pre-image resistance are referred to as One-Way Hash Functions (OWHF). Those with second pre-image resistance and collision resistance are referred to as Collision Resistance Hash Functions (CRHF) [Pre94].

Chapter 4

k-Time Proxy Signature

Proxy signature, which allows an original signer to delegate his/her signing right to another party (or proxy signer), is very useful in many applications. Conventional proxy signature only allows the original signer to specify in the warrant the validity time period of the delegation but not the number of proxy signatures the proxy signer can generate. To address this problem, we provide a formal treatment for k-time proxy signature in this chapter. Such a scheme allows a designated proxy signer to produce only a fixed number of proxy signatures on behalf of the original signer. We provide the formal definitions and adversary models for k-time proxy signature, and propose an efficient construction which is provably secure against different types of adversaries. The original scheme was presented in *ProSec 2013*.

4.1 Introduction

One of the key issues in proxy signature is to ensure that a proxy signer will not misuse the signing right obtained from an original signer. In the seminal work by Mambo et al. [MUO96], a validity period is specified in a warrant in order to restrict the signing capability of a proxy signer. This approach has been used in almost all the following works on proxy signature. However, if the proxy signer is malicious, even in a very short time, the malicious proxy signer can still produce as many proxy signatures as he/she wishes. To address this problem, we provide a formal and comprehensive treatment for k-time proxy signature where the proxy signer can only generate a fixed number of proxy signatures on behalf of the original signer.

There have been a number of works (e.g., [BTT03, HKLL03, PWX04, KYC10]) on restricting the signing capability of a signer in normal digital signature schemes. In [HKLL03], Hwang et al. proposed a multiple-time digital signature scheme, which gives an upper bound on the number of signatures a signer can produce. Shortly after that, Pieprzyk et al. [PWX04] proposed a more general multiple-time signature scheme based on one-way functions and cover-free families. Kim et al. [KYC10] then extended multiple-time signature to a new primitive named metered signature, which allows a signer to produce a fixed number of signatures in a designated time period.

However, a formal and complete treatment for multi-time (or k-time) proxy signature is still missing. In [MH05], Mehta and Harn proposed a one-time proxy signature scheme, which is less useful than a more general k-time proxy signature

scheme. There is a multi-time proxy signature scheme presented in [CKK03], however, no formal security model or proof has been provided. In [HC09], Hong and Chen presented a multiple-time proxy signature scheme based on a binary hash tree. However, their security analysis is incomplete since it does not cover all the possible attacks against a multiple-time proxy signature scheme.

Our Contributions. In this chapter, we provide a formal and complete treatment for multi-time (or k-time) proxy signature schemes. We first provide a formal security model for such schemes. In our model, we will consider three types of adversaries, namely outsiders, proxy signer, and original signer. Our model aims to capture the exact security goal of a k-time proxy signature scheme, that is only a proxy signer, who has been delegated the signing right from an original signer, can produce *at most* k valid proxy signatures. We then propose a new k-time proxy signature scheme based on the Schnorr signature scheme and verifiable secret sharing. In our scheme, the original signer can specify in the warrant the number of proxy signatures a proxy signer can produce. If the proxy signer produces more than predetermined number of proxy signatures, his/her private key can be computed by the public. That means the original signer does not need to monitor the behavior of the proxy signer. It is worth noting that such a feature is not supported in Hong and Chen's scheme [HC09]. In their scheme, the proxy signer's private key can only be computed by the original signer rather than by any third party verifier when the proxy signer misbehaves.

Organization of This Chapter. The rest of this chapter is organized as follows. We introduce the definition of k-time proxy signature in Section 4.2. A formal security model for k-time proxy signature is presented in Section 4.3. We then give our new proxy signature scheme in Section 4.4 and prove its security in Section 4.5. This chapter is concluded in Section 4.6.

4.2 Formal Definition

A k-time (or multi-time) proxy signature scheme consists of a tuple of algorithms $(\mathcal{ST}, \mathcal{KG}, \mathcal{DSK}\ \mathcal{PKG}, \mathcal{PS}, \mathcal{PV}, \mathcal{R})$:

- Setup–($\mathcal{ST}$): This algorithm takes 1^κ as input where κ is a security parameter and returns the public parameters $params$.
- KeyGen–($\mathcal{KG}$): The Key Generation algorithm takes $params$ as input and outputs a user key pair (pk, sk).
- DskGen–($\mathcal{DKG}$): This algorithm takes (sk_o, pk_o, pk_p, m_w) as input and outputs a delegation key dsk. Here m_w denotes a warrant which specifies the predetermined

number of proxy signatures that can be generated by the proxy signer.

- PskGen–($\mathcal{PKG}$): This algorithm takes dsk and sk_p as input and outputs a proxy signing key psk.
- ProSig–($\mathcal{PS}$): The proxy signing algorithm takes the proxy signing key psk and a message m in the message space $\mathbb{M}$ as input, and outputs a proxy signature σ.
- ProVer–($\mathcal{PV}$): The proxy signature verification algorithm takes the public keys pk_o and pk_p, a warrant m_w, a message m, and a proxy signature σ as input, and outputs either 1 or 0.
- Reveal–($\mathcal{R}$): Given pk_o, pk_p, a warrant m_w, and $k+1$ different message and proxy signature pairs, where k is the number specified in the warrant m_w, this algorithm either outputs a private key sk_p of the proxy signer or a special symbol '$\perp$'.

Correctness. We require that for any message space $\mathbb{M} \subseteq \{0,1\}^*$ and any security parameter $\kappa \in \mathbb{N}$, if $params \leftarrow \mathcal{ST}(1^\kappa)$, $(sk_o, pk_o) \leftarrow \mathcal{KG}(params)$, $(sk_p, pk_p) \leftarrow \mathcal{KG}(params)$, $dsk \leftarrow \mathcal{DKG}(sk_o, pk_o, pk_p, m_w)$, $psk \leftarrow \mathcal{PKG}(dsk, sk_p)$, then

$$\mathcal{PV}(pk_o, pk_p, m_w, m, \mathcal{PS}(psk, m)) = 1.$$

4.3 Security Model

In a k-time proxy signature scheme, the security consideration is different from that for the traditional proxy signature [YMS+12] or k-time signature [HKLL03]. According to the definition, the security of a k-time proxy signature should be defined in three aspects, which are summarized below.

1. Type I: the Type I attacker $\mathcal{A}_I$ (an outsider) possesses the public keys of the original signer and the proxy signer, and tries to forge a proxy signature.
2. Type II: the Type II attacker $\mathcal{A}_{II}$ (proxy signer) possesses the public keys of the original signer and the proxy signer. In addition, he also possesses the private key sk_p. We can further divide $\mathcal{A}_{II}$ into $\mathcal{A}_{II1}$ and $\mathcal{A}_{II2}$. $\mathcal{A}_{II1}$ tries to forge a valid proxy signature without obtaining delegation from the original signer, and $\mathcal{A}_{II2}$ has a valid delegation from the original signer and tries to produce more than predetermined number of proxy signatures.
3. Type III: the Type III attacker $\mathcal{A}_{III}$ (the original signer) possesses the public keys of the original signer and the proxy signer. In addition, he has the private key sk_o of the original signer. $\mathcal{A}_{III}$ tries to forge a valid proxy signature without knowing the private key sk_p of the proxy signer.

It is obvious that if a k-time proxy signature scheme is secure against $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$, it is also secure against $\mathcal{A}_I$. So we will only focus on the adversarial models with regards to $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ in the rest of this chapter.

Before we formally define each adversarial model, we first introduce two types of queries that may appear in the models:

- Delegation query: $\mathcal{A}$ can query the delegation oracle $\mathcal{O}_{DKG}(sk_o, pk_o, pk_p, \cdot)$ with any warrant m_w. The corresponding delegation key dsk is then generated and returned to $\mathcal{A}$.
- Proxy signing query: $\mathcal{A}$ can query the proxy signing oracle $\mathcal{O}_{PS}(psk, \cdot)$ with any message m of his choice. A valid proxy signature on m is generated and returned to $\mathcal{A}$.

Type II1 Adversary

We define the adversarial game between a Type II1 adversary $\mathcal{A}_{II1}$ and an simulator $\mathcal{S}$ as follows:

- **Setup:** The Simulator $\mathcal{S}$ runs $\mathcal{ST}$ to generate public parameters $params$.
- **KeyGen** The Simulator $\mathcal{S}$ runs $\mathcal{KG}$ to generate the key pairs of the original signer (sk_o, pk_o) and a proxy signer (sk_p, pk_p). $\mathcal{S}$ sends pk_o, pk_p and sk_p to the adversary $\mathcal{A}_{II1}$.
- **Delegation queries:** $\mathcal{A}_{II1}$ chooses any warrant m_w of his/her choice and queries the delegation oracle $\mathcal{O}_{DKG}$. $\mathcal{S}$ generates the delegation key $dsk \leftarrow \mathcal{DKG}(sk_o, pk_o, pk_p, m_w)$ and returns dsk to $\mathcal{A}_{II1}$.
- **Proxy signing queries:** $\mathcal{A}_{II1}$ chooses a warrant m_w and a message m, and queries the proxy signing oracle $\mathcal{O}_{PS}$. If m_w has appeared in a Delegation Query, a special symbol '$\perp$' is returned to $\mathcal{A}$. Otherwise, $\mathcal{S}$ generates $dsk \leftarrow \mathcal{DKG}(sk_o, pk_o, pk_p, m_w)$, $psk \leftarrow \mathcal{PKG}(dsk, sk_p)$, $\sigma \leftarrow \mathcal{PS}(psk, m)$, and returns σ to $\mathcal{A}_{II1}$.
- Finally, $\mathcal{A}_{II1}$ outputs (m_w^*, m^*, σ^*). We say $\mathcal{A}_{II1}$ wins the game if
  - $\mathcal{PV}(pk_o, pk_p, m_w^*, m^*, \sigma^*) = 1$;
  - $\mathcal{A}_{II1}$ did not make a query to $\mathcal{O}_{DKG}$ on m_w^*;
  - $\mathcal{A}_{II1}$ did not make a query to $\mathcal{O}_{PS}$ on (m_w^*, m^*).

  Define the advantage of a Type II1 adversary as

$$Adv_{\mathcal{A}_{II1}}^{cwcma}(\kappa) = \Pr[A_{II1} \text{ Wins the game}].$$

Definition 4.1. *We say a k-time proxy signature scheme is secure against the Type II1 chosen warrant and chosen message attacks if for any probabilistic polynomial time $\mathcal{A}_{II1}$, $Adv^{cwcma}_{\mathcal{A}_{II1}}(\kappa)$ is negligible in κ.*

Type II2 Adversary

We define the adversarial game between a Type II2 adversary $\mathcal{A}_{II2}$ and an simulator $\mathcal{S}$ as follows:

- **Setup:** The Simulator $\mathcal{S}$ runs $\mathcal{ST}$ to generate public parameters *params*.
- **KeyGen** The Simulator $\mathcal{S}$ runs $\mathcal{KG}$ to generate the key pairs of an original signer (sk_o, pk_o) and a proxy signer (sk_p, pk_p). $\mathcal{S}$ sends pk_o, pk_p and sk_p to the adversary $\mathcal{A}_{II2}$.
- **Delegation queries:** $\mathcal{A}_{II2}$ chooses any warrant m_w of his/her choice and queries the delegation oracle $\mathcal{O}_{DKG}$. $\mathcal{S}$ generates the delegation key $dsk \leftarrow \mathcal{DKG}(sk_o, pk_o, pk_p, m_w)$ and returns dsk to $\mathcal{A}_{II2}$.
- Finally, $\mathcal{A}_{II2}$ outputs a warrant m_w which contains a predetermined number k, and $k+1$ message-signature pairs (m_i, σ_i) $(1 \leq i \leq k+1)$ where $m_i \neq m_j$ for $i \neq j$. We say $\mathcal{A}_{II2}$ wins the game if
  - $\mathcal{PV}(pk_o, pk_p, m_w, m_i, \sigma_i) = 1$ for all $i \in [1, k+1]$;
  - $\mathcal{R}(pk_o, pk_p, m_w, (m_1, \sigma_1), \cdots, (m_{k+1}, \sigma_{k+1})) = \perp$.

  Define the advantage of a Type II2 adversary as

$$Adv^{cwa}_{\mathcal{A}_{II2}}(\kappa) = \Pr[A_{II2} \text{ Wins the game}].$$

Definition 4.2. *We say a k-time proxy signature scheme is secure against the Type II2 chosen warrant attacks if for any probabilistic polynomial time $\mathcal{A}_{II2}$, $Adv^{cwa}_{\mathcal{A}_{II2}}(\kappa)$ is negligible in κ.*

Type III Adversary

The adversarial game between a Type III adversary $\mathcal{A}_{III}$ and an simulator $\mathcal{S}$ is defined as follows:

- **Setup:** The Simulator $\mathcal{S}$ runs $\mathcal{S}$ to generate public parameters *params* and gives *params* to the adversary.
- **KeyGen** The Simulator $\mathcal{S}$ runs $\mathcal{KG}$ to generate the key pairs of the original signer (sk_o, pk_o) and a proxy signer (sk_p, pk_p). $\mathcal{S}$ sends sk_o, pk_o and pk_p to the adversary $\mathcal{A}_{III}$.

- **Proxy signing queries:** $\mathcal{A}_{III}$ queries the proxy signing oracle $\mathcal{O}_{PS}$ by providing a warrant m_w generated according to the scheme, a valid delegation key dsk for m_w, and a message m. $\mathcal{S}$ generates $psk \leftarrow \mathcal{PKG}(dsk, sk_p)$, $\sigma \leftarrow \mathcal{PS}(psk, m)$, and returns σ to $\mathcal{A}_{III}$.

- Finally, $\mathcal{A}_{III}$ outputs (m_w^*, m^*, σ^*). We say $\mathcal{A}_{III}$ wins the game if
  - $\mathcal{PV}(pk_o, pk_p, m_w^*, m^*, \sigma^*) = 1$;
  - For any warrant m_w with a predetermined number k, $\mathcal{A}_{III}$ makes at most k proxy signing queries;
  - $\mathcal{A}_{III}$ did not make a query to $\mathcal{O}_{PS}$ on (m_w^*, m^*).

Define the advantage of a Type III adversary as

$$Adv_{\mathcal{A}_{III}}^{cma}(\kappa) = \Pr[A_{III} \text{ Wins the game}].$$

Definition 4.3. *We say a k-time proxy signature scheme is secure against the Type III chosen message attacks if for any probabilistic polynomial time $\mathcal{A}_{III}$, $Adv_{\mathcal{A}_{III}}^{cma}(\kappa)$ is negligible in κ.*

4.4 Proposed Scheme

In this section, we present a new k-time proxy signature scheme based on the Schnorr signature [Sch89] and secret sharing. Our k-time proxy signature scheme works as follows:

1. $\mathcal{ST}$: given a security parameter $\kappa \in \mathbb{N}$, generate the parameters $params = (G, g, q)$ such that $|q| = \kappa$ and a hash function $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$.

2. $\mathcal{KG}$: randomly choose $x \in \mathbb{Z}_q$ and compute $y = g^x$. Output $(sk, pk) = (x, y)$.

3. $\mathcal{DKG}$: given a warrant $m_w = (k, B = \{b_1, b_2, \cdots, b_k\})$[a], where k is a number selected by the original signer and $b_i = g^{a_i}$ $(1 \leq i \leq k)$ are generated by the proxy signer and sent to the original singer via a secure channel, the original signer first chooses a random number $k_o \in \mathbb{Z}_q$, and then computes $K_o = g^{k_o}$, $\sigma_o = sk_o \cdot h(m_w \| K_o) + k_o \bmod q$. The original signer then sets $dsk = (K_o, \sigma_o)$ as the delegation key for m_w.

4. $\mathcal{PKG}$: given a delegation key $dsk = (K_o, \sigma_o)$ for a warrant m_w, the proxy signer computes $S_p = \sigma_o + sk_p \bmod q$ and outputs the proxy signing key $psk = (K_o, S_p, sk_p)$.

[a] It is worth noting that we can put additional information, such as the validity time period and the type of message the proxy signer is allowed to sign, in the warrant.

5. $\mathcal{PS}$: given a message m to be signed, and a proxy signing key $psk = (K_o, S_p, sk_p)$, the proxy signer chooses a random number $k_p \in \mathbb{Z}_q$, and computes $K_p = g^{k_p}$ and $\sigma_p = S_p \cdot h(h(m_w \| K_o) \| m \| K_p) + k_p \bmod q$. The proxy signer also computes $f(\omega) = sk_p + a_1\omega + a_2\omega^2 + ... + a_k\omega^k \bmod q$ where $\omega = h(m_w, m, \sigma_p)$. The proxy signature is $\sigma = (K_o, K_p, \sigma_p, f(\omega))$.

6. $\mathcal{PV}$: given public keys pk_o and pk_p, a warrant $m_w = (k, B = \{b_1, b_2, \cdots, b_k\})$, a message m and a proxy signature $\sigma = (K_o, K_p, \sigma_p, f(\omega))$, the verifier checks if the following equation holds

   - $g^{\sigma_p} = K_p \cdot (pk_p \cdot K_o \cdot pk_o^{h(m_w \| K_o)})^{h(h(m_w \| K_o) \| m \| K_p)}$;
   - $g^{f(\omega)} = pk_p \cdot b_1^{\omega} \cdot b_2^{\omega^2} \cdots b_k^{\omega^k}$.

   If both equations hold, output 1; otherwise, output 0.

7. $\mathcal{R}$: given $pk_o, pk_p, m_w = (k, B = \{b_1, b_2, \cdots, b_k\})$, and $k+1$ message signature pairs (m_i, σ_i), solve the following equations

$$\begin{aligned} f(\omega_1) &= sk_p + a_1\omega_1 + a_2\omega_1^2 + ... + a_k\omega_1^k \\ f(\omega_2) &= sk_p + a_1\omega_2 + a_2\omega_2^2 + ... + a_k\omega_2^k \\ &\cdots \\ f(\omega_{k+1}) &= sk_p + a_1\omega_{k+1} + a_2\omega_{k+1}^2 + ... + a_k\omega_{k+1}^k \end{aligned}$$

   for variables $(sk_p, a_1, \cdots, a_k)$. If a solution is found, output sk_p, otherwise, output '$\perp$'.

The correctness of the scheme can be verified as follows

$$\begin{aligned} g^{\sigma_p} &= g^{S_p \cdot h(h(m_w \| K_o) \| m \| K_p) + k_p} \\ &= (g^{\sigma_o + sk_p})^{h(h(m_w \| K_o) \| m \| K_p)} \cdot g^{k_p} \\ &= (g^{sk_o \cdot h(m_w \| K_o) + k_o} \cdot g^{sk_p})^{h(h(m_w \| K_o) \| m \| K_p)} \cdot K_p \\ &= (pk_o^{h(m_w \| K_o)} \cdot K_o \cdot pk_p)^{h(h(m_w \| K_o) \| m \| K_p)} \cdot K_p \\ g^{f(\omega)} &= g^{sk_p + a_1\omega + a_2\omega^2 + ... + a_k\omega^k} \\ &= pk_p \cdot (g^{a_1})^{\omega} \cdot (g^{a_2})^{\omega^2} \cdots (g^{a_k})^{\omega^k} \\ &= pk_p \cdot b_1^{\omega} \cdot b_2^{\omega^2} \cdots b_k^{\omega^k} \end{aligned}$$

4.5 Security Analysis

In this section we analyse the security of the above k-time proxy signature scheme against $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ adversaries.

Theorem 4.1. *The proposed k-time proxy signature scheme is secure against the Type II1 chosen warrant and chosen message attacks if the Discrete Logarithm Problem is hard.*

Proof. The proof is by contradiction. Given an adversary $\mathcal{A}_{II1}$ that can win the Type $II1$ game, we construct another algorithm $\mathcal{B}$ that can solve the DLP.

Given $(g, y^* = g^{x^*})$ for some unknown $x^* \in \mathbb{Z}_q$, $\mathcal{B}$ simulates the Type II1 game for $\mathcal{A}_{II1}$ as follows. $\mathcal{B}$ sets the original signer's public key as $pk_o = y^*$ and maintains a H-table to record all the hash queries and the corresponding answers.

Hash queries: For each hash query with an input message msg, $\mathcal{B}$ first checks the H-table:

- If there exists an item (msg, h) in the H-table,where msg refers to the messages queried before, $\mathcal{B}$ returns h as the answer to $\mathcal{A}_{II1}$.
- Otherwise, $\mathcal{B}$ chooses a random $h \in \mathbb{Z}_q$, sends h to $\mathcal{A}_{II1}$ as the answer for the hash query, and adds (msg, h) into the H-table.

Delegation queries: When $\mathcal{A}_{II1}$ makes a delegation query on a warrant $m_w = (k, B = (b_1, b_2, \cdots, b_k))$, $\mathcal{B}$ answers the query as follows.

- Choose randomly $h_o, \sigma_o \in Z_q$, compute $K_o = g^{\sigma_o}/pk_o^{h_o}$, and set $h(m_w \| K_o) = h_o$ by adding $(m_w \| K_o, h_o)$ into the H-table.
- Return (K_o, σ_o) as the delegation key to $\mathcal{A}_{II1}$.

Proxy signing queries: When $\mathcal{A}_{II1}$ makes a proxy signing query on a warrant $m_w = (k, B = (b_1, b_2, \cdots, b_k))$, and a message m, $\mathcal{B}$ responds the query as follows:

- Generate a delegation key $dsk = (K_o, \sigma_o)$ for the warrant m_w by applying the same approach as described in answering delegation queries.
- Use the derived dsk and sk_p to produce the proxy signing key psk by running the $\mathcal{PKG}$ algorithm, and then use psk to generate the proxy signature for message m by running the $\mathcal{PS}$ algorithm.

Assume $\mathcal{A}_{II1}$ can forge a valid proxy signature $\sigma^* = (K_o^*, K_p^*, \sigma_p^*, f(\omega^*))$ for a warrant m_w^* and a message m^* such that

$$g^{\sigma_p^*} = K_p^* \cdot (pk_p \cdot K_o^* \cdot pk_o^{h(m_w^* \| K_o^*)})^{h(h(m_w^* \| K_o^*) \| m^* \| K_p^*)}.$$

Then according to the Forking Lemma [PS96], by rewinding the adversary and providing a new hash value for $h(m_w^* \| K_o^*) \| m^* \| K_p^*$, $\mathcal{B}$ can obtain $S_p^* = \sigma_o^* + sk_p \bmod$

q and $\sigma_o^* = S_p^* - sk_p \bmod q$ which satisfies

$$g^{\sigma_o^*} = K_o^* \cdot pk_o^{h^*}$$

where $h^* = h(m_w^* \| K_o^*)$.

After that, $\mathcal{B}$ repeats the above simulation for $\mathcal{A}_{II1}$ except that a new value $\hat{h}^*$ is chosen as the hash value for $m_w^* \| K_o^*$. Again, due to the Forking Lemma [PS96], $\mathcal{B}$ can obtain a new $\hat{\sigma}_o^*$ which satisfies

$$g^{\hat{\sigma}_o^*} = K_o^* \cdot pk_o^{\hat{h}^*}.$$

$\mathcal{B}$ can then compute $x^* = sk_o = (\sigma_o^* - \hat{\sigma}_o^*)/(h^* - \hat{h}^*)$ and solve the Discrete Logarithm Problem. This completes the proof for Theorem 1.

Theorem 4.2. *The proposed k-time proxy signature scheme is secure against the Type II2 chosen warrant attacks.*

Proof. According to our scheme, if a signature $\sigma = (K_o, K_p, \sigma_p, f(\omega))$ is valid with regards to a warrant $m_w = (k, B = (b_1, b_2, \cdots, b_k))$ and message m , then

$$g^{f(\omega)} = pk_p \cdot {b_1}^{\omega} \cdot {b_2}^{\omega^2} \cdots {b_k}^{\omega^k}.$$

Suppose an adversary $\mathcal{A}_{II2}$ have produced $k+1$ proxy signatures with regards to a warrant m_w and different messages $\{m_1, m_2, \cdots, m_{k+1}\}$, then we have

$$\begin{cases} f(\omega_1) = sk_p + a_1\omega_1 + a_2\omega_1^2 + ... + a_k\omega_1^k \\ f(\omega_2) = sk_p + a_1\omega_2 + a_2\omega_2^2 + ... + a_k\omega_2^k \\ ... \\ f(\omega_{k+1}) = sk_p + a_1\omega_{k+1} + a_2\omega_{k+1}^2 + ... + a_k\omega_{k+1}^k \end{cases}$$

where $\omega_i = h(m_w, m_i, \sigma_{p_i})$ for $1 \leq i \leq k+1$. Since the hash function is modelled as a random oracle, each ω_i is a random element in $\mathbb{Z}_q$. Therefore, with overwhelming probability, the reveal algorithm $\mathcal{R}$ can recover the unique solution $(sk_p, a_1, a_2, \cdots, a_k)$ that satisfies the above equations.

Theorem 4.3. *The proposed k-time proxy signature scheme is secure against the Type III chosen message attacks if the Discrete Logarithm Problem is hard.*

Proof. The proof is similar to the proof for Theorem 1, that is, if there exists an adversary $\mathcal{A}_{III}$ which can win the Type III game, we can construct another algorithm $\mathcal{B}$ which can solve the Discrete Logarithm Problem.

Given $(g, y^* = g^{x^*})$ where $x^* \in \mathbb{Z}_q$ is randomly chosen from $\mathbb{Z}_q$, $\mathcal{B}$ simulates the Type III game for $\mathcal{A}_{III}$ as follows. $\mathcal{B}$ generates sk_o, pk_o and sets the proxy signer's public key as $pk_p = y^*$. $\mathcal{B}$ answers hash queries by maintaining a H-table as in the proof of Theorem 1.

When a new warrant m_w with a predetermined number k is to be created, $\mathcal{B}$ generates the values of $B = (b_1, b_2, \cdots, b_k)$ as follows. $\mathcal{B}$ randomly chooses $\omega_i, s_i \in \mathbb{Z}_q$ for $1 \leq i \leq k$. Then based on the result in [Ped91], $\mathcal{B}$ can calculate b_i $(1 \leq i \leq k) \in G$ that satisfies $g^{s_i} = y^* \cdot \prod_{j=1}^{k} b_j^{\omega_i^j}$ for all $1 \leq i \leq k$. $\mathcal{B}$ saves the values of $\{\omega_i, s_i\}_{1 \leq i \leq k}$ with regards to m_w for later use.

Proxy signing queries: To answer the ℓ-th $(1 \leq \ell \leq k)$ proxy signing query on a warrant m_w, $\mathcal{B}$ first finds out the values of (ω_ℓ, s_ℓ) that have been computed when generating the warrant m_w. $\mathcal{B}$ then computes the proxy signature as follows:

- Randomly choose $\sigma_p, \tau \in \mathbb{Z}_q$;
- Compute $K_p = g^{\sigma_p}/(pk_p \cdot K_o \cdot pk_o^{h(m_w \| K_o)})^{\tau}$;
- Set $h(h(m_w \| K_o) \| m \| K_p) = \tau$;
- Set $h(m_w \| m \| \sigma_p) = \omega_\ell$;
- Return $\sigma = (K_o, K_p, \sigma_p, s_\ell)$.

It is easy to verify that σ can successfully pass the signature verification.

Suppose $\mathcal{A}_{III}$ outputs a forgery $(m_w^*, m^*, \sigma^* = (K_o^*, K_p^*, \sigma_p^*, s^*))$ which satisfies

$$g^{\sigma_p^*} = K_p^* \cdot (y^* \cdot K_o^* \cdot pk_o^{h(m_w^* \| K_o^*)})^{h(h(m_w^* \| K_o^*) \| m^* \| K_p^*)}$$

where $dsk^* = (K_o^*, \sigma_o^*)$ is the delegation key provided by $\mathcal{A}_{III}$ for the warrant m_w^*. According to the Forking Lemma, by rewinding $\mathcal{A}_{III}$ and providing a new hash value of $h(h(m_w^* \| K_o^*) \| m^* \| K_p^*)$, $\mathcal{B}$ can obtain another valid signature $\hat{\sigma}^* = (K_o^*, K_p^*, \hat{\sigma}_p^*, \hat{s}^*))$ for (m_w^*, m^*). Then $\mathcal{B}$ can derive

$$S_p^* = (\sigma_p^* - \hat{\sigma}_p^*)/(h^* - \hat{h^*}) \bmod q$$

where h^* and $\hat{h}^*$ are the hash values for $h(m_w^* \| K_o^*) \| m^* \| K_p^*$ in the two executions. Finally, $\mathcal{B}$ can compute $x^* = S_p^* - \sigma_o^* \bmod q$ and solve the DLP.

4.6 Summary

In this chapter, we presented a formal security model and an efficient construction of k-time proxy signature scheme. Our model has considered different types of

potential adversaries against a k-time proxy signature scheme, and is to date the first complete formal security model for such schemes. We then presented a practical k-time proxy signature scheme based on the Schnorr signature and verifiable secret sharing. One interesting feature of our scheme is that the proxy signer's secret key can be discovered by the public if the proxy signer misbehaves. We also provided formal security proofs to demonstrate that the proposed scheme is provably secure in the proposed security model. We leave the problem of constructing a secure k-time proxy signature scheme without random oracles as our future work.

Chapter 5

Attribute-Based Signing Right Delegation

Attribute-based signature and proxy signature are both very useful in many real-world applications. In this chapter, we present an attribute-based proxy signature scheme benefiting from both proxy signature and attribute-based signature. In the proposed scheme, an original signer, who possesses a set of attributes, can delegate his/her signing right to a designated proxy signer. By verifying the signature, a verifier can be convinced that the signature is generated by the proxy signer who has obtained the delegation from a legitimate signer whose attributes satisfy a predicate. However, the verifier cannot tell from the signature who is the original signer. We provide the formal definition and adversarial models for attribute-based proxy signature, and an efficient scheme that supports threshold predicates. The original scheme was presented in *NSS 2014*.

5.1 Introduction

In this chapter, we are interested in signing right delegation under the attribute-based setting environment. The proposed scheme can be regarded as a variant of attribute-based proxy signature schemes(ABPS). ABPS has many potential applications, for example, attribute-based authentication [MPR11]. Consider a database whose access control is described in a policy such that only users who hold authorised attribute keys can access it. An authorised user can delegate his/her signing rights to another user so that the latter can also access the database and collect information when the former is not available. The delegated signer is called a proxy of the original authorised signer. In our proposed scheme, the verifier can be convinced that a valid proxy signer holds the right delegation from an original signer and therefore can access the database. The attributed based proxy signature can be regarded as a certificate for accessing the database. We noticed that a paper regarding attribute-based proxy signature has recently been proposed in [LMX+13]. However, the adversarial models in [LMX+13] are not properly defined. In addition, the application scenario is different from ours.

Organization of This Chapter. The rest of this chapter is organized as follows. The formal definition and security model of ABPS are presented in Section 5.2 and Section 5.3 separately. We then present our ABPS scheme in Section 5.4 and prove

its security in Section 5.5. This chapter is concluded in Section 5.6.

5.2 Formal Definition

An attribute-based proxy signature scheme is parameterized by a universe of possible attributes $\mathbb{A}$, a warrant space $\mathbb{M}_\omega$, and a message space $\mathbb{M}$. It consists of the following algorithms.

- **ABPS.Setup**: takes a security parameter 1^κ as input and outputs the public parameters $params$ and a master secret key MSK for the central authority.
- **ABPS.KeyGen**: takes $params$ as input and outputs a proxy key pair (pk, sk).
- **ABPS.AttrKeyGen**: takes $(MSK, params, \omega)$ as input where $\omega \subseteq \mathbb{A}$ is the attribute set of a user and outputs an attribute key sk_ω.
- **ABPS.DskGen**: takes $(sk_\omega, m_w \in \mathbb{M}_w, \Upsilon)$ as input, where m_w is a warrant specified by the original signer, Υ is a predicate such that there exists $\omega' \subseteq \omega$ which satisfies $\Upsilon(\omega') = 1$, and outputs a delegation key dsk.
- **ABPS.ProSig**: takes $(dsk, sk, m \in \mathbb{M})$ as input, and outputs a proxy signature σ.
- **ABPS.ProVer**: takes $(\Upsilon, pk, m_w, m, \sigma)$ as input, and outputs 1 ('accept') or 0 ('reject').

Correctness: We require that for any warrant and message spaces $\mathbb{M}_w, \mathbb{M} \subseteq \{0,1\}^*$ and any security parameter $\kappa \in \mathbb{N}$, if

$$(params, MSK) \leftarrow \textbf{ABPS.Setup}(1^\kappa),$$

$$(pk, sk) \leftarrow \textbf{ABPS.KeyGen}(params),$$

$$sk_\omega \leftarrow \textbf{ABPS.AttrenKenGen}(MSK, params, \omega),$$

$$dsk \leftarrow \textbf{ABPS.DskGen}(sk_\omega, m_w, \Upsilon),$$

then

$$\textbf{ABPS.ProVer}(\Upsilon, pk, m_w, m, \textbf{ABPS.ProSig}(dsk, sk, m)) = 1.$$

5.3 Security Model

In an attribute-based proxy signature scheme, the security consideration is different from that for a traditional proxy signature or attribute-based signature. According

to the definition of attribute-based proxy signature, we consider three different types of adversaries:

1. $\mathcal{A}_I$: an outsider attacker who only has the universe of attributes $\mathbb{A}$ and the public key pk_p of the proxy signer and tries to forge a valid proxy signature σ.

2. $\mathcal{A}_{II}$: a malicious proxy signer that possesses the private key sk_p and a valid warrant m_w from the original signer, and tries to forge a valid proxy signature σ for another warrant m_w^*.

3. $\mathcal{A}_{III}$: a malicious original signer that possesses the attribute key sk_ω and the public key pk_p of the proxy signer, and tries to forge a valid proxy signature σ without knowing the private key sk_p of the proxy signer.

It is obvious that if an attribute-based proxy signature scheme is secure under $\mathcal{A}_{II}$ or $\mathcal{A}_{III}$, it is also secure against $\mathcal{A}_I$. Thus we will only focus on the adversarial models with regards to $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ in the rest of this chapter. Before we formally define each adversarial model, we first introduce three types of oracle queries that will appear in the models:

- **Attribute Key Generation Query**: $\mathcal{A}$ can query the attribute key for an attribute set $\omega \subseteq \mathbb{A}$ of his choice to the attribute key generation oracle $\mathcal{O}_{AKG}(\cdot)$. The corresponding attribute key sk_ω is then generated and returned to $\mathcal{A}$.

- **Delegation Query**: $\mathcal{A}$ can query the delegation oracle $\mathcal{O}_{DKG}(sk_\omega, \cdot, \cdot)$ with any warrant m_w and access structure Υ of his choice. The corresponding delegation key dsk is generated and returned to $\mathcal{A}$.

- **Proxy Signing Query**: $\mathcal{A}$ can query the proxy signing oracle $\mathcal{O}_{PS}(dsk, sk_p, \cdot)$ with any message m of his choice. A valid proxy signature on m is generated and returned to $\mathcal{A}$.

We define the selective adversarial game between a malicious proxy signer $\mathcal{A}_{II}$ and a simulator $\mathcal{S}$ as follows:

- **Initial Phase**: $\mathcal{A}_{II}$ chooses and outputs a challenge predicate Υ^* that will be used in forging a proxy signature.

- **ABPS.Setup Phase**: The simulator $\mathcal{S}$ runs **ABPS.Setup** to generate the $params$ and MSK, and sends $params$ to $\mathcal{A}_{II}$.

- **ABPS.KeyGen Phase**: The simulator $\mathcal{S}$ also runs the **ABPS.KeyGen** to generate the key pairs (pk_p, sk_p) of the proxy signer, and sends (pk_p, sk_p) to $\mathcal{A}_{II}$.

- **Attribute Key Generation Queries**: $\mathcal{A}_{II}$ selects an attribute set $\omega \in \mathbb{A}$, the simulator $\mathcal{S}$ runs $sk_\omega \leftarrow$ **ABPS.AttrKeyGen**$(MSK, params, \omega)$ and returns sk_ω to $\mathcal{A}_{II}$.

- **Delegation Queries Phase**: $\mathcal{A}_{II}$ chooses any predicate Υ such that $\Upsilon \neq \Upsilon^*$ and any warrant m_w of his choice and queries the delegation oracle $\mathcal{O}_{DKG}$. $\mathcal{S}$ generates the delegation key $dsk \leftarrow$ **ABPS.DskGen**$(sk_\omega, \Upsilon, m_w)$ and sends dsk to $\mathcal{A}$.

- **Proxy Signing Queries Phase**: $\mathcal{A}_{II}$ chooses a warrant $m_w \in \mathbb{M}_W$ and a message $m \in \mathbb{M}$ and queries the proxy signing oracle $\mathcal{O}_{PS}$. If m_w has appeared in a Delegation Query, a special symbol '$\perp$' is returned to $\mathcal{A}_{II}$. Otherwise, $\mathcal{S}$ generates

$$dsk \leftarrow \textbf{ABPS.DskGen}(sk_\omega, \Upsilon, m_w),$$

$$\sigma \leftarrow \textbf{ABPS.ProSign}(dsk, sk_p, m_w, m)$$

  and returns σ to $\mathcal{A}_{II}$.

- **Forgery Phase**: Finally, $\mathcal{A}$ outputs a proxy signature σ^* on message m^* for a warrant m_w^* and the predicate Υ^*.

We say $\mathcal{A}_{II}$ wins the game if

- **ABPS.ProVer**$(\Upsilon^*, pk_p, m_w^*, m^*, \sigma^*) = 1$;

- (m_w^*, Υ^*) has not been queried to $\mathcal{O}_{DSK}$;

- Attribute sets ω^* satisfying $\Upsilon^*(\omega^*) = 1$ have not been submitted to the attribute key generation oracle $\mathcal{O}_{AKG}$.

Define the advantage of a malicious adversary $\mathcal{A}_{II}$ in winning the game as

$$Adv_{\mathcal{A}_{II}}^{spcwcma}(\kappa) \;=\; \Pr[\mathcal{A}_{II} \text{ Wins the game}].$$

Definition 5.1. *We say an attribute-based proxy signature scheme is secure against the $\mathcal{A}_{II}$ under the selective-predicate and chosen warrant and message attacks if for any probabilistic polynomial time $\mathcal{A}_{II}$, $Adv_{\mathcal{A}_{II}}^{spcwcma}(\kappa)$ is negligible in κ.*

The adversarial game between a malicious original signer $\mathcal{A}_{III}$ and a simulator $\mathcal{S}$ is defined as follows:

- **ABPS.Setup Phase**: The simulator $\mathcal{S}$ runs the **ABPS.Setup** to generate the $params$ and MSK, and sends $params$ and MSK to $\mathcal{A}_{III}$.

- **ABPS.KeyGen Phase**: The simulator generates

$$(pk_p, sk_p) \leftarrow \textbf{ABPS.KeyGen}$$

and sends pk_p to $\mathcal{A}_{III}$.

- **Proxy Signing Queries Phase**: $\mathcal{A}_{III}$ queries the proxy signing oracle $\mathcal{O}_{PS}$ by providing a warrant m_w, a valid delegation key dsk for m_w, and a message m of his choice. The simulator $\mathcal{S}$ generates the proxy signature $\sigma \leftarrow$ **ABPS.ProSign**(dsk, sk_p, m_w, m) and returns σ to $\mathcal{A}_{III}$.

- **Forgery Phase**: Finally, $\mathcal{A}_{III}$ outputs a proxy signature σ^* on message m^* for a warrant m_w^* and predicate Υ^*.

We say $\mathcal{A}_{III}$ wins the game if

- **ABPS.PorVer**$(\Upsilon^*, pk_p, m_w^*, m^*, \sigma^*) = 1$;

- (m_w^*, m^*) has not been queried to $\mathcal{O}_{PS}$;

Define the advantage of a malicious adversary $\mathcal{A}_{III}$ in winning the game as

$$Adv_{\mathcal{A}_{III}}^{cma}(\kappa) \quad = \quad \Pr[\mathcal{A}_{III} \text{ Wins the game}].$$

Definition 5.2. *We say an attribute-based proxy signature scheme is secure against the $\mathcal{A}_{III}$ under chosen message attacks if for any probabilistic polynomial time $\mathcal{A}_{III}$, $Adv_{\mathcal{A}_{III}}^{cma}(\kappa)$ is negligible in κ.*

5.4 Proposed Scheme

In our system, the original signer holds a set of attributes and delegates his signing rights to a proxy signer with a normal public/private key pair.

1. **ABPS.Setup**: First, define the universe of attributes U as elements in $\mathbb{Z}_p$. Let the $d-1$ default set of attributes from $\mathbb{Z}_p$ which has no intersection with U be $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_{d-1}\}$ and let ω^* be another default attribute set with $\omega^* \subseteq U$. Select a random generator $g \in_R \mathbb{G}_1$ and a random number $x \in \mathbb{Z}_p^*$, set $g_1 = g^x$. Pick random elements g_2 and compute $Z = e(g_1, g_2)$. Select a collision resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$. The public parameters are $params = (g, g_1, g_2, Z, H)$. The master secret key is $MSK = x$.

2. **ABPS.KeyGen**: The user selects one random number $x_p \in_R \mathbb{Z}_p^*$ and set the private and public key pair as $(sk_p, pk_p) = (x_p, g^{x_p})$.

3. **ABPS.AttrKeyGen**: To generate a private key for an attribute set ω, proceed as follows:

   - Choose a $d-1$ polynomial q such that $q(0) = x$;
   - Generate a new set of attribute $\hat{\omega} = \omega \cup \Omega$. For each $i \in \hat{\omega}$, choose $r_i \in_R \mathbb{Z}_p$ and compute $d_{i0} = g_2^{q(i)} \cdot H(attr_i)^{r_i}$ and $d_{i1} = g^{r_i}$;
   - The private key $D_i = \{(d_{i0}, d_{i1})\}$, $i \in \hat{\omega}$.

4. **ABPS.DskGen**: Given a warrant m_w, the original signer selects a k-element subset $\omega' \subseteq \omega \cap \omega^*$ and the delegation signing key is generated as follows:

   - The original signer selects a default attribute subset $\Omega' \subseteq \Omega$ with $|\Omega'| = d-k$, chooses $n+d-k$ random values $r_i' \in \mathbb{Z}_p$, where $i \in \omega^* \cup \Omega'$;
   - The original signer chooses a random value $s \in \mathbb{Z}_p$ and computes $\sigma_0 = \prod_{i \in \omega \prime \cup \Omega'} d_{i0}^{\Delta_{i,S}(0)} \prod_{i \in \omega^* \cup \Omega'} H(attr_i)^{r_i'} H(m_w)^s$, $\{\sigma_i = d_{i1}^{\Delta_{i,S}(0)} g^{r_i'}\}_{i \in \omega' \cup \Omega'}$, $\{\sigma_i = g^{r_i'}\}_{\omega^*/\omega'}$, $\sigma_0' = g^s$;
   - The delegation signing key is $dsk = (\sigma_0, \{\sigma_i\}_{i \in \omega^* \cup \Omega'}, \sigma_0')$.

5. **ABPS.ProSign**: Given dsk, sk_p and a message $m \in \{0,1\}^*$. The proxy signature $\sigma_M = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ is generated as follows:

   - Compute $\sigma_{M_1} = \sigma_0 \cdot H(m)^{sk_p}$, $\sigma_{M_2} = \{\sigma_i\}_{i \in \omega^* \cup \Omega'}$, $\sigma_{M_3} = \sigma_0'$.

6. **ABPS.Verification**: Given $\sigma_M = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, m_w and Υ_{k,ω^*}, the verifier first check whether the proxy signer follow the rules specified in the warrant. If no, output reject, otherwise, the verifier checks the following equation:

$$\frac{e(g, \sigma_{M_1})}{\prod_{i \in \omega^* \cup \Omega'} e(H(attr_i), \sigma_i) e(H(m_w), \sigma_{M_3}) e(pk_p, H(m))} \stackrel{?}{=} Z.$$

   If the equation holds, output accept, where it can be assured that the signature is generated form some user possessing k attributes among ω^*, otherwise, output reject.

- **Correctness:** The correctness of the verification is justified by the following

equations:

$$
\begin{aligned}
&\frac{e(g,\sigma_{M_1})}{\prod_{i\in\omega^*\cup\Omega'} e(H(attr_i),\sigma_i)e(H(m_w),\sigma_{M_3})e(pk_p,H(m))} \\
&= \frac{e(g,\sigma_0\cdot H(m)^{sk_p})}{\prod_{i\in\omega^*\cup\Omega'} e(H(attr_i),\sigma_i)e(H(m_w),\sigma_{M_3})e(pk_p,H(m))} \\
&= \frac{e(g,\prod_{i\in\omega\prime\cup\Omega'} d_{i0}^{\Delta_{i,S}(0)}\prod_{i\in\omega^*\cup\Omega'} H(attr_i)^{r'_i}H(m_w)^s\cdot H(m)^{sk_p})}{\prod_{i\in\omega^*\cup\Omega'} e(H(attr_i),\sigma_i)e(H(m_w),\sigma_{M_3})e(pk_p,H(m))} \\
&= \frac{e(g,\prod_{i\in\omega\prime\cup\Omega'} d_{i0}^{\Delta_{i,S}(0)})e(g,\prod_{i\in\omega^*\cup\Omega'} H(attr_i)^{r'_i})e(H(m_w),g^s)e(pk_p,H(m))}{\prod_{i\in\omega^*\cup\Omega'} e(H(attr_i),\sigma_i)e(H(m_w),\sigma_{M_3})e(pk_p,H(m))} \\
&= \frac{e(g,\prod_{i\in\omega\prime\cup\Omega'} d_{i0}^{\Delta_{i,S}(0)})e(g,\prod_{i\in\omega^*\cup\Omega'} H(attr_i)^{r'_i})}{\prod_{i\in\omega^*\cup\Omega'} e(H(attr_i),\sigma_i)} \\
&= \frac{e(g,\prod_{i\in\omega\prime\cup\Omega'} g_2^{q(i)\cdot\Delta_{i,S}(0)})e(g,\prod_{i\in\omega\prime\cup\Omega'} H(attr_i)^{r_i\cdot\Delta_{i,S}(0)+r'_i})e(g,\prod_{i\in\omega^*/\omega'} H(attr_i)^{r'_i})}{e(g,\prod_{i\in\omega\prime\cup\Omega'} H(attr_i)^{r_i\cdot\Delta_{i,S}(0)+r'_i})e(g,\prod_{i\in\omega^*/\omega'} H(attr_i)^{r'_i})} \\
&= e(g,g_2^{q(0)}) \\
&= e(g,g_2^x) \\
&= e(g^x,g_2) \\
&= Z.
\end{aligned}
$$

5.5 Security Analysis

In this section we analyse the security of the above attribute-based proxy signature scheme against $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ adversaries.

Theorem 5.1. *Our attribute-based proxy signature scheme is secure against the $\mathcal{A}_{II}$ chosen warrant and chosen message attacks if the CDH Problem is hard.*

Proof. The proof is by contradiction in the selective predicate security model. Suppose that an adversary $\mathcal{A}_{II}$ has an advantage ϵ in attacking the proposed scheme, then we can build an algorithm $\mathcal{B}$ that use $\mathcal{A}_{II}$ to solve the CDH problem. Let $\mathbb{G}_1$ be a bilinear pairing group of prime order p, $\mathcal{B}$ is given $g, g^\alpha, g^\beta \in \mathbb{G}_1$ which is a random instance of the CDH problem. Its goal is to compute $g^{\alpha\beta}$. Algorithm $\mathcal{B}$ will simulate the challenger and interact with the forger $\mathcal{A}_{II}$ as described below, let's recall the definition of $\mathcal{A}_{II}$, $\mathcal{A}_{II}$ is a malicious proxy signer possessing the private key of the proxy signer. With this in mind, the simulation is as follows:

1. **Initial Phase**: $\mathcal{A}_{II}$ chooses a predicate Υ^*_{k,ω^*} as the challenge predicate.

2. **Setup**: Let the default attribute set Ω be $\{\Omega_1,\ldots,\Omega_{d-1}\}$. $\mathcal{B}$ sets $g_1 = g^\alpha, g_2 = g^\beta$, where g^α, g^β are inputs of the CDH problem. $\mathcal{B}$ sets the public parameters as:

- $\mathcal{B}$ chooses random $x_p \in_R \mathbb{Z}_p^*$ and sets $sk_p = x_p, pk_p = g^{x_p}$.
- $\mathcal{B}$ sends $(G_1, G_2, e, p, g, g_1, g_2, H)$ and (sk_p, pk_p) to $\mathcal{A}_{II}$.

3. **Hash queries**: In order to make the simulation easy to follow, we regard the attribute, warrant and message queries as H_1,H_2 and H_3 queries respectively. Assume $\mathcal{B}$ keeps hash tables T_1, T_2 and T_3 for the queries.

   (a) H_1 **Query**: Assume $\mathcal{A}_{II}$ makes q_{H1} attribute queries, for each query on attribute $attr_i$, $\mathcal{B}$ simulates as follows:

   - If $attr_i$ have existed in T_1, a same value $H(attr_i)$ is returned to $\mathcal{A}_{II}$.
   - Otherwise,
     - If $attr_i \in \omega^* \cup \Omega^*$, $\mathcal{B}$ chooses random $a_i \in \mathbb{Z}_p$ and returns $H(attr_i) = g^{a_i}$ to $\mathcal{A}_{II}$. $\mathcal{B}$ adds $(attr_i, H(attr_i))$ to T_1.
     - If $attr_i \notin \omega^* \cup \Omega^*$, $\mathcal{B}$ chooses random $a_i, b_i \in \mathbb{Z}_p$ and returns $H(attr_i) = g^{-a_i} g^{b_i}$ to $\mathcal{A}_{II}$. $\mathcal{B}$ adds $(attr_i, H(attr_i))$ to T_1.

   (b) H_2 **Query**: Assume $\mathcal{A}_{II}$ makes q_{H2} warrant queries, $\mathcal{B}$ selects a random number $\delta \in (0, q_{H2})$, for each query on warrant m_{w_i}, $\mathcal{B}$ simulates as follows:

   - If m_{wi} have existed in T_2, a same value $H(m_{w_i})$ is returned to $\mathcal{A}_{II}$.
   - Otherwise,
     - If $i \neq \delta$, $\mathcal{B}$ chooses random $a_i', b_i' \in_r \mathbb{Z}_p$ and returns $H(m_{w_i}) = g_1^{b_i'} g^{a_i'}$. $\mathcal{B}$ adds $(m_{w_i}, H(m_{w_i}))$ to T_2.
     - If $i = \delta$, $\mathcal{B}$ chooses random b_i' and returns $H(w_i) = g^{a_i'}$. $\mathcal{B}$ adds $(m_{w_i}, H(m_{w_i}))$ to T_2

   (c) H_3 **Query**: Assume $\mathcal{A}_{II}$ makes q_{H3} message queries, for each query on message m_i, $\mathcal{B}$ simulates as follows:

   - If m_i has existed in T_3, a same value $H(m_i)$ is returned to $\mathcal{A}_{II}$.
   - Otherwise, $\mathcal{B}$ chooses random $r_i \in_R \mathbb{Z}_p$ and returns $H(m_i) = g^{r_i}$. $\mathcal{B}$ adds $(m_i, H(m_i))$ to T_3.

4. **Attribute key extraction queries**: Assume $\mathcal{A}_{II}$ issues an attribute key extraction query on an attribute set ω such that $|\omega^* \cap \omega| < k$. Following the analysis in [LAS+10], we first define three sets Γ, Γ', S in the following manner: $\Gamma = (\omega \cap \omega^*) \cup \Omega^*$ and $\Gamma \subseteq \Gamma' \subseteq S$ with $|\Gamma'| = d - 1$. Let $S = \Gamma' \cup \{0\}$. The simulation on the attribute key D_i is as follows:

   - For $i \in \Gamma'$: $D_i = (g_2^{\tau_i} H(attr_i)^{r_i}, g^{r_i})$, where $\tau_i, r_i \in_R \mathbb{Z}_p$.

- For $i \notin \Gamma'$, D_i could be simulated as:

$$D_i = (g_2^{\frac{\Delta_{0,S}(i)b_i}{a_i} + \sum_{j\in\Gamma'} \Delta_{j,S}(i)q(j)})(g_1^{-a_i} g^{b_i})^{r'_i}, g_2^{\frac{\Delta_{0,S(i)}}{a_i}} g^{r'_i}),$$

where $r'_i \in_R \mathbb{Z}_p$. It is a correct key because it implicitly sets

$$r_i = \frac{\Delta_{j,S}(i)q(j)}{a_i}\beta + r'_i.$$

As we know,

$$q(i) = \sum_{j\in\Gamma'} \Delta_{j,S}(i)q(j) + \Delta_{0,S}(i)q(0),$$

thus we have,

$$g_2^{q(i)} H(attr_i)^{r_i} = g_2^{\frac{\Delta_{0,S}(i)b_i}{a_i} + \sum_{j\in\Gamma'} \Delta_{j,S}(i)q(j)} H(attr_i)^{r'_i}$$

and

$$g^{r_i} = g_2^{\frac{\Delta_{0,S}(i)}{a_i}} g^{r'_i}.$$

5. **Delegation signing key queries**: $\mathcal{A}_{II}$ can also issue a query for a warrant W for an attribute set ω with k' values out of an n'-value attribute set ω. The delegation signing key query could be simulated as follows:

   - If $|\omega \cup \omega^*| < k$, $\mathcal{B}$ can generate a simulated private key for ω as in the attribute key simulation and get a signature for ω on W normally.
   - If $|\omega \cap \omega^*| > k$, $\mathcal{B}$ selects a random $(d - k')$-element subset Ω' from Ω. If $H(W) \neq g^{a_i}$, in order to simulate $(g_2^x \prod_{i\in\omega\cup\Omega'} H(attr_i)^{r_i} H(w)^{r_a}$, $\{g^{r_i}\}_{i\in\omega\cup\Omega'}, g^{r_a})$
     - Choose $r'_a \in \mathbb{Z}_p$ and set $r'_a = \frac{1}{c}\beta + r_a$. Then

$$g_2^x \prod_{i\in\omega\cup\Omega'} H(attr_i)^{r_i} H(w)^{r_a} = (g_1^c g^{a_i})^{r'_a} \prod_{i\in\omega\cup\Omega'} H(attr_i)^{r_i} g_2^{-\frac{a_i}{c}},$$

$$g^{r_a} = g_2^{\frac{-1}{c}} g^{r'_a}$$

     when $H(W) = g_1^c g^{a_i}$.

6. **Proxy signing queries**: Assume $\mathcal{A}_{II}$ makes q_{ps} proxy signing queries. If $\mathcal{A}_{II}$ issues a proxy signature queries for a message $m \in \{0,1\}^*$ under a warrant W for a predicate Υ, in order to simulate $\sigma = \sigma_0 \cdot H(m)^{sk_p}$, $\mathcal{B}$ generates the delegation signing key σ_0 as in the **delegation signing queries** and answers the proxy signing queries as follows:

- If m_i has existed in T_3, then return $\sigma = \sigma_0 \cdot pk_p^{r_i}$ as the proxy signature to $\mathcal{A}_{II}$, where $H(m_i) = g^{r_i}$ exists in T_3.
- If m_i does not appear in T_3, then choose random $r_i \in \mathbb{Z}_p$ and return $\sigma = \sigma_0 \cdot pk_p^{r_i}$ as the proxy signature to $\mathcal{A}_{II}$. $\mathcal{B}$ adds $(m_i, H(m_i))$ to T_3.

7. **Forgery**: Assume $\mathcal{A}_{II}$ outputs a valid proxy signature

$$\sigma^* = (\sigma_0^*, \{\sigma_i^*\}_{i \in \omega^* \cup \Omega^*}, \sigma_0')$$

for predicate Υ_{k,ω^*}^*. If $H(m_w) \neq g^{a'_\delta}$ or $\overline{\Omega^*} \neq \Omega^*$ where $\overline{\Omega^*}$ are the dummy attributes, $\mathcal{B}$ will abort. Therefore

$$\begin{aligned}\sigma^* &= (\sigma_0^*, \{\sigma_i^*\}_{i \in \omega^* \cup \Omega^*}, \sigma_0') \\ &= (g_2^\alpha \prod_{i \in \omega^* \cup \Omega^*} H(attr_i)^{r_i} H(m_w)^{r_a} H(m)^{sk_p}, \{g^{r_i}\}_{i \in \omega^* \cup \Omega^*}, g^{r_a}).\end{aligned}$$

Thus $\mathcal{B}$ can compute

$$g^{\alpha\beta} = \frac{\sigma_0^*}{\prod_{i \in \omega^* \cup \Omega^*} (\sigma_i^*)^{a_i} (\sigma_0')^{a'_\delta} (pk_p)^{r_i}}$$

because $H(attr_i) = g^{a_i}$, $H(m_w) = g^{a'_\delta}$.

Next, we analysis the success probability of $\mathcal{B}$, $\mathcal{B}$ will not abort if the following conditions holds:

- $H(m_w) = g^{a_\delta}$.
- Correct guess of $d - k$ elements Ω^* from Ω.

Therefore the success probability of $\mathcal{B}$ in solving CDH problem is:

$$Succ_{\mathcal{B}}^{CDH} = \frac{\epsilon}{q_{H2} \mathrm{C}_{d-1}^{d-k}}.$$

Theorem 5.2. *Our attribute-based proxy signature scheme is secure against the $\mathcal{A}_{III}$ chosen message attacks if the CDH Problem is hard.*

Proof. Let $\mathbb{G}_1$ be a bilinear pairing group of prime order p. Algorithm $\mathcal{B}$ is given $g, g^\alpha, g^\beta \in \mathbb{G}_1$ which is a random instance of the CDH problem. Its goal is to compute $g^{\alpha\beta}$. Algorithm $\mathcal{B}$ will simulate the challenger and interact with the adversary $\mathcal{A}_{III}$ as described below.

Let's recall the definition of the adversary $\mathcal{A}_{III}$. $\mathcal{A}_{III}$ has the attribute key of the original signer as well as the public of the proxy signer, thus the attribute key

extraction and delegation queries are not needed here. The simulation is performed as follows:

1. **Setup**: $\mathcal{B}$ sets the public keys of the users and the common parameter as :
   - $\mathcal{B}$ selects a random generator $g \in_R \mathbb{G}_1$ and two random number $x, g_2 \in_R \mathbb{Z}_p^*$, then $\mathcal{B}$ chooses a $d-1$ degree polynomial q with $q(0) = x$ and computes $g_1 = g^x$. $\mathcal{B}$ sets $sk_p = \alpha, pk_p = g^\alpha$, where g^α, g^β are inputs of the CDH problem.
   - $\mathcal{B}$ then sends $(\mathbb{G}_1, \mathbb{G}_2, e, p, g, x, g_1, g_2, H)$ and pk_p to $\mathcal{A}_{III}$.
2. **Hash queries**:Assume $\mathcal{A}_{III}$ makes q_{H1}, q_{H2}, q_{H3} times for attribute, warrant and message queries, respectively. $\mathcal{B}$ maintains hash tables T_1, T_2, T_3 for attribute, warrant and message queries. For the hash queries for the attribute and warrant, $\mathcal{B}$ performs the same as in Theorem 1. For the message query on any m of $\mathcal{A}_{III}$'s choice, $\mathcal{B}$ chooses a random number $I \in (1, q_{H3})$, for each query on message m_i, if $(m_i, H(m_i))$ exits in hash table, $\mathcal{B}$ just returns $H(m_i)$ to $\mathcal{A}_{III}$, otherwise, the simulation is performed as follows:
   - If $m_i \neq m_I$, $\mathcal{B}$ chooses random $r_i \in_R \mathbb{Z}_p$, returns $H(m_i) = g^{r_i}$ and adds $(m_i, H(m_i))$ to T.
   - If $m_i = m_I$, $\mathcal{B}$ chooses random $r_I \in_R \mathbb{Z}_p$, return $H(m_I) = (g^\beta)^{r_I}$.
3. **Proxy Signing Queries**: Suppose $\mathcal{A}_{III}$ issues a proxy signing query for a message $M \in \{0,1\}^*$ under a warrant W with predicate Υ_{k,ω^*}. $\mathcal{B}$ first generates the attribute key sk_ω using the same method as the **attribute key extraction queries** in Theorem 1. Then $\mathcal{B}$ generates the delegation key $dsk = (\sigma_0, \{\sigma_i\}_{i\in\omega^*\cup\Omega'}, \sigma_0')$ using the method same as **Delegation signing key queries** in Theorem 1. Then $\mathcal{B}$ simulates the proxy signature queries as follows:
   - If $M \in T_3$, assume $H(M) = g^{r_M}$, $\mathcal{B}$ simulates the proxy signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ where $\sigma_1 = \sigma_0 \cdot pk_p^{r_M}$, $\sigma_2 = \{\sigma_i\}_{i\in\omega^*\cup\Omega'}$ and $\sigma_3 = \sigma_0'$.
   - If $M \notin T_3$, $\mathcal{B}$ chooses random $r^* \in_R \mathbb{Z}_p$ and simulates the proxy signature as $\sigma_1 = \sigma_0 \cdot pk_p^{r^*}$, $\sigma_2 = \{\sigma_i\}_{i\in\omega^*\cup\Omega'}$ and $\sigma_3 = \sigma_0'$. $\mathcal{B}$ adds $(M, H(M))$ to hash table T.
4. **Forgery**: Assume that the adversary $\mathcal{A}_{III}$ can output a proxy signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*,)$ of the message M^* under the warrant W^* for predicate Υ^* such that:
   - (M^*, W^*) has not been submitted as one of the proxy signing queries.

- $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ is a valid proxy signature.

In this case, if $M^* = m_I$, $\mathcal{B}$ can compute:

$$g^{\alpha\beta} = \left(\frac{\sigma_1^*}{g_2^x \prod_{i \in \omega \cup \Omega^*} (\sigma_2^*)^{a_i} (\sigma_3^*)^{a'_\delta}}\right)^{\frac{1}{r_I}}$$

Next, we analyse the success probability of $\mathcal{B}$. $\mathcal{B}$ will not abort if the following conditions hold:

- $H(m_w) = g^{a_\delta}$.
- $H(M) = (g^\beta)^{r_I}$.
- Correct guess of $d - k$ elements Ω^* from Ω.

Therefore,

$$Succ_{\mathcal{B}}^{CDH} = \frac{\epsilon}{q_{H2}(q_{H3} + q_{ps})\mathrm{C}_{d-1}^{d-k}}.$$

5.6 Summary

In this chapter, we studied attribute-based proxy signature (ABPS) for threshold predicates. We presented a formal security model and a concrete construction of ABPS scheme. Our model has considered different types of potential adversaries against an ABPS scheme. An interesting feature of our scheme is that it offers original signer privacy, that is even the proxy signer cannot find out who is the original signer except that the original signer's attributes satisfy a pre-claimed predicate. We leave the problem of building ABPS for other types of predicates as our future work.

Chapter 6

Security of Delegation-by-Warrant Proxy Signature Schemes

In this chapter, we identify a new attack that has been neglected by many existing proven secure proxy signature schemes. We demonstrate this attack by launching it against an identity-based proxy signature scheme which is proven secure. We then propose one method that can effectively prevent this attack. The weakness in some other proxy signature schemes can also be fixed by applying the same method.

6.1 Introduction

The concept of proxy signature was introduced by Mambo, Usuda and Okamoto in 1996 [MUO96]. They presented three different types of proxy signature, namely full delegation, partial delegation, and delegation by warrant in their seminal work. Shortly after Mambo et al.'s work, Kim et al. [KPW97] proposed a new type of proxy signature combing partial delegation and warrant. They demonstrated that schemes combining partial delegation and warrant can provide a higher level of security than schemes based on partial delegation or warrant separately. Since then, proxy signature has been extensively researched in different settings, such as blind proxy signature [ZSNL], anonymous proxy signature [SW02, FP08], and identity-based proxy signature [WMS+07].

These delegation-by-warrant proxy signature schemes can be further classified into two categories according to whether the proxy signature is generated by the proxy signer using his own private key or not. In the first type, the proxy signer generates a new proxy signing key using the delegation information and his own private key. The proxy signatures are generated under the new proxy signing key. The proxy signature schemes in [Zha97, LKK01b, LKK01a, Wan05, LYMW13] fall into the first type. In the second type, the proxy signer issues a proxy signature using his own private key. The proxy signatures are essentially combinations of the original signer's signature on the warrant and the proxy signer's signature on the message. Such proxy signature schemes could be found in [HSMW06, WMS+07, LKZC07, SXYM11, LMY14a]

On the security modeling of proxy signature, Boldyreva et al. [BPW12] proposed a comprehensive security model for the delegation by warrant proxy signature, where an original signer can also perform self-delegation. Malkin et al. [MOY04] extended

the security model to allow fully hierarchical proxy signatures. They also proved that proxy signatures are essentially equivalent to key-insulated signatures. The security model proposed in [BPW12, MOY04] is in the registered key model, which means the adversary has to submit every public and private key pair in the security game except the challenge one. Later, Schuldt et al. [SMP08] proposed an enhanced security model for proxy signature by allowing the adversary to query arbitrary proxy signing keys.

Our Contributions. We revisit proxy signature and show an attack that has been neglected by the second type of proxy signature schemes [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a] that have been proven secure. In these schemes, a proxy signature is essentially the combination of the original signer's standard signature on a warrant and the proxy signer's standard signature on a message. In the security analysis, it is assumed that an adversary has access to the original signer and proxy signer's standard signature oracles. We show that under such a circumstance, some proxy signature schemes [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a] that have been previous proved secure are in fact not secure.

We demonstrate a new attack by launching it against an identity-based proxy signature scheme [WMS$^+$07] that has been proven secure. We show that a malicious adversary can create a proxy signature on a message, if he has access to the standard signature of the original signer and proxy signer, which is as defined in the security models in [HSMW06, WMS$^+$07]. Thus, these proxy signature schemes [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a] , which we believe is not a complete list, are in fact not secure. We propose an efficient solution by revising the identity-based proxy signature scheme [WMS$^+$07] to thwart this attack. It is worth noticed that the same method can also be applied to [HSMW06, LKZC07, SXYM11, LMY14a] to resist this attack.

Organization of This Chapter. The rest of this chapter is organized as follows. We present a new attack in some proxy signature schemes in Section 6.2 by attacking an identity-based proxy signature scheme. The security model for proxy signature that captures the attack is presented in Section 6.3. We then revise the identity-based proxy signature scheme in Section 6.4 and prove its security in Section 6.5. This chapter is concluded in Section 6.6.

6.2 An Attack in Some Proxy Signature Schemes

In this section, we present an attack that has been neglected by many existing proxy signature schemes [HSMW06, LKZC07, WMS$^+$07, SXYM11, LMY14a]. To better explain how an attacker works, we demonstrate this attack via a concrete example.

Before we start to introduce the attack, we first review an identity-based proxy signature scheme proposed in [WMS+07].

An Identity-Based Proxy Signature Scheme

1. **Setup**: Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear pairing map where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. Let P be a generator of $\mathbb{G}_1$. Chooses a random number $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. Select three collision-resistant hash functions H_0, H_1, H_2 such that $H_0, H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$. The system parameters $params = \{e, \mathbb{G}_1, \mathbb{G}_2, q, P_{pub}, H_0, H_1, H_2\}$, the master secret key $Msk = s$.

2. **KeyExtract**: On input a user's identity ID, outputs the secret key for this identity $sk_{ID} = sH_0(ID)$.

3. **StandardSign**: On input a message m, the standard signature on m under identity ID is $\sigma = (\sigma_1, \sigma_2)$ such that $\sigma_1 = sk_{ID} + rH_1(M)$ and $\sigma_2 = rP$ where $r \in \mathbb{Z}_q$.

4. **StandardVer**: On input a standard signature $\sigma = (\sigma_1, \sigma_2)$ of message m under identity ID, outputs '1' if $e(\sigma_1, P) = e(H_0(ID), P_{pub})e(H_1(m), \sigma_2)$; Otherwise, output '0'.

5. **DelegationGen**: Let w be a warrant that includes the delegation information such as the identities of the original signer and the designated proxy signer, the delegation period, the types of messages that a proxy signer can sign and so on. Then the original signer with identity ID_A generates the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ such that $\sigma_{W_1} = sk_{ID_A} + r_A H_1(m_w)$ and $\sigma_{W_2} = r_A P$ where $r_A \in \mathbb{Z}_q$. The original signer sends the delegation signing key σ_w to the proxy signer.

6. **ProSign**: Upon receiving the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ and w from the original signer, the proxy signer with identity ID_B generates a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ on a message m such that $\sigma_{M_1} = \sigma_{W_1} + sk_{ID_B} + r_B H_2(m)$, $\sigma_{M_2} = \sigma_{W_2}$, $\sigma_{M_3} = r_B P$.

7. **ProVer**: On input the identities ID_A, ID_B of the original signer and proxy signer, a warrant $w \in \{0,1\}^*$ and a message $m \in \{0,1\}^*$ and the proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, outputs '1' if

$$e(\sigma_{M_1}, P) = e(H_0(ID_1), P_{pub})e(H_1(w), \sigma_{M_2})e(H_0(ID_B), P_{pub})e(H_2(m), \sigma_{M_3}).$$

Otherwise, output '0'.

An Attack Against the ID-based Proxy Signature Scheme

Wu et al.'s identity-based proxy signature scheme [WMS+07] is proven secure. However, we show below that if the original signer and proxy signer also use their private keys to generate standard signatures, which is just as defined in their security models, then their scheme could be broken by a malicious outsider attacker. Assume the identities of the original signer and proxy signer are ID_A, ID_B respectively, in the security model in [WMS+07], three types of adversaries are defined, namely,

- $\mathcal{A}_I$, which is an outsider adversary that has knowledge of (ID_A, ID_B).
- $\mathcal{A}_{II}$, which is a malicious proxy signer that has knowledge of (ID_A, ID_B, sk_{ID_B}).
- $\mathcal{A}_{III}$, which is a malicious original signer that has knowledge of (ID_A, sk_{ID_A}, ID_B).

Since the original signer and proxy signer could use the same key pairs to generate normal signatures using the standard signature scheme introduced in [WMS+07]. Suppose $\mathcal{A}_I$ aims to generate a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ on a message m with a warrant w, it is worth noticing that $\mathcal{A}_I$ might obtain such a genius warrant w when verifies a valid proxy signature. Then $\mathcal{A}_I$ acts as follows:

- $\mathcal{A}_I$ requires a standard signature $(\sigma_{A_1}, \sigma_{A_2})$ on warrant w of the original signer with identity ID_A, where w is a warrant containing the delegation information. The original signer chooses a random $r_A \in \mathbb{Z}_q$ and generates the standard signature $(\sigma_{A_1}, \sigma_{A_2})$ such that $\sigma_{A_1} = sk_{ID_A} + r_A H_1(w)$ and $\sigma_{A_2} = r_A P$.
- Upon receiving the standard signature $(\sigma_{A_1}, \sigma_{A_2})$ on w from the original signer. $\mathcal{A}_I$ aborts if $e(\sigma_{A_1}, P) \neq e(H_0(ID_A), P_{pub})e(H_1(w), \sigma_{A_2})$.
- $\mathcal{A}_I$ requires a standard signature $(\sigma_{B_1}, \sigma_{B_2})$ on message $w||m$ of the proxy signer with identity ID_B, where m is a message. The proxy signer chooses a random $r_B \in \mathbb{Z}_q$ and generates the standard signature $(\sigma_{B_1}, \sigma_{B_2})$ such that $\sigma_{B_1} = sk_{ID_B} + r_B H_2(w, m)$ and $\sigma_{B_2} = r_B P$.
- Upon receiving the standard signature $(\sigma_{B_1}, \sigma_{B_2})$ on m from the proxy signer. $\mathcal{A}_I$ aborts if $e(\sigma_{B_1}, P) \neq e(H_0(ID_B), P_{pub})e(H_2(w, m), \sigma_{B_2})$.
- If both $(\sigma_{A_1}, \sigma_{A_2})$ and $(\sigma_{B_1}, \sigma_{B_2})$ are valid. $\mathcal{A}_I$ outputs a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ on message m with warrant w such that $\sigma_{M_1} = \sigma_{A_1} + \sigma_{B_1} = sk_{ID_A} + r_A H_1(w) + sk_{ID_B} + r_B H_2(w, m)$, $\sigma_{M_2} = \sigma_{A_2} = r_A P$ and $\sigma_{M_3} = \sigma_{B_2} = r_B P$.

It can be verified that $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$ is a valid proxy signature. Thus, the proposed identity-based proxy signature is insecure, since given a proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, it might come from a malicious adversary. The proposed attack is a practical attack since a malicious adversary could launch such an attack without notice of both the original signer and the proxy signer. Besides the scheme mentioned in this section, we have found that the proxy signature schemes in [HSMW06, LKZC07, SXYM11, LMY14a] are also subjected to this attack.

6.3 Security Model

We revise the security model for identity-based proxy signature defined in [WMS+07] to capture the new attack in this section. In the security model for proxy signature, the capability of an adversary is modelled by its ability to query different oracles. Before we formally define each adversarial game, we first introduce four types of oracle queries that will appear in the models:

- **Key extract query**: $\mathcal{A}$ can query an identity $ID \in \mathcal{ID}$, where $\mathcal{ID}$ represents the identity space, to the key extract oracle $\mathcal{O}_{KE}(\cdot)$. The corresponding key sk_{ID} is then generated and returned to $\mathcal{A}$.

- **Original signer's standard signing query**: $\mathcal{A}$ can query the original signer's signing oracle $\mathcal{O}_{OS'S}(\cdot)$ with any warrant $w \in \mathcal{W}$ under the original signer's identity $ID \in \mathcal{ID}$, where $\mathcal{W}$ represents the warrant space. The private key sk_{ID} on identity ID is generated using the key extraction algorithm. The corresponding original signer's signature σ_o on warrant w is generated and returned to $\mathcal{A}$.

- **Proxy signing query**: $\mathcal{A}$ can query the proxy signing oracle $\mathcal{O}_{PS}(\cdot)$ with any message $m \in \mathcal{M}$ with warrant $w \in \mathcal{W}$ of his choice under the original signer's identity ID_A and the proxy signer's identity ID_B such that $ID_A, ID_B \in \mathcal{ID}$, where $\mathcal{M}$ represents the message space. The private keys sk_{ID_A} and sk_{ID_B} on identities ID_A, ID_B are generated using the key extraction algorithm. A valid proxy signature on m is then generated and returned to $\mathcal{A}$.

- **Proxy signer's signing Query**: $\mathcal{A}$ can query the standard signature with any message $m \in \mathcal{M}$ of his choice to the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}(\cdot)$. A valid standard signature of the proxy signer σ_p on m under the proxy signer's identity is then generated and returned to $\mathcal{A}$.

According to the information held by an attacker, three different types of adversaries are defined:

1. $\mathcal{A}_I$: an outsider attacker who only has the identities of the original signer and the proxy signer that aims to forge a valid proxy signature.

2. $\mathcal{A}_{II}$: a malicious proxy signer who possesses the private key sk_{ID_B} of the proxy signer and the identity of the original signer, and tries to forge a valid proxy signature σ without knowledge of the private key sk_{ID_A} of the original signer.

3. $\mathcal{A}_{III}$: a malicious original signer that possesses the private key sk_{ID_A} of the original signer and the identity ID_B of the proxy signer, and tries to forge a valid proxy signature σ without knowing the private key sk_{ID_B} of the proxy signer.

Adversarial Game with A Malicious Outsider Adversary

We first define the adversarial game between a malicious outsider adversary $\mathcal{A}_I$ and a simulator $\mathcal{S}$ as follows:

- **Setup**: The simulator $\mathcal{S}$ runs **Setup** algorithm to generate the $params$ and MSK, and sends $params$ to $\mathcal{A}_I$ as well as keeps MSK secret.

- **Original signer's standard signing queries**: $\mathcal{A}_I$ can choose any warrant $w \in \mathcal{W}$ with the original signer's identity ID_A and queries the original signer's standard signing oracle $\mathcal{O}_{OS'S}$. $\mathcal{S}$ generates the private key sk_{ID_A} using the key extract algorithm $sk_{ID_A} \leftarrow$ **KeyExtract**$(MSK, ID_A,$ $params)$, then $\mathcal{S}$ generates the delegation information $\sigma_o \leftarrow$ **StandardSign** $(sk_{ID_A}, w, params)$ and sends σ_o to $\mathcal{A}_I$.

- **Proxy Signer's Standard Signature Queries**: $\mathcal{A}_I$ queries the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}$ with a message $m \in \mathcal{M}$ of his choice under the proxy signer's identity $ID_B \in \mathcal{ID}$. $\mathcal{S}$ generates the private key sk_{ID_B} using the key extract algorithm $sk_{ID_B} \leftarrow$ **KeyExtract**$(MSK, ID_B,$ $params)$, then $\mathcal{S}$ generates the standard signature $s\sigma \leftarrow$ **StandardSign** $(sk_{ID_B}, m, params)$ and sends $s\sigma$ to $\mathcal{A}_I$.

- **Forgery Phase**: Finally, $\mathcal{A}_I$ outputs a proxy signature σ^* on message M^* for a warrant W^* with the original signer's identity ID_A and the proxy signer's identity ID_B.

We say $\mathcal{A}_{II}$ wins the game if

- **ProVer**$(\sigma^*, ID_A, ID_B, W^*, M^*) = 1$;

- (W^*, ID_A) has been queried to the original signer's standard signing oracle $\mathcal{O}_{OS'S}$;

- (W^*, M^*, ID_B) has been queried to the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}$.

Define the advantage of a malicious adversary $\mathcal{A}_I$ in winning the game as

$$Adv_{\mathcal{A}_I}(\kappa) \quad = \quad \Pr[\mathcal{A}_I \text{ Wins the game}].$$

Definition 6.1. *We say an identity-based proxy signature scheme is secure against an outsider adversary $\mathcal{A}_I$ if for any probabilistic polynomial time $\mathcal{A}_I$, $Adv_{\mathcal{A}_I}(\kappa)$ is negligible in κ.*

Adversarial Game with A Malicious Proxy Signer

We first define the adversarial game between a malicious proxy signer $\mathcal{A}_{II}$ and a simulator $\mathcal{S}$ as follows:

- **Setup**: The simulator $\mathcal{S}$ runs **Setup** algorithm to generate the $params$ and MSK, and sends $params$ to $\mathcal{A}_{II}$ as well as keeps MSK secret.

- **Key extract queries**: $\mathcal{A}_{II}$ selects an identity ID such that $ID \in \mathcal{ID}$, the simulator $\mathcal{S}$ runs $sk_{ID} \leftarrow$ **KeyExtract**$(MSK, ID, params)$ and returns sk_{ID} to $\mathcal{A}_{II}$.

- **Original signer's standard signing queries**: $\mathcal{A}_{II}$ can choose any warrant $w \in \mathcal{W}$ with an identity $ID \in \mathcal{ID}$ and queries original signer's standard signing oracle $\mathcal{O}_{OS'S}$. $\mathcal{S}$ generates the private key sk_{ID} using the key extract algorithm $sk_{ID} \leftarrow$ **KeyExtract**$(MSK, ID, params)$, then $\mathcal{S}$ generates the original signer's standard signature $\sigma_o \leftarrow$ **StandardSign**$(sk_{ID},$
  $w, params)$ and sends σ_o to $\mathcal{A}_{II}$.

- **Proxy signing queries**: $\mathcal{A}_{II}$ chooses a warrant $w \in \mathcal{W}$ and a message $m \in \mathcal{M}$ and queries the proxy signing oracle $\mathcal{O}_{PS}$ with the original signer's identity ID_1 and the proxy signer's identity ID_2. $\mathcal{S}$ generates

$$sk_{ID_1}, sk_{ID_2} \leftarrow \textbf{KeyExtract}(MSK, ID_1, ID_2, params)$$

$$\sigma_w \leftarrow \textbf{DelegationGen}(sk_{ID_1}, w, params),$$

$$\sigma \leftarrow \textbf{ProSign}(\sigma_w, sk_{ID_2}, m, params)$$

  and returns σ to $\mathcal{A}_{II}$.

- **Forgery Phase**: Finally, $\mathcal{A}$ outputs a proxy signature σ^* on message M^* for a warrant W^* with the original signer's identity ID_A and the proxy signer's identity ID_B.

We say $\mathcal{A}_{II}$ wins the game if

- **ProVer**$(\sigma^*, ID_A, ID_B, W^*, M^*) = 1$;
- ID_A has not been queried to the key extraction oracle $\mathcal{O}_{KE}(\cdot)$.
- (W^*, ID_A) has not been queried to the delegation oracle $\mathcal{O}_{DG}$;
- (W^*, M^*, ID_A, ID_B) has not been queried to the proxy signing oracle $\mathcal{O}_{PS}$.

Define the advantage of a malicious adversary $\mathcal{A}_{II}$ in winning the game as

$$Adv_{\mathcal{A}_{II}}(\kappa) \quad = \quad \Pr[\mathcal{A}_{II} \text{ Wins the game}].$$

Definition 6.2. *We say an identity-based proxy signature scheme is secure against the $\mathcal{A}_{II}$ under chosen identity and warrant attacks if for any probabilistic polynomial time $\mathcal{A}_{II}$, $Adv_{\mathcal{A}_{II}}(\kappa)$ is negligible in κ.*

Adversarial Game with A Malicious Original Signer

The adversarial game between a malicious original signer $\mathcal{A}_{III}$ and a simulator $\mathcal{S}$ is defined as follows:

- **Setup**, **Key Extract Queries** and **Proxy Signing Queries** are same as those in the adversarial game against a malicious proxy signer.
- **Proxy Signer's Standard Signature Queries**: $\mathcal{A}_{III}$ queries the proxy signer's standard signing oracle $\mathcal{O}_{ps's}$ with a message $m \in \mathcal{M}$ of his choice under an identity $ID \in \mathcal{ID}$. $\mathcal{S}$ generates the private key sk_{ID} using the key extract algorithm $sk_{ID} \leftarrow$ **KeyExtract**$(MSK, ID, params)$, then $\mathcal{S}$ generates the standard signature $\sigma_p \leftarrow$ **StandardSign**$(sk_{ID}, m, params)$ and sends σ_p to $\mathcal{A}_{III}$.
- **Forgery Phase**: Finally, $\mathcal{A}_{III}$ outputs a proxy signature σ^* on message M^* for a warrant W^* with the original signer's identity ID_A and the proxy signer's identity ID_B.

We say $\mathcal{A}_{III}$ wins the game if

- **ProVer**$(\sigma^*, ID_A, ID_B, W^*, M^*) = 1$;
- ID_B has not been queried to the key extraction oracle $\mathcal{O}_{KE}$;
- (W^*, M^*, ID_B) has not been queried to the proxy signer's standard signing oracle $\mathcal{O}_{PS'S}$.
- (W^*, M^*, ID_A, ID_B) has not been queried to the proxy signing oracle $\mathcal{O}_{PS}$.

Define the advantage of a malicious adversary $\mathcal{A}_{III}$ in winning the game as

$$Adv_{\mathcal{A}_{III}}(\kappa) \quad = \quad \Pr[\mathcal{A}_{III} \text{ Wins the game}].$$

Definition 6.3. *We say an identity-based proxy signature scheme is secure against the $\mathcal{A}_{III}$ under chosen identity and message attacks if for any probabilistic polynomial time $\mathcal{A}_{III}$, $Adv_{\mathcal{A}_{III}}(\kappa)$ is negligible in κ.*

6.4 The Revised Identity-Based Proxy Signature Scheme

We present the revised ID-based proxy signature scheme that efficiently thwarts the proposed attack in this section.

1. **Setup**: Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear pairing map where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. Let P be a generator of $\mathbb{G}_1$. Chooses a random number $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. Select three collision-resistant hash functions H_0, H_1, H_2 such that $H_0, H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$. The system parameters $params = \{e, \mathbb{G}_1, \mathbb{G}_2, q, P_{pub}, H_0, H_1, H_2\}$, the master secret key $Msk = s$.

2. **KeyExtract**: On input a user's identity ID, outputs the secret key for this identity $sk_{ID} = sH_0(ID)$.

3. **StandardSign**: On input a message m, the standard signature on m under identity ID is $\sigma = (\sigma_1, \sigma_2)$ such that $\sigma_1 = sk_{ID} + rH_1(m)$ and $\sigma_2 = rP$, where $r \in \mathbb{Z}_q^*$.

4. **StandardVer**: On input a standard signature $\sigma = (\sigma_1, \sigma_2)$ of message m under identity ID, output '1' if $e(\sigma_1, P) = e(H_0(ID), P_{pub})e(H_1(m), \sigma_2)$; Otherwise, output '0'.

5. **DelegationGen**: Let w be a warrant that includes the delegation information such as the identities of the original signer and the designated proxy signer, the delegation period, the types of messages that a proxy signer can sign and so on. Then the original signer with identity ID_A generates the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ such that $\sigma_{W_1} = sk_{ID_A} + r_A H_1(w)$ and $\sigma_{W_2} = r_A P$ where $r_A \in \mathbb{Z}_q$. The original signer sends the delegation information σ_w to the proxy signer.

6. **ProSign** Upon receiving the delegation information $\sigma_w = (\sigma_{W_1}, \sigma_{W_2})$ and w from the original signer, the proxy signer with identity ID_B generates a proxy

signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3}$ on a message m such that $\sigma_{M_1} = \sigma_{W_1} + sk_{ID_B} + r_B H_2(w, m) + r_B H_1(w)$, $\sigma_{M_2} = \sigma_{W_2} + r_B P$, $\sigma_{M_3} = r_B P$.

7. **ProVer**: On input the identities ID_A, ID_B of the original signer and proxy signer, a warrant w and a message m and the proxy signature $\sigma = (\sigma_{M_1}, \sigma_{M_2}, \sigma_{M_3})$, outputs '1' if

$$e(\sigma_{M_1}, P) = e(H_0(ID_A), P_{pub})e(H_1(w)), \sigma_{M_2}) \cdot e(H_0(ID_B), P_{pub})e(H_2(w, m), \sigma_{M_3})$$

Otherwise, output '0'.

6.5 Security Analysis

In this section we analyse the security of the revised ID-based proxy signature scheme against $\mathcal{A}_I$, $\mathcal{A}_{II}$ and $\mathcal{A}_{III}$ adversaries.

Theorem 6.1. *The revised ID-based proxy signature scheme is secure against an outsider adversary $\mathcal{A}_I$ if the CDH assumption holds.*

Proof. The proof is by contradiction under the random oracle model. Suppose that exists an outsider adversary $\mathcal{A}_I$ that has a non-negligible advantage ϵ in attacking the proposed scheme, then we can build another algorithm $\mathcal{B}$ that uses $\mathcal{A}_I$ to solve the CDH problem. Let $\mathbb{G}_1$ be a bilinear pairing group of prime order q, $\mathcal{B}$ is given $P, aP, bP \in \mathbb{G}_1$ which is a random instance of the CDH problem. Its goal is to compute abP. Algorithm $\mathcal{B}$ will simulate the challenger and interact with the forger $\mathcal{A}_I$ as described below.

1. **Setup**: $\mathcal{B}$ selects a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. $\mathcal{B}$ chooses a generator P of $\mathbb{G}_1$. Let (P, aP, bP) be the inputs of the CDH problem. $\mathcal{B}$ sets the master public key $P_{pub} = sP$ where $s \in \mathbb{Z}_q^*$. $\mathcal{B}$ selects three collision-resistant hash functions $H_0, H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$. $\mathcal{B}$ sends $(e, \mathbb{G}_1, \mathbb{G}_2, q, P, P_{pub}, H_0, H_1, H_2)$ to $\mathcal{A}_{II}$.

2. **Hash queries**: In the security proof, the hash functions H_0, H_1, H_2 are modelled as random oracles. We regard the identity, warrant and message queries as H_0, H_1 and H_2 queries respectively. Assume $\mathcal{B}$ keeps hash tables T_0, T_1 and T_2 for these queries.

   (a) H_0 **Query**: For each query on identity ID_i, if ID_i has existed in T_0, a same value $H_0(ID_i)$ is returned to $\mathcal{A}_{II}$. Otherwise, $\mathcal{B}$ chooses a random $c_i \in \mathbb{Z}_q$ and sets $H_0(ID_i) = c_i P$. $\mathcal{B}$ sends $c_i P$ to $\mathcal{A}_I$ as well as stores $(ID_i, c_i, H_0(ID_i))$ to T_0.

(b) H_1 **Query**: Assume $\mathcal{A}_I$ makes q_{H_1} warrant queries, $\mathcal{B}$ selects a random number $\beta \in (1, q_{H_1})$, for each query on warrant w_i such that $1 \leq i \neq \beta \leq q_{H_1}$, if w_i has existed in T_1, a same value $H_1(w_i)$ is returned to $\mathcal{A}_I$. Otherwise,

- If $w_i \neq w_\beta$, $\mathcal{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_\beta) = k_i P$. $\mathcal{B}$ sends $H_1(w_\beta)$ to $\mathcal{A}_I$ as well as stores $(w_\beta, k_i, H_1(w_\beta))$ to T_1.
- If $w_i = w_\beta$, $\mathcal{B}$ sets $H_1(w_\beta) = aP$. $\mathcal{B}$ sends $H_1(w_\beta)$ to $\mathcal{A}_I$.

(c) H_2 **Query**: For each query on message m_i accompanying with a warrant w_i, if $H_2(w_i, m_i)$ has existed in T_2, a same value $H_2(w_i, m_i)$ is returned to $\mathcal{A}_I$. Otherwise, $\mathcal{B}$ chooses a random $u_i \in \mathbb{Z}_q$ and sets $H_2(w_i, m_i) = u_i P$. $\mathcal{B}$ sends $H_2(w_i, m_i)$ to $\mathcal{A}_I$ as well as stores $((w_i, m_i), u_i, H_2(w_i, m_i))$ to T_2.

3. **Original signer's standard signing queries**: $\mathcal{A}_I$ can query the original signer's standard signature on a warrant w_i. Assume $\mathcal{A}_I$ makes $q_{os's}$ queries with the original signer's identity ID_A, for each query on w_i, assume $H_0(ID_A)$ and $H_1(w_i)$ have existed in T_0 and T_1, if they are not the cases, $\mathcal{B}$ performs the above algorithms to assign values for $H_0(ID_A)$ and $H_1(w_i)$. Assume $H_0(ID_A) = c_A P$, $\mathcal{B}$ simulates as follows:

- If $w_i \neq w_\beta$, assume $H_1(w_i) = k_i P$, then $\mathcal{B}$ chooses randomly $r_{A_i} \in \mathbb{Z}_q$ and sets $\sigma_{w_i} = (\sigma_{w_{i1}}, \sigma_{w_{i2}})$ such that $\sigma_{w_{i1}} = c_A s P + r_{A_i} k_i P = s H_0(ID_A) + r_{A_i} H_1(w_i)$ and $\sigma_{w_{i2}} = r_{A_i} P$.
- If $w_i = w_\beta$, then $\mathcal{B}$ chooses randomly $r_{A_\beta} \in \mathbb{Z}_q$ and sets $\sigma_\beta = (\sigma_{w_{\beta 1}}, \sigma_{2_{\beta 2}})$ such that $\sigma_{w_{\beta 1}} = c_A s P + r_{A_\beta} b P = s H_0(ID_A) + r_{A_\beta} H_1(w_\beta)$ and $\sigma_{w_{\beta 2}} = r_{A_\beta} P$.

4. **Proxy signer's standard signing queries**: Assume $\mathcal{A}_I$ makes $q_{ps's}$ standard signature queries under the proxy signer's identity ID_B. For each query on $M_i = w_i || m_i$, assume $H_0(ID_B)$ and $H_2(M_i)$ have existed in T_0 and T_2, if they are not the cases, $\mathcal{B}$ performs the above algorithms to assign values for $H_0(ID_A)$ and $H_2(M_i)$. Assume $H_0(ID_B) = c_B P$, $\mathcal{B}$ chooses a number $\delta \in (1, q_{ps's})$ and simulates as follow:

- If $M_i \neq M_\delta$, assume $H_2(M_2) = u_i P$, then $\mathcal{B}$ chooses randomly $r_{B_i} \in \mathbb{Z}_q$ and sets $\sigma_{p_i} = (\sigma_{p_{i1}}, \sigma_{p_{i2}})$ such that $\sigma_{p_{i1}} = c_B s P + r_{B_i} k_i P = s H_0(ID_B) + r_{B_i} H_2(M_i)$ and $\sigma_{p_{i1}} = r_{B_i} P$.
- If $M_i = M_\delta$, assume $H_2(M_\delta) = u_\delta P$, then $\mathcal{B}$ sets $dsk_\delta = (\sigma_{B1_\delta}, \sigma_{B2_\delta})$ such that $\sigma_{B1_\delta} = c_B s P + b u_\delta P = s H_0(ID_B) + b H_2(M_\delta)$ and $\sigma_{B2_i} = bP$

5. **Forgery**: Assume $\mathcal{A}_I$ outputs a valid proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$ on message M^* under a warrant W^* with the proxy signer's identity ID_A and the proxy signer's identity ID_B. Besides,

- (ID_A, W^*) has been queried in the original signer's standard signing queries.
- (ID_B, W^*, M^*) has been queried in the proxy signer's standard signing queries.

If $W^* \neq w_\beta$ or $M^* \neq M_\delta$, $\mathcal{B}$ will abort. Otherwise, given the the forged proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$. $\mathcal{B}$ can solve the CDH problem

$$abP = \sigma_{M_1^*} - \sigma_{A1_\beta} - \sigma_{B1_\delta}$$

$\mathcal{B}$ will not abort when $W^* = w_\beta$ and $M^* = M_\delta$. Thus, if there exists an outsider adversary $\mathcal{A}_I$ that has a non-negligible probability ϵ in breading the proposed identity-based proxy signature scheme. Then there exist another probabilistic polynomial time algorithm $\mathcal{B}$ that has a probability

$$Succ^{CDH}_{\mathcal{B},\mathbb{G}_1} = \frac{\epsilon}{q_{os's} \cdot q_{ps's}}$$

which is non-negligible. Thus, we reach a contradiction.

Theorem 6.2. *The revised ID-based proxy signature scheme is secure against the $\mathcal{A}_{II}$ chosen identity and chosen warrant attacks if the CDH assumption holds.*

Proof. Let's recall the definition of $\mathcal{A}_{II}$, $\mathcal{A}_{II}$ is a malicious proxy signer possessing the private key of the proxy signer. With this in mind, the simulation is as follows:

1. **Setup**: $\mathcal{B}$ selects a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are of prime order q. $\mathcal{B}$ chooses a generator P of $\mathbb{G}_1$. Let (P, aP, bP) be the inputs of the CDH problem. $\mathcal{B}$ sets the master public key $P_{pub} = aP$. $\mathcal{B}$ selects three collision-resistant hash functions $H_0, H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$. $\mathcal{B}$ sends $(e, \mathbb{G}_1, \mathbb{G}_2, q, P, P_{pub}, H_0, H_1, H_2)$ to $\mathcal{A}_{II}$.

2. **Hash queries**: Regard the identity, warrant and message queries as H_0, H_1 and H_2 queries respectively. $\mathcal{B}$ keeps hash tables T_0, T_1 and T_2 for these queries.

   (a) H_0 **Query**: Assume $\mathcal{A}_{II}$ makes q_{H_0} identity queries, choose $\alpha \in (1, q_{H_0})$, for each query on identity ID_i such that $1 \leq i \neq \alpha \leq q_{H_0}$, if ID_i has existed in T_0, a same value $H_0(ID_i)$ is returned to $\mathcal{A}_{II}$. Otherwise,

   - If $i \neq \alpha$, $\mathcal{B}$ chooses a random $c_i \in \mathbb{Z}_q$ and sets $H_0(ID_i) = c_i P$. $\mathcal{B}$ sends $c_i P$ to $\mathcal{A}_{II}$ as well as stores $(ID_i, c_i, H_0(ID_i))$ to T_0.
   - If $i = \alpha$, $\mathcal{B}$ sets $H_0(ID_\alpha) = bP + c_\alpha P$ where $c_\alpha \in \mathbb{Z}_q$ and returns $H_0(ID_I)$ to $\mathcal{A}_{II}$. $\mathcal{B}$ adds $(ID_\alpha, c_\alpha, H_0(ID_\alpha))$ to T_0.

(b) H_1 **Query**: Assume $\mathcal{A}_{II}$ makes q_{H_1} warrant queries, $\mathcal{B}$ selects a random number $\beta \in (1, q_{H_1})$, for each query on warrant w_i such that $1 \leq i \neq \beta \leq q_{H_1}$, if w_i has existed in T_1, a same value $H_1(w_i)$ is returned to $\mathcal{A}_{II}$. Otherwise,

- If $w_i \neq w_\beta|_{ID_\alpha \to o}$, which means ID_α is included in w_i and the user with identity ID_α plays the role of original signer in the system. $\mathcal{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_i) = k_iP - bP$. $\mathcal{B}$ sends $H_1(w_i)$ to $\mathcal{A}_{II}$ as well as stores $(w_i, b_i, H_1(w_i))$ to T_1.
- If $w_i \neq w_\beta|_{ID_\alpha \to p}$, which means ID_α is included in w_i and the user with identity ID_α plays the role of proxy signer in the system. $\mathcal{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_i) = k_iP$. $\mathcal{B}$ sends $H_1(w_i)$ to $\mathcal{A}_{II}$ as well as stores $(w_i, k_i, H_1(w_i))$ to T_1.
- If $w_i = w_\beta$, $\mathcal{B}$ chooses a random $k_i \in \mathbb{Z}_q$ and sets $H_1(w_\beta) = k_iP$. $\mathcal{B}$ sends $H_1(w_\beta)$ to $\mathcal{A}_{II}$ as well as stores $(w_\beta, k_i, H_1(w_\beta))$ to T_1.

(c) H_2 **Query**: Assume $\mathcal{A}_{II}$ makes q_{H_2} message queries, $\mathcal{B}$ selects a random number $\delta \in (1, q_{H_1})$, for each query on message m_i accompanying with a warrant w_i such that $1 \leq i \neq \delta \leq q_{H_2}$, if $H_2(w_i, m_i)$ has existed in T_2, a same value $H_2(w_i, m_i)$ is returned to $\mathcal{A}_{II}$. Otherwise,

- If $w_i \neq w_\beta, m_i \neq m_\delta$, $\mathcal{B}$ chooses a random $u_i \in \mathbb{Z}_q$ and sets $H_2(w_i, m_i) = u_iP + aP$. $\mathcal{B}$ sends $H_2(w_i, m_i)$ to $\mathcal{A}_{II}$ as well as stores $((w_i, m_i), c_i, H_2(w_i, m_i))$ to T_2.
- If $w_i = w_\beta, m_i \neq m_\delta$, same as the case when $w_i \neq w_\beta, m_i \neq m_\delta$.
- If $w_i \neq w_\beta, m_i = m_\delta$, same as the case when $w_i \neq w_\beta, m_i \neq m_\delta$.
- If $w_i = w_\beta, m_i = m_\delta$, $\mathcal{B}$ chooses a random $u_i \in \mathbb{Z}_q$ and sets $H_2(w_\beta, m_\delta) = u_iP$. $\mathcal{B}$ sends $H_2(w_\beta, m_\beta)$ to $\mathcal{A}_{II}$ as well as stores $((w_\beta, m_\delta), u_i, H_2(w_\beta, m_\delta))$ to T_2.

3. **Key extraction queries**: $\mathcal{A}_{II}$ can make key extraction queries on any identity $ID \in \mathcal{ID}$ such that $ID \neq ID_\alpha$. If $\mathcal{A}_{II}$ makes key extraction query on identity ID_α, $\mathcal{B}$ just terminates the simulation and reports a failure. Assume $\mathcal{A}_{II}$ makes q_k key extractions queries, for each query on identity ID_i for $1 \leq i \leq q_k$.

   - If ID_i has existed in table T_0, assume $H_0(ID_i) = c_iP$, then $\mathcal{B}$ returns $sk_{ID_i} = c_iaP = aH_0(ID_i)$ to $\mathcal{A}_{II}$.
   - Otherwise, $\mathcal{B}$ chooses a random $c_i \in \mathbb{Z}_q$ and sets $H_0(ID_i) = c_iP$. $\mathcal{B}$ returns $sk_{ID_i} = c_iaP$ to $\mathcal{A}_{II}$ and adds $(ID_i, c_i, H_0(ID_i))$ to T_0.

4. **Original signer's standard signing queries**: $\mathcal{A}_{II}$ can query original signer's standard signature on a warrant $w_i \in \mathcal{W}$ under an identity $ID_i \in \mathcal{ID}$. Assume

$\mathcal{A}_{II}$ makes $q_{os's}$ original signer's standard signing queries. For each query, assume ID_i and w_i have been submitted to the H_0 and H_1 queries respectively. If they are not the cases, $\mathcal{B}$ performs the above algorithms to set values for $H_0(ID_i)$ and $H_1(w_i)$, then $\mathcal{B}$ simulates σ_{w_i} as follows:

- If $ID_i \neq ID_\alpha$ and $w_i \neq w_\beta|_{ID_\alpha \to o}$, assume $H_0(ID_i) = c_iP$ and $H_1(w_i) = k_iP - bP$ respectively, then $\mathcal{B}$ chooses a random $r_i \in \mathbb{Z}_q$ and returns the original signer's standard signature $\sigma_{w_i} = (\sigma w_{i1}, \sigma w_{i2})$ such that $\sigma w_{i1} = c_iP_{pub} + r_i(k_iP - bP) = sk_{ID_i} + r_iH_1(w_i)$ and $\sigma w_{i2} = r_iP$ and to $\mathcal{A}_{II}$.
- If $ID_i \neq ID_\alpha$ and $w_i \neq w_\beta|_{ID_\alpha \to p}$, assume $H_0(ID_i) = c_iP$ and $H_1(w_i) = k_iP$ respectively, then $\mathcal{B}$ chooses a random $r_i \in \mathbb{Z}_q$ and returns original signer's standard signature $\sigma_{w_i} = (\sigma w_{i1}, \sigma w_{i2})$ such that $\sigma w_{i1} = c_iP_{pub} + r_ik_iP = sk_{ID_i} + r_iH_1(w_i)$ and $\sigma w_{i2} = r_iP$ to $\mathcal{A}_{II}$.
- If $ID_i = ID_\alpha$ and $w_i \neq w_\beta|_{ID_\alpha \to o}$, assume $H_0(ID_i) = bP + c_iP$ and $H_1(w_i) = k_iP - bP$ respectively, then $\mathcal{B}$ simulates the original signer's standard signature $\sigma_{w_i} = (\sigma w_{i1}, \sigma w_{i2})$ by setting $\sigma w_{i2} = r_iP = l_iP + aP$ where $l_i \in_R \mathbb{Z}_q^*$ and $\sigma w_{i1} = (c_i + k_i)P_{pub} + k_il_iP - l_ibP$. It can be verified that $(\sigma w_{i1}, \sigma w_{i2})$ is a correct simulation since:

$$\begin{aligned} \sigma w_{i1} &= (c_i + k_i)P_{pub} + k_il_iP - l_ibP \\ &= abP + c_iaP + k_iaP + k_il_iP - l_ibP - abP \\ &= a(c_iP + bP) + (a + l_i)(k_iP - bP) \\ &= aH_0(ID_\alpha) + r_iH_1(w_i) \end{aligned}$$

- If $ID_i = ID_\alpha$ and $w_i \neq w_\beta|_{ID_\alpha \to p}$, since we do not consider self-delegation in our scheme, then $\mathcal{B}$ just terminates the simulation and reports failure.
- If $ID_i = ID_\alpha$ and $w_i = w_\beta$, $\mathcal{B}$ terminates the simulation and reports failure.

5. **Proxy signing queries**: $\mathcal{A}_{II}$ can query a proxy signature on a message $m_i \in \mathcal{M}$ under a warrant $w_i \in \mathcal{W}$ with the proxy signer's identity ID_{1_i} and the original signer's identity ID_{2_i} such that $ID_{1_i}, ID_{2_i} \in \mathcal{ID}$. Assume ID_{1_i}, ID_{2_i} have been submitted to the H_0 query and w_i and $w_i||m_i$ have been submitted to the H_1 and H_2 queries respectively. If they are not the cases, the above algorithms will be performed to assign new values $H_0(ID_{1_i})$, $H_0(ID_{2_i})$, $H_1(w_i)$ and $H_2(w_i, m_i)$. Assume $\mathcal{A}_{II}$ makes q_{ps} proxy signing queries. For each queries on a message m_i with warrant w_i such that $1 \leq i \leq q_{ps}$, $\mathcal{B}$ simulates the corresponding proxy signature as follows:

   (a) If $ID_{1_i} \neq ID_\alpha$, $ID_{2_i} \neq ID_\alpha$ assume $H_0(ID_{1_i}) = c_{1_i}P$, $H_0(ID_{2_i}) = c_{2_i}P$, then $\mathcal{B}$ chooses two random numbers $r_{1_i}, r_{2_i} \in \mathbb{Z}_q^*$ and returns the proxy

signature $\sigma_i = (\sigma_{M_{i1}}, \sigma_{M_{i2}}, \sigma_{M_{i3}})$ such that $\sigma_{M_{i1}} = c_{1_i}aP + r_{1_i}H_1(w_i) + c_{2_i}aP + r_{2_i}H_2(w_i, m_i) + r_{2_i}H_2(m_i)$, $\sigma_{M_{i2}} = (r_{1_i} + r_{2_i})P$ and $\sigma_{M_{i3}} = r_{2_i}P$ to $\mathcal{A}_{II}$. It is a correct simulation since

$$\begin{aligned} & e(\sigma_{M_{i1}}, P) \\ &= e(c_{1_i}aP + r_{1_i}H_1(w_i) + c_{2_i}aP + r_{2_i}H_2(w_i, m_i) + r_{2_i}H_1(w_i), P) \\ &= e(c_{1_i}P, aP)e(H_1(w_i), (r_{1_i} + r_{2_i})P)e(c_{2_i}P, aP)e(H_2(m_i, w_i), r_{2_i}P) \\ &= e(H_0(ID_{1_i}), P_{pub})e(H_1(w_i), \sigma_{M_{i2}})e(H_0(ID_{2_i}), P_{pub})e(H_2(m_i, w_i), \\ & \sigma_{M_{i3}}) \end{aligned}$$

(b) If $ID_{1_i} \neq ID_\alpha$, $ID_{2_i} = ID_\alpha$, assume $H_0(ID_{1_i}) = c_{1_i}P$, $H_0(ID_{2_i}) = c_\alpha P + bP$, then

i. If $w_i \neq w_\beta|_{ID_\alpha \to o}, m_i \neq m_\delta$ or $w_i \neq w_\beta|_{ID_\alpha \to o}, m_i = m_\delta$, $\mathcal{B}$ terminates the simulation and reports failure.

ii. If $w_i \neq w_\beta|_{ID_\alpha \to p}$ and $m_i \neq m_\delta$, assume $H_1(w_i) = k_iP$ and $H_2(w_i, m_i) = u_iP + aP$, $\mathcal{B}$ simulates the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ by setting $\sigma_{M_{i_3}} = r_{2_i}P = v_iP - bP$, $\sigma_{M_{i_2}} = r_{1_i}P + v_iP - bP$ and $\sigma_{M_{i_1}} = (c_{1_i} + c_\alpha + v_i)P_{pub} + r_{1_i}H_2(w_i) + k_i(v_iP - bP) + u_i(v_iP - bP)$ where $v_i, r_{1_i} \in \mathbb{Z}_q$. It can be verified that it is a correct simulation since

$$\begin{aligned} & e(\sigma_{M_{i1}}, P) \\ &= e((c_{1_i} + c_\alpha + v_i)P_{pub} + r_{1_i}H_2(w_i) + k_i(v_iP - bP) + u_i(v_iP - bP), \\ & , P) \\ &= e(c_{1_i}P, aP)e(H_1(w_i), (r_{1_i} + r_{2_i})P)e(abP + c_\alpha aP + v_iaP + u_iv_iP \\ & - u_ibP - abP, P) \\ &= e(c_{1_i}P, aP)e(H_1(w_i), (r_{1_i} + r_{2_i})P)e(a(bP + c_\alpha P, P)e((v_i - b)(u_iP \\ & + aP), P) \\ &= e(H_0(ID_{1_i}), P_{pub})e(H_1(w_i), \sigma_{M_{i_2}})e(H_0(ID_{2_i}), P_{pub})e(H_2(w_i, m_i), \\ & \sigma_{M_{i_3}}) \end{aligned}$$

iii. If $w_i \neq w_\beta|_{ID_\alpha \to p}, m_i = m_\delta$ or $w_i = w_\beta, m_i \neq m_\delta$, $\mathcal{B}$ performs same as that in case *ii*.

iv. If $w_i = w_\beta$ and $m_i = m_\delta$, $\mathcal{B}$ terminates the simulation and reports failure.

(c) If $ID_{1_i} = ID_\alpha$, $ID_{2_i} \neq ID_\alpha$, assume $H_0(ID_{1_i}) = c_\alpha P + bP$, $H_0(ID_{2_i}) = c_{2_i}P$, then

i. If $w_i \neq w_\beta|_{ID_\alpha \to o}$ and $m_i \neq m_\delta$, assume $H_1(w_i) = k_iP - bP$ and $H_2(w_i, m_i) = u_iP + aP$. $\mathcal{B}$ chooses $l_i, r_{2_i} \in \mathbb{Z}_q^*$ and simulates the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ by setting $\sigma_{M_{i_3}} = r_{2_i}P$, $\sigma_{M_{i_2}} = v_iP - bP + r_{2_i}P$ and $\sigma_{M_{i_1}} = (c_\alpha + k_i + c_{2_i})P_{pub} + l_i(k_iP - bP) + r_{2_i}(k_iP - bP) + r_{2_i}(u_iP + aP)$. It is a correct simulation since

$$
\begin{aligned}
& e(\sigma_{M_{i_1}}, P) \\
&= e(c_\alpha + k_i + c_{2_i})P_{pub} + l_i(k_iP - bP) + r_{2_i}(k_iP - bP) + r_{2_i}(u_iP + \\
&\quad aP), P) \\
&= e(abP + ac_\alpha P + l_ik_iP - l_ibP + ak_iP - abp + r_{2_i}(k_iP - bP), P) \\
&\quad e(c_{2_i}P_{pub}, P)e(r_{2_i}(u_iP + aP), P) \\
&= e(a(c_\alpha P + bP), P)e((l_i + a + r_{2_i})(k_iP - bP), P)e(c_{2_i}P, aP)e(u_iP \\
&\quad + aP, r_{2_i}P) \\
&= e(H_0(ID_{1_i}), P_{pub})e(H_1(w_i), \sigma_{M_{i_2}})e(H_0(ID_{2_i}), P_{pub})e(H_2(w_i, m_i), \\
&\quad \sigma_{M_{i_3}})
\end{aligned}
$$

ii. If $w_i \neq w_\beta|_{ID_\alpha \to p}, m_i \neq m_\delta$ or $w_i \neq w_\beta|_{ID_\alpha \to p}, m_i = m_\delta$, $\mathcal{B}$ terminates the simulation and reports failure.

iii. If $w_i \neq w_\beta|_{ID_\alpha \to 0}$ and $m_i = m_\delta$, assume $H_1(w_i) = k_iP - bP$ and $H_2(w_i, m_\beta) = u_iP + aP$, $\mathcal{B}$ performs same as that in case i.

iv. If $w_i = w_\beta$ and $m_i \neq m_\delta$, assume $H_1(w_\beta) = k_iP$ and $H_2(w_\beta, m_i) = u_iP + aP$, $\mathcal{B}$ chooses $v_i, r_{1_i} \in \mathbb{Z}_q^*$ and simulates the proxy signature $\sigma_i = (\sigma_{M_{i_1}}, \sigma_{M_{i_2}}, \sigma_{M_{i_3}})$ by setting $\sigma_{M_{i_3}} = v_iP - bP$, $\sigma_{M_{i_2}} = v_iP - bP + r_{1_i}P$ and $\sigma_{M_{i_1}} = (c_\alpha + c_{2_i} + v_i)P_{pub} + r_{1_i}k_iP + k_i(v_iP - bP) + u_i(v_iP - bP)$. It is a correct simulation since

$$
\begin{aligned}
& e(\sigma_{M_{i_1}}, P) \\
&= e((c_\alpha + c_{2_i} + v_i)P_{pub} + r_{1_i}k_iP + k_i(v_iP - bP) + u_i(v_iP - bP), P) \\
&= e((c_\alpha + b)aP + c_{2_i}aP + r_{1_i}k_iP + k_i(v_iP - bP) + (v_i - b)aP + u_i \\
&\quad (v_iP - bP), P) \\
&= e(c_\alpha P + bP, aP)e(k_iP, r_{1_i}P + v_iP - bP)e(c_{2_i}P, aP)e(u_iP, v_iP - \\
&\quad bP) \\
&= e(H_0(ID_{1_i}), P_{pub})e(H_1(w_i), \sigma_{M_{i_2}})e(H_0(ID_{2_i}), P_{pub})e(H_2(w_i, m_i), \\
&\quad \sigma_{M_{i_3}})
\end{aligned}
$$

v. If $w_i = w_\beta$ and $m_i = m_\delta$, $\mathcal{B}$ terminates the simulation and reports failure.

(d) If $ID_{1_i} = ID_\alpha$, $ID_{2_i} = ID_\alpha$, $\mathcal{B}$ terminates the simulation and reports failure.

6. **Forgery**: Assume $\mathcal{A}_{II}$ outputs a valid proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$ on message M^* under a warrant W^* with the proxy signer's identity ID_A and the proxy signer's identity ID_B. Besides,

   - ID_A has not been queried in the key extraction queries.
   - (ID_A, W^*) has not been queried in the delegation queries.
   - (ID_A, ID_B, W^*, M^*) has not been queried in the proxy signing queries.

   If $H_0(ID_A) \neq bP + c_\alpha P$ or $H_1(W^*) \neq k_\beta P$ or $H_2(W^*, M^*) \neq u_\delta P$, $\mathcal{B}$ will abort. Otherwise, given the the forged proxy signature $\sigma^* = (\sigma^*_{M_1}, \sigma^*_{M_2}, \sigma^*_{M_3})$. $\mathcal{B}$ can solve the CDH problem

$$abP = \sigma^*_{M_1} - c_\alpha aP - k_\beta \sigma^*_{M_2} - c_{2_i} aP - u_\delta \sigma^*_{M_3}$$

   when $H_0(ID_A) = bP + c_\alpha P$, $H_1(ID_B) = k_\beta P$ and $H_2(W^*, M^*) = u_\delta P$.

Next, we analysis the success probability of $\mathcal{B}$, $\mathcal{B}$ will not abort if the following conditions holds:

- $ID_A = ID_\alpha$.
- $W^* = w_\beta$.
- $M^* = m_\delta$.

Therefore if $\mathcal{A}_{II}$ has a non-negligible probability ϵ in breaking the proposed ID-based proxy signature scheme, then the success probability of $\mathcal{B}$ in solving CDH problem is:

$$Succ^{CDH}_{\mathcal{B},\mathbb{G}_1} \geq \frac{\epsilon}{(q_{H_0} + q_k + q_{os's} + 2q_{ps})(q_{H_1} + q_{os's} + q_{ps})(q_{H_2} + q_{ps})}.$$

which is non-negligible. Thus, we reach a contradiction.

Theorem 6.3. *The revised ID-based proxy signature scheme is secure against the $\mathcal{A}_{III}$ chosen message and identity attack if the CDH assumption holds.*

Proof. The security is similar to that in Theorem 2. Thus, we just describe it briefly.

1. **Setup**, **Hash queries** and **Key extract** queries are same as those in the security proof against a malicious proxy signer.

2. **Proxy signer's standard signing queries** and **Proxy signing queries** are similar to the **Original signer's stand signing queries** and **Proxy signing queries** in the security for Theorem 2.

Through simulation, it can be reduced that if there exists a malicious original signer that can break the proposed scheme with a non-negligible probability ϵ, then we can build another probabilistic polynomial time algorithm $\mathcal{B}$ that can solve the CDH problem with a non-negligiblre probability $Succ^{CDH}_{\mathcal{B},\mathbb{G}_1}$ such that

$$Succ^{CDH}_{\mathcal{B},\mathbb{G}_1} \geq \frac{\epsilon}{(q_{H_0} + q_k + q_{ps's} + 2q_{ps})(q_{H_1} + q_{ps's} + q_{ps})(q_{H_2} + q_{ps})}$$

, where $q_{ps's}$ refers to the number of proxy signer's standard signing queries. Thus, we reach a contradiction.

6.6 Summary

In this chapter, we introduced a practical attack which has not been considered by some existing proxy signature schemes. In particular, we took an identity-based proxy signature scheme to describe how this attack works. We also presented an enhanced security model that can capture this attack. Our model has considered different types of potential adversaries against an identity-based proxy signature scheme and allowed the adversary to query the individual signatures of both the original signer and the proxy signer. The proposed new scheme inherits the good features of the original scheme, and at the same time can effectively prevent the attack. The proposed method can also be applied in other proxy signature schemes [HSMW06, LKZC07, SXYM11, LMY14a] to ensure an improved security.

Chapter 7

An Efficient Privacy-Preserving E-Coupon System

Previous work on electronic coupon (e-coupon) systems mainly focused on security properties such as unforgeability, double-redemption detection, and anonymity. However, achieving both traceability against dishonest users and anonymity for honest users without involving any third party is an open problem that has not been solved by the previous work. Another desirable feature of an e-coupon system that has not been studied in the literature is user privacy, which means the shop cannot identify the good (among all the choices specified in the coupon) that has been chosen by the customer during the redemption process. In this chapter, we present a novel e-coupon system that can achieve all these desirable properties. We define the formal security models for these new security requirements, and show that our new e-coupon system is proven secure in the proposed models. The original scheme was presented in *Inscrypt 2014*.

7.1 Introduction

We have reviewed several e-coupon system in the literature [CES+05, Ngu06b, CGH06]. Besides those properties already mentioned in the literature, there are some desirable features that should be taken into consideration in an e-coupon system. For example, privacy of purchase is an important issue that has been neglected in existing e-coupon systems [CES+05, Ngu06b, CGH06]. It is important to keep the choices and buying habits of users private from the coupon issuer. Another issue is that there is no efficient mechanism to trace malicious users in existing e-coupon systems without involving a trusted third party.

Our Contributions. In this chapter, we propose a new e-coupon system, which can achieve all the desirable properties mentioned above, namely unforgeability, anonymity for honest users, double redemption detection, traceability against dishonest users, and user privacy. It is worth noticing that different from a fair e-cash system [SPC95, FTY98, MNV01], the traceability in our e-coupon system is performed by the merchant (i.e., coupon issuer) rather than the bank, which makes the task more challenging. In order to achieve unforgeability, anonymity for honest users, double redemption dectection and traceability against dishonest users,

we design a new variant of blind signature which allows the signer (which is essentially the coupon issuer) to issue a signature (coupon) on a message without seeing its content. However, different from conventional blind signature schemes [Cha82, CPS94, JLO97, CKW04, AO00], our scheme involves an extra dynamic challenge-response verification in the verification phase to ensure that if a coupon is redeemed more than once, the identity of the coupon holder can be calculated. In order to achieve user privacy, we employ an oblivious transfer scheme [CT05] in the redemption protocol.

Organization of This Chapter. The rest of this chapter is organized as follows. We provide some definitions in Section 7.2. The formal security model for our e-coupon system is presented in Section 7.3. We then present our construction in Section 7.4 and prove its security in Section 7.5. This chapter is concluded in Section 7.6.

7.2 Formal Definition

An e-coupon system consists of two participants, namely, a user and a coupon issuer, which is also a service provider or shop. Our e-coupon system consists of the following algorithms.

1. **ParamGen**: On input a security parameter κ, the parameter generation algorithm outputs the public parameters.

$$params \leftarrow \textbf{ParamGen}(1^{\kappa})$$

.

2. **KeyGen**: On input the public parameter $params$, the key generation algorithm outputs a key pair for a user or a service provider.

$$(pk, sk) \leftarrow \textbf{KeyGen}(params)$$

.

3. **Issue**: The issue algorithm is an interactive protocol between the service provider $\mathcal{S}$ and a user $\mathcal{U}$,

$$C \leftarrow \textbf{Issue}(\mathcal{S}(pk_{\mathcal{S}}, sk_{\mathcal{S}}), \mathcal{U}(pk_{\mathcal{U}}, params)).$$

The output is an e-coupon for the user.

4. **Redeem**: The redeem algorithm is an interactive protocol between a user $\mathcal{U}$ and the service provider $\mathcal{S}$, taking as input an e-coupon C, the public key $pk_{\mathcal{S}}$ of the service provider, the public parameters $params$, a challenge c from the service provider $\mathcal{S}$ and a corresponding response R from the user $\mathcal{U}$. The output of this algorithm for the service provider is accept or reject.

$$\text{`}accept\text{'} \text{ or } \text{`}reject\text{'} \leftarrow \mathbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C, c, R).$$

The output of this algorithm for the user is the item $item_i$ of his choice or a failure symbol.

$$\text{`}\perp\text{'} \text{ or } item_i \leftarrow \mathbf{Redeem}_{\mathcal{U}}(pk_{\mathcal{S}}, params, C, c, R).$$

5. **Reveal**: The reveal algorithm is executed by the service provider, taking a sequence of the challenge-response pairs and the corresponding redeemed coupon $\{(c_1, R_1), (c_2, R_2), C\}$ and the public parameters $params$ as input, outputs the identity $ID_{\mathcal{U}}$ of the corresponding user or a failure symbol '$\perp$'.

$$ID_{\mathcal{U}} \text{ or } \text{`}\perp\text{'} \leftarrow \mathbf{Reveal}((c_1, R_1), (c_2, R_2), C, params).$$

7.3 Security Model

We formalize four security requirements for our e-coupon system, that is unforgeability, user anonymity, double-redemption detection, and user privacy.

Unforgeability

Unforgeability requires that an adversary $\mathcal{A}$ (could be a malicious user) cannot forge a new valid coupon that can be redeemed successfully with an honest service provider $\mathcal{S}$. The adversarial game for unforgeability between an adversary $\mathcal{A}$ and a simulator $\mathcal{B}$ is defined as follows:

1. **ParamGen**: The simulator $\mathcal{B}$ runs algorithm **ParamGen** to generate public parameters $params$.

2. **KeyGen**: The simulator $\mathcal{B}$ generates two key pairs $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ and $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$, $\mathcal{B}$ sends $pk_{\mathcal{S}}$ and $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ to $\mathcal{A}$ and keeps $sk_{\mathcal{S}}$ secret.

3. **Issue queries**: Assume $\mathcal{A}$ makes q_s issue queries to the issuing oracle $\mathcal{I}(\cdot)$, for the i-th query, $1 \leq i \leq q_s$, $\mathcal{A}$ runs the issue protocol with $\mathcal{B}$ in an interactive

manner, after each query, $\mathcal{A}$ obtains a coupon

$$C_i \leftarrow \textbf{Issue}(\mathcal{S}(pk_{\mathcal{S}}, sk_{\mathcal{S}}), \mathcal{U}(pk_{\mathcal{U}}, params)).$$

4. **Challenge**: Finally, $\mathcal{A}$ outputs a new coupon C^*. We say $\mathcal{A}$ wins the game if this coupon has not appeared in any issue query but can be redeemed successfully by $\mathcal{A}$, i.e.,
   - '*accept*' $\leftarrow \textbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C^*, c, R)$.
   - $C^* \neq C_i$, for $1 \leq i \leq q_s$.

Define the advantage of a adversary $\mathcal{A}$ in winning the unforgeability game as

$$\textbf{Adv}_{\mathcal{A}}^{unf}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}]$$

Definition 7.1. *An e-coupon system is said to be unforgeable if* $\boldsymbol{Adv}_{\mathcal{A}}^{unf}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

Anonymity

Anonymity requires that if one user follows the protocol honestly, even a malicious service provider cannot link one redeemed coupon to the identity of the user. The adversarial game between $\mathcal{A}$ and simulator $\mathcal{B}$ for anonymity is defined as follows:

1. **ParamGen**: The simulator $\mathcal{B}$ runs algorithm **ParamGen** to generate public parameters *params*.
2. **KeyGen**: The simulator $\mathcal{B}$ generates key pairs for a service provider $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ and two users $\mathcal{U}_0$ $(pk_{\mathcal{U}_0}, sk_{\mathcal{U}_0})$ and $\mathcal{U}_1$ $(pk_{\mathcal{U}_1}, sk_{\mathcal{U}_1})$ respectively, $\mathcal{B}$ sends $(pk_{\mathcal{S}}, sk_{\mathcal{S}}, pk_{\mathcal{U}_0}, pk_{\mathcal{U}_1})$ to $\mathcal{A}$.
3. **Issue queries**: Assume $\mathcal{A}$ runs **Issue** algorithm q times with $\mathcal{U}_0$ and $\mathcal{U}_1$ respectively. Let $C^0 = \{C^1_{\mathcal{U}_0}, C^2_{\mathcal{U}_0}, \ldots, C^q_{\mathcal{U}_0}\}$ and $C^1 = \{C^1_{\mathcal{U}_1}, C^2_{\mathcal{U}_1}, \ldots, C^q_{\mathcal{U}_1}\}$ be the coupon set obtained by $\mathcal{U}_0$ and $\mathcal{U}_1$.
4. **Challenge**: $\mathcal{A}$ outputs an index $1 \leq i \leq q$. $\mathcal{B}$ flips a coin to decide a value $b^* \in \{0, 1\}$, and returns $C^i_{\mathcal{U}_{b^*}}$ to $\mathcal{A}$. $\mathcal{A}$ makes a guess b' of the value b^*.

We say $\mathcal{A}$ wins the game if $b' = b^*$. Define the advantage of a adversary $\mathcal{A}$ in winning the game as

$$\textbf{Adv}_{\mathcal{A}}^{Ano}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}] - \frac{1}{2}$$

Definition 7.2. *An e-coupon system is said to provide anonymity if* $\boldsymbol{Adv}_{\mathcal{A}}^{Ano}(k)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

Double-redemption detection

Detection of double-redemption is a major concern for any digital coupon system. An e-coupon system is said to provide double-redemption detection if one user cannot redeem one coupon twice with the same service provider without being caught. In our e-coupon system, if one coupon is redeemed twice, the service provider can find a polynomial time algorithm to trace the identity of the user with overwhelming probability. The adversarial game for double-redemption detection is defined as follows:

1. **ParamGen**: The simulator $\mathcal{B}$ runs algorithm **ParamGen** to generate public parameters $params$.

2. **KeyGen**: The simulator $\mathcal{B}$ generates two key pairs $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ and $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$, $\mathcal{B}$ sends $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ and $pk_{\mathcal{S}}$ to $\mathcal{A}$.

3. **Issue queries**: Assume $\mathcal{A}$ makes q_d coupon issuing queries. $\mathcal{S}$ runs the **Issue** algorithm with $\mathcal{A}$ to issue a sequence of coupons $\{C_1, C_2, \ldots, C_{q_d}\}$ for $\mathcal{A}$.

4. **Redeem queries**: $\mathcal{A}$ runs the redeem protocol with $\mathcal{S}$ with any coupon of his choice.

5. **Challenge**: $\mathcal{A}$ outputs two pairs (C^*, c_1^*, R_1^*) and (C^*, c_2^*, R_2^*). We say $\mathcal{A}$ wins the game if

   - $(C^*, c_1^*, R_1^*) \neq (C^*, c_2^*, R_2^*)$.
   - $\mathbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C^*, c_1^*, R_1^*) = 1$ and $\mathbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C^*, c_2^*, R_2^*) = 1$.
   - '$\perp$'$\leftarrow$**Reveal**$((c_1^*, R_1^*), (c_2^*, R_2^*), C^*, params)$

Define the advantage of $\mathcal{A}$ in winning the adversarial game above as

$$\mathbf{Adv}_{\mathcal{A}}^{drd}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}]$$

Definition 7.3. *An e-coupon system is said to provide double-redemption detection if* $\boldsymbol{Adv}_{\mathcal{A}}^{drd}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$.

User privacy

We formalize a new security property which has not been considered in previous e-coupon systems. When a valid user redeems an e-coupon with the service provider, it is desirable that the service provider cannot make a connection between the coupon from the user and the service that is redeemed by the user if the coupon can be used

to redeem an item from a list of options. The adversarial game for user privacy is defined as follows.

1. **ParamGen**: The simulator $\mathcal{B}$ runs algorithm **ParamGen** to generate public parameters *params*.

2. **KeyGen**: The simulator $\mathcal{B}$ generates two key pairs $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ and $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$, $\mathcal{B}$ sends $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ and $pk_{\mathcal{U}}$ to $\mathcal{A}$.

3. **Issue queries**: $\mathcal{A}$ runs the **Issue** algorithm with $\mathcal{B}$ to generate a set of coupons $C = \{C_1, C_2, \ldots, C_{q_R}\}$.

4. **Guess**: $\mathcal{A}$ outputs an index $1 \leq i \leq q_R$. $\mathcal{B}$ then redeems C_i with $\mathcal{A}$ to choose an item m_{b^*} from $\{m_1, m_2, \ldots, m_n\}$, which is the set of items that can be redeemed by $\mathcal{B}$. Finally, $\mathcal{A}$ makes a guess $b' \in [1, n]$ for b^*.

We say $\mathcal{A}$ wins the game if $b' = b^*$. Define the success probability of the adversary $\mathcal{A}$ in making a successful guess about the service that the user choose as

$$\mathbf{Adv}^{up}_{\mathcal{A}}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}] - \frac{1}{n}$$

Definition 7.4. *An e-coupon system is said to provide user privacy if* $\boldsymbol{Adv}^{up}_{\mathcal{A}}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

7.4 Proposed Scheme

We denote in our system the service provider by $\mathcal{S}$ and a user by $\mathcal{U}$. Denote $\{m_1, m_2, \ldots, m_n\}$ the set of items that can be redeemed. The detail description of our e-coupon system is as follows.

1. **ParamGen**: On input a security parameter $\kappa \in \mathbb{N}$, generates the system parameters $paras = (G, g, p, q, H_1, H_2)$, where G_q is the subgroup of $\mathbb{Z}_p$ with prime order q and g is a generator of G_q, where $p = 2q + 1$ is also prime. $H_1 : \{0,1\}^* \to G_q$ and $H_2 : G_q \to \{0,1\}^\kappa$ are two collision-resistant hash functions.

2. **KeyGen**: On input a security parameter $\kappa \in \mathbb{N}$ and the public parameter *params*, randomly choose $x, y \in_R \mathbb{Z}_q^*$ and calculate g^x, g^y and output the private and public key pairs $(sk_{\mathcal{U}} = x, pk_{\mathcal{U}} = g^x)$ and $(sk_{\mathcal{S}} = y, pk_{\mathcal{S}} = g^y)$ for the user and service provider respectively.

3. **Issue**: The issue protocol is performed through interactive communications between the service provider $\mathcal{S}$ and a user $\mathcal{U}$. The result of the issue protocol is that $\mathcal{S}$ generates a valid coupon for a user $\mathcal{U}$.

- On receiving a request from $\mathcal{U}$, $\mathcal{S}$ chooses $k \in_R \mathbb{Z}_q^*$ and computes $\delta_1 \leftarrow pk_{\mathcal{U}}^k$ and $\delta_2 \leftarrow g^k$ sends (δ_1, δ_2) to $\mathcal{U}$.
- After receiving (δ_1, δ_2) from $\mathcal{S}, \mathcal{U}$ checks whether $\delta_1 = \delta_2^{sk_{\mathcal{U}}}$. If the verification fails, $\mathcal{U}$ stops; otherwise, $\mathcal{U}$ chooses $x_1 \in \mathbb{Z}_p^*$ and computes $\alpha \leftarrow (g^{xy})^{x_1}$, $\beta \leftarrow (g^x)^{x_1}$ and $\lambda = g^{x_1}$, $m \leftarrow H_1(\alpha, \beta, \lambda)$. $\mathcal{U}$ chooses two different random number a, b and computes $r \leftarrow m\beta^a \delta_1^{\frac{bx_1}{a}}$ and $m' \leftarrow \frac{aH_1(m,r)}{b}$, $\mathcal{U}$ sends m' to $\mathcal{S}$.
- $\mathcal{S}$ computes the signature $s' = m'y + k$ on the blind message m' and sends s' to $\mathcal{U}$.
- $\mathcal{U}$ verifies if $g^{s'} \equiv Y^{m'}\delta_2 \mod p$, if the equation holds, $\mathcal{U}$ removes the blind factor b by calculating $s = \frac{s'b}{a} + a$ and stores $(\alpha, \beta, \lambda, r, s)$; otherwise, abort.

4. **Redeem**: The redeem protocol is performed as follows:
   - After receiving a redeem request from the user, $\mathcal{S}$ generates a challenge $c = H_1(ID_{\mathcal{S}}||Date||Time)$ and sends c to $\mathcal{U}$.
   - After receiving c, $\mathcal{U}$ computes $R = x_1 + cx_1x$ and choose $\sigma_i \in \{1, 2, \ldots, n\}$ and a random number $a_i \in \mathbb{Z}_q^*$, $w_{\sigma_i} = H_1(\sigma_i)$ and $A = w_{\sigma_i} g^{a_i}$. $\mathcal{U}$ sends $(c, R, \alpha, \beta, \lambda, r, s)$ and A to $\mathcal{S}$.
   - $\mathcal{S}$ checks if $H_1(\alpha, \beta, \lambda) = \beta^{-s}\alpha^{H_1(H_1(\alpha,\beta,\lambda),r)}r$ and $g^R = \lambda\beta^c$. If the equation not holds, aborts; otherwise, $\mathcal{S}$ computes $D = A^y$, $w_i = H_1(i)$ and $c_i = m_i \oplus H_2(w_i^y)$, $i = 1, 2, \ldots, n$. $\mathcal{S}$ sends D and $c_1, c_2, \ldots, c_n$ to $\mathcal{U}$.
   - $\mathcal{U}$ computes $K = D/Y^{a_i}$ and recover $m_{\sigma_i} = c_{\sigma_i} \oplus H_2(K)$.
5. **Reveal**: Assume the coupon $C = (\alpha, \beta, \lambda, r, s)$ is redeemed twice, the $\mathcal{S}$ could get two challenge-response pairs (R_1, c_1) and (R_2, c_2) on C such that $R_1 = x_1 + c_1x_1x$ and $R_2 = x_1 + c_2x_1x$. It is obvious that $\mathcal{S}$ could calculate x and x_1, thus the identity of $\mathcal{U}$ is traced by $\mathcal{S}$.
6. **Correctness**: The correctness check for validity of the coupon is as follows:

$$
\begin{aligned}
&\beta^{-s}\alpha^{H_1(H_1(\alpha,\beta,\lambda),r)}r \\
&= (pk_{\mathcal{U}}^{x_1})^{-s}(g^{xyx_1})^r r \\
&= (pk_{\mathcal{U}}^{x_1})^{-H_1(H_1(\alpha,\beta,\lambda),r)y - \frac{kb}{a} - a}(pk_{\mathcal{U}}^{x_1})^{H_1(H_1(\alpha,\beta,\lambda),r)y} m (pk_{\mathcal{U}}^{x_1})^a (pk_{\mathcal{U}}^{x_1})^{\frac{kb}{a}} \\
&= m \\
&= H_1(\alpha, \beta, \lambda)
\end{aligned}
$$

The correctness check for a user $\mathcal{U}$ to recover the correct message is as follows:

$$
\begin{aligned}
& c_{\sigma_i} \oplus H_2(K) \\
&= m_{\sigma_i} \oplus H_2(w_{\sigma_i}^y) \oplus H_2(A^y / Y^{a_i}) \\
&= m_{\sigma_i} \oplus H_2(w_{\sigma_i}^y) \oplus H_2((w_{\sigma_i} g^{a_i})^y / Y^{a_i}) \\
&= m_{\sigma_i} \oplus H_2(w_{\sigma_i}^y) \oplus H_2((w_{\sigma_i})^y) \\
&= m_{\sigma_i}
\end{aligned}
$$

7.5 Security Analysis

Theorem 7.1. *The proposed e-coupon system is unforgeable.*

Proof. The security proof is by contradiction. We will prove that if there exists a PPT adversary $\mathcal{A}$ that can forge a coupon, then there exists another algorithm $\mathcal{B}$ that can break the OMDL assumption with a non-negligible probability. Suppose there exists a polynomial time forge adversary $\mathcal{A}$ which can break the unforgeability of our system with a non-negligible probability ϵ. $\mathcal{B}$ is the simulator in our proof and has access to two types of oracles. The first is discrete logarithm oracle $DLog_{G_{q,g}}(\cdot)$ which takes $P_i \in G_q$ as input and returns $p_i \in \mathbb{Z}_q$ such that $g^{p_i} = P_i$. The second is a challenge oracle $C(\cdot)$ which takes noting as input, but for each time it is invoked it returns a challenge $P \in G_q$. Besides, $\mathcal{B}$ maintains an H-table to record all the hash queries and the corresponding answers. Assume $\mathcal{A}$ makes q_h hash queries and q_s coupon issuing queries, the simulation is as follows:

1. **ParamGen**: $\mathcal{B}$ runs algorithm **ParamGen** to generate public parameters (G, p, q, g, H_1, H_2).

2. **KeyGen**: $\mathcal{B}$ runs **KeyGen** to generate a key pair $(sk_{\mathcal{U}}, pk_{\mathcal{U}})$. $\mathcal{B}$ queries the challenge oracle $C(\cdot)$ and sets the response P_0 as the public key of the shop $pk_{\mathcal{S}} = P_0$. $\mathcal{B}$ sends $(p, g, pk_{\mathcal{S}})$ and $(sk_{\mathcal{U}}, pk_{\mathcal{U}})$ to $\mathcal{A}$.

3. **Hash queries**: For each hash query with an input message m, $\mathcal{B}$ first checks the H-table:

   - If there exists a pair (m, h) in the H-table,where m refers to the message queried before, $\mathcal{B}$ returns h as the answer to $\mathcal{A}$.
   - Otherwise, $\mathcal{B}$ chooses a random $h \in \mathbb{Z}_q$, sends h to $\mathcal{A}$ as the answer for the hash query, and adds (m, h) into the H-table.

4. **Issue queries**: Upon receiving an issuing query, $\mathcal{B}$ make a query to the challenge oracle $C(\cdot)$ and obtains a challenge P_i. $\mathcal{B}$ then sets $(\delta_1, \delta_2) = (P_i^{sk_{\mathcal{U}}}, P_i)$

and sends (δ_1, δ_2) to the adversary. After receiving a message m_i, $\mathcal{B}$ sends $P_i P_0^{m_i}$ to the discrete logarithm oracle $DLog(\cdot)$ and gets a response z_i, and sends z_i to $\mathcal{A}$. Since $z_i = DLog_{G_{q,g}}(P_i P_0^{m_i}) = DLog_{G_{q,g}}(P_i) + m_i DLog_{G_{q,g}}(P_0)$. In $\mathcal{A}$'s view, $\mathcal{B}$ simulates the signer perfectly.

5. **Challenge**: Suppose $\mathcal{A}$ can successfully forge a new coupon $C^* = (\alpha^*, \beta^*, \lambda^*, r^*, s^*)$ where $s^* = ep_0 + r'$, and C^* can pass the redemption protocol. According to the Forking lemma [PS96] by rewinding $\mathcal{A}$ to the step where $H_1(m^*, r^*) = e$ is determined and providing a new hash value for $H_1(m^*, r^*) = \hat{e}$. $\mathcal{B}$ can generate another valid coupon $\hat{C}^* = (\alpha^*, \beta^*, \lambda^*, r^*, \hat{s}^*)$ where $\hat{s}^* = e'p_0 + r'$. Then $\mathcal{B}$ can compute

$$p_0 = DLog_{G_{q,g}}(P_0) = \frac{s^* - \hat{s}^*}{e - \hat{e}}.$$

Once $\mathcal{B}$ obtains p_0, for each challenge P_i from the challenge oracle $C(\cdot)$, it can calculate $p_i = z_i - m_i p_0$ for each P_i. Therefore, $\mathcal{B}$ can successfully solve the OMDL problem.

Theorem 7.2. *The proposed e-coupon system provides anonymity.*

Proof. Anonymity of the user requires the service provider cannot link a redeemed coupon to an honest user. The proof is by contradiction, suppose that there exists a PPT adversary $\mathcal{A}$ which can break anonymity of our e-coupon system with a non-negligible probability ϵ, then we can build an algorithm $\mathcal{B}$ that use $\mathcal{A}$ to solve the DDH problem with a non-negligible probability. Let (g, g^a, g^b, g^z) be an instance of the DDH problem, the purpose of $\mathcal{B}$ is to decide whether $g^z = g^{ab}$. The simulation is as follows:

1. **ParamGen**: $\mathcal{B}$ runs algorithm **ParamGen** to generate public parameters (G, p, q, g, H_1, H_2).

2. **KeyGen**: $\mathcal{B}$ choose two random number $s^*, r_0 \in \mathbb{Z}_p^*$ and computes $S^* = g^{s^*}$, $\mathcal{S}$ sets key pair of the service provider as $(pk_{\mathcal{S}}, sk_{\mathcal{S}}) = (S^*, s^*)$ and the public keys of two valid users $\mathcal{U}_0$ and $\mathcal{U}_1$ as $pk_{\mathcal{U}_0} = g^{r_0}$ and $pk_{\mathcal{U}_1} = g^b$ respectively. $\mathcal{B}$ sends $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ and $pk_{\mathcal{U}_0}$, $pk_{\mathcal{U}_1}$ to $\mathcal{A}$.

3. **Issuing queries**: $\mathcal{B}$ performs **Issuing queries** with $\mathcal{A}$ as follows.

   (a) For $\mathcal{U}_0$, $\mathcal{B}$ knows the private key of $\mathcal{U}_0$. Thus $\mathcal{B}$ just follows the **Issue** protocol to obtains a set of coupons $\{C_{\mathcal{U}_0}^1, C_{\mathcal{U}_0}^2, \ldots, C_{\mathcal{U}_0}^{q_c}\}$;

   (b) For $\mathcal{U}_1$, $\mathcal{B}$ simulates the queries as follows:

- Upon receiving a pair $(\delta_1, \delta_2) = (pk_{\mathcal{U}_1}^{k_i}, g^{k_i})$ from $\mathcal{A}$. $\mathcal{B}$ executes the extractor defined in the KEA assumption [BP04] to extract the value k_i. If $\mathcal{A}$ misbehaves to generate a fake pair (δ_1', δ_2'). The extractor will return a failure symbol '$\perp$' and thus $\mathcal{B}$ stops this query. Otherwise, $\mathcal{B}$ chooses a random number r_i and sets $\alpha_i = g^{(z)s^* r_i}, \beta_i = g^{(z)r_i}, \lambda_i = g^{(a)r_i}$ and computes $m_i = H_1(\alpha_i, \beta_i, \lambda_i)$.
- $\mathcal{B}$ chooses two random number $a_i, b_i \in \mathbb{Z}_q$ and computes $r = m_i g^{z r_i a_i} \cdot g^{\frac{z k_i b_i}{a_i}}$ and $m' = \frac{a_i H_1(m_i, r)}{b_i}$. $\mathcal{B}$ sends m' to $\mathcal{A}$.
- On receiving an $\bar{s}$ from $\mathcal{A}$, $\mathcal{B}$ calculates $s = \frac{\bar{s} b_i}{a_i} + a_i$, and stores $(\alpha_i, \beta_i, \lambda_i, r, s)$.

Let $C^0 = \{C_{\mathcal{U}_0}^1, C_{\mathcal{U}_0}^2, \ldots, C_{\mathcal{U}_0}^{q_c}\}$ and $C^1 = \{C_{\mathcal{U}_1}^1, C_{\mathcal{U}_1}^2, \ldots, C_{\mathcal{U}_1}^{q_c}\}$ be the q coupons generated for $\mathcal{U}_0$ and $\mathcal{U}_1$ respectively in this phase.

4. **Challenge**: After receiving the index i, $\mathcal{B}$ flips a coin to decide a value $b^* \in \{0, 1\}$ and returns $C_{\mathcal{U}_{b^*}}^i$ to $\mathcal{A}$. $\mathcal{A}$ finally returns b'. $\mathcal{B}$ outputs '1' if $b' = b^*$. Otherwise, $\mathcal{B}$ outputs '0'.

We finish the simulation for the e-coupon system. Assume a PPT $\mathcal{A}$ have a non-negligible probability ϵ in breaking anonymity of our scheme. Then the probability of $\mathcal{B}$ to solve the DDH problem $\mathcal{A}_{\mathcal{B}}^{DDH}(\kappa)$ can be calculated as follows:

$$
\begin{aligned}
&\mathcal{A}_{\mathcal{B}}^{DDH}(\kappa) \\
&= \Pr[\mathcal{A} \text{ wins} | g^z = g^{ab}] - \Pr[\mathcal{A} \text{ wins} | g^z = g^r] \\
&= \Pr[b^* = b' | g^z = g^{ab}] - \Pr[b^* = b' | g^z = g^r] \\
&= \frac{1}{2} + \epsilon - (\Pr[b^* = b' | g^z = g^r \wedge b^* = 0] \Pr[b^* = 0] + \Pr[b^* = b' | g^z = g^r \wedge b^* = 1] \\
&\quad \cdot \Pr[b^* = 1]) \\
&= \frac{1}{2} + \epsilon - \frac{1}{2}(\frac{1}{2} + \epsilon + \frac{1}{2}) \\
&= \frac{1}{2}\epsilon
\end{aligned}
$$

which is non-negligible. Thus, we reach a contradiction.

Theorem 7.3. *The proposed e-coupon system provides double-redemption detection.*

Proof. According to our e-coupon system, if $\mathcal{U}$ has double-redeemed a coupon $(\alpha, \beta, \lambda, r, s)$, then $\mathcal{B}$ obtains two different challenge-response pairs (c_1, R_1) and (c_2, R_2) on the coupon where $R_1 = x_1 + c_1 x_1 x$ and $R_2 = x_1 + c_2 x_1 x$, therefore, the secret key x of $\mathcal{U}$ can be easily calculated as follows:

$$x = \frac{R_2 - R_1}{c_2 R_1 - c_1 R_2}$$

Thus, the public key of the user could be obtained by the service provider by further calculating $y = g^x \mod p$.

Theorem 7.4. *The proposed e-coupon system provides unconditionally user privacy.*

Proof. User privacy of our e-coupon system can be prove by following the receiver's privacy in the oblivious transfer scheme proposed in [CT05]. For any $A = w_{\sigma_i} g^{a_i}$, there exists w_l and a'_l such that $l \neq \sigma_i$, but $A = w_l g^{a'_l}$. Thus in the service provider's view, A could be a masked value of any index. Thus, the user's choices are unconditionally secure.

7.6 Summary

In this chapter, we proposed a practical e-coupon system which enables the coupon issuer to trace the identity of misbehaving users, while maintain the anonymity for the honest users. We achieved this without requiring any third party in the system. In addition, we formalized the notion of user privacy during the coupon redemption process and proved that our new e-coupon system also satisfied this property.

Chapter 8

Efficient Oblivious Transfer with Retrievable Receiver's Privacy

To achieve traceability against users' redemption privacy in an e-coupon system, we construct two oblivious transfer schemes with some new properties in this chapter. Oblivious transfer (OT) has been applied widely in privacy-sensitive systems such as on-line transactions and electronic commerce to protect users' sensitive information. Traceability is a desirable feature of such systems where the privacy of the honest users are protected unconditionally while the misbehaving users' privacy can be traced. However, previous research on OT mainly focused on designing protocols with unconditional receiver's privacy. Thus, traditional OT schemes cannot fulfill the traceability requirements in the aforementioned applications. In this chapter, we address this problem by presenting a novel OT with retrievable receiver's privacy (OTRRP) without involvement of any trusted third party. In the new system, an honest receiver is able to make a fixed number of choices with perfect receiver privacy. However, if the receiver misbehaves and tries to request more than a pre-fixed number of choices, then all his previous choices could be traced by the sender. We first define the formal definition and security model for OTRRP, and then propose two efficient OTRRP schemes and prove their security under the proposed security model.

8.1 Introduction

As reviewed in Section 2.3, all the previous research on OT aimed to design OT schemes with unconditional receiver and sender privacy. In real-world applications [AIR01, LMY14b], it is desirable for the sender to trace the choices of the receiver if they misbehave. Thus, the previous OT schemes are not suitable in these scenarios. To the best of our knowledge, there is only one work [MXZ11] on constructing OT scheme with retrievable receiver's privacy. Unfortunately, this OT scheme involves a trusted time server that publishes trapdoors on a time basis, and using the trapdoor the privacy of all the receivers, including the honest ones, will be lost. The motivation of this work is to propose a new OT with retrievable receiver's privacy such that the privacy of an honest receiver is protected unconditionally while all the previous choices of a misbehaving receiver can be revealed by the sender if the receiver makes more than the pre-determined number of choices in the OT protocol.

Our Contributions. In this chapter, we present two novel oblivious transfer schemes that allow a sender to reveal the dishonest receivers' privacy without the help of any trusted third party. To be more specific, our proposed OT_n^k schemes can achieve the following properties:

1. The receiver can only obtain a fix number of messages $m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}$ from the message set $\{m_1, m_2, \ldots, m_n\}$ held by the sender where $\sigma_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq k$. The receiver's choice is hidden from the sender.

2. The receiver cannot learn anything on message m_i such that $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ for $1 \leq i \leq n$.

3. If receiver makes more than k requests, then all his previous choices $(m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k})$ could be traced by the sender.

We construct two efficient adaptive OT_n^k schemes with retrievable receiver's privacy and prove its security under the half-simulation model [NP05].

Organization of This Chapter. The rest of this chapter is organized as follows. We introduce the formal definition in Section 8.2 and the security model in Section 8.3 separately. Two ORTTP schemes and their security analysis are presented in Section 8.4 and Section 8.5 separately. This chapter is concluded in Section 8.6.

8.2 Formal Definition

We present the formal definition of OTRRP in this section. There are two participants in an OT system, namely, a sender S and a receiver R. S possesses a set of messages $\{m_1, m_2, \ldots, m_n\}$ and R makes a set of choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ such that $\sigma_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq k$.

Formal Definitions of OTRRP

An OTRRP scheme is essentially an interactive protocol consisting of a tuple of PPT algorithms (*Setup*, *Commitment* (Optional), *Request*, *Response*, *Extract*, *Reveal*).

1. *Setup*: Taking as input of a security parameter κ, the setup algorithm outputs the system public parameters.

$$params \leftarrow Setup(1^\kappa)$$

2. *KeyGen*: Taking as input of the public parameter *params*, the key generation algorithm outputs a retrievable key pair (rpk, rsk) for the receiver and a one-

time key pair for the sender.

$$(rpk, rsk) \leftarrow KeyGen(params)$$
$$(opk, osk) \leftarrow KeyGen(params)$$

3. *Commitment* (Optional): Taking as input of the system parameters $params$, the retrievable public key rpk of the receiver, the messages $m_1, m_2, \ldots, m_n$ and one-time secret key osk of the sender, the commitment algorithm outputs a set of ciphertext $c_1, c_2, \ldots, c_n$.

$$c_1, c_2, \ldots, c_n \leftarrow Commitment(rpk, m_1, m_2, \ldots, m_n, osk, params)$$

4. *Request*: Taking as input of the intended indexes σ, the retrievable private key rsk and $params$, this algorithms outputs the commitment of the user's choice.

$$A_\sigma \leftarrow Request(\sigma; rsk; params)$$

5. *Response*: Taking as input of the commitment A_σ from the receiver, the secret of the sender, the output of the algorithm is response of the sender.

$$D_\sigma \leftarrow Response(A_\sigma, sk, params)$$

If there is no *Commitment* algorithm in the scheme, then there are some additional outputs $c_1, c_2, \ldots, c_n$ of the *Response* algorithm such that:

$$c_1, c_2, \ldots, c_n, D_\sigma \leftarrow Response(A_\sigma, rpk, m_1, m_2, \ldots, m_n, osk, params)$$

where c_1, c_2, c_n are ciphertexts of $m_1, m_2, \ldots, m_n$.

6. *Extract*: Taking as input of the response D_σ from the sender, the cipertext c_α and the system parameters $params$, output the message of the receiver's choice.

$$m_\sigma \leftarrow Extract(D_\sigma, c_\sigma, params)$$

7. *Reveal*: The *Reveal* algorithm is performed by the sender, taking as input of the $k+1$ transcripts $A_{\sigma_1}, A_{\sigma_2}, \ldots, A_{\sigma_{k+1}}$ from a receiver, the retrievable public key rpk and $params$, outputs the receiver's choice $\sigma_1, \sigma_2, \ldots, \sigma_k$.

$$\sigma_1, \sigma_2, \ldots, \sigma_k \leftarrow Reveal(A_{\sigma_1}, A_{\sigma_2}, \ldots, A_{\sigma_{k+1}}; rpk; params)$$

Correctness: We require that for any security parameter $\kappa \in \mathbb{N}$, if $params \leftarrow ParamGen(1^\kappa)$, $(rpk, rsk) \leftarrow KeyGen(params)$, $(opk, osk) \leftarrow KeyGen(params)$, $c_1, c_2, \ldots, c_n \leftarrow Commitment(rpk, m_1, m_2, \ldots, m_n, osk, params)$, $A_\sigma \leftarrow Request(\sigma; rsk, params)$, $D_\sigma \leftarrow Response(A_\sigma, osk; parmas)$,

or $params \leftarrow ParamGen(1^\kappa)$, $(rpk, rsk) \leftarrow KeyGen(params)$, $(opk, osk) \leftarrow KeyGen(params)$, $A_\sigma \leftarrow Request(\sigma; rsk, params)$, $c_1, c_2, \ldots, c_n, D_\sigma \leftarrow Response(A_\sigma, rpk, m_1, m_2, \ldots, m_n, osk, params)$, then

- The receiver can extract the correct message.

$$\Pr(m_\sigma \leftarrow Extract(D_\sigma, rsk, params)) = 1.$$

- If the receiver makes less than $k+1$ requests, then the sender cannot obtain any information about the receiver's choice.

$$\Pr(\text{`} \perp \text{'} \leftarrow Reveal(A_{\sigma_1}, A_{\sigma_2}, \ldots, A_{\sigma_\delta}; rpk; params | \delta \leq k)) = 1.$$

- If the receiver makes more than k requests, then the sender can trace the previous choice of the receiver.

$$\Pr(\sigma_1, \sigma_2, \ldots, \sigma_\delta \leftarrow Reveal(A_{\sigma_1}, A_{\sigma_2}, \ldots, A_{\sigma_\delta}; rpk; params | \delta > k)) = 1.$$

8.3 Security Model

In this section, we revise the half-simulation model proposed in [NP05] to evaluate the security of the proposed OTRRP scheme. Besides the sender and receiver's privacy, we define a new security property named retrievability to capture the new feature of OTRRP. In the half-simulation model, the security of the sender and receiver is considered separately. Assume the sender S holds a set of messages $\{m_1, m_2, \ldots, m_n\}$ and the receiver possesses a set of choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ such that $\sigma_i \in \{1, 2, \ldots, n\}$ for $1 \leq i \leq k$. A secure OTRRP scheme should meet the following security requirements:

1. *Receiver's Privacy*:

   - If R makes less than $k+1$ requests, then S cannot obtain any information about R's choice.
   - For any two different choice sets $\mathcal{C} = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ and $\mathcal{C}'$, the transcripts $\mathcal{A} = \{A_{\sigma_1}, A_{\sigma_2}, \ldots, A_{\sigma_k}\}$ and $\mathcal{A}' = \{A'_{\sigma_1}, A'_{\sigma_2}, \ldots, A'_{\sigma_k}\}$ received by S corresponding to $\mathcal{M} = \{m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}\}$ and $\mathcal{M}' = \{m'_{\sigma_1}, m'_{\sigma_2}, \ldots, m'_{\sigma_k}\}$ are

indistinguishable if the received messages $\mathcal{M} = \{m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}\}$ and $\mathcal{M}' = \{m'_{\sigma_1}, m'_{\sigma_2}, \ldots, m'_{\sigma_k}\}$ are identically distributed.

2. *Sender's Privacy*:

   - R cannot obtain any information on m_i, $i \notin \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ for $1 \leq i \leq n$.
   - In the half-simulation model, the security of R is defined by the real-world/ideal-world paradigm. In the real world, R and S execute the protocol. In the ideal world, the protocol is implemented with the help a trusted third party (TTP). S sends all the messages $m_1, m_2, \ldots, m_n$ to the TTP. While R sends his choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$ adaptively to the TTP. If $\{\sigma_1, \sigma_2, \ldots, \sigma_k\} \in \{1, 2, \ldots, n\}$ the TTP sends messages $\{m_{\sigma_1}, m_{\sigma_2}, \ldots, m_{\sigma_k}\}$ to the receiver. An OTRRP scheme is said to provide the privacy of the sender if for any receiver R in real world, there exists an probabilistic polynomial-time (PPT) R' in the ideal world such that the output of R and R' are indistinguishable.

3. *Retrievability*:

   If a dishonest receiver R makes $k+1$ choices $\{\sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}\}$ from S, suppose $\mathcal{A} = \{A_{\sigma_1}, A_{\sigma_2}, \ldots, A_{\sigma_k}, A_{\sigma_{k+1}}\}$ is the transcript set of the $k+1$ choices, then S is able to trace R's choices through an efficient PPT algorithm *Reveal*.

8.4 Our First Scheme and Security Analysis

Construction overview

In this section, we present one oblivious transfer scheme that can achieve all the desirable features. To be more specific, in our system, each receiver is associated with a retrievable key pair. During each execution of the OT protocol, the public retrievable key is used to hide the choice of the receiver. Meanwhile, the receiver has to create a share of the secret retrievable key using Shamir's Secret Sharing scheme and then include the share in the transaction. This is achieved through an efficient Non-interactive Proof of Knowledge which allows the prover to check that a correct share is indeed sent by the receiver. If the receiver performs more than the allowed transactions, then the sender can retrieve the receiver's secret key and use it to recover the choices the receiver has made in all the transactions.
In order to better understand the proposed scheme, we first introduce some notations used in the construction.

Table 8.1: Notations Used in The Proposed OTRRP Scheme

$s_1, \ldots, s_k$	k random secrets
$S_1, \ldots, S_k$	commitments on $s_1, \ldots, s_k$
A_i	transcript on the receiver's choice
B_i	commitment on the random number r_i
B_i'	commitment on the receiver's α_i
$f(B_i)$	one verifiable secret share

8.4.1 The First Proposed OTRRP Scheme

The proposed scheme consists of a tuple of PPT algorithms as follows.

1. **Setup**: Let G_q denote a subgroup of $\mathbb{Z}_p$ with prime order q and $g, h_1, h_2, \ldots, h_n$ be generators of G_q, where $p = 2q+1$ is also prime. Choose two collision resistant hash functions H, H_1 such that $H : \mathbb{N} \to Z_q^*$ and $H_1 : G_q \to G_q$. The system parameters $params = (G_q, p, q, g, h_1, h_2, \ldots, h_n, H, H_1)$.

2. **KeyGen**: The receiver R chooses a random number $s \in \mathbb{Z}_q^*$ and generates a retrievable key pair $(rpk, rsk) = (g^s, s)$. R chooses k random numbers $s_1, s_2, \ldots, s_k \in_R \mathbb{Z}_q$ and computes $S_1 = g^{s_1}, S_2 = g^{s_2}, \ldots, S_k = g^{s_k}$. S chooses a random number $z \in_R \mathbb{Z}_q^*$ and generates a one-time key pair $(opk, osk) = (g^z, z)$. R publishes rpk and $S_1, S_2, \ldots, S_k$ and S publishes opk.

3. **Commitment Phase**: S computes the ciphertext of $m_1, m_2, \ldots, m_n$ as $c_i = H_1((rpk \cdot h_i^{H(i)})^z) \cdot m_i,\ 1 \le i \le n$, S sends $c_1, c_2, \ldots, c_n$ to R.

4. **Request**: In the i-th round,

   - R chooses $r_i \in_R \mathbb{Z}_q^*$, and computes $B_i = g^{r_i}$, $B_i' = h_{\alpha_i}^{r_i}$ and $A_i = (g^{r_i})^s (h_{\alpha_i}^{r_i})^{H(\alpha_i)}$, where $\alpha_i \in_R \{1, 2, \ldots, n\}$ is the receiver's choice and $f(B_i) = s + s_1 B_i + \ldots + s_k B_i^k$.
   - R sends $(B_i, B_i', f(B_i), A_i)$ to S, and simultaneously does the following proof of knowledge. $PoK\{(H(\alpha_i), s) : A = B_i^s B_i'^{H(\alpha_i)} \wedge rpk = g^s\}$.

5. **Response**: S first verifies B_i, the secret share $f(B_i)$ and the PoK by checking:

   - S checks whether B_i appears in previous session.
   - $g^{f(B_i)} \stackrel{?}{=} rpk \cdot S_1^{B_i} \cdot S_2^{B_i^2} \cdot \ldots \cdot S_k^{B_i^k}$. If this equation holds,
   - S verifies $PoK\{(H(\alpha_i), s) : A_i = B_i^s B_i'^{H(\alpha_i)} \wedge rpk = g^s\}$.

   If either of the verification fails, S aborts; Otherwise, S stores $(B_i, B_i', f(B_i), A_i)$ and S generates $D_i = A_i^z$ and sends D_i to R.

6. **Extract**: Upon receiving D_i from S, R computes $K_{\alpha_i} = D_i^{\frac{1}{r_i}}$ and extracts the intended message $m_{\alpha_i} = c_{\alpha_i}/H_1(K_{\alpha_i})$.

7. **Reveal**: Once R and S execute the OT for $k+1$ times, S obtains $k+1$ shares of the secret. S is able to recover s from secret sharing. Once s is calculated, for the previous commitments $A_i = B_i^s B_i'^{H(\alpha_i)}$, given B_i, B_i' for $1 \leq i \leq k$. S is able to retrieve α_i for $1 \leq i \leq k$.

The proof of knowledge $PoK\{(H(\alpha_i), s) : A_i = B_i^s B_i'^{H(\alpha_i)} \wedge rpk = g^s\}$ can be implemented as follows:

1. R randomly chooses two random numbers $t_1, t_2 \in \mathbb{Z}_p$, computes $T_1 = B_i^{t_1} B_i'^{t_2}$, $T_2 = g^{t_1}$, $c = H(f(B_i), B_i, B_i', T_1, T_2)$, $v_1 = t_1 - cs$ and $v_2 = t_2 - cH(\alpha_i)$. R sends v_1, v_2, T_1, T_2 to S.

2. S accepts if both $A_i^c B_i^{v_1} B_i'^{v_2} = T_1$ and $rpk^c g^{v_1} = T_2$ hold.

8.4.2 Security Analysis

Theorem 8.1. *The proposed OTRRP scheme is correct.*

Proof. The correctness checks of the proposed scheme are as follows:

1. **Correctness of PoK**: If R is honest, then R has knowledge of $H(\alpha_i)$ and s, R computes $v_1 = t_1 - cs$ and $v_2 = t_2 - cH(\alpha_i)$. S can verify correctly that:

$$A_i^c B_i^{v_1} B_i'^{v_2} = B_i^{sc} B_i'^{H(\alpha_i)c} B_i^{t_1 - cs} B_i'^{t_2 - cH(\alpha_i)} = B_i^{t_1} B_i'^{t_2} = T_1.$$

and

$$rpk^c g^{v_1} = g^{sc} g^{t_1 - cs} = g^{t_1} = T_2.$$

2. **Correctness of extracting the message**:

$$m_{\alpha_i} = \frac{c_{\alpha_i}}{H_1(K_{\alpha_i})} = \frac{m_{\alpha_i} H_1(rpk \cdot h_{\alpha_i}^{H(\alpha_i)})^z)}{H_1((g^{r_i s z} h_{\alpha_i}^{r_i H(\alpha_i) z})^{\frac{1}{r_i}})} = \frac{m_{\alpha_i} H_1(g^{sz} h_{\alpha_i}^{H(\alpha_i) z})}{H_1(g^{sz} h_{\alpha_i}^{H(\alpha_i) z})} = m_{\alpha_i}$$

Theorem 8.2. *The proposed OTRRP scheme provides receiver's privacy for honest receivers.*

Proof. Suppose an honest receiver runs the OT protocol with the sender for k times. The sender could obtain k pairs of transcripts $\{(A_1, B_1, B_1'), (A_2, B_2, B_2'), \ldots, (A_k, B_k, B_k')\}$ such that $A_1 = (g^{r_1})^s (h_{\alpha_1}^{r_1})^{H(\alpha_1)}, A_2 = (g^{r_2})^s (h_{\alpha_2}^{r_2})^{H(\alpha_2)}, \ldots, A_k = (g^{r_k})^s (h_{\alpha_k}^{r_k}$

$)^{H(\alpha_k)}$, where $\alpha_1, \alpha_2, \ldots, \alpha_k \in \{1, 2, \ldots, n\}$ are the user's choice and $r_1, r_2, \ldots, r_k \in_R \mathbb{Z}_q^*$. Given $B_j = g^{r_j}, rpk = g^s$ for some random $r_j \in \mathbb{Z}_q^*$, it is computation-infeasible to decide the masked value equals $g^{r_j s}$ or a random value Z in G_q, thus for any two transcripts A_j and A_i such that $1 \leq i \neq j \leq k$ from the user, they are computationally indistinguishable to the service provider as long as the DDH problem is hard in G_q.

Claim 1. *The proposed encryption scheme is semantic secure.*

Proof. The security proof is performed using random oracle. Suppose the simulator $\mathcal{B}$ maintains a table T_1 for the hash queries. $\mathcal{B}$ obtains $n+1$ values $Z, Y_1, Y_2, \ldots, Y_n$ from the challenge oracle $C(\cdot)$. $\mathcal{B}$ sets the one-time public key of the sender $opk = Z$ and sends $Z, Y_1, Y_2, \ldots, Y_n$ to a PPT adversary $\mathcal{A}$. Assume $\mathcal{A}$ queries on a message m_i for $1 \leq i \leq n-1$. $\mathcal{B}$ first obtain the diffie-hellman value of (Z, Y_i) with help of $DH(\cdot)$ oracle. Then $\mathcal{A}$ checks if $DH(Z, Y_i)$ has existed in T_1. If not, $\mathcal{B}$ chooses a new random $Z_i \in G_q$ and stores $(DH(Z, Y_i), Z_i)$ to T_1. Otherwise, assume $H_1(DH(Z, Y_i)) = Z_i$, $\mathcal{B}$ returns $c_i = Z_i \cdot m_i$ as the ciphertext on m_i. After $n-1$ queries, $\mathcal{A}$ sends two challenge messages m_0^*, m_1^*, $\mathcal{B}$ chooses $b \in \{0, 1\}$ and a random number $Z_n \in G_q$. $\mathcal{A}$ sets the ciphertext c_b^* on m_b^* as $c_b^* = Z_n \cdot m_b^*$. If $\mathcal{A}$ has a non-negligible probability ϵ in distinguishing c_b^* than random guess. Then with an overwhelming probability that $DH(Z, Y_n)$ has been submitted in the hash queries. Thus $\mathcal{B}$ breaks the OMDH assumption, we reach a contradiction. Therefore the proposed encryption scheme is semantic secure.

Theorem 8.3. *The proposed OTRRP scheme provides sender's privacy.*

Proof. Suppose an honest receiver runs the OT protocol with the sender k times. For any probabilistic polynomial-time malicious receiver $\hat{R}$ in the real-world model, we are able to construct a probabilistic polynomial-time malicious receiver $\hat{R}^*$ in the ideal model such that the outputs of $\hat{R}$ and $\hat{R}^*$ are indistinguishable.

Briefly, the ideal-world cheating receiver $\hat{R}^*$ can extract α from the proof of knowledge. This enables him to obtain the message m_α form the TTP. $\hat{R}^*$ simulates the honest sender S in the real-world and interacts with $\hat{R}$ as follows:

1. S sends $m_1, m_2, \ldots, m_n$ to the trusted third party TTP.
2. $\hat{R}^*$ sends $c_1^*, c_2^*, \ldots, c_n^*$ to TTP such that $c_i^* \in_R G_q$ for $i = 1, 2, \ldots, n$.
3. $\hat{R}^*$ monitors the outputs $A_{\alpha_1}, A_{\alpha_2}, \ldots, A_{\alpha_k}$ of $\hat{R}$, $\hat{R}^*$ chooses $A_{\alpha_1}^*, A_{\alpha_2}^*, \ldots, A_{\alpha_k}^* \in_R G_q$.
4. After $\hat{R}$ runs *Request* protocol, if the verification of PoK fails, $\hat{R}^*$ sends a value $\alpha_i \notin \{1, 2, \ldots, n\}$ to TTP.

5. If the verification of PoK successes, $\hat{R}^*$ extracts $\hat{R}$'s choice α_i from the PoK and gets back $D^*_{\sigma_1}, D^*_{\sigma_2}, \ldots, D^*_{\sigma_k}$ such that $D^*_{\sigma_i} = A^{z^*}_{\alpha_i}$ for $i = 1, 2, \ldots, k$.
6. If $\hat{R}$ can compute $K_{\alpha_i} = g^{sz} h_{\alpha_i}^{H(\alpha_i)z}$, $\hat{R}^*$ sends α_i to TTP, TTP returns $\frac{c^*_{\alpha_i}}{m_{\alpha_i}}$.
7. $\hat{R}^*$ outputs $(A^*_{\alpha_1}, A^*_{\alpha_2}, \ldots, A^*_{\alpha_k}, D^*_{\sigma_1}, D^*_{\sigma_2}, \ldots, D^*_{\sigma_k}, c^*_1, c^*_2, \ldots, c^*_n)$.

We can see from Theorem 8.2 and Claim 1 that $\{A_{\alpha_1}, A_{\alpha_2}, \ldots, A_{\alpha_k}\}$ and $\{c_1, c_2, \ldots, c_n\}$ are indistinguishable from random elements in G_q. In addition, the sets of $\{D_{\sigma_1}, D_{\sigma_2}, \ldots, D_{\sigma_k}\}$ and $\{D^*_{\sigma_1}, D^*_{\sigma_2}, \ldots, D^*_{\sigma_k}\}$ are identically distributed. Therefore, no distinguishers can distinguish the outputs of $\hat{R}$ and $\hat{R}'$ with a non-negligible probability.

Theorem 8.4. *The proposed OTRRP provides retrievable privacy for the receiver.*

Proof. After running the protocol $k+1$ times with the receiver, the sender obtains $k+1$ shares of the retrievable private key s with respect to the unknown integers $s_1, s_2, \ldots, s_k$ such that

$$f(B_i) = s + s_1 B_i + s_2 B_i^2 \ldots + s_k B_i^k, 1 \leq i \leq k+1.$$

The corresponding linear equations in a matrix form are as follows:

$$\begin{pmatrix} 1 & B_1 & B_1^2 & \cdots & B_1^k \\ 1 & B_2 & B_2^2 & \cdots & B_2^k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & B_{k+1} & B_{k+1}^2 & \cdots & B_{k+1}^k \end{pmatrix} * \begin{pmatrix} s \\ s_1 \\ \vdots \\ s_k \end{pmatrix} = \begin{pmatrix} f(B_1) \\ f(B_2) \\ \vdots \\ f(B_{k+1}) \end{pmatrix}$$

As we can see the coefficient matrix is a Vandermonde matrix or a non-singular matrix. The determinant of such a matrix is not equal to zero. Thus the equations have a unique solution to $s, s_1, s_2, \ldots, s_k$.
Once the sender obtains the value of the retrievable private key rsk. For previous commitments on receiver's choice $A_i = B_i^{rsk} B_i'^{H(\alpha_i)}$ for $1 \leq i \leq k$. Since S has store the values of B_i and B_i' in the i-th round. Thus, the sender could trace the receiver choice $\alpha_i = j$ in the i-th round by checking that $A_i = B_i^{rsk} B_i'^{H(\alpha_i)} = B_i^{rsk} B_i'^{H(j)}$ for $1 \leq j \leq n$.

8.5 Our Second Scheme and Security Analysis

In this section, we propose another oblivious transfer scheme with retrievable receiver's privacy which we will use together with a blind signature scheme to construct our second e-coupon systems in next chapter.

8.5.1 The Second Proposed OTRRP Scheme

The proposed oblivious transfer scheme consists a tuple of PPT algorithms as follows:

1. **Setup**: On input a security parameter $\kappa \in \mathbb{N}$, a trusted third party generates the system parameters $params = (G_q, g, h, p, q, H)$, where G_q is the subgroup of $\mathbb{Z}_p$ with prime order q and $p = 2q + 1$ is also prime, g, h are generators of G_q. $H : \{0,1\}^* \rightarrow G_q$ is a collision resistant hash function.

2. **KeyGen**: A user $\mathcal{U}$ chooses a random number $x \in \mathbb{Z}_q^*$ and sets the key pair $(rpk, rsk) = (g^x, x)$. $\mathcal{U}$ chooses k random numbers $s_1, s_2, \ldots, s_k \in_R \mathbb{Z}_q$ and computes $S_1 = g^{s_1}, S_2 = g^{s_2}, \ldots, S_k = g^{s_k}$. $\mathcal{U}$ publishes rpk and $S_1, S_2, \ldots, S_k$. The service provider S randomly chooses $y \in \mathbb{Z}_q^*$ and sets the public key as $pk = g^y$.

3. **Request**: In the i-th round,

   - R chooses $r_i \in_R \mathbb{Z}_q^*$, and computes $B_i = g^{r_i}$ and $A_i = B_i^x h^{\alpha_i}$ where α_i is its choice, and $f(B_i) = x + s_1 B_i + \ldots s_k B_i^k$.
   - R sends $(B_i, f(B_i), A_i)$ to S, and simultaneously does the following PoK: $PoK\{(\alpha_i, s) : A_i = B_i^x h^{\alpha_i} \wedge rpk = g^s\}$.

4. **Response**: S first verifies B_i, the secret share $f(B_i)$ and the PoK by checking:

   - B_i has not been used in a previous session;
   - $g^{f(B_i)} = rpk \cdot S_1^{B_i} \cdot S_2^{B_i^2} \cdot \ldots \cdot S_k^{B_i^k}$; and
   - $PoK\{(\alpha, x) : A_i = B_i^x h^{\alpha_i} \wedge rpk = g^x\}$ is valid.

   If any verification fails, S aborts; otherwise, S stores $(B_i, f(B_i), A_i)$ and

   - generates $c_i = ((rpk)^{k_i}, m_i(A_i/h^i)^{k_i})$ where $k_i \in_R \mathbb{Z}_q$, for $1 \leq i \leq n$; and
   - sends $(c_1, c_2, \ldots, c_n)$ to R.

   Then R can obtain $m_{\alpha_i} = m_{\alpha_i}(A_i/h^{\alpha_i})^{k_{\alpha_i}}/((rpk)^{k_{\alpha_i}})^{r_i}$.

5. **Reveal**: Once R and S execute the OT for $k + 1$ times, S obtains $k + 1$ shares of the secret x. Then S is able to calculate x, and retrieve h^{α_i} from $A_i = B_i^x h^{\alpha_i}$.

The proof of knowledge $PoK\{(\alpha_i, x) : A_i = B_i^x h^{\alpha_i} \wedge rpk = g^x\}$ can be done non-interactively as follows:

1. R randomly chooses two random numbers $t_1, t_2 \in \mathbb{Z}_p$, computes $T_1 = B_i^{t_1} h^{t_2}, T_2 = g^{t_1}$, $c_i' = H(f(B_i), B_i, T_1, T_2)$, $s_1' = t_1 - c_i' x$ and $s_2' = t_2 - c_i' \alpha_i$. R sends s_1', s_2', T_1, T_2 to S.

2. S accepts if both $A_i^{c_i'} B_i^{s_1'} h^{s_2'} = T_1$ and $rpk^{c_i'} g^{s_1'} = T_2$ hold.

8.5.2 Security Analysis

We analyze the security of the proposed oblivious transfer scheme under half-simulation model [NP05].

Theorem 8.5. *The proposed OTRRP scheme is correct.*

Proof. The correctness checks of the proposed scheme are as follows.

1. **Correctness of PoK**:

$$A_i^{c_i'} B_i^{s_1'} h^{s_2'} = B_2^{x c_i'} h^{\alpha_i c_i'} B_i^{t_1 - c_i' x} h^{t_2 - c_i' \alpha_i} = B_i^{t_1} h^{t_i} = T_1.$$

and

$$rpk^{c'} g^{s_1'} = g^{x c'} g^{t_1 - c' x} = g^{t_1} = T_2.$$

2. **Correctness of Extracting the Message**:

$$\begin{aligned} & m_{\alpha_i} (A_{\alpha_i} / h^{\alpha_i})^{k_{\alpha_i}} / (rpk)^{k_{\alpha_i} r_{\alpha_i}} \\ &= (B_{\alpha_i}^x h^{\alpha_i} / h^{\alpha_i})^{k_{\alpha_i}} / (rpk)^{k_{\alpha_i} r_{\alpha_i}} \\ &= m_{\alpha_i} g^{x r_{\alpha_i} k_{\alpha_i}} / g^{x r_{\alpha_i} k_{\alpha_i}} \\ &= m_{\alpha_i} \end{aligned}$$

Theorem 8.6. *The proposed OTRRP scheme provides receiver's privacy for honest receivers.*

Proof. Suppose an honest receiver runs the OT protocol with the sender for k times. The sender could obtain k pairs of transcripts $\{(A_1, B_1, f(B_1)), (A_2, B_2, f(B_2)), \ldots, (A_k, B_k, f(B_k))\}$ such that $A_1 = g^{r_1 x} h^{\alpha_1}, A_2 = g^{r_2 x} h^{\alpha_2}, \ldots, A_k = g^{r_k x} h^{\alpha_k}$, where $\alpha_1, \alpha_2, \ldots, \alpha_k \in \{1, 2, \ldots, n\}$ are the user's choice and $r_1, r_2, \ldots, r_k \in_R \mathbb{Z}_q^*$. Given $B_j = g^{r_j}, rpk = g^x$ for some random $r_j \in \mathbb{Z}_q^*$, it is computation-infeasible to decide the masked value equals $g^{r_j x}$ or a random value Z in G_q, thus for any two transcripts A_j and A_i such that $1 \leq i \neq j \leq k$ from the user, they are computationally indistinguishable to the service provider as long as the DDH problem is hard in G_q.

Claim 2. *The proposed encryption scheme is semantic secure.*

Proof. As can be seen in the proposed OT scheme, the cipertext is $c_i = ((rpk)^{k_i}, m_i (A_i/h^i)^{k_i})$ where $k_i \in_R \mathbb{Z}_q$, for $1 \leq i \leq n$. The proposed encryption scheme in our OT scheme is a variant of ElGamal encryption. Therefore the encryption scheme is semantic secure.

Theorem 8.7. *The proposed OTRRP scheme provides sender's privacy.*

Proof. Suppose an honest receiver runs the OT protocol with the sender k times. For any probabilistic polynomial-time malicious receiver $\hat{R}$ in the real-world model, we are able to construct a probabilistic polynomial-time malicious receiver $\hat{R}^*$ in the ideal model such that the outputs of $\hat{R}$ and $\hat{R}^*$ are indistinguishable.

Briefly, the ideal-world cheating receiver $\hat{R}^*$ can extract α from the proof of knowledge. This enables him to obtain the message m_α form the TTP. $\hat{R}^*$ simulates the honest sender S in the real-world and interacts with $\hat{R}$ as follows:

1. S sends $m_1, m_2, \ldots, m_n$ to the trusted third party TTP.
2. $\hat{R}^*$ sends $c_1^*, c_2^*, \ldots, c_n^*$ to TTP such that $c_i^* \in_R G_q$ for $i = 1, 2, \ldots, n$.
3. $\hat{R}^*$ monitors the outputs $A_{\alpha_1}, A_{\alpha_2}, \ldots, A_{\alpha_k}$ of $\hat{R}$, $\hat{R}^*$ chooses $A_{\alpha_1}^*, A_{\alpha_2}^*, \ldots, A_{\alpha_k}^* \in_R G_q$.
4. After $\hat{R}$ runs *Request* protocol, if the verification of PoK fails, $\hat{R}^*$ sends a value $\alpha_i \notin \{1, 2, \ldots, n\}$ to TTP.
5. If the verification of PoK successes, $\hat{R}^*$ extracts $\hat{R}$'s choice α_i from the PoK and gets back $c_{\sigma_1}^*, c_{\sigma_2}^*, \ldots, c_{\sigma_k}^*$ such that $c_{\sigma_i}^* \in_R G_q$ for $i = 1, 2, \ldots, k$.
6. If $\hat{R}$ can compute $g^{xr\alpha_i}$, $\hat{R}^*$ sends α_i to TTP, TTP returns $\frac{c_{\alpha_i,2}^*}{m_{\alpha_i}}$.
7. $\hat{R}^*$ outputs $(A_{\alpha_1}^*, A_{\alpha_2}^*, \ldots, A_{\alpha_k}^*; c_1^*, c_2^*, \ldots, c_n^*)$.

We can see from Theorem 8.6 and Claim 2 that $\{A_{\alpha_1}, A_{\alpha_2}, \ldots, A_{\alpha_k}\}$ and $\{c_1, c_2, \ldots, c_n\}$ are indistinguishable from random elements in G_q. Therefore, no distinguishers can distinguish the outputs of $\hat{R}$ and $\hat{R}'$ with a non-negligible probability.

Theorem 8.8. *The proposed OTRRP provides retrievable privacy for the receiver.*

Proof. The security proof is similar to that in Theorem 8.4. Thus we omit it.

8.6 Summary

In this chapter, we proposed two novel oblivious transfer schemes that can achieve retrievable receiver's privacy without the help of a trusted third party. The misbehaving receivers' choices could be traced while the honest receivers' privacy is unconditionally protected. We also proved the security of the schemes under the proposed security model.

Chapter 9

Two E-Coupon Systems with Strong User Privacy

In this chapter, we propose another two e-coupon system allowing multiple use of an e-coupon. To be specific, the propose e-coupon systems can achieve the following properties:

1. The coupon issuer (or service provider) can trace the identity of a dishonest user while the identity privacy (or anonymity) of an honest user is still well protected.

2. An honest user's redemption privacy (i.e., the items chosen when redeeming an e-coupon) is well protected from the service provider.

3. If a dishonest user redeems an e-coupon for more than the pre-determined number of times, then the user will lose the redemption privacy (i.e., all the choices the user has made in the previous redemptions can be revealed).

Our first e-coupon system achieves the first two properties without the involvement of any trusted third party. Then we use a novel oblivious transfer scheme proposed in the previous chapter to construct the second e-coupon system that can achieve all the properties given above. Compared with the e-coupon system in Chapter 7, two major improvements have been made in the proposed e-coupon systems. First, the proposed e-coupon systems allow an honest user to redeem a valid coupon for a fix number of times (more than twice). Besides, the adoption of an OTRRP scheme in the second scheme makes it possible to trace the choices of the dishonest users in addition to their identities. We define the formal security models for these new security requirements, and show that our new e-coupon systems are proven secure in the proposed models.

9.1 Introduction

We have constructed a new e-coupon system with new properties in Chapter 7. The proposed e-coupon system can ensure desired security properties such as unforgeability, anonymity, double-redemption detection and user privacy (redemption privacy). However, it only permits an honest user to spend a valid coupon twice, which might limit its usage in practice. In addition, another interesting feature

that has not been considered in our previous e-coupon system is traceability against user's choice (redemption privacy). That is, if a user is detected as a malicious user, then all the previous choices related to his coupon could be traced by the coupon issuer.

Though some existing e-coupon systems [CES+05, Ngu06b, CGH06] have already supported multiple use of an electronic coupon, the new properties of redemption privacy and traceability have not been considered in these systems. Chen et al.'s e-coupon system [CES+05] allows a user to redeem a coupon for a predefined number of times. Their system can provide unlinkability among different redemptions performed by a user using the same multi-coupon. Meanwhile, the number of remaining redemptions can also be hidden from the coupon issuer. To reduce the cost for issuing and redeeming coupons, Nguyen [Ngu06b] later presented another more efficient e-coupon system which has constant communication and computation costs. Nguyen's e-coupon system provides another interesting feature named revocability, which allows the coupon issuer to revoke (i.e., terminate) an e-coupon. In an independent work [CGH06], Canard et al. also proposed another multi-coupon system with a different feature: a coupon holder can transfer some values in his/her multi-coupon to another user.

Our Contributions. In this chapter, we propose two new e-coupon systems, which are based on a blind signature scheme and a new OT scheme. Our first e-coupon system can achieve unforgeability, anonymity for honest users, k-time redemption detection, traceability against dishonest users' identities, and user privacy. The user privacy in our first e-coupon system is achieved by employing an normal OT scheme in the redemption phase.

In order to trace the items redeemed by a dishonest user, we apply a novel OT scheme proposed in last chapter which has the following properties:

1. The privacy (i.e., choices) of an honest receiver is well protected against the sender.

2. The receiver cannot gain any information other than his choice during the OT protocol.

3. If the receiver runs the OT protocol for more than the pre-determined number of times, then all the previous choices made by the receiver can be revealed by the sender.

As we will show later, our OT scheme incurs very little computation and communication overhead in order to achieve the third (new) property. We then use the new OT scheme to construct the second e-coupon system which allows both the identity and the choices of a dishonest user to be revealed.

Organization of This Chapter. The rest of this chapter is organized as follows. We provide some definitions in Section 9.2. The formal security models for our e-coupon systems are presented in Section 9.3. We then present our first construction with security analysis in Section 9.4. The second construction and its security analysis are presented in Section 9.5. We compare our e-coupon systems with existing systems and analyze the efficiency of them in section 9.6 and this chapter is concluded in Section 9.7.

9.2 Formal Definition

We present the formal definition of our e-coupon system in this section. An e-coupon system involves two participants, namely, a user $\mathcal{U}$ and a coupon issuer $\mathcal{S}$, which is also a service provider. An e-coupon system consists of a tuple of non-interactive or interactive probabilistic polynomial-time (PPT) algorithms as follows.

1. **Setup**: On input of a security parameter κ, a trusted third party runs the setup algorithm and generates the system parameters.

$$params \leftarrow \mathbf{Setup}(1^{\kappa}).$$

2. **KeyGen**: On input of the public parameter $params$, the key generation algorithm outputs a key pair for a user or a service provider. It is worth noticing that we assume a public key infrastructure (PKI) binding the public key to user's identity in our system.

$$(pk_{\mathcal{U},sk_{\mathcal{U}}}),\ (pk_{\mathcal{S}}, sk_{\mathcal{S}}) \leftarrow \mathbf{KeyGen}(params).$$

3. **Issue**: The issue algorithm is an interactive protocol between the service provider $\mathcal{S}$ and a user $\mathcal{U}$, the output is an e-coupon for the user.

$$C \leftarrow \mathbf{Issue}(\mathcal{S}(pk_{\mathcal{S}}, sk_{\mathcal{S}}, params), \mathcal{U}(pk_{\mathcal{U}}, params)).$$

4. **Redeem**: The redeem algorithm is an interactive protocol between a user $\mathcal{U}$ and the service provider $\mathcal{S}$, taking as input an e-coupon C, the public key $pk_{\mathcal{S}}$ of the service provider and the public parameters $params$. Suppose $(m_1, m_2, \ldots, m_n)$ is a set of goods or services possessed by $\mathcal{S}$, where n is the number of services decided by $\mathcal{S}$. The output of this algorithm for the service provider is accept or reject.

$$\text{`}accept\text{'} \text{ or } \text{`}reject\text{'} \leftarrow \mathbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C).$$

The output of this algorithm for the user is the service m_i of his choice or a failure symbol.

$$\text{`}\perp\text{'} \text{ or } m_i \leftarrow \mathbf{Redeem}_{\mathcal{U}}(pk_{\mathcal{S}}, params, C).$$

5. **Reveal**: The reveal algorithm is executed by the service provider, taking a sequence of the challenge-response pairs and the corresponding redeemed coupon $\{(c_1, R_1), (c_2, R_2), \ldots, (c_{k+1}, R_{k+1}); C\}$ and the public parameters $params$ as input, and outputs the identity $ID_{\mathcal{U}}$ of a user or a failure symbol '$\perp$'.

$$ID_{\mathcal{U}} \text{ or } \text{`}\perp\text{'} \leftarrow \mathbf{Reveal}((c_1, R_1), (c_2, R_2), \ldots, (c_{k+1}, R_{k+1}), C, params).$$

If the e-coupon system also supports traceability of items against dishonest users, then the reveal algorithm returns $(ID_{\mathcal{U}}, \{m_1, m_2, \ldots, m_{k+1}\})$ or '$\perp$'.

Correctness: We require that for any security parameter $\kappa \in \mathbb{N}$, if $params \leftarrow \mathbf{Setup}(1^{\kappa})$, $(pk_{\mathcal{U}}, sk_{\mathcal{U}}) \leftarrow \mathbf{KeyGen}(params)$, $(pk_{\mathcal{S}}, sk_{\mathcal{S}}) \leftarrow \mathbf{KeyGen}(params)$, $C \leftarrow \mathbf{Issue}(\mathcal{S}(pk_{\mathcal{S}}, sk_{\mathcal{S}}), \mathcal{U}(pk_{\mathcal{U}}, params))$, then

- $$\Pr(\text{`}accept\text{'} \leftarrow \mathbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C, (c, R))) = 1$$

  and

  $$\Pr(item_i \leftarrow \mathbf{Redeem}_{\mathcal{U}}(pk_{\mathcal{S}}, params, C, (c, R))) = 1.$$

- If the user redeems the coupon for at most k times, then the service provider cannot obtain any private information about the user.

  $$\Pr(\text{`}\perp\text{'} \leftarrow \mathbf{Reveal}((c_1, R_1), (c_2, R_2), \ldots, (c_{\delta}, R_{\delta}); C; params) |\ \delta \leq k) = 1.$$

- If the user redeems the coupon more than k times, then the service provider can trace the identity (and all the choices, if supported) of the user.

  $$\Pr(ID_{\mathcal{U}} \leftarrow \mathbf{Reveal}((c_1, R_1), (c_2, R_2), \ldots, (c_{\delta}, R_{\delta}); C; params) | k < \delta) = 1$$

  or

  $$\Pr((ID_{\mathcal{U}}, \{m_1, m_2, \ldots, m_{\delta}\}) \leftarrow \mathbf{Reveal}((c_1, R_1), (c_2, R_2), \ldots, (c_{\delta}, R_{\delta}); C;$$
  $$params) |\ k < \delta) = 1.$$

9.3 Security Model

We formalize four security requirements for our e-coupon system, that are unforgeability, user anonymity, k-time redemption detection, and user privacy.

Unforgeability

The adversarial game of unforgeability is same as that in Section 7.3, thus we omit it. Define the advantage of a adversary $\mathcal{A}$ in winning the unforgeability game as

$$\mathbf{Adv}_{\mathcal{A}}^{unf}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}]$$

Definition 9.1. *An e-coupon system is said to be unforgeable if* $\boldsymbol{Adv}_{\mathcal{A}}^{unf}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

Anonymity

The adversarial game regarding anonymity is same as that in Section 7.3, thus we omit it. Define the advantage of a adversary $\mathcal{A}$ in winning the game as

$$\mathbf{Adv}_{\mathcal{A}}^{Ano}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}] - \frac{1}{2}$$

Definition 9.2. *An e-coupon system is said to provide anonymity if* $\boldsymbol{Adv}_{\mathcal{A}}^{Ano}(k)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

k-time Redemption Detection

Detection of over spending is a major concern for any digital coupon system. An e-coupon system is said to provide k-time redemption detection if one user cannot redeem one coupon more than pre-determined number of times with the same service provider without being caught.The adversarial game for k-time redemption detection is defined as follows:

1. **Setup**: The simulator $\mathcal{B}$ runs algorithm **Setup** to generate public parameters $params$.

2. **KeyGen**: The simulator $\mathcal{B}$ generates two key pairs $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ and $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$, $\mathcal{B}$ sends $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ and $pk_{\mathcal{S}}$ to $\mathcal{A}$.

3. **Issue queries**: Assume $\mathcal{A}$ makes q_d coupon issuing queries. $\mathcal{S}$ runs the **Issue** algorithm with $\mathcal{A}$ to issue a sequence of coupons $\{C_1, C_2, \ldots, C_{q_d}\}$ for $\mathcal{A}$.

4. **Redeem queries**: $\mathcal{A}$ runs the redeem protocol with $\mathcal{S}$ with any coupon of his choice.

5. **Challenge**: $\mathcal{A}$ outputs a sequence $(C^*, (c_1^*, R_1^*), (c_2^*, R_2^*), \ldots, (c_\delta^*, R_\delta^*))$ such that $k < \delta$. We say $\mathcal{A}$ wins the game if

   - $\mathbf{Redeem}_{\mathcal{S}}(pk_{\mathcal{S}}, params, C^*, (c_i^*, R_i^*)) = 1$ for $1 \leq i \leq \delta$.
   - '$\perp$'$\leftarrow$**Reveal**$((c_1^*, R_1^*), (c_2^*, R_2^*), \ldots, (c_\delta^*, R_\delta^*), C^*, params)$

Define the advantage of $\mathcal{A}$ in winning the adversarial game above as

$$\mathbf{Adv}_{\mathcal{A}}^{krd}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}]$$

Definition 9.3. *An e-coupon system is said to provide k-time redemption detection if* $\boldsymbol{Adv}_{\mathcal{A}}^{krd}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

User privacy

The adversarial game regarding user privacy is same as that in Section 7.3, thus we omit it. Define the success probability of the adversary $\mathcal{A}$ in breaking users' privacy as

$$\mathbf{Adv}_{\mathcal{A}}^{up}(\kappa) = \Pr[\mathcal{A} \text{ wins the game}] - \frac{1}{n}$$

Definition 9.4. *An e-coupon system is said to provide user privacy if* $\boldsymbol{Adv}_{\mathcal{A}}^{up}(\kappa)$ *is negligible for any PPT adversary* $\mathcal{A}$*.*

9.4 Our First Privacy-preserving E-coupon System

We apply a blind signature scheme that involves a challenge-response verification to ensure that if a coupon is redeemed more than pre-determined number of times then the identity of the misbehaving user could be traced. We combine the proposed blind signature with the oblivious transfer scheme in [CT05] to construct the first e-coupon system with user privacy.

9.4.1 PPE-COUPON-I

We denote in our system the service provider by $\mathcal{S}$ and a user by $\mathcal{U}$. Denote $\{m_1, m_2, \ldots, m_n\}$ the set of items that can be redeemed. A user with a valid coupon is allowed to make $k < n$ choices from them. The detailed description of our e-coupon system is as follows.

1. **Setup**: On input of a security parameter $\kappa \in \mathbb{N}$, a trusted party generates the system parameters $params = (G_q, g, p, q, H_1, H_2)$, where G_q is the subgroup of $\mathbb{Z}_p$ with prime order q and g is a generator of G_q, where $p = 2q + 1$ is also prime. $H_1 : \{0,1\}^* \rightarrow G_q$ and $H_2 : G_q \rightarrow \{0,1\}^\kappa$ are two collision-resistant hash functions.

2. **KeyGen**: A user $\mathcal{U}$ chooses a random number $x \in \mathbb{Z}_q^*$ and sets the key pair $(pk_\mathcal{U}, sk_\mathcal{U})$. $\mathcal{U}$ publishes $pk_\mathcal{U} = g^x$ and keeps $sk_\mathcal{U} = x$ secret. The service provider S randomly chooses $y \in \mathbb{Z}_q^*$ and sets the public key as $pk_\mathcal{S} = Y = g^y$.

3. **Issue**: The issue protocol is performed by interactive communications though a secure channel between the service provider $\mathcal{S}$ and a user $\mathcal{U}$. The result of the issue protocol is that $\mathcal{S}$ generates a valid coupon for a user $\mathcal{U}$. We assume there exists a PKI binding the public keys to the identities of users in the system.

   - On receiving a request from $\mathcal{U}$, $\mathcal{S}$ chooses a random number $k' \in_R \mathbb{Z}_q^*$ and computes $\delta_1 \leftarrow pk_\mathcal{U}^{k'}$ and $\delta_2 \leftarrow g^{k'}$ sends (δ_1, δ_2) to $\mathcal{U}$
   - After receiving (δ_1, δ_2) from $\mathcal{S}, \mathcal{U}$ checks whether $\delta_1 = \delta_2^{sk_\mathcal{U}}$. If the verification fails, $\mathcal{U}$ stops; otherwise, $\mathcal{U}$ chooses $x_1 \in \mathbb{Z}_q^*$ and computes $\alpha \leftarrow (g^{xy})^{x_1}$, $\beta \leftarrow (g^x)^{x_1}$, $\lambda = g^{x_1}$ and $m \leftarrow H_1(\alpha, \beta, \lambda)$. $\mathcal{U}$ chooses two different random number $a, b \in \mathbb{Z}_q^*$ and computes $r \leftarrow m\beta^a \delta_1^{\frac{bx_1}{a}}$ and $m' \leftarrow \frac{aH_1(m,r)}{b}$, $\mathcal{U}$ sends m' to $\mathcal{S}$.
   - $\mathcal{S}$ computes $s' = m'y + k'$ and sends s' to $\mathcal{U}$.
   - $\mathcal{U}$ verifies if $g^{s'} \equiv Y^{m'}\delta_2 \mod p$, if the equation holds, $\mathcal{U}$ removes the blind factor b by calculating $s = \frac{s'b}{a} + a$; otherwise, abort. $\mathcal{U}$ stores $(\alpha, \beta, \lambda, r, s)$ and the secret x_1 in a safe place (i.e. tamper proof memory).

4. **Redeem**: At the start of the redeem phase, $\mathcal{U}$ chooses k random numbers $s_1, s_2, \ldots, s_k \in_R \mathbb{Z}_q$ and computes $S_1 = g^{s_1}, S_2 = g^{s_2}, \ldots, S_k = g^{s_k}$. $\mathcal{U}$ sends $S_1, S_2, \ldots, S_k$ to the service provider $\mathcal{S}$. S stores these commitments $S_1, S_2, \ldots, S_k$. In the i-th round, the redeem protocol is performed as follows:

   - After receiving a redeem request from the user, $\mathcal{S}$ generates a challenge $c_i = H_1(ID_\mathcal{S}||Date||Time)$ and sends c_i to $\mathcal{U}$, where $ID_\mathcal{S}$ is the identity of the service provider.
   - After receiving the challenge c_i from the service provider and $\mathcal{U}$ computes $R_i = x_1 + s_1c_i + s_2c_i^2 + \ldots + s_kc_i^k$ and choose $\sigma_i \in \{1, 2, \ldots, n\}$, where σ_i is the index of the service, and a random number $b_i \in \mathbb{Z}_q^*$, $\mathcal{U}$ computes $w_{\sigma_i} = H_1(\sigma_i)$ and $A_{\sigma_i} = w_{\sigma_i} g^{b_i}$. $\mathcal{U}$ sends $(c_i, R_i, \alpha, \beta, \lambda, r, s)$ and A_{σ_i} to $\mathcal{S}$.

- $\mathcal{S}$ checks if
  - (c_i, R_i) has appeared in a previous session.
  - $H_1(\alpha, \beta, \lambda) \stackrel{?}{=} \beta^{-s}\alpha^{H_1(H_1(\alpha,\beta,\lambda),r)}r$.
  - $g^{R_i} \stackrel{?}{=} \lambda S_1^{c_i} S_2^{c_i^2} \ldots S_k^{c_i^k}$.

  If either of the check fails, abort; Otherwise, $\mathcal{S}$ computes $D_{\sigma_i} = {A_i}^y$, $w_j = H_1(i)$ and $c_j = m_i \oplus H_2(w_i^y)$, $i = 1, 2, \ldots, n$. $\mathcal{S}$ sends D_{σ_i} and $c_1, c_2, \ldots, c_n$ to $\mathcal{U}$.
- $\mathcal{U}$ computes $K_{\sigma_i} = D_{\sigma_i}/Y^{b_i}$ and recover $m_{\sigma_i} = c_{\sigma_i} \oplus H_2(K_{\sigma_i})$.

5. **Reveal**: Assume the coupon $C = (\alpha, \beta, \lambda, r, s)$ is redeemed by a dishonest user $\mathcal{U}'$ for $k+1$ times, $\mathcal{S}$ could obtain $k+1$ shares about the secret x_1. Once $\mathcal{S}$ obtains the value of x_1, $\mathcal{S}$ could trace the identity of $\mathcal{U}'$ by making an exhaustive search in his database to determine the public key of the misbehaving user such that ${pk_{\mathcal{U}'}}^{x_1} = \beta$.

6. **Correctness**: The correctness check for validity of the coupon is as follows:

$$
\begin{aligned}
&\beta^{-s}\alpha^{H_1(H_1(\alpha,\beta,\lambda),r)}r \\
&= (pk_{\mathcal{U}}^{x_1})^{-s}(g^{xyx_1})^r r \\
&= (pk_{\mathcal{U}}^{x_1})^{-H_1(H_1(\alpha,\beta,\lambda),r)y-\frac{k'b}{a}-a}(pk_{\mathcal{U}}^{x_1})^{H_1(H_1(\alpha,\beta,\lambda),r)y} \\
&\quad m(pk_{\mathcal{U}}^{x_1})^a(pk_{\mathcal{U}}^{x_1})^{\frac{k'b}{a}} \\
&= m \\
&= H_1(\alpha, \beta, \lambda)
\end{aligned}
$$

The correctness check for a user $\mathcal{U}$ to recover the correct message is as follows:

$$
\begin{aligned}
&c_{\sigma_i} \oplus H_2(K_{\sigma_i}) \\
&= m_{\sigma_i} \oplus H_2(w_{\sigma_i}^y) \oplus H_2(A_{\sigma_i}^y/Y^{b_i}) \\
&= m_{\sigma_i} \oplus H_2(w_{\sigma_i}^y) \oplus H_2((w_{\sigma_i}g^{b_i})^y/Y^{b_i}) \\
&= m_{\sigma_i} \oplus H_2(w_{\sigma_i}^y) \oplus H_2((w_{\sigma_i})^y) \\
&= m_{\sigma_i}
\end{aligned}
$$

9.4.2 Security Analysis

In this section, we prove the security of PPE-COUPON-I under the proposed security models.

Theorem 9.1. *The proposed PPE-COUPON-I system is unforgeable.*

Proof. The security analysis is similar as that in theorem 7.1, thus we omit it.

Theorem 9.2. *The proposed PPE-COUPON-I system provides anonymity.*

Proof. The security analysis is similar as that in 7.2, thus we omit it.

Theorem 9.3. *The proposed PPE-COUPON-I system provides k-time redemption detection.*

Proof. According to our e-coupon system, if $\mathcal{U}$ has redeemed a coupon $(\alpha, \beta, \lambda, r, s)$ for $k+1$ times, then $\mathcal{B}$ obtains $k+1$ different challenge-response pairs $(c_1, R_1), (c_2, R_2) \ldots, (c_{k+1}, R_{k+1})$ where

$$R_i = x_1 + s_1 c_i + s_2 c_i^2 \ldots + s_k c_i^k, 1 \leq i \leq k+1.$$

We can represent these linear equations in the matrix form as follows:

$$\begin{pmatrix} 1 & c_1 & c_1^2 & \cdots & c_1^k \\ 1 & c_2 & c_2^2 & \cdots & c_2^k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & c_{k+1} & c_{k+1}^2 & \cdots & c_{k+1}^k \end{pmatrix} * \begin{pmatrix} x_1 \\ s_1 \\ \vdots \\ s_k \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_{k+1} \end{pmatrix}$$

Since the challenges $c_1, c_2, \ldots, c_{k+1}$ are random integers, the coefficient matrix is a Vandermonde matrix. Thus the equations have a unique solution for $x_1, s_1, s_2, \ldots, s_k$. Once x_1 is calculated by $\mathcal{S}$, $\mathcal{S}$ could trace the identity of the misbehaving user $\mathcal{U}'$ by making an exhaustive search in his database to determine the public key of the misbehaving user such that ${pk_{\mathcal{U}'}}^{x_1} = \beta$.
It remains to show that the public key revealed in such a method is the correct public key of the misbehaving user. The proof is straightforward. Since the coupon issuer has specified the public key $pk_{\mathcal{U}'}$ in the pair (δ_1, δ_2). Only the user $\mathcal{U}'$ possessing the corresponding private key $sk_{\mathcal{U}'}$ can obtain a valid coupon (A cheating user without knowledge of $sk_{\mathcal{U}'}$ can not compute the component α of a coupon). In this way, it can be ensured that a public key revealed through this method is indeed the public key of the misbehaving user.

Theorem 9.4. *The proposed PPE-COUPON-I system provides unconditional user privacy.*

Proof. The security analysis regarding user privacy is similar as that in theorem 7.4, thus we omit it.

9.5 Our Second Privacy-Preserving E-coupon System

We construct the second e-coupon system that can achieve traceability against a misbehaving user's choices in addition to all the security properties mentioned above. Since the user's privacy relies on oblivious transfer in our construction, in order to reveal a misbehaving user's privacy, we apply one of the new OT schemes with retrievable receiver's privacy proposed in last chapter.

9.5.1 PP-ECOUPON-II

We combine the proposed blind signature and oblivious transfer schemes to construct the second e-coupon system that can achieve all the desirable properties.

1. **Setup**: On input a security parameter $\kappa \in \mathbb{N}$, a trusted third party generate the system parameters $paras = (G_q, g, h, p, q, H_1)$, where G_q is the subgroup of $\mathbb{Z}_p$ with prime order q and $p = 2q + 1$ is also prime, g, h are generators of G_q, and $H_1 : \{0,1\}^* \rightarrow G_q$ is a collision-resistant hash function.

2. **KeyGen**: On input a security parameter $\kappa \in \mathbb{N}$ and the public parameter $params$, randomly choose $x, y \in_R \mathbb{Z}_q^*$ and calculate g^x, g^y and output the private and public key pairs $(sk_{\mathcal{U}} = x, pk_{\mathcal{U}} = g^x)$ and $(sk_{\mathcal{S}} = y, pk_{\mathcal{S}} = Y = g^y)$ for the user and service provider respectively.

3. **Issue**: The issue protocol is performed through interactive communications between the service provider $\mathcal{S}$ and a user $\mathcal{U}$. The result of the issue protocol is that $\mathcal{S}$ generates a valid coupon for a user $\mathcal{U}$.

   - On receiving a request from $\mathcal{U}$, $\mathcal{S}$ chooses a random number $k' \in_R \mathbb{Z}_q^*$ and computes $\delta_1 \leftarrow pk_{\mathcal{U}}^{k'}$ and $\delta_2 \leftarrow g^{k'}$ sends (δ_1, δ_2) to $\mathcal{U}$.
   - After receiving (δ_1, δ_2) from $\mathcal{S}, \mathcal{U}$ checks whether $\delta_1 = \delta_2^{sk_{\mathcal{U}}}$. If the verification fails, $\mathcal{U}$ stops; otherwise, $\mathcal{U}$ chooses $x_1 \in \mathbb{Z}_p^*$ and computes $\alpha \leftarrow (g^{xy})^{x_1}$, $\beta \leftarrow (g^x)^{x_1}$ and $\lambda = g^{x_1}$, $m \leftarrow H_1(\alpha, \beta, \lambda)$. $\mathcal{U}$ chooses two different random number a, b and computes $r \leftarrow m\beta^a \delta_1^{\frac{bx_1}{a}}$ and $m' \leftarrow \frac{aH_1(m,r)}{b}$, $\mathcal{U}$ sends m' to $\mathcal{S}$.
   - $\mathcal{S}$ computes the signature $s' = m'y + k'$ on the blinded message m' and sends s' to $\mathcal{U}$.
   - $\mathcal{U}$ verifies if $g^{s'} \equiv Y^{m'} \delta_2 \mod p$, if the equation holds, $\mathcal{U}$ removes the blind factor b by calculating $s = \frac{s'b}{a} + a$. Otherwise, abort. $\mathcal{U}$ stores $(\alpha, \beta, \lambda, r, s)$ and x_1 in a safe place (i.e. tamper proof memory);

4. **Redeem**: $\mathcal{U}$ chooses k random number $s_1, s_2, \ldots, s_k \in_R \mathcal{G}_q$ and computes $S_1 = g^{s_1}, S_2 = g^{s_2}, \ldots, S_k = g^{s_k}$. $\mathcal{U}$ sends $S_1, S_2, \ldots, S_k$ to $\mathcal{S}$ at the start of the redeem phase. $\mathcal{S}$ stores these commitments $S_1, S_2, \ldots, S_k$. In the i-th round, the redeem protocol is performed as follows:

   - After receiving a redeem request from the user, $\mathcal{S}$ generates a challenge $c_i = H_1(ID_{\mathcal{S}}||Date||Time)$ and sends c_i to $\mathcal{U}$.
   - After receiving c_i, $\mathcal{U}$ computes $R_i = x_1 + s_1c_i + s_2c_i^2 + \ldots + s_kc_i^k$ and chooses $\alpha_i \in \{1, 2, \ldots, n\}$ and a random number $r_{\alpha_i} \in \mathbb{Z}_q^*$, $\mathcal{U}$ computes $B_i = g^{r_{\alpha_i}}$ and $A_i = B_i^{x_1}h^{\alpha_i}$. $\mathcal{U}$ sends $(c_i, R_i, \alpha, \beta, \lambda, r, s)$ and (A_i, B_i) to $\mathcal{S}$.
   - $\mathcal{S}$ checks if
     - R_i has not been used in a previous session.
     - $H_1(\alpha, \beta, \lambda) = \beta^{-s}\alpha^{H_1(H_1(\alpha,\beta,\lambda),r)}r$.
     - $g^{R_i} = \lambda S_1^{c_i} S_2^{c_i^2} \ldots S_k^{c_i^k}$.
     - The proof of knowledge $PoK\{(x_1, \alpha_i) : A_i = B_i^{x_1}h^{\alpha_i} \wedge \lambda = g^{x_1}\}$ is valid.

     If any of the checks fails, aborts; otherwise, $\mathcal{S}$ chooses n different random number $k_1, k_2, \ldots, k_n$
     $\in_R \mathbb{Z}_q^*$ and computes $c_i = (c_{i,1}, c_{i,2}) = (\lambda^{k_i}, m_i \cdot (A_i/h^i)^{k_i})$. $\mathcal{S}$ sends $c_1, c_2, \ldots, c_n$ to $\mathcal{U}$.
   - $\mathcal{U}$ extracts the intended message $m_{\alpha_i} = c_{\alpha_i,2}/c_{\alpha_i,1}^{r_{\alpha_i}}$.

5. **Reveal**: Assume the coupon $C = (\alpha, \beta, \lambda, r, s)$ is redeemed by a dishonest user $\mathcal{U}'$ for $k+1$ times, the $\mathcal{S}$ could obtain $k+1$ shares about the secret x_1. Once $\mathcal{S}$ obtains the value of x_1,

   - $\mathcal{S}$ could trace the identity of $\mathcal{U}'$ by making an exhaustive search in his database to determine the public key of the misbehaving user such that $pk_{\mathcal{U}'}{}^{x_1} = \beta$.
   - For all the previous transcripts $B_1^{x_1}h^{\alpha_1}, B_2^{x_1}h^{\alpha_2}, \ldots, B_k^{x_1}h^{\alpha_k}$, given $B_1, B_2, \ldots, B_k$, then all the previous choice $\alpha_1, \alpha_2, \ldots, \alpha_k$ could also be decided.

6. **Correctness**:

   - The correctness check for validity of the coupon is same as that in PPE-COUPON-I.
   - The correctness check for a user $\mathcal{U}$ to extract the correct message is as

follows:

$$
\begin{aligned}
& m_{\alpha_i} \\
&= c_{\alpha_i,2} / c_{\alpha_i,1}^{r_{\alpha_i}} \\
&= m_{\alpha_i} (A_i / h^{\alpha_i})^{k_{\alpha_i}} / (\lambda^{k_{\alpha_i}})^{r_{\alpha_i}} \\
&= m_{\alpha_i} (B_i^{x_1} h^{\alpha_i} / h^{\alpha_i})^{k_{\alpha_i}} / (g^{x_1 k_{\alpha_i}})^{r_{\alpha_i}} \\
&= m_{\alpha_i} g^{r_{\alpha_i} x_1 k_{\alpha_i}} / g^{x_1 k_{\alpha_i} r_{\alpha_i}} \\
&= m_{\alpha_i}
\end{aligned}
$$

9.5.2 Security Analysis

Theorem 9.5. *The proposed PPE-COUPON-II system is unforgeable.*

Proof. The security proof for unforgeability is same as the security proof for PP-COUPON-I.

Theorem 9.6. *The proposed PPE-COUPON-II system provides anonymity.*

Proof. The security proof for anonymity is same as the security proof for PP-COUPON-I.

Theorem 9.7. *The proposed PPE-COUPON-II system provides k-time redemption detection.*

Proof. The security proof for anonymity is same as the security proof for PP-COUPON-I.

Theorem 9.8. *The proposed PPE-COUPON-II system provides user privacy for honest users.*

Proof. For the transcripts received by the service provider $A_1 = (g^{r_1})^{x_1} h^{\alpha_2}, A_2 = (g^{r_2})^{x_1} h^{\alpha_2}, \ldots, A_k = (g^{r_k})^{x_1} h^{\alpha_k}$, where $\alpha_1, \alpha_2, \ldots, \alpha_k \in \{1, 2, \ldots, n\}$ are the user's choice and $r_1, r_2, \ldots, r_k \in_R \mathbb{Z}_q^*$. Given $B_j = g^{r_j}, \lambda = g^{x_1}$ for some random $r_j \in \mathbb{Z}_q^*$, it is computation-infeasible to decide the masked value A_j equals $g^{r_j x_1}$ or a random value Z in G_q, thus for any two transcripts A_j and A_i such that $1 \leq i \neq j \leq k$ from the user, they are computationally indistinguishable to the service provider as long as the DDH problem is hard in G_q.

Theorem 9.9. *The proposed e-coupon system provides retrievable user privacy for dishonest users.*

Proof. As we have demonstrated in previous security proof, once a malicious user $\mathcal{U}'$ tried to redeem a coupon for $k+1$ times, then the service provider can reconstruct the secret x_1. For all the previous choice of $\mathcal{U}'$, given the transcripts $A_j = B_j^{x_1} h^{\alpha_j}$ and $B_j = g^{r_j}$ where $1 \leq j \leq k$, once x_1 be revealed, each α_j could also be decided. In order to increase the computation speed, the service provider could pre-compute the set $\{h^1, h^2, \ldots, h^n\}$.

9.6 Efficiency Analysis

We compare the proposed e-coupon systems with existing e-coupon systems in the literature in terms of computation efficiency and security properties.

Comparison in terms of Security Properties

We can see in Table 9.1 that our e-coupon systems provide traceability and privacy of purchase, which have not been considered in previous e-coupons systems. However, our e-coupon systems only achieves anonymity, which is weaker than unlinkability in [CES+05, Ngu06b]. We leave the construction of e-coupon systems with both traceability and unlinkability as future work. In order to save space, in Table 9.1, the abbreviations 'DM' denotes 'Detection of Misuse' while 'PP' is short for 'Privacy of Purchase'.

Table 9.1: Comparison with Other E-Coupon Systems in terms of Security Properties

Scheme	Security Properties				
	Unforgeability	DM	Anonymity	PP	Traceability
Scheme 1 [CES+05]	√	√	*strong*	×	×
Scheme 2 [Ngu06b]	√	√	*strong*	×	×
PPE-COUPON-I	√	√	*weak*	√	√
PPE-COUPON-II	√	√	*weak*	√	√

Efficiency Analysis

We present a detailed analysis regarding the computation efficiency of our e-coupon systems and present the comparison with existing e-coupons systems in Table 9.2. In the table, E represents an exponentiation operation, P refers to a pairing operation; n is the number of goods or services from the service provider and k is the number of choices that could be made by a user. From the efficiency analysis, we can see that the computation cost of our e-coupon systems is quite low. Specially, our second e-coupon system achieves traceability against misbehaving users' choices without introducing significant computation cost.

Table 9.2: Comparison with Other E-Coupon Systems in terms of Computation Cost

Scheme	Computation cost					
	KeyGen		Issue		Redeem	
	U	S	U	S	U	S
Scheme 1 [CES+05]	0	0	$4E$	$2E$	$(k+3)E$	0
Scheme 2 [Ngu06b]	0	$(k+2)E$	$3E+2P$	$11E+5P$	$20E+2P$	$7E+5P$
PPE-COUPON-I	$(k+1)E$	$1E$	$10E$	$3E$	$5E$	$(2n+6)E$
PPE-COUPON-II	$(k+1)E$	$1E$	$10E$	$3E$	$6E$	$(2n+7)E$

9.7 Summary

In this chapter, we formalized the notion of user privacy for e-coupon systems. We proposed two practical e-coupon systems that can maintain anonymity and user privacy for honest users, while the identity (and also the choices in the second system) of a misbehaving user could be traced by the service provider. We achieved these without requiring any third party in the system. In addition, we formalized the security models for our e-coupon systems and proved that our new e-coupon systems are proven secure under the proposed security models.

Chapter 10

Conclusion

In this chapter, we present the summary of our contributions in this thesis and some potential directions of future work.

10.1 Summary of Contributions

In this thesis, we investigate several cryptographic primitives including proxy signature, oblivious transfer and e-coupon system, in which the users only have restricted capabilities. The contributions of this thesis can be summarized in the following six aspects: k-time proxy signature that only allows a proxy signer to generate a constant number proxy signatures in the name of an original signer; attribute-based signing right delegation benefiting from both attribute-based signature and proxy signature; we identify a practical attack that has been neglected in one type of delegation-by-warrant proxy signature scheme and propose one generic solution to thwart the attack; a practical e-coupon system is proposed which enables the coupon issuer to trace the identities of misbehaving users without involvement of a trusted third party; an OT scheme with retrievable receiver's privacy; and another multiple-use e-coupon system which is based on our OT scheme and allows the coupon issuer to trace both the identities and choices of dishonest users. The specific contributions in each aspect listed above are summarized as follows.

In chapter 4, we propose a k-time proxy signature scheme, in which a legitimate proxy signer is only able to issue a pre-determined number of signature in the name of an original signer. The proposed scheme performs better than existing proxy signature schemes in prevent proxy signers from misusing their delegated signing ability. Since in conventional delegation-by-warrant proxy signature schemes, the only trick used in restricting the proxy signers' signing ability is that the original signers specify the valid time period in the warrants. We define the formal definition and security model for such a type of proxy signature. We analyse the security of the proposed k time proxy signature scheme and prove that it is secure under the proposed security model.

In chapter 5, we construct an attribute-based proxy signature scheme, in which the original signers are legitimate users with a valid set of attributes delegate the signing right to the proxy signers with normal public keys. The proposed ABPS scheme benefits from both proxy signature and attribute-based signature. By verifying an attribute-based proxy signature, a public verifier can be convinced that the

proxy signature is generated by the proxy signer who has obtained the delegation from the original signer whose attributes satisfy a pre-claimed predicate. However, the verifier cannot tell from the signature who is the original signer. Thus, the proposed scheme ensures an enhanced privacy for the original signer. We provide the formal definition and adversarial models for attribute-based proxy signature and prove the security of the proposed scheme using random oracle model.

In chapter 6, we present a practical attack that has been neglected in one type of delegation-by-warrant proxy signature schemes. By launching the attack to an identity-based proxy signature scheme [WMS+07], We demonstrate that a malicious adversary can successfully forge a proxy signature without being noticed by the original signer or proxy signer. We revise the existing security model to capture this attack. Then one solution is proposed to resist this attack. We prove the security of the revised identity-based proxy signature scheme under the revised security model. Though we work on one concrete scheme, the weakness in some other proxy signature schemes can also be fixed by applying the same method.

In chapter 7, we construct an efficient privacy-preserving e-coupon system. Besides all the desirable properties (unforgeability, anonymity and double redemption detection), we formalize a new property of e-coupon system named user privacy (privacy of purchase). Another new feature of the proposed e-coupon system is achieving traceability against dishonest users without involvement of trusted third party in the tracing phase. We prove the security of the e-coupon system under the proposed security model and show that the proposed e-coupon system meets all the secure requirements.

In chapter 8, we propose two novel oblivious transfer schemes, which can be applied in privacy-sensitive systems to trace the dishonest users' choices, while the privacy of honest users is well protected. We revise the half-simulation model [NP05] to evaluate the security of the proposed OT schemes. Detailed security analysis is presented to show that the proposed OT schemes are secure under the proposed security model.

In chapter 9, two novel privacy-preserving e-coupon systems supporting multiple use of an electronic coupon are proposed. We use our new oblivious transfer scheme with retrievable receiver's privacy together with a new blind signature in designing the e-coupon systems. Besides the basic desirable security properties, the first e-coupon system allows the coupon issuer to trace the identities of misbehaving users while the second e-coupon system permits the coupon issuer to reveal both the identities and choices of the dishonest users. In both e-coupon systems, the privacy of honest users is well protected. We prove the security of proposed e-coupon systems under the defined security model and compare them with existing e-coupon systems. The efficiency analysis shows the proposed e-coupon systems provide comparable

efficiency.

10.2 Future Work

Though all the proposed schemes have achieved the pre-claimed goals, our work in this thesis could be further extended in the following aspects.

1. The security of the proposed k-time proxy signature and attribute-based proxy signature schemes are based on random oracles. Though random oracle has been proved to be practical in [BR93], there is still controversy in using random oracle in cryptographic constructions. In fact, there have been some artificial cryptographic schemes proven secure using random oracles but are trivially insecure when any real function is substituted for the random oracle [CGH98, GR04]. Thus, it is of independent interesting to design k-time proxy signature and attribute-based signature schemes without random oracles to ensure an enhanced security.

2. We propose an OTRRP scheme and analyse its security under the half-simulation model [NP05]. In half-simulation model, the security of the sender and receiver is considered separately. An OT scheme proven secure under the half-simulation model achieves simulatable security for sender privacy and computationally indistinguishability for receiver privacy. Recently, the full-simulation model achieving simulatable security for both the receiver and sender has been proposed in [CNS07, KN09]. We leave the work of constructing OTRRP schemes that could be proven secure under full-simulation model as one of our future work.

3. We can see in the comparison between our proposed e-coupon systems with existing e-coupon systems [CES$^+$05, Ngu06b] that our e-coupon systems provide traceability and privacy of purchase, which have not been considered in previous e-coupons systems. However, our e-coupon systems only achieves anonymity, which is weaker than unlinkability in [CES$^+$05, Ngu06b]. It has been an open problem to achieve traceability and unlinkability simultaneously in one scheme. We leave the construction of e-coupon systems with both traceability and unlinkability as another future work.

Bibliography

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, pages 119–135, 2001.

[AO00] Masayuki Abe and Tatsuaki Okamoto. Provably secure partially blind signatures. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, pages 271–286, 2000.

[AWSM07] Man Ho Au, Qianhong Wu, Willy Susilo, and Yi Mu. Compact e-cash from bounded accumulator. In *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings*, pages 178–195, 2007.

[BCR86] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, pages 234–238, 1986.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, pages 26–45, 1998.

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, pages 390–420, 1992.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. *J. Cryptology*, 16(3):185–215, 2003.

[Bon98] Dan Boneh. The decision diffie-hellman problem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, pages 48–63, 1998.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 273–289, 2004.

[BPW12] Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. Secure proxy signature schemes for delegation of signing rights. *J. Cryptology*, 25(1):57–115, 2012.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993.*, pages 62–73, 1993.

[Bra93] Stefan Brands. Untraceable off-line cash in wallets with observers (extended abstract). In *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 302–318, 1993.

[BTT03] Kemal Bicakci, Gene Tsudik, and Brian Tung. How to construct optimal one-time signatures. *Computer Networks*, 43(3):339–349, 2003.

[Buc04] Thomas M. Buchsbaum. E-voting: International developments and lessons learnt. In *Electronic Voting in Europe - Technology, Law, Politics and Society, Workshop of the ESF TED Programme together with GI and OCG, July, 7th-9th, 2004, in Schloß Hofen / Bregenz, Lake of Constance, Austria, Proceedings*, pages 31–42, 2004.

[CDN09] Jan Camenisch, Maria Dubovitskaya, and Gregory Neven. Oblivious transfer with access control. In *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, pages 131–140, 2009.

[CDNZ11] Jan Camenisch, Maria Dubovitskaya, Gregory Neven, and Gregory M. Zaverucha. Oblivious transfer with hidden access control policies. In *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, pages 192–209, 2011.

[CES+05] Liqun Chen, Matthias Enzmann, Ahmad-Reza Sadeghi, Markus Schneider, and Michael Steiner. A privacy-protecting coupon system. In

Financial Cryptography and Data Security, 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica, February 28 - March 3, 2005, Revised Papers, pages 93–108, 2005.

[CFN88] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, pages 319–327, 1988.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 209–218, 1998.

[CGH06] Sébastien Canard, Aline Gouget, and Emeline Hufschmitt. A handy multi-coupon system. In *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, pages 66–81, 2006.

[CGH09] Scott E. Coull, Matthew Green, and Susan Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In *Public Key Cryptography - PKC 2009, 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings*, pages 501–520, 2009.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982.*, pages 199–203, 1982.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[CKK03] Chul-Joon Choi, Zeen Kim, and Kwangjo Kim. Schnorr signature scheme with restricted signing capability. 2003.

[CKW04] Jan Camenisch, Maciej Koprowski, and Bogdan Warinschi. Efficient blind signatures without random oracles. In *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, pages 134–148, 2004.

[CNS07] Jan Camenisch, Gregory Neven, and Abhi Shelat. Simulatable adaptive oblivious transfer. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications*

of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings, pages 573–590, 2007.

[CPS94] Jan Camenisch, Jean-Marc Piveteau, and Markus Stadler. Blind signatures based on the discrete logarithm problem. In *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, pages 428–432, 1994.

[CT05] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *Public Key Cryptography - PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, January 23-26, 2005, Proceedings*, pages 172–183, 2005.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 542–552, 1991.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DSP07] Manik Lal Das, Ashutosh Saxena, and Deepak B. Phatak. Proxy signature scheme with effective revocation using bilinear pairings. *I. J. Network Security*, 4(3):312–317, 2007.

[EGL85] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[FKTT98] Ian T. Foster, Carl Kesselman, Gene Tsudik, and Steven Tuecke. A security architecture for computational grids. In *CCS '98, Proceedings of the 5th ACM Conference on Computer and Communications Security, San Francisco, CA, USA, November 3-5, 1998.*, pages 83–92, 1998.

[FP08] Georg Fuchsbauer and David Pointcheval. Anonymous proxy signatures. In *Security and Cryptography for Networks, 6th International Conference, SCN 2008, Amalfi, Italy, September 10-12, 2008. Proceedings*, pages 201–217, 2008.

[FTY98] Yair Frankel, Yiannis Tsiounis, and Moti Yung. Fair off-line e-cash made easy. In *Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information*

Security, Beijing, China, October 18-22, 1998, Proceedings, pages 257–270, 1998.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.

[GR04] Craig Gentry and Zulfikar Ramzan. Eliminating random permutation oracles in the even-mansour cipher. In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, pages 32–47, 2004.

[HC09] Xuan Hong and Kefei Chen. Secure multiple-times proxy signature scheme. *Computer Standards & Interfaces*, 31(1):19–23, 2009.

[HKLL03] Jung Yeon Hwang, Hyun-Jeong Kim, Dong Hoon Lee, and Jong In Lim. Digital signature schemes with restriction on signing capability. In *Information Security and Privacy, 8th Australasian Conference, ACISP 2003, Wollongong, Australia, July 9-11, 2003, Proceedings*, pages 324–335, 2003.

[HSMW06] Xinyi Huang, Willy Susilo, Yi Mu, and Wei Wu. Proxy signature without random oracles. In *Mobile Ad-hoc and Sensor Networks, Second International Conference, MSN 2006, Hong Kong, China, December 13-15, 2006, Proceedings*, pages 473–484, 2006.

[HSMY12] Jinguang Han, Willy Susilo, Yi Mu, and Jun Yan. Efficient oblivious transfers with access control. *Computers & Mathematics with Applications*, 63(4):827–837, 2012.

[JLO97] Ari Juels, Michael Luby, and Rafail Ostrovsky. Security of blind digital signatures (extended abstract). In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 150–164, 1997.

[KN09] Kaoru Kurosawa and Ryo Nojima. Simple adaptive oblivious transfer without random oracle. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 334–346, 2009.

[KPW97] Seungjoo Kim, Sangjoon Park, and Dongho Won. Proxy signatures, revisited. In *Information and Communication Security, First International Conference, ICICS'97, Beijing, China, November 11-14, 1997, Proceedings*, pages 223–232, 1997.

[KYC10] Woo-Hwan Kim, HyoJin Yoon, and Jung Hee Cheon. Metered signatures: How to restrict the signing capability. *Journal of Communications and Networks*, 12(3):201–208, 2010.

[LAS+10] Jin Li, Man Ho Au, Willy Susilo, Dongqing Xie, and Kui Ren. Attribute-based signature and its applications. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 60–69. ACM, 2010.

[LHH05] Eric Jui-Lin Lu, Min-Shiang Hwang, and Cheng-Jian Huang. A new proxy signature scheme with revocation. *Applied Mathematics and Computation*, 161(3):799–806, 2005.

[LK08] Jin Li and Kwangjo Kim. Attribute-based ring signatures. *IACR Cryptology ePrint Archive*, 2008:394, 2008.

[LKK01a] Byoungcheon Lee, Heesun Kim, and Kwangjo Kim. Secure mobile agent using strong non-designated proxy signature. In *Information Security and Privacy, 6th Australasian Conference, ACISP 2001, Sydney, Australia, July 11-13, 2001, Proceedings*, page 474, 2001.

[LKK01b] Byoungcheon Lee, Heesun Kim, and Kwangjo Kim. Strong proxy signature and its applications. In *Proc of SCIS*, volume 1, pages 603–608, 2001.

[LKZC07] Jin Li, Kwangjo Kim, Fangguo Zhang, and Xiaofeng Chen. Aggregate proxy signature and verifiably encrypted proxy signature. In *Provable Security*, pages 208–217. Springer, 2007.

[LMX+13] Ximeng Liu, Jianfeng Ma, Jinbo Xiong, Tao Zhang, and Qi Li. Personal health records integrity verification using attribute based proxy signature in cloud computing. In *Internet and Distributed Computing Systems - 6th International Conference, IDCS 2013, Hangzhou, China, October 28-30, 2013, Proceedings*, pages 238–251, 2013.

[LMY14a] Weiwei Liu, Yi Mu, and Guomin Yang. Attribute-based signing right delegation. In *Network and System Security - 8th International Conference, NSS 2014, Xi'an, China, October 15-17, 2014, Proceedings*, pages 323–334, 2014.

[LMY14b] Weiwei Liu, Yi Mu, and Guomin Yang. An efficient privacy-preserving e-coupon system. In *Information Security and Cryptology - 10th International Conference, Inscrypt 2014, Beijing, China, December 13-15, 2014, Revised Selected Papers*, pages 3–15, 2014.

[LYMW13] Weiwei Liu, Guomin Yang, Yi Mu, and Jiannan Wei. k-time proxy signature: Formal definition and efficient construction. In *Provable Security - 7th International Conference, ProvSec 2013, Melaka, Malaysia, October 23-25, 2013. Proceedings*, pages 154–164, 2013.

[McC90] Kevin S McCurley. The discrete logarithm problem. In *Proc. of Symp. in Applied Math*, volume 42, pages 49–74, 1990.

[MH05] Manish Mehta and Lein Harn. Efficient one-time proxy signatures. In *Communications, IEE Proceedings-*, volume 152, pages 129–133. IET, 2005.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, pages 417–426, 1985.

[MNV01] Yi Mu, Khanh Quoc Nguyen, and Vijay Varadharajan. A fair electronic cash scheme. In *Topics in Electronic Commerce, Second International Symposium, ISEC 2001 Hong Kong, China, April 26-28, 2001, Proceedings*, pages 20–32, 2001.

[MOY04] Tal Malkin, Satoshi Obana, and Moti Yung. The hierarchy of key evolving signatures and a characterization of proxy signatures. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 306–322, 2004.

[MPR11] Hemanta K. Maji, Manoj Prabhakaran, and Mike Rosulek. Attribute-based signatures. In *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, pages 376–392, 2011.

[MUO96] Masahiro Mambo, Keisuke Usuda, and Eiji Okamoto. Proxy signatures for delegating signing operation. In *CCS '96, Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, March 14-16, 1996.*, pages 48–57, 1996.

[MXZ11] Xu Ma, Lingling Xu, and Fangguo Zhang. Oblivious transfer with timed-release receiver's privacy. *Journal of Systems and Software*, 84(3):460–464, 2011.

[MZV02] Yi Mu, Junqi Zhang, and Vijay Varadharajan. m out of n oblivious transfer. In *Information Security and Privacy, 7th Australian Conference, ACISP 2002, Melbourne, Australia, July 3-5, 2002, Proceedings*, pages 395–405, 2002.

[Neu93] B. Clifford Neuman. Proxy-based authorization and accounting for distributed systems. In *Proceedings of the 13th International Conference on Distributed Computing Systems, Pittsburgh, Pennsylvania, USA, May 25-28, 1993*, pages 283–291, 1993.

[Ngu06a] Lan Nguyen. Efficient dynamic k-times anonymous authentication. In *Progressin Cryptology - VIETCRYPT 2006, First International Conferenceon Cryptology in Vietnam, Hanoi, Vietnam, September 25-28, 2006, Revised Selected Papers*, pages 81–98, 2006.

[Ngu06b] Lan Nguyen. Privacy-protecting coupon system revisited. In *Financial Cryptography and Data Security, 10th International Conference, FC 2006, Anguilla, British West Indies, February 27-March 2, 2006, Revised Selected Papers*, pages 266–280, 2006.

[NMV97] Khanh Quoc Nguyen, Yi Mu, and Vijay Varadharajan. A new digital cash scheme based on blind nyberg-rueppel digital signature. In *Information Security, First International Workshop, ISW '97, Tatsunokuchi, Japan, September 17-19, 1997, Proceedings*, pages 313–320, 1997.

[NP99] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 573–590, 1999.

[NP05] Moni Naor and Benny Pinkas. Computationally secure oblivious transfer. *J. Cryptology*, 18(1):1–35, 2005.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 427–437, 1990.

[OT14] Tatsuaki Okamoto and Katsuyuki Takashima. Efficient attribute-based signatures for non-monotone predicates in the standard model. *IEEE T. Cloud Computing*, 2(4):409–421, 2014.

[Ped91] Torben P. Pedersen. Distributed provers with applications to undeniable signatures. In *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, pages 221–242, 1991.

[Pre94] Bart Preneel. Cryptographic hash functions. *European Transactions on Telecommunications*, 5(4):431–448, 1994.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, pages 387–398, 1996.

[PWX04] Josef Pieprzyk, Huaxiong Wang, and Chaoping Xing. Multiple-time signature schemes against adaptive chosen message attacks. In *Selected Areas in Cryptography*, pages 88–100. Springer, 2004.

[Rab81] Michael O Rabin. How to exchange secrets with oblivious transfer. 1981.

[RS91] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, pages 433–444, 1991.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 239–252, 1989.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 47–53, 1984.

[SMP08] Jacob C. N. Schuldt, Kanta Matsuura, and Kenneth G. Paterson. Proxy signatures secure against proxy key exposure. In *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory*

in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings, pages 141–161, 2008.

[SPC95] Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. Fair blind signatures. In *Advances in Cryptology - EUROCRYPT '95, International Conference on the Theory and Application of Cryptographic Techniques, Saint-Malo, France, May 21-25, 1995, Proceeding*, pages 209–219, 1995.

[SW02] K. Shum and Victor K. Wei. A strong proxy signature scheme with proxy signer privacy protection. In *11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2002), 10-12 June 2002, Pittsburgh, PA, USA*, pages 55–56, 2002.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 457–473, 2005.

[SXYM11] Ying Sun, Chunxiang Xu, Yong Yu, and Yi Mu. Strongly unforgeable proxy signature scheme secure in the standard model. *Journal of Systems and Software*, 84(9):1471–1479, 2011.

[TFS04] Isamu Teranishi, Jun Furukawa, and Kazue Sako. k-times anonymous authentication (extended abstract). In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, pages 308–322, 2004.

[Wan05] Guilin Wang. Designated-verifier proxy signature schemes. In *Security and Privacy in the Age of Ubiquitous Computing, IFIP TC11 20th International Conference on Information Security (SEC 2005), May 30 - June 1, 2005, Chiba, Japan*, pages 409–424, 2005.

[WMS+07] Wei Wu, Yi Mu, Willy Susilo, Jennifer Seberry, and Xinyi Huang. Identity-based proxy signature from pairings. In *Autonomic and Trusted Computing, 4th International Conference, ATC 2007, Hong Kong, China, July 11-13, 2007, Proceedings*, pages 22–31, 2007.

[XZF05] Jing Xu, Zhenfeng Zhang, and Dengguo Feng. Id-based proxy signature using bilinear pairings. In *Parallel and Distributed Processing and*

Applications - ISPA 2005 Workshops, ISPA 2005 International Workshops AEPP, ASTD, BIOS, GCIC, IADS, MASN, SGCA, and WISA, Nanjing, China, November 2-5, 2005, Proceedings, pages 359–367, 2005.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982.

[YMS+12] Yong Yu, Yi Mu, Willy Susilo, Ying Sun, and Yafu Ji. Provably secure proxy signature scheme from factorization. *Mathematical and Computer Modelling*, 55(3-4):1160–1168, 2012.

[Zha97] Kan Zhang. Threshold proxy signature schemes. In *Information Security, First International Workshop, ISW '97, Tatsunokuchi, Japan, September 17-19, 1997, Proceedings*, pages 282–290, 1997.

[ZSNL] Fangguo Zhang, Reihaneh Safavi-Naini, and Chih-Yin Lin. New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairing.