New security notions for identity based encryption

Sriramkrishnan Srinivasan

Technical Report
RHUL–MA–2011–3
21 February 2011

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England

http://www.rhul.ac.uk/mathematics/techreports

NEW SECURITY NOTIONS FOR IDENTITY BASED ENCRYPTION

By

Sriramkrishnan Srinivasan

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
ROYAL HOLLOWAY, UNIVERSITY OF LONDON
EGHAM, SURREY TW20 0EX
2010

ROYAL HOLLOWAY, UNIVERSITY OF LONDON

Author: **Sriramkrishnan Srinivasan**

Title: **New Security Notions For Identity Based Encryption**

Department: **Information Security Group**

Faculty: **Department of Mathematics**

Degree: **Ph.D.**

I hereby affirm that this is my original work and that all sources and references have been acknowledged.

Sriramkrishnan Srinivasan

Peace

Table of Contents

Abstract

We study extended security notions for Identity Based Encryption (IBE) in settings where multiple Trusted Authorities (TAs) share some common parameters, as distinct from most existing research considering a single TA that issues keys to users in a system. We extend current notions of security for IBE to the multi-TA setting, and in addition, formalize the notion of TA anonymity. We study the security properties of natural multi-TA analogues of existing IBE schemes in both the Random Oracle Model (ROM) and the Standard Model with respect to these new notions. We give a modified multi-TA version of a Fujisaki-Okamoto transform (in the ROM), and multi-TA versions of the Canetti-Halevi-Katz and Boneh-Katz transforms (in the Standard Model), which are used to build strong cryptographic schemes, from schemes meeting weaker notions of security. Using our new ideas, we also give the first generic methods to construct Key Private Public Key Encryption schemes in the Standard model.

We consider the applications of IBE schemes that not only share common parameters, but in addition share additional public parameters in such a way that a ciphertext created for an identity and a particular TA can be read by a recipient with the same identity, but with a private key issued by another TA. This gives us extensions to the basic IBE primitive which enable flexible and secure communications in coalition environments.

We also present an extended security model for Identity Based Non Interactive Key Distribution (ID-NIKD). In addition, we present a transform that takes an ID-NIKD scheme that is secure in this extended security model and satisfies some mild technical conditions, and produces an IND-ID-CPA secure IBE scheme. These results shed light on the relationships between existing ID-NIKD and IBE schemes.

Acknowledgements

My supervisor Kenny Paterson made this thesis possible. Kenny already supervised my master's thesis during my earlier stay at the Information Security Group at Royal Holloway. He encouraged my earliest desires to undertake research and played a big part in my returning to Royal Holloway to pursue a Ph.D. program. Over these last few years Kenny has supported me throughout my research and guided me in every possible way. His passion and attention to detail have never ceased to amaze me and it has been a pleasure to work with him.

My research was sponsored by a Dorothy Hodgkin Scholarship with generous funding from EPSRC and Vodafone. Many academic visits were undertaken with funding associated with my scholarship and funding from the ECRYPT project. My master's thesis was sponsored by a British Chevening Scholarship. I am indebted to these organizations, the staff at the British Council in India and the three interviewers who selected me for the Chevening Scholarship.

Many individuals have influenced my academic career over the years. Staff and fellow researchers at the Information Security Group at Royal Holloway have been wonderful. Steven Galbraith, Alex Dent, Keith Martin, Fred Piper, Chez Ciechanowicz, Peter Wild, Pauline Stoner, Jenny Lee, Jon Hart and Tristan Findley deserve special mention. I am grateful to colleagues at Adastral Park Research Centre in Ipswich where I was briefly a research intern and teachers at the Centre for Development of Advanced Communication in Pune. Zachariah C. Alex, Elizabeth Rufus, Dharani Bai and many others at the Vellore Institute of Technology in Katpadi where I did my undergraduate studies went out of their way for me. Teachers at La Martiniere and St. Xavier's in Kolkata, have all shaped my academic and personal life.

Over the last three years my house mates (at the aptly named House(s) of Crypto),

James and Gaven, and regular visitors, among them David and Christian, have provided endless entertainment, food for thought and support. Fedor and Qin provided many opportunities for discussing life. Thank you for reminding me that the world is as it is and that a perfect world may not be possible, but a better one is. Gaven, Sujith and Pooja have always gone out of their way for me and I am grateful to them.

Ani, Mayur, Gargi, Sumana, Prasenjit, Devakumar, Karthik, Sathya, Ramesh, Anand, Arul, Parthasarathy, Ganesh and many others from India have continued to care and support. I am indebted to all of them for their gift of knowledge and insight and for putting up with my quirks. My friend Gaurav and his family (Asha Aunty in particular) have been an extended family over the years. I have not forgotten your kindness.

My parents have joyfully sacrificed everything so that my brother and I could have the upbringing and education they could not have for themselves. My brother has always been there for me when I needed him. Thank you for your perseverance, your faith and most importantly your love.

I met Yenn when I was first at Royal Holloway for my master's studies and it has been five years now and she has supported me every step of the way. This thesis is as much due to her as it is for her.

I would like to express my gratitude to open source and free software enthusiasts who make high quality software available for free, and to academics who make academic papers, course notes and books available freely. Your contributions keep the spirit of education alive. Thank you.

Sriramkrishnan Srinivasan

Chapter 1

Introduction

Cryptography is an ancient yet living subject. Kahn's encyclopedic work [76] discusses the evolution of the subject from ancient times till the time of the publication of the book in 1967. The year 1976 marked the beginning of a period of rapid innovation in the field with the announcement of the concept of Public Key Cryptography (PKC) [51] in the open cryptographic community. The symmetric cipher DES was also announced the same year and has a rich body of knowledge surrounding it. However, we will be dealing mostly with public key techniques in this thesis. The simple yet revolutionary idea of using a "public" key to encrypt, and a different "private" key, mathematically related to the public key, yet created in such a way that it is infeasible to deduce the private key without knowledge of some secret or trapdoor information, to decrypt, marked a fundamental change in the general cryptographic mindset.

PKC (and Public Key Encryption (PKE) in particular) was born out of a need to solve one of the fundamental problems of cryptography, viz. key-management. Prior to the advent of PKE, the only way for two or more parties to communicate securely was for them to agree upon a common key in advance or use the services of a Trusted Third Party (TTP). This common or symmetric key is used to both encrypt and decrypt and must be distributed and stored securely. The need for a secure channel to distribute keys and the fact that keys need to be set up securely between every

pair of communicating parties were significant hurdles to the widespread use of secure communication. With the advent of PKE, data could be sent securely with knowledge only of the recipient's public key, which as the name suggests is made readily available to everyone. The encrypted data could only be decrypted by the recipient who is in possession of the appropriate private key.

While PKC solves some of the problems it sets out to, it introduces others. On the one hand, it provides the flexibility to be able to communicate securely with an intended recipient so long as an appropriate public key is available, eliminating the need to establish a shared secret in some way. On the other hand, it introduces the problem of authenticating keys. How is a sender to know that the public key he is encrypting with belongs to the recipient he has in mind? Assuming the key was originally generated by the intended recipient, how is a sender to know that the key has not since been compromised?

To ensure that a public key is genuine it must be signed by a Trusted Third Party (TTP). At the most basic level, signing a public key, produces an assertion by the TTP of the form, "This is the public key corresponding to user A". Such an assertion is called a certificate and the TTP is called a Certificate Authority (CA). Signature schemes are also enabled by PKC and so long as the signature scheme is cryptographically sound, a certificate on a public key cannot be forged. However, the solution is more complex that it seems. The user must then be able to obtain a valid verification key for the CA and more importantly, he must trust this CA. In practice, there are "root CAs" that are trusted by hardware and software manufacturers and the verification keys of these root CAs are embedded in products. These root verification keys are used in turn to sign the verification keys of other lower level CAs. A user can be assured that the public key he intends to use is genuine if he can find a chain of certificates, leading to a root CA (or some other CA that he trusts). This still does not solve the problem of compromised public keys and directories of compromised

public keys (or alternately, valid public keys) must be maintained, which the sender must look up before sending an encrypted message. The introduction of additional infrastructure to certify the authenticity of public keys and to revoke keys lead to a different, yet equally difficult kind of key-management problem.

Geer [61] argues rather succinctly that, "both symmetric cryptosystems, like Kerberos, and asymmetric cryptosystems, like RSA, do the same thing – that is to say they do key distribution – but the semantics are quite different. The fundamental security-enabling activity of a secret key system is to issue fresh keys at low latency and on demand. The fundamental security-enabling activity of an asymmetric key system is to verify the as-yet-unrevoked status of a key already in circulation, again with low latency and on demand. This is key-management and it is a systems cost; a secret key system like Kerberos has incurred nearly all its costs by the moment of key issuance. By contrast, a public key system incurs nearly all its costs with respect to key revocation. Hence, a rule of thumb: The cost of key issuance plus the cost of key revocation is a constant, just yet another version of, 'You can pay me now or you can pay me later'."

Eliminating certificates, and the associated processing and management overheads from PKC, has been one of the biggest challenges for modern cryptographers. A step in this direction was taken when the concept of Identity Based Cryptography (IBC) was first introduced by Shamir [104]. In IBC, arbitrary identifying strings such as e-mail addresses or IP addresses can be used to form public keys, with the corresponding private keys being created by a Trusted Authority (TA) who is in possession of a system-wide master secret. Then a party Alice who wishes, for example, to encrypt to a party Bob need only know Bob's identifier and the system-wide public parameters. This approach eliminates certificates and the associated processing and management overheads from PKC.

While IBC potentially removes the problem of trust in the public keys, it introduces trust in the TA, which by virtue of issuing private keys to users, using its knowledge of the master secret key, is now automatically a key-escrow agency. While, this is unacceptable to many users and in many application scenarios, this is exactly the property desirable in closed military, government and corporate infrastructures.

Although Shamir introduced the concept of Identity Based Encryption (IBE) in 1984, the first efficient and secure constructions of IBE schemes were not forthcoming till the pairing based solutions of Sakai, Ohigishi and Kasahara in 2000 [101] and Boneh and Franklin in 2001 [22], and work of Cocks [44], also published in 2001. Boneh and Franklin [22] also proposed the first security models for IBE and gave schemes that were proven secure in the Random Oracle Model (ROM) [14]. Since the publication of these first results, there has been an explosion of interest in IBE and related cryptographic primitives.

1.1 Motivation and Contributions

1.1.1 Sharing of Parameters

In almost all the existing literature on IBE, with a small number of exceptions (as we will discuss in Section 1.3), there is a single global TA issuing keys to all users in the system, and all ciphertexts are created using the public parameters of that single global TA. This TA is also known as the private key generator (PKG) in the literature.

In practice however, it is unlikely that there will be a single global TA, even within a single organization. It is more likely that there will be multiple TAs, each issuing private keys to a set of users. In addition, some users may have keys issued by more than one TA. In such scenarios, it is not unreasonable to assume that the different

TAs may share some common system parameters. In fact, there are several reasons why this may actually be the case in practice.

- The complexity of setting up an IBE infrastructure, for example, generating the public parameters of the TA for currently known IBE schemes that are practical to implement, far exceeds the complexity of setting up an El-Gamal or RSA based PKE scheme. Almost all well known IBE schemes are pairing based (we will not be discussing the scheme of Cocks [44] and Boneh *et al.* [23] which are not based on the mathematics of pairings) and in these schemes a number of complex issues need to be addressed – for example, appropriate elliptic curves need to be generated, the pairing map, the input and target groups and appropriate hash functions need to be defined. In addition, the representation of the various elements in the system needs to be unambiguously defined. If any kind of interoperability is desired, all these activities must be supported by appropriate standardization efforts, which leads us to the next point.

- Cryptographic schemes and related parameters are often standardized by bodies like ISO, NIST and IEEE, and then used in multiple domains by different authorities. Indeed, the IEEE P1363.3 working group is aiming to produce a set of standards specific to IBC.

- Even when the standards are developed and perhaps even made freely available, it is not feasible to expect individual TAs to generate these parameters in a manner that is secure and inter-operable. Technology companies generate these parameters, over which they may hold exclusive rights, by virtue of holding patents over elliptic curves or other mathematical objects and algorithms on which they are based, and sell licences for their use. The cost of

these licences often represents a significant investment for any corporate entity or governmental agency looking at a potential IBE deployment.

A corporate entity or agency with more than one potential IBE deployment will want to reuse these standard parameters across deployments so as to maximize its investment. Such a scenario where parameters are reused across deployments is not captured in existing security models. This thesis deals mainly with IBE in the setting where multiple TAs sharing some common system parameters.

1.1.2 Anonymous Communication

Anonymous encryption was historically motivated in the context of mobile communication. In the standard public key setting, entities $\mathcal{A}$ and $\mathcal{B}$ exchange encrypted messages using each others' public keys over a broadcast medium where eavesdroppers can see all ciphertexts on the network. It is reasonable to assume that $\mathcal{A}$ and $\mathcal{B}$ will want to keep their identities hidden from such eavesdroppers and this is possible only when ciphertexts do not leak information about the public keys used to create them, a notion formalized as Key Privacy in [9].

In the IBE setting, the security notion equivalent to Key Privacy is that of Recipient Anonymity: the ciphertext should not leak the identity of the (intended) recipient. The systematic study of Recipient Anonymity was initiated in [2], motivated both by its intrinsic interest in IBE and for its application in constructing Public Key Encryption with Keyword Search (PEKS) schemes. Since then, Recipient Anonymity has quickly become a standard security property for IBE schemes.

It is possible to envisage scenarios as above but with multiple, independent TAs (perhaps sharing some common system parameters). In some applications, each user may only have a single private key issued by one of the TAs, while in others, users could have multiple private keys for the same identity string with the different private

keys being issued by different TAs. In both settings, in addition to the usual IBE security notions of indistinguishability and Recipient Anonymity, the notion of TA Anonymity arises as being both natural and of fundamental importance. Here, we desire that an adversary should find it difficult to distinguish ciphertexts produced using different TA master public keys, even if the ciphertext is for the same message and identity string. As well as being a natural security notion for the multi-TA setting, TA Anonymity may have practical significance. For example, we can imagine a coalition of TAs operating in a wireless setting where all ciphertexts can be captured from the network by an adversary. In such a scenario, if the ciphertext were to somehow leak the identity of the TA, this would open up avenues for traffic analysis. In a hostile environment, traffic analysis can lead to the leakage of information relating to which entities are communicating and how frequently, which can often reveal important intelligence.

As we will see, resistance to traffic analysis is not the only security concern in the multi-TA setting. We will also study the cryptographic implications of TA Anonymity on schemes that use IBE as a building block, later in this thesis.

1.2 Structure of the Thesis

In Chapter 2 we start by fixing some basic notation and introducing basic concepts that are used heavily throughout the thesis. Additional background information is introduced as and when applicable in later chapters to facilitate presentation.

In Chapter 3 we extend the usual indistinguishability and Recipient Anonymity notions for IBE to the multi-TA setting, and in addition, formalize the notion of TA Anonymity. This chapter forms the foundations for much of what follows.

Fujisaki and Okamoto give generic transforms [59, 58] that build PKE schemes which are secure in a very strong sense from PKE schemes which are secure in a weaker

sense, in the ROM. In Chapter 4 we introduce a modified version of the Fujisaki-Okamoto transform [58] for the multi-TA IBE setting, proving that our modified transform lifts security and anonymity properties from the chosen plaintext attack (CPA) setting to the chosen ciphertext attack (CCA) setting in the ROM. We then apply these results to study the security and anonymity of the Boneh-Franklin [22] and the Sakai-Kasahara [102] schemes in the multi-TA setting.

In Chapter 5 we study the TA Anonymity properties of multi-TA versions of some known Standard Model IBE schemes. We show that multi-TA versions of the two Standard Model schemes of Boneh and Boyen in [19], termed $\mathtt{BB}_1$ and $\mathtt{BB}_2$ in the literature, and multi-TA versions of schemes related to the $\mathtt{BB}_1$ scheme, such as those of Waters [117], trivially do not meet the notion of TA Anonymity. We also prove that a multi-TA version of the scheme of Gentry [62] is TA Anonymous.

Canetti *et al.* [33] give a generic transform, which we will refer to as the CHK transform, that converts an IBE scheme meeting a weak security notion into a PKE scheme meeting a very strong notion of security. In Chapter 6 we first consider the CHK transform in the setting of multiple public keys. The Key Privacy notion, formalized in [9], models the requirement that ciphertexts do not leak information about the public keys used to create them. Considering the CHK transform in the setting of multiple public keys as required to study the Key Privacy of the PKE scheme resulting from the application of the CHK transform, quite naturally gives rise to a multi-TA IBE setting of the type considered in this thesis. We show how to modify the CHK transform to reflect this setting. We then prove that the Key Privacy of the PKE scheme resulting from our modified CHK transform follows from a weak form of TA Anonymity for the underlying multi-TA IBE scheme. Our result gives the first generic method of constructing a PKE scheme in the Standard Model that is Key Private, as well as being IND-CCA secure. We also prove similar results for the Boneh-Katz transform [24], which builds on the ideas of [33] to give a more

efficient construction of PKE from IBE.

In Chapter 7 we consider the problem of how entities operating under distinct roots of trust in a coalition environment can flexibly and securely communicate with one another. In keeping with the theme of this thesis, we consider the identity based setting, with each entity being pre-configured with a private key from a particular TA, but where multiple, independent TAs are involved in the coalition. Our solution to the problem adapts the `BasicIdent` scheme of Boneh and Franklin [22]. It allows any entity to securely communicate with any other entity, even without knowing the TA with which the intended recipient is associated. To enable this, we assume that the TAs share some common parameters along the lines of the schemes discussed in the earlier chapters, and in addition co-operate to distribute certain additional public information to all entities. This allows entities to decrypt a ciphertext that was composed using the public parameters of one TA, using a private key issued by another. We also include a security analysis of our new approach.

Finally, in Chapter 8, we investigate the relationship between Identity Based Non Interactive Key Distribution (ID-NIKD) and IBE. We provide an extended security model for ID-NIKD, and a construction that converts an ID-NIKD scheme that is secure in this extended security model and satisfies some mild technical conditions, and produces an IND-ID-CPA secure IBE scheme. This conversion is used to explain the relationship between the ID-NIKD scheme of Sakai, Ohgishi and Kasahara [101] and the `BasicIdent` IBE scheme of Boneh and Franklin [22].

1.3 Related Work

In the following paragraphs, we outline some of the papers in the literature where multiple TAs have been considered and highlight the differences from our work. We also explore some of the related literature.

The earliest treatment of multiple TAs in IBE can be found in the papers by Chen *et al.* [41, 42, 43]. In these early works, the authors propose a number of interesting applications involving multiple TAs, that exploit the properties of the `BasicIdent` scheme of Boneh and Franklin [22]. Although no explicit mention is made in these works of the different TAs sharing common parameters, this is a prerequisite for the schemes to work. Also, these early works do not propose concrete definitions or any security analysis of the proposed schemes.

Secret Handshakes and Hidden Credentials

Secret Handshakes, introduced by Balfanz *et al.* [5], allow two parties to mutually and secretly authenticate, if and only if they are both members of the same group. A third party who is not a member of the group is not able to tell whether an entity is a member of the group by engaging them in the protocol. Thus, Secret Handshakes can be used in situations where members of a group need to identify each other, without revealing their group affiliations. Balfanz *et al.* [5] construct Secret Handshakes secure under the BDH assumption in the ROM, by essentially using the Sakai *et al.* ID-NIKD scheme [101] with pseudonymous identities. The application scenarios considered in [5] involve Secret Handshakes between parties that have been issued keys by different TAs (for example, a driver who has been issued a license in the form of a private key by a licensing authority and a policeman who has been issued a private key by a traffic authority) and for the schemes to work, these different authorities must share common parameters.

Hidden Credentials were introduced in [72, 28]. In contrast to a Secret Handshake scheme which requires both parties to mutually authenticate using their credentials, Hidden Credentials allow a sender to send a recipient a message that depends only on the recipient's credentials (senders need not even have any credentials of their own). The authors construct Hidden Credentials using IBE as the underlying primitive.

In [72, 28] the authors introduce the desirable security properties of a Hidden Credentials scheme. A policy is defined as a list of attributes (possibly along with the public parameters of a TA, although they refer to a TA using the more general term, Credential Authority). They define Credential Indistinguishability to be the indistinguishability of ciphertexts corresponding to different single credential policies. For complex policies, possibly incorporating public parameters of different Credential Authorities, a security notion termed Policy Indistinguishability is defined.

In the concrete Hidden Credential scheme presented in [72, Section 4.2], the authors assume explicitly that the different CAs share common parameters. Here, Credential Indistinguishability roughly corresponds to Recipient Anonymity of the IBE scheme and Policy Indistinguishability to our definition of TA Anonymity.

Motivated by these earlier works on Hidden Credentials, Holt considered the security of IBE in the multi-TA setting [71]. Two notions of Key Privacy for IBE were outlined in [71]. The first, termed "identity based indistinguishability of identity under chosen plaintext attack" (ID-II-CPA), is just the standard single-TA Recipient Anonymity notion. The second is termed "identity based indistinguishability of key generator under chosen plaintext attack" (ID-IKG-CPA), and is roughly similar to the notion of TA anonymity under chosen plaintext attack (m-TAA-CPA) which we introduce in Section 3.2.4. However, the ID-IKG-CPA security model in [71], while allowing corruption of TAs, does not allow the adversary to extract any user private keys at all. Our m-TAA-CPA model is strictly stronger, allowing both corruption of TAs and extraction of private keys (even for the challenge TA). Holt's work allows the adversary to dynamically instantiate new TAs during its attack but without any adversarial input to the set up process, while we set up all the TAs at the start of the security games. These two approaches are easily seen to have equivalent strength. Moreover, [71] only considers the CPA setting, showing that the `BasicIdent` scheme of [22] has ID-II-CPA and ID-IKG-CPA security. However, even the proofs for these

CPA cases are at best informal. In this thesis, we will consider the CCA setting, use stronger security notions, and give rigourous proofs.

Secret handshakes are also constructed in [36] from what the authors call CA-Oblivious PKE. A CA-Oblivious PKE scheme is an encryption scheme such that neither the public key, nor the ciphertext, reveals any information about the Certification Authority (CA), which certified the public key. The authors of [36] construct a CA-Oblivious PKE scheme based on the CDH assumption in the ROM, and consequently, a Secret Handshake scheme. Another credential system that differs from the basic Secret Handshake (which is defined as a key-agreement protocol) and Hidden Credentials (which is defined as an IBE scheme) is the Oblivious Signature-Based Envelope (OSBE) [84] which is an interactive protocol. A comparison of Secret Handshakes, Secret Handshakes from CA-Oblivious PKE, Hidden Credentials and OSBE can be found in [73].

Multi-Recipient IBE

Wang and Cao [112] gave examples of IBE schemes enjoying reduced ciphertext expansion and reduced computation when the sender sends the same message to a single identity using multiple, different master public keys belonging to different TAs (which share common parameters), such that the message can be recovered with a private key issued for that identity by any one of the TAs. However, the security models presented in [112] are the standard single-TA, indistinguishability based security models, and no consideration is given to how security may be affected by encrypting the same message using multiple master public keys. In addition, the schemes of [112] reuse randomness to enhance efficiency, and this is not formally addressed in the security analysis. Barbosa and Farshim [7] consider the security of multi-recipient IBE with randomness re-use, but only in the single-TA setting.

Multi-Authority ABE

Chase [37] has considered Attribute Based Encryption (ABE), a generalization of IBE, in the setting of multiple authorities. In her work, a user is equipped with private keys corresponding to attributes from different TAs and the user is only able to decrypt a ciphertext if in possession of a threshold of attributes from different TAs. Chase does not consider the issue of TA Anonymity.

Security Notions for IB-KEM with an Untrusted TA

Izabachene and Pointcheval [75] introduced two new notions of security for Identity Based KEMs (IB-KEMs), which they termed "Key Anonymity with respect to the Authority" and "Identity Based Non-Malleability" respectively. In these security models, the TA is untrusted. They gave a construction for an IB-KEM that meets both new notions of security and showed how it can be used to build a Password-Based Key-Exchange (PBKE) protocol. We note that the security notions we consider in our work are different from those introduced in [75] in that we assume the presence of multiple TAs and that we do not consider security notions where the TA is untrusted by the recipient.

Incomparable Public Key Encryption

Towards the goal of achieving anonymous encryption, Waters *et al.* [118] define a primitive which they call an Incomparable Public Key Encryption scheme (IPKE). Here a large number of public keys can be generated corresponding to a single private key. The different public keys must be unlinkable. In addition, Key Privacy is a desirable security property so that it is not possible to determine if different ciphertexts were produced using the same public key.

The KEM-DEM Paradigm

In practice, only small messages are encrypted using asymmetric cryptography. To perform encryption of large messages efficiently, asymmetric techniques are used to encrypt a one-time symmetric key and symmetric key techniques are used to encrypt the actual message. This approach is known as the KEM-DEM paradigm in the literature and formalized in [47]. The KEM or Key encapsulation Mechanism represents the public key part and the DEM or Data Encapsulation Mechanism represents the symmetric key part. The approach is attractive not only from the point of view of efficiency, but also for its modular design providing clear separation between various parts of the encryption scheme.

Bentahar *et al.* [15] extend the KEM-DEM paradigm to the identity based setting and formalize the notion of an ID-KEM. Combining an ID-KEM with an appropriate DEM gives rise to efficient IBE schemes capable of handling arbitrary length messages. Bentahar *et al.* [15] also give a generic method to construct an identity based KEM that is secure in a very strong sense from any IBE scheme that is secure in a weak sense and illustrate by instantiating their generic transform with the `BasicIdent` IBE scheme. Subsequently, Chen *et al.* [39] used the techniques from [15] to build an ID-KEM based on the Sakai-Kasahara IBE scheme [102]. In this thesis, we have focussed on formalizing the TA Anonymity security properties for multi-TA IBE and we have not considered similar extensions to the (multi-TA) identity based KEM-DEM setting.

1.4 Publications Associated with this Thesis

This thesis contains material from publications in association with Kenneth G. Paterson [94, 95, 96] and [18] in association with Kenneth G. Paterson, Kent D. Boklan and Zev Klagsbrun. The co-operation of all my co-authors is gratefully acknowledged.

Chapter 3 and Chapter 4 are based on [94]. Chapter 5 and Chapter 6 are based on [95]. Chapter 7 is based on [18]. Chapter 8 is based on [96].

Chapter 2

Background and Definitions

2.1 Introduction

In this chapter we introduce some basic notation, relevant concepts and background material that are used throughout. Cryptography is a subject that borrows heavily and freely from many different areas and it is not possible to provide an extensive coverage of all the relevant concepts. Nevertheless, we provide the minimum possible background required to read this thesis in a manner that is as self-contained as possible. References are given for further background information. Additional material is introduced as needed in later chapters to facilitate presentation.

2.2 Notation

We use $\bigoplus$ to denote the exclusive-or operation. If x is chosen uniformly at random from the set Y, we denote this as $x \xleftarrow{\$} Y$. The symbol $\perp$ denotes an error message. $\{0,1\}^*$ denotes the set of all bit strings and for a string $s \in \{0,1\}^*$, we denote the length of s as $|s|$. $\{0,1\}^n$ denotes the set of bit strings of length n. $s||t$ denotes the concatenation of two bit strings s and t. If s and t are not bit strings, then $s||t$ denotes the concatenation of their bit representations.

2.3 Basic Cryptographic Terminology

In this section we will introduce some basic terminology used in Cryptography, and Information Security in general. Although these terms are used rather fluidly in practice, it helps to have them defined from a given reference point and we will use the ISO 7498-2 standard [55] as our point of reference. Cryptography aims to achieve certain security goals, termed **security services**, via **security mechanisms**.

2.3.1 Cryptographic Services

The ISO 7498-2 defines five main categories of security services.

- **Authentication:** Corroboration that the entity at the other end of a communication link is the one claimed (entity authentication) and corroboration that the source of the data received is as claimed (data origin authentication).
- **Access Control:** Prevention of unauthorized use.
- **Data Confidentiality:** Keeping data secret from all but intended recipients.
- **Data Integrity:** Detection of manipulation by unauthorized entities.
- **Non-repudiation:** Preventing denial of actions and commitments.

2.3.2 Cryptographic Mechanisms

Security mechanisms exist to provide and support security services. ISO 7498-2 categorizes security mechanisms into eight types of specific security mechanisms and five types of pervasive security mechanisms. The eight specific security mechanisms, as the name suggests, are used to provide particular services. We are not concerned with pervasive security mechanisms in our work. The eight types of specific security mechanisms are

- **Encipherment mechanisms**
- **Digital Signature mechanisms**
- **Data Integrity mechanisms**
- **Authentication mechanisms**
- **Access Control mechanisms**
- **Traffic Padding mechanisms**
- **Routing Control mechanisms**
- **Notarization mechanisms**.

The term "Encipherment mechanisms" as used in the ISO 7498-2 standard conflicts with accepted usage in the cryptographic community. In this thesis, following the style in [50], it is used as a synonym for encryption. The first three kinds of security mechanisms appear frequently in this work.

We now introduce some terminology that appears frequently in cryptographic literature. A **cryptographic primitive** is a basic tool used to provide some security service. For example, encryption is a cryptographic primitive that is used to provide confidentiality. A **cryptographic scheme** is a specific set of algorithms used to provide some security service. For example, a digital signature scheme is used to provide authentication.

A **cryptographic protocol** is a distributed algorithm defined by a sequence of steps carried out by two or more entities so as to achieve a specific security objective. For example, SSL is a protocol that is used to achieve secure communication between clients and servers. An **entity** is a person or device that uses a cryptographic primitive or participates in any cryptographic scheme or protocol. Unfortunately, there is no

general consensus on how these terms are used in practice, with different authors having slightly different viewpoints. Often, these terms are used rather fluidly in practice, a luxury that we will also indulge in, to facilitate presentation. For example, we will most often use the term *scheme* in this work, referring to encryption schemes (rather than encryption primitives) and digital signature schemes (rather than digital signature mechanisms).

Further details on the ISO 7498-2 standard as well as other information regarding usage of cryptographic terminology can be found in the book by Dent and Mitchell [50].

2.4 Complexity Theory

Complexity theory provides mechanisms by which computational problems may be classified in terms of the resources, usually time, and storage space, required to solve them. Complexity theory is a vast and rich subject in its own right. In this section, we provide some basic terminology and definitions that we need.

Definition 2.1. *An **algorithm** is any computational procedure that takes some variable input and terminates with some output. An algorithm that follows the same execution path each time that it is executed with the same input is said to be a **deterministic algorithm**. An algorithm whose execution path may differ each time that it is executed with the same input is said to be a **randomized** or **probabilistic algorithm**.*

Definition 2.2. *The **running time** of an algorithm on a particular input is the number of steps or primitive operations executed before the algorithm terminates. The running time is usually expressed as a function of the input size. The **worst***

case running time *of an algorithm is an upper bound on the running time for any input. The* ***expected running time*** *of an algorithm is the average running time of an algorithm over all inputs of a specific size.*

We note here that we have not given a definition for a "primitive operation". Indeed this cannot be done unless we define appropriate models of computation. Unfortunately we are unable to devote a discussion to these topics in this thesis. It is usually non-trivial to derive the exact running time of algorithms and we then rely on approximations of the running time. The big-$\mathcal{O}$ notation is useful in this regard.

Definition 2.3. *For any two functions $f(l)$ and $g(l)$, we say that $f(l) = \mathcal{O}(g(l))$, if there exists a constant $c \geq 0$ and $l_c \geq 0$ such that $0 \leq f(l) \leq c.g(l)$ for every $l \geq l_c$.*

Definition 2.4. *The complexity of algorithms is measured with respect to the input size k. In the case of cryptographic algorithms, we call this the* ***security parameter****.*

Definition 2.5. *A function $\epsilon(k)$ is said to be* ***negligible*** *if, for every $c \geq 0$, there exists $k_c \geq 0$ such that $\epsilon(k) \leq k^{-c}$ for every $k \geq k_c$.*

Definition 2.6. *A* ***polynomial time algorithm*** *(PTA) is an algorithm whose worst case running time is of the form $\mathcal{O}(k^c)$, where k is the input size and c is a constant. A PTA that follows the same execution path each time it is executed with the same input is called* ***deterministic polynomial time****. A PTA that has access to a source of randomness such that its execution path may differ each time that it is executed with the same input, is called* ***probabilistic polynomial time*** *(PPT).*

We regard PTAs as being efficient. Algorithms whose running time cannot be bounded as in the case of PTAs are called non-polynomial time algorithms. These algorithms are said to be inefficient.

Definition 2.7. *Problems may be of the decisional or the computational type. Problems for which no PTAs are known to exist are said to be* ***infeasible*** *or* ***intractable****. Note that a problem that is currently intractable need not be intractable in the future. Many conjectures in complexity theory such as the* $P \neq NP$ *conjecture rest on the assumption that certain classes of problems are intractable.*

Further information on the topics we have touched upon in this section can be found in [108, 110].

2.5 Provable Security

Provable Security marks a departure from the traditional cryptographic method which was similar to the traditional iterative software development life cycle. Traditional software development was predominantly characterized by iterative design, build, deploy and patch cycles. In the traditional cryptographic approach, a cryptographic scheme would be designed, an ad-hoc security analysis conducted and it would then be deployed and assumed to be secure so long as no one found an attack against it. Often, the more complex the design, the more secure a scheme was assumed to be, although there may have been no actual basis for this assumption. If an attack was found, and assuming the original designers were made aware of the attack, it was either redesigned and redeployed, or discarded in favour of a newer scheme if it was considered irreparably "broken".

There have been attempts to introduce scientific rigour into cryptography. Shannon proved information theoretically that the one-time pad encryption scheme is unconditionally secure so long as the message is encrypted with a randomly generated key which is as long as the message and as long as the key is never re-used [105]. During the height of the cold war, the hot line between the Presidents of the U.S.A

and U.S.S.R. was allegedly encrypted using a one-time pad and it has been used in banks to secure high value transactions. However, using the one-time pad in general is infeasible given the impractical key-management issues.

Provable Security moves away from both ad-hoc design principles and the realm of unconditional security, and provides a concrete scientific framework to analyse cryptographic protocols against computationally bounded adversaries. The Provable Security paradigm is characterized by the following basic steps.

- The **precise specification** of a cryptographic scheme is first given. The specification captures the input/output behaviour of the various algorithms.
- An **adversarial model** is formally specified. The exact specifications of what a computationally bounded adversary is allowed (or not allowed) to do when launching an attack is given. It attempts to answer fundamental questions such as what it means for a cryptographic scheme to be "broken" and what it means for a cryptographic scheme to be secure.
- A **concrete instantiation** of the cryptographic scheme is given.
- Finally, a **reduction** is given that shows that any computationally bounded adversary with non-negligible advantage against the scheme can be converted into an adversary against an underlying "hard" problem.
- The scheme is then assumed to be secure based on the assumed security of the underlying hard problem, i.e. **security is conditional** on the security of the underlying hard problem.

There are a few important points to note. The security guarantee of a scheme is derived from the hardness assumptions of presumably well studied hard problems like factoring, or computing discrete logarithms in appropriate groups. So long as

we believe our assumptions to be true, we have reason to believe that no adversary that follows the adversarial model can break the scheme. The "proof" in Provable Security is essentially a reduction from the security of the scheme to an instance of a hard problem. We note that in the description above, we have defined the reduction asymptotically i.e. the reduction states that any computationally bounded adversary with non-negligible advantage against the scheme can be converted into an adversary against an underlying hard problem with non-negligible probability. There is no information on the tightness of the reduction i.e. the security of a scheme cannot be related in a concrete fashion to the hardness of the underlying hard problem. As such, the proof is conditional on the quality of the model, the hardness of the hard problem, the assumption that adversaries are computationally bounded and the tightness of the reduction, rather than being an absolute guarantee.

Provable Security has both its proponents and its critics. Proponents of Provable Security argue that it provides a rigourous framework in which cryptographic schemes can be designed and this is a concrete step forward in the absence of any other scientific method.

Critics on the other hand argue that the "proofs" in Provable Security are cumbersome and give false security guarantees. There are two frequent criticisms. Firstly, many new schemes are proven secure under new and unstudied hardness assumptions. Many of the hardness assumptions we state and use in this thesis for example have not been subject to the levels of scrutiny that the integer factorization problem or the discrete logarithm problem have been subjected to. It can be argued that as the field of Identity Based Cryptography is in its infancy, this is inevitable. It remains to be seen if many of these hardness assumptions withstand scrutiny. A second criticism is that as reductions are often stated asymptotically, there is no information on the tightness of the reduction i.e. the security of a scheme cannot be related in a concrete fashion to the hardness of the underlying hard problem.

The Practice Oriented Provable Security paradigm [8] aims to address some of these criticisms. In contrast to the asymptotic approach, in the Practice Oriented Provable Security approach [8] one attempts to provide a more exact or "concrete" reduction. While an asymptotic security reduction may be stated by saying that "scheme A is secure if B is hard", in the concrete setting, an attempt is made to convey more information about the reduction by stating the reduction along the lines of "scheme A is (t', ϵ') secure if problem B is (t, ϵ) secure." The concrete reduction conveys the information that any adversary against scheme A that runs in time t' and has success probability ϵ' can be converted into an adversary against the hard problem B that runs in time t with success probability ϵ. As the gap between t and t' and ϵ and ϵ' increases, the reduction is said to become less tight or weak. This has implications in practice as a weak reduction means that to get the same level of security, larger keys may need to be used. It is therefore desirable to aim for reductions as tight as possible.

This thesis follows the Provable Security paradigm. We will rigourously define both the functionalities of the cryptographic primitives as well as the adversarial models. Where we study concrete schemes, we will provide reductions to appropriate hard problems. We also strive to provide concrete security reductions.

A number of works deal with the Provable Security paradigm and its advantages and disadvantages. Interesting debates can be found in [80, 81, 64]. Further details on Provable Security, its pros and cons, as well as additional references can be found in the works by Dent [49, 48]. We do not discuss this issue further here.

2.6 Abstract Algebra

We will make use of basic concepts from abstract algebra freely. We present here some definitions and theorems, all stated without proofs. Further information can be found in introductory texts on the subject such as [69].

Throughout, $\mathbb{G}$ denotes a group, which is a set with an associated binary operation that satisfies the group axioms.

Definition 2.8. *The order of a group $\mathbb{G}$ is the number of elements in the group and is denoted by $|\mathbb{G}|$. The group $\mathbb{G}$ is said to be finite if $|\mathbb{G}|$ is finite.*

In this work, we will write the group operation multiplicatively and denote the group as $(\mathbb{G}, *)$ or simply $\mathbb{G}$ for the sake of clarity.

Definition 2.9. *A group $\mathbb{G}$ is cyclic if there exists an element $g \in \mathbb{G}$ such that for every element $x \in \mathbb{G}$, there exists an integer i such that $g^i = x$. Element g is called a generator of the group $\mathbb{G}$.*

We denote the set of natural numbers by $\mathbb{N}$ and the set of integers by $\mathbb{Z}$. For any positive integer n, we write $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ to denote the (ring of) integers modulo n. The group of units of $\mathbb{Z}_n$ is represented by $\mathbb{Z}_n^*$. These are the elements of $\mathbb{Z}_n$ that are relatively prime to n and therefore have an inverse under multiplication.

Definition 2.10. *The Euler totient function $\phi(n)$ is the order of $\mathbb{Z}_n^*$.*

A *field* is a set with two binary operations defined on it, satisfying the field axioms. We denote a field by $(\mathbb{F}, +, *)$ or simply $\mathbb{F}$. We do not discuss fields in detail here. Further information can be found in [69].

2.7 Elliptic Curves and Bilinear Pairings

Elliptic Curve Cryptography (ECC) has become an active area of research since the seminal works of Koblitz and Miller [88, 79]. Since the work of Boneh and Franklin [22], pairing based cryptography has also become one of the most active areas of research in cryptography. These are rich and non-trivial areas of specialization and although this thesis deals with pairing based IBE schemes, we do so from a cryptographic rather than from a mathematical viewpoint. To this end, we now provide a minimal introduction to these topics.

2.7.1 Elliptic Curves

Let q be a large prime and m an integer with $m \geq 1$. Let $\mathbb{F}_{q^m}$ be the finite field with q^m elements. Here, q denotes the *characteristic* of the field and m the *extension degree*. The multiplicative group of $\mathbb{F}_{q^m}$ is denoted by $\mathbb{F}^*_{q^m}$.

Then, the elliptic curve E over $\mathbb{F}_{q^m}$ is denoted by $E/\mathbb{F}_{q^m}$ and is defined to be the set of elements $(x, y) \in \mathbb{F}_{q^m} \times \mathbb{F}_{q^m}$ satisfying an equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

where $a_i \in \mathbb{F}_{p^m}$ for $i = 1, 2, 3, 4, 6$.

A point $P = (x, y) \in \mathbb{F}_{q^m} \times \mathbb{F}_{q^m}$ is said to be on the curve if it satisfies the above equation. $E(\mathbb{F}_{q^m})$ represents the set of points on the curve and together with a point at infinity denoted by ∞, forms an additive abelian group.

2.7.2 Bilinear Pairings

Now, suppose that $E(\mathbb{F}_{q^m})$ has a cyclic subgroup $\mathbb{G}$ of prime order p. We define the embedding degree or security multiplier to be the least integer $k \geq 0$ such that $p|q^{km} - 1$ and $p \not| q^l - 1$ for all $0 \leq l \leq k$. We let $\mathbb{G}_T$ denote the cyclic subgroup of $\mathbb{F}^*_{q^m}$

of prime order p. Then, an admissible bilinear pairing is a function e which maps a pair of elliptic curve points in $\mathbb{G}$ to an element in $\mathbb{G}_T$,

$$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T,$$

having the following properties:

- **Bilinearity:** $\forall g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, $e(g^a, g^b) = e(g, g)^{ab}$
- **Non-degenerate:** $e(g_1, g_2) \neq 1$ for some $g_1, g_2 \in \mathbb{G}$
- **Efficiently Computable:** There must be an efficient algorithm that computes the map e for any pair of inputs.

We note that although $\mathbb{G}$ is a subgroup of $E(\mathbb{F}_{q^m})$, itself an additive group, we have used multiplicative notation to denote the group operation in $\mathbb{G}$ and we will use this style throughout this thesis. In the literature, both additive and multiplicative notation have been used to denote the group operation in $\mathbb{G}$.

We note that in the background material on elliptic curves and pairings that we introduced, k denoted the embedding degree or security multiplier. Henceforth we use k to denote a security parameter in keeping with the style of standard cryptographic literature. Varying the size of the input parameter k, varies the key lengths and group sizes and in turn affects the security level of the cryptographic schemes in question. Where algorithms require the security parameter as input, it is given in unary notation, i.e. for a security parameter k, inputs to algorithms are given as 1^k. This is to ensure that algorithms running in polynomial time in their input size are in fact polynomial in k.

Admissible pairings can be derived from the modified Weil pairing or the Tate pairing [22, 57]. Some recent surveys on the applications of pairings in cryptography can be found in [92, 93].

We will deal with pairings in the abstract and define a pairing-friendly group generator so as to simplify presentation.

Definition 2.11. *A pairing-friendly group generator* `PairingGen` *is a polynomial time algorithm with input* 1^k *and output a tuple* $(\mathbb{G}, \mathbb{G}_T, e, p, g)$*. Here* $(\mathbb{G}, \mathbb{G}_T)$ *are groups of prime order* p*,* g *generates* $\mathbb{G}$*, and* $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ *is a bilinear, non-degenerate and efficiently computable map.*

For ease of presentation, we work exclusively in the setting where e is symmetric; our definitions and results can be generalized to the asymmetric setting where $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, with $\mathbb{G}_1$ and $\mathbb{G}_2$ being different groups. Further details concerning the basic choices that are available when using pairings in cryptography can be found in [60].

The cryptographic schemes we will consider in this thesis will have security that is based on hardness assumptions for problems in groups equipped with a pairing. For example, the BDH and DBDH problems are defined as follows.

Definition 2.12. *We define the advantage of an algorithm* $\mathcal{A}$ *in solving the Bilinear Diffie-Hellman (BDH) problem in* $(\mathbb{G}, \mathbb{G}_T)$ *to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{BDH}(k) = \Pr[\mathcal{A}(g^a, g^b, g^c) = e(g,g)^{abc}]$$

where $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$*. Here, we implicitly assume that parameters* $(\mathbb{G}, \mathbb{G}_T, e, p, g)$ *as output by* `PairingGen` *on input* 1^k*, are given to* $\mathcal{A}$ *as additional inputs. We say that the BDH problem is* hard *in* $(\mathbb{G}, \mathbb{G}_T)$ *if no PTA that solves the BDH problem in* $(\mathbb{G}, \mathbb{G}_T)$ *has a non-negligible advantage, as a function of the security parameter* k*.*

Definition 2.13. *We define the advantage of an algorithm* $\mathcal{A}$ *in solving the Decisional Bilinear Diffie-Hellman (DBDH) problem in* $(\mathbb{G}, \mathbb{G}_T)$ *to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{DBDH}(k) = \left|\Pr[\mathcal{A}(g^a, g^b, g^c, e(g,g)^{abc}) = 1] - \Pr[\mathcal{A}(g^a, g^b, g^c, Z) = 1]\right|$$

where $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$ *and* $Z \xleftarrow{\$} \mathbb{G}_T$. *Here, we implicitly assume that parameters* $(\mathbb{G}, \mathbb{G}_T, e, p, g)$ *as output by* `PairingGen` *on input* 1^k *are given to* $\mathcal{A}$ *as additional inputs. We say that the DBDH problem is* hard *in* $(\mathbb{G}, \mathbb{G}_T)$ *if no PTA that solves the DBDH problem in* $(\mathbb{G}, \mathbb{G}_T)$ *has a non-negligible advantage, as a function of the security parameter* k.

2.8 Identity Based Encryption

Although IBE followed PKE historically, we first introduce IBE formally as this is the focus of the thesis.

Definition 2.14. *An IBE scheme is defined in terms of four algorithms:*

- `Setup`: On input 1^k, outputs a master public key *mpk* which includes system parameters *params*, and a master secret key *msk*. We assume that *params* contains descriptions of the message and ciphertext spaces, MsgSp and CtSp. This algorithm is randomized.
- `KeyDer`: A key derivation algorithm that on input *mpk*, *msk* and identifier *id*, returns a private key usk_{id}. This algorithm may or may not be randomized.
- `Enc`: An encryption algorithm that on input *mpk*, identifier *id* and message $m \in$ MsgSp, returns a ciphertext $c \in$ CtSp. This algorithm is usually randomized. We will write $c = \texttt{Enc}(mpk, id, m)$ in general. When we wish to emphasize that randomness r (drawn from some space RSp) is used when performing an encryption, we will write $c = \texttt{Enc}(mpk, id, m; r)$.
- `Dec`: A decryption algorithm that on input *mpk*, a private key usk_{id} and a ciphertext $c \in$ CtSp, returns either a message $m \in$ MsgSp or a failure symbol $\perp$.

We assume identities are bit strings of arbitrary length, i.e. $id \in \{0,1\}^*$. However, concrete schemes may require identities to be drawn from some restricted sets. In such situations, hashing of bit-strings onto appropriate sets can be used to allow the use of arbitrary bit strings as identities.

These algorithms must satisfy the standard consistency requirement that decryption undoes encryption: $\forall m \in \mathsf{MsgSp}$, $\forall id \in \{0,1\}^*$, $\forall usk_{id} = \mathtt{KeyDer}(mpk, msk, id)$, if $c = \mathtt{Enc}(mpk, id, m)$ then $\mathtt{Dec}(mpk, usk_{id}, c) = m$.

2.8.1 Basic Security Notions for IBE

In the traditional single-TA identity based setting, the notions of security analogous to the IND-CPA and IND-CCA security notions for PKE were first formalized in [22]. In the IND-ID-CPA and IND-ID-CCA games for IBE, the adversary is also given access to a private key extraction oracle with suitable restrictions on its use. For the remainder of this thesis, we will suppress "ID" wherever it is clear from the context or in the associated text that we are dealing with identity based schemes.

We describe the IND-ID-CCA security notion which captures the property of message indistinguishability against chosen ciphertext attackers. This notion is defined in terms of the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The challenger $\mathcal{C}$ takes as input the security parameter 1^k, runs algorithm `Setup` of the IBE scheme, gives $\mathcal{A}$ mpk, and keeps msk to itself. $\mathcal{A}$ then runs in two phases:

- **Phase 1**: $\mathcal{A}$ issues a series of adaptively selected key derivation and decryption queries on identities id and identifier/ciphertext combinations (id, c) of its choice. These are replied to by $\mathcal{C}$ by using algorithms `KeyDer` and `Dec` and knowledge of msk.

- **Challenge:** After $\mathcal{A}$ decides to end Phase 1, it outputs an identity id^* and two equal length messages m_0 and m_1.

$\mathcal{C}$ selects $b \xleftarrow{\$} \{0,1\}$, sets $c^* = \texttt{Enc}(mpk, id^*, m_b)$ and gives c^* to $\mathcal{A}$. We require that id^* not be the subject of any key derivation query in Phase 1.

- **Phase 2**: This phase proceeds as in Phase 1, with the constraint that id^* not be the subject of any key derivation query and that (id^*, c^*) not be the subject of any decryption query.

- **Guess**: $\mathcal{A}$ outputs a bit b' and wins the game if $b' = b$.

The advantage of $\mathcal{A}$ against the IBE scheme in the above IND-ID-CCA security game is defined to be:

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-ID-CCA}}(k) = |\Pr[b' = b] - 1/2|$$

where the probability is measured over the random choices of coins of $\mathcal{A}$ and $\mathcal{C}$. A scheme is said to be IND-ID-CCA secure if the advantage of all PPT adversaries is negligible as a function of the security parameter k. Removing the adversary's access to the decryption oracle gives the weaker IND-ID-CPA security notion.

In the remainder of this thesis, security games such as the one above will be described in algorithmic notation, in terms of experiments, based on the choice of a bit b. To facilitate comparison, we give the IND-ID-CCA experiment, in algorithmic notation below.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{IND-ID-CCA-}b}(k)$

$(mpk, msk) \leftarrow \texttt{Setup}(1^k)$

$IDSet \leftarrow \emptyset,\ CSet \leftarrow \emptyset$

$(m_0, m_1, id^*, state) \leftarrow$

$\mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, mpk)$

$c^* \leftarrow \texttt{Enc}(mpk, id^*, m_b)$

$CSet \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $id^* \in IDSet$

`Or` $\{(id^*, c^*)\} \in CSet$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{KeyDer}(id)$

$IDSet \leftarrow IDSet \cup \{id\}$

$usk_{id} \leftarrow \texttt{KeyDer}(mpk, msk, id)$

`Return` usk_{id}

Oracle $\texttt{Dec}(id, c)$

$CSet \leftarrow CSet \cup \{(id, c)\}$

$usk_{id} \leftarrow \texttt{KeyDer}(mpk, msk, id)$

$m \leftarrow \texttt{Dec}(mpk, usk_{id}, c)$

`Return` m

The IND-ID-CCA Security Experiment.

Strictly speaking, we need to make sure that the messages are drawn from appropriate message spaces, that messages are distinct and where messages are assumed to be bit strings, they are of equal length. To simplify the presentation, we will assume these conditions as standard and not include them explicitly in our experiments. Where an oracle and an algorithm have the same name, their respective functions will be clear from the context and their inputs.

We will also use an alternative definition for the advantage of the adversary. The advantage of the adversary $\mathcal{A}$ in the IND-ID-CCA security experiment is defined to be:

$$\begin{aligned} & \mathbf{Adv}_{\mathcal{A}}^{\text{IND-ID-CCA}}(k) \\ = \ & |\Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{IND-ID-CCA-1}}(k) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{IND-ID-CCA-0}}(k) = 1]| \end{aligned}$$

i.e. the probability that an adversary $\mathcal{A}$ outputs the correct bit, minus the probability

that the adversary $\mathcal{A}$ outputs an incorrect bit. A scheme is said to be IND-ID-CCA secure if the advantage of all PPT adversaries $\mathcal{A}$ is negligible as a function of the security parameter k.

It can be shown in a fairly straightforward manner that both the game based and experiment based formulations are in fact equivalent up to a factor of 2 in the advantage. Consider the advantage of an adversary $\mathcal{A}$ in the IND-ID-CCA game we defined earlier.

We have, by definition

$$\mathbf{Adv}_{\mathcal{A}}(k) = |\Pr[b' = b] - 1/2|.$$

Now,

$$\begin{aligned}
& 2 \cdot \mathbf{Adv}_{\mathcal{A}}(k) \\
= \;& 2|\Pr[b' = b] - 1/2| \\
= \;& 2|\Pr[b' = 1|b = 1] \cdot \Pr[b = 1] + \Pr[b' = 0|b = 0] \cdot \Pr[b = 0] - 1/2| \\
= \;& 2|\tfrac{1}{2}\Pr[b' = 1|b = 1] + \tfrac{1}{2}\Pr[b' = 0|b = 0] - 1/2| \\
= \;& |\Pr[b' = 1|b = 1] - (1 - \Pr[b' = 0|b = 0])| \\
= \;& |\Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0]|.
\end{aligned}$$

The right hand side of the last equation is the probability that the adversary outputs the correct bit minus the probability that the adversary outputs an incorrect bit, which by definition is the advantage of the adversary in the IND-ID-CCA experiment. A similar argument holds for other indistinguishability based games and experiments in the remainder of this thesis.

So far, as per the provable security paradigm, we have provided a precise specification of IBE, given adversarial models and defined what it means for an IBE scheme to be secure in these models. To complete the discussion and give a flavour of the provable security style that we will use throughout this thesis, we will show a specific instantiation of an IBE scheme from the literature, that is secure in an appropriate security model and based on an appropriate hardness assumption.

2.8.2 Boneh-Franklin `BasicIdent` IBE

$\texttt{Setup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \texttt{PairingGen}(1^k)$.

Set $params =$
$(\mathbb{G}, \mathbb{G}_T, e, p, g, H_1, H_2, l)$
where $H_1 : \{0,1\}^* \to \mathbb{G}$,
$H_2 : \mathbb{G}_T \to \{0,1\}^l$ for some $l = l(k)$.
$\mathsf{MsgSp} = \{0,1\}^l$, $\mathsf{CtSp} = \mathbb{G} \times \{0,1\}^l$,
$\mathsf{RSp} = \mathbb{Z}_p^*$.

Set $s \xleftarrow{\$} \mathbb{Z}_p^*, h = g^s$.

Set $mpk = (params, h)$.

Set $msk = s$.

$\texttt{KeyDer}(mpk, msk, id)$:

Output
$usk_{id} = H_1(id)^{msk}$.

$\texttt{Enc}(mpk, id, m)$:

Parse mpk as $(params, h)$.

Set $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $T = e(H_1(id), h)^r$.

Output
$c = (u, v) = (g^r, m \oplus H_2(T))$.

$\texttt{Dec}(mpk, usk_{id}, c)$:

Parse c as $(u, v) \in \mathbb{G} \times \{0,1\}^l$.

Set $T = e(usk_{id}, u)$.

Output $m = v \oplus H_2(T)$.

The scheme `BasicIdent`.

The `BasicIdent` scheme [22] shown above is a concrete instantiation of an IBE scheme. To be able to say something "provably" about this scheme, we need to show a reduction from this scheme, to an instance of a hard problem.

In [22] the `BasicIdent` scheme is proved to be IND-ID-CPA secure in the Random

Oracle Model (ROM) [14], assuming the hardness of the BDH problem in groups output by `PairingGen`.

The above reduction is an asymptotic security reduction that conveys the idea that any IND-ID-CPA adversary against the `BasicIdent` scheme that succeeds with non-negligible success probability can be converted into an adversary against the BDH problem that succeeds with non-negligible success probability. However, as we believe the BDH problem to be hard i.e. we assume that no adversary exists that can solve the BDH problem with non-negligible success probability, no adversary against the `BasicIdent` scheme exists that can succeed with non-negligible success probability and the scheme is thus shown to be secure. We note that a concrete reduction in given in [22].

2.8.3 Recipient Anonymity for IBE

Key Privacy captures the security requirement that ciphertexts do not leak information about the public keys used to create them [9]. Key Privacy has surfaced as a desirable property in a number of applications [103, 30, 67, 1]. Specific PKE schemes such as ElGamal, Cramer-Shoup and RSA based schemes are known to be Key Private [9]. We will define Key Privacy for PKE schemes in Section 6.2.1. In the IBE setting, the equivalent notion to Key Privacy is that of Recipient Anonymity: the ciphertext should not leak the identity of the (intended) recipient. We describe the IND-RA-CCA security notion which simultaneously captures both the property of ciphertext indistinguishability and the property of Recipient Anonymity in IBE.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{IND-RA-CCA-}b}(k)$

$(mpk, msk) \leftarrow \texttt{Setup}(1^k)$

$IDSet \leftarrow \emptyset,\ CSet \leftarrow \emptyset$

$(m_0, m_1, id_0, id_1, state) \leftarrow$

$\quad \mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, mpk)$

$c^* \leftarrow \texttt{Enc}(mpk, id_b, m_b)$

$CSet \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $id_0 \in IDSet$ `Or` $id_1 \in IDSet$

`Or` $\{(id_0, c^*)\} \in CSet$

`Or` $\{(id_1, c^*)\} \in CSet$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{KeyDer}(id)$

$IDSet \leftarrow IDSet \cup \{id\}$

$usk_{id} \leftarrow \texttt{KeyDer}(mpk, msk, id)$

`Return` usk_{id}

Oracle $\texttt{Dec}(id, c)$

$CSet \leftarrow CSet \cup \{(id, c)\}$

$usk_{id} \leftarrow \texttt{KeyDer}(mpk, msk, id)$

$m \leftarrow \texttt{Dec}(mpk, usk_{id}, c)$

`Return` m

The IND-RA-CCA Security Experiment for IBE.

We can describe other security notions by placing suitable restrictions on the IND-RA-CCA security notion. Removing the adversary's access to the decryption oracle gives the IND-RA-CPA security notion. Considering an adversary that selects $id_0 = id_1$ in the output from the `find` stage gives us the IND-ID-CCA security notion. We will say in brief that setting $id_0 = id_1$ gives the IND-ID-CCA security notion. Setting $id_0 = id_1$ and removing the adversary's access to the decryption oracle gives the IND-ID-CPA security notion. Setting $m_0 = m_1$ gives the RA-CCA security notion. Setting $m_0 = m_1$ and removing the adversary's access to the decryption oracle gives the RA-CPA security notion. We note that what we term the RA-CPA security notion, has in the literature been referred to as the IBE-ANO-CPA security notion [2]. Here, we use "RA" in place of "ANO" because we will study

two forms of anonymity, *viz* Recipient Anonymity (RA) and TA Anonymity (TAA). The advantage of the adversary in these security models is defined as for the IND-ID-CCA security experiment in Section 2.8.1 and we will assume the same for the security models given in the remainder of this chapter. As a consequence, we no longer give explicit definitions for the adversary's advantage in the remainder of this chapter. IBE schemes known to offer Recipient Anonymity in the ROM include the CPA secure `BasicIdent` scheme of Boneh and Franklin [22] (the proof for this can be found in [2]). In the Standard Model, the IBE schemes of Gentry [62] enjoy Recipient Anonymity in both the CPA and CCA setting.

2.8.4 Selective Identity Security for IBE

Weaker models of security for IBE are the selective-id security models where the adversary commits to the identity that it will use in the challenge phase before it is given the TA's parameters.

Selective-id security notions (for the single TA setting) were first introduced by Canetti *et al.* [32]. They introduced the notion of binary tree encryption (BTE) schemes and showed how a Hierarchical IBE (HIBE) scheme (see Section 2.8.5) can be constructed from any BTE scheme. They constructed a HIBE scheme that is secure in the Standard Model in a selective-id security model. This construction is mainly of theoretical interest as there is a very large computational overhead in performing decryption – one pairing computation for each bit of the identity. Subsequently, Canetti *et al.* [33] gave a construction that builds an IND-CCA secure PKE scheme from a selective-id IND-CPA secure IBE scheme and a strongly secure one-time signature scheme. We will study this result in Chapter 6 with a view to building Key Private PKE schemes and we will give there an appropriate selective-id security notion for IBE in the setting of multiple TAs. The $\mathtt{BB}_1$ and $\mathtt{BB}_2$ IBE schemes

in [19] are proved to be selective-id IND-CPA secure. We will consider the Recipient Anonymity of these schemes, as well as the TA Anonymity of multi-TA analogues of these schemes in Chapter 5.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{s-id IND-CCA-}b}(k)$

$id^* \leftarrow \mathcal{A}(1^k)$

$(mpk, msk) \leftarrow \texttt{Setup}(1^k)$

$IDSet \leftarrow \emptyset,\ CSet \leftarrow \emptyset$

$(m_0, m_1, state) \leftarrow$

$\mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, mpk)$

$c^* \leftarrow \texttt{Enc}(mpk, id^*, m_b)$

$CSet \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $id^* \in IDSet$

`Or` $\{(id^*, c^*)\} \in CSet$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{KeyDer}(id)$

$IDSet \leftarrow IDSet \cup \{id\}$

$usk_{id} \leftarrow \texttt{KeyDer}(mpk, msk, id)$

`Return` usk_{id}

Oracle $\texttt{Dec}(id, c)$

$CSet \leftarrow CSet \cup (id, c)$

$usk_{id} \leftarrow \texttt{KeyDer}(mpk, msk, id)$

$m \leftarrow \texttt{Dec}(mpk, usk_{id}, c)$

`Return` m

The selective-id IND-CCA Security Experiment.

Removing the adversary's access to the decryption oracle gives the selective-id IND-CPA security notion which we will refer to in later chapters.

2.8.5 Hierarchical IBE

Hierarchical Identity Based Encryption (HIBE) is an extension to IBE first proposed by Horwitz and Lynn [74]. Similar to a regular IBE, there is a TA that can generate a private key usk_{id_1} for an identity id_1 using its master secret key (we assume for convenience that identities are bit strings of length l i.e. $id \in \{0,1\}^l$). In addition, a private key usk_{id_1} can be used to generate a private key usk_{id_1,id_2} for an identity vector $(id_1, id_2) \in (\{0,1\}^l)^2$ and so on for identity vectors of maximum length $h = h(k)$ corresponding to the maximum depth of the HIBE scheme. Such a HIBE scheme is called an h-level HIBE scheme.

Again, as in the case of IBE schemes, a sender can encrypt a message for any identity vector $v = (id_1, id_2, \ldots, id_{h'})$ with $h' \leq h$, using only the master public key of the "root" TA, such that the ciphertext can be decrypted by anyone who possesses the secret key $usk_{id_1,id_2,\ldots,id_{h'}}$.

The security model for HIBE resembles that for regular IBE, except that the model needs to account for the fact that the challenger must not be in possession of secret keys corresponding to identity vectors that are prefixes of the challenge identity vector (as this would allow the challenger to derive a private key for the challenge identity).

Horwitz and Lynn [74] first proposed a construction for a 2-level HIBE scheme in the ROM. The first efficient constructions for an h-level HIBE scheme in the ROM were proposed in [63], building on the IBE scheme of [22].

2.9 Public Key Encryption

As described in the introduction, in a PKE scheme, a public key is used to encrypt a message and the message can only be recovered by a recipient in possession of a private key corresponding to the public key used to perform the encryption. The private key must be mathematically related to the public key and it must be infeasible to recover the private key from knowledge of the public key only. The concept of PKC (also known as asymmetric cryptography) was made known in the open cryptographic community in the year 1976 [51] and marked a shift in the thinking of the cryptographic community, which was up to that time dominated by symmetric cryptography.

We present below a formal definition of a PKE scheme. Our definition is slightly non-standard in that our definition includes a `CommonSetup` algorithm that outputs a set of common parameters. This facilitates presentation of the Key Privacy notion later in this thesis. Our presentation can incorporate the standard definitions by having the `CommonSetup` algorithm output just the security parameter 1^k.

Definition 2.15. *A PKE scheme is defined in terms of four algorithms:*

- `CommonSetup`: A "common parameter generation" algorithm that takes as input 1^k and returns some common parameters $\mathcal{I}$. This algorithm is usually randomized. (We note that $\mathcal{I}$ may simply be 1^k or include some additional information, for example in a discrete-log based scheme this may be a description of a group of prime order and a generator for that group.)

- `KeyGen`: A randomized key generation algorithm that on input $\mathcal{I}$, outputs a public key PK and a matching private key SK. Associated to each public/private key pair are a message space MsgSp and a ciphertext space CtSp.

- `Enc`: An encryption algorithm that on input PK and message m returns a ciphertext c. This algorithm is usually randomized.
- `Dec`: A decryption algorithm that on input a secret key SK and a ciphertext c, returns either a message m or a failure symbol $\perp$.

These algorithms must satisfy the standard consistency requirement that decryption undoes encryption: i.e. $\forall (PK, SK) \leftarrow \texttt{KeyGen}(\mathcal{I})$, $\forall m \in \mathsf{MsgSp}$, $\forall c = \texttt{Enc}(PK, m)$, $\texttt{Dec}(SK, c) = m$.

2.9.1 Basic Security Notions for PKE

We define the IND-CCA security experiment for PKE next. The IND-CCA security notion is widely believed to be the minimum level of security required for a PKE scheme that is to be used as part of a wider system. Shoup [106] provides an accessible justification for requiring IND-CCA security for PKE schemes used in practice.

Experiment $\mathbf{Exp}^{\text{IND-CCA-}b}_{\mathcal{A}}(k)$	
$\mathcal{I} \xleftarrow{\$} \texttt{CommonSetup}(1^k)$	
$(PK, SK) \xleftarrow{\$} \texttt{KeyGen}(\mathcal{I})$	
$CSet \leftarrow \emptyset$	Oracle $\texttt{Dec}(c)$
$(m_0, m_1, state) \leftarrow \mathcal{A}^{\texttt{Dec}}(\texttt{find}, PK)$	$CSet \leftarrow CSet \cup \{c\}$
$c^* \leftarrow \texttt{Enc}(PK, m_b)$	$m \leftarrow \texttt{Dec}(SK, c)$
$CSet \leftarrow \emptyset$	`Return` m
$b' \leftarrow \mathcal{A}^{\texttt{Dec}}(\texttt{guess}, c^*, state)$	
`If` $c^* \in CSet$ `Then Return` 0	
`Else Return` b'.	

The IND-CCA Security Experiment for PKE.

A weaker notion of security for PKE is the IND-CPA security notion which can be obtained by removing the adversary's access to the decryption oracle in the IND-CCA experiment.

2.10 Hash Functions

A cryptographic hash function is defined as follows.

Definition 2.16. *A hash function H is an efficient algorithm that maps an input $x \in \{0,1\}^*$ of arbitrary but finite bit length, to a fixed length output $H(x)$ in a finite range.*

From the definition, it is clear that collisions are unavoidable in a hash function, yet we require that such collisions are hard to find in practice. A hash function may be required to meet one or more of the following informal security requirements.

- **Pre-image resistance:** Given a pre specified output y it should be infeasible to find a pre-image x such that $H(x) = y$.
- **Second pre-image resistance:** Given x, it is infeasible to find $x' \neq x$ such that $H(x) = H(x')$.
- **Collision resistance:** It should be infeasible to find any two distinct inputs x and x' such that $H(x) = H(x')$.

Security requirements for Hash Functions have been formalized in [100]. Secure hash functions are notoriously difficult to design and a number of attacks have been found on hash functions used in practice [115, 116, 114, 113, 16].

We note that we have given a rather limited definition of a hash function, in that the output of a hash function has been defined over the set of all strings of a specified

length. In practice, the output space may have a more complex structure. In this thesis, for example, we will encounter hash functions that map an arbitrary input onto a point in an elliptic curve group and such a mapping may require a deterministic encoding and intermediate mapping functions.

Hash functions are central in the design of cryptographic schemes, especially in the design of efficient schemes. However, it has proved difficult to obtain proofs of security for schemes that employ hash functions. This brings us to the Random Oracle Model.

2.11 The Random Oracle Model

Bellare and Rogaway [14] introduced the Random Oracle Model (ROM) which attempts to capture the concept of an ideal hash function. In this model, proofs of security are obtained by replacing hash functions with "Random Oracles" that output truly random values for every distinct input. Proofs of security are often easier to construct in the ROM, rather than in a model without Random Oracles (termed the Standard Model). Schemes proven secure in the Standard Model tend to be less efficient than schemes proven secure in the ROM and it is argued that proofs of security in the ROM provide security guarantees so long as the hash function that is used to instantiate the Random Oracle in an actual implementation has no obvious weaknesses.

The ROM has become a popular tool in the construction of security proofs, yet it is not without its critics [10, 65, 91]. Canetti *et al.* [31] show "contrived" schemes that can be proven secure in the ROM, yet are insecure when instantiated with any concrete hash function. In this thesis, we will refrain from entering the debate of whether or not the Standard Model is better than the ROM, but rather present constructions in both settings.

Chapter 3

Security Notions for Multi-TA IBE

3.1 Introduction

As discussed in Section 1.1, in almost all the existing literature on IBE, with a small number of exceptions as discussed in Section 1.3, there is a single global TA issuing keys to all users in the system, and all ciphertexts are created using the public parameters of that single global TA.

To formally study the setting of multi-TA IBE, we extend the usual indistinguishability and Recipient Anonymity notions for IBE security to the multi-TA setting, and in addition, formalize the notion of TA Anonymity. This chapter provides much of the conceptual framework for what follows in the remainder of the thesis.

3.2 Multi-TA IBE and Multi-TA Security

A multi-TA IBE scheme is defined in terms of five algorithms:

- `CommonSetup`: On input 1^k, this algorithm outputs *params*, a set of system parameters.
- `TASetup`: On input *params*, this algorithm outputs a master public key *mpk* (which includes *params*), and a master secret key *msk*. This algorithm is randomized and executed independently for each TA.
- `KeyDer`, `Enc`, `Dec`: These are all as per a normal IBE scheme (see Definition 2.14).

To facilitate the presentation of security notions and schemes in the multi-TA setting, we will let $\mathcal{T} = \{ta_i : 1 \leq i \leq n\}$ represent the set of (labels of) TAs, where $n = n(k) \in \mathbb{N}$ is the number of TAs. For example, we will refer to the master secret key and master public key for a particular $ta \in \mathcal{T}$ as msk_{ta} and mpk_{ta} and the private key for a user with identity *id* corresponding to a particular $ta \in \mathcal{T}$ as $usk_{id,ta}$.

Note that we explicitly include a `CommonSetup` algorithm which outputs *params*, a set of system parameters shared by all TAs. The different TAs will of course have different master public keys and master secret keys obtained by running `TASetup` on input *params*, for each TA. Our model is capable of handling situations where no such common system parameters are used, simply by setting *params* to be the security parameter 1^k. For the concrete schemes considered in this paper, common parameters are needed in order to achieve our notion of TA Anonymity; doing so without having some (non-trivial) common parameters is an interesting open problem.

In the security experiments defined below, the adversary has access to one or more of the following oracles. In the multi-TA IBE setting, an adversary is able to corrupt one or more TAs, which is modelled by giving the adversary access to a `Corrupt`

oracle which returns the master secret key corresponding to a specified TA. This is in contrast to the single TA setting where no reasonable security exists if the single TA is corrupted. In addition, similar to the single-TA setting, the adversary may have access to a key derivation oracle `KeyDer` which returns the private key for an identity, corresponding to a particular TA, and to a decryption oracle `Dec` which returns the decryption of a ciphertext, corresponding to a particular identity and TA. Suitable restrictions on the use of these oracles will be specified in individual security experiments so that the adversary is not able to win by performing trivial queries.

TASet represents the set of TAs that have been compromised, i.e. queried for their master secret keys, $IDSet_{ta}$ represents the set of identities queried for private keys for each $ta \in \mathcal{T}$, while $CSet_{ta}$ represents the set of identity/ciphertext pairs on which decryption queries have been performed for each $ta \in \mathcal{T}$. In these experiments, $MPK = \{mpk_{ta} : ta \in \mathcal{T}\}$ and $MSK = \{msk_{ta} : ta \in \mathcal{T}\}$ represent the set of all master public keys and all master secret keys, respectively.

For each experiment defined below, we define the advantage of the adversary for a given "notion-attack" combination to be:

$$\mathbf{Adv}_{\mathcal{A}}^{\text{notion-atk}}(k) = \left|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{notion-atk-1}}(k) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{notion-atk-0}}(k) = 1]\right|.$$

A scheme is said to be "notion-atk" secure if the advantage of any PPT adversary is negligible as a function of the security parameter k.

We focus below on chosen ciphertext attacks (CCA) for three different security notions: indistinguishability of messages, Recipient Anonymity and TA Anonymity. In each of the first two cases (namely, indistinguishability of messages and Recipient Anonymity), setting $n = 1$ and removing access to the `Corrupt` oracle gives us a security notion that coincides with a known (single-TA) IBE security notion. Formally, to obtain a (single-TA) IBE scheme, we need to combine the `CommonSetup` and `TASetup` algorithms of the multi-TA scheme into a single `Setup` algorithm. In what follows,

we will refer to this scheme as being the *corresponding single-TA IBE scheme*. In the third case, TA Anonymity, the security notion is inappropriate for the single-TA setting.

Removing adversarial access to decryption oracles gives the same notions of security against a chosen plaintext attack (CPA). Later in this thesis we will also discuss security notions obtained by removing adversarial access to the TA Corrupt oracle, and variants requiring the adversary to specify the identities used in the Challenge phase ahead of time.

3.2.1 m-IND-CCA Security

We first define the m-IND-CCA security notion that captures indistinguishability of messages under chosen ciphertext attacks in the multi-TA setting. The m-IND-CCA security notion is a straightforward extension of the IND-CCA security notion for regular (single-TA) IBE. The adversary has access to a key derivation oracle, a decryption oracle and in addition, a TA corruption oracle. In the challenge phase, the adversary is given a ciphertext corresponding to one of two messages chosen by the adversary and corresponding to a specified identity and TA. The adversary's task is to determine which message was used to create the ciphertext. Suitable restrictions are placed on the use of the oracles so as to prevent the adversary from winning the game by making a trivial oracle query. We define the m-IND-CCA security notion formally below. Removing the adversary's access to the decryption oracle gives the m-IND-CPA security notion.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-IND-CCA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m_0, m_1, id^*, ta^*, state) \leftarrow$

$\mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK)$

$c^* \leftarrow \texttt{Enc}(mpk_{ta^*}, id^*, m_b)$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $ta^* \in TASet$

`Or` $id^* \in IDSet_{ta^*}$

`Or` $\{(id^*, c^*)\} \in CSet_{ta^*}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Corrupt}(ta)$

$TASet \leftarrow TASet \cup \{ta\}$

`Return` msk_{ta}

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The m-IND-CCA Security Experiment.

The following theorem relates the security of a multi-TA IBE scheme to the security of the corresponding single-TA IBE scheme.

Theorem 3.1. *Let $atk \in \{CPA, CCA\}$. Then for any m-IND-atk adversary $\mathcal{A}$ against a multi-TA IBE scheme with n TAs, having advantage ε and running in time t, there exists an IND-atk adversary $\mathcal{B}$ against the corresponding single-TA IBE scheme with advantage $\frac{\varepsilon}{n}$ and running in time $O(time(\mathcal{A}))$.*

Proof. Suppose there is an m-IND-atk adversary $\mathcal{A}$ against a multi-TA IBE scheme having advantage ε and running in time t. We show how to construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the IND-atk security of the corresponding single-TA IBE scheme.

$\mathcal{B}$'s input from its challenger is the public key *mpk* of the single-TA scheme which, by our definitions, includes some public parameters *params* that are output by the `CommonSetup` part of the `Setup` algorithm of the single-TA scheme. $\mathcal{B}$'s task is to break the IND-atk property of the scheme and it does this by acting as a challenger for $\mathcal{A}$

$\mathcal{B}$ first sets up a multi-TA IBE scheme. It does this by first taking *params* from the public key of the single-TA scheme. If n is the number of TAs in the multi-TA setting, it first picks $i \xleftarrow{\$} \{1, \ldots n\}$ and sets $mpk_{ta_i} = mpk$ (note it does not know the corresponding master secret key for this TA). For the remaining $n-1$ TAs it generates the master public keys and master secret keys itself using the `TASetup` algorithm. $\mathcal{B}$ now gives the set of n master public keys to $\mathcal{A}$.

$\mathcal{A}$ then makes a series of TA corrupt queries, extraction queries (and decryption queries in the CCA setting) which $\mathcal{B}$ answers using either its knowledge of the relevant master secret key or by relaying queries to its own challenger. If $\mathcal{A}$ makes a corrupt query on ta_i then $\mathcal{B}$ aborts the simulation.

$\mathcal{A}$ also makes a single query in the challenge phase; if $\mathcal{A}$'s selected TA in this phase is not ta_i, then $\mathcal{B}$ aborts, otherwise $\mathcal{B}$ again uses its own challenger to answer the query. When $\mathcal{A}$ terminates by outputting a bit b', $\mathcal{B}$ simply relays this bit to its challenger.

This completes our description of $\mathcal{B}$'s simulation. Note that $\mathcal{A}$'a view of the

simulation is identical to its view in a real attack, unless $\mathcal{B}$ aborts. Moreover $\mathcal{B}$'s output b' is correct if $\mathcal{A}$'s is. It is easy to see that $\mathcal{B}$ aborts with probability $1/n$ and that $\mathcal{B}$ runs in time $\mathcal{O}(time(\mathcal{A}))$. The result follows.

□

3.2.2 m-RA-CCA Security

Halevi [68] provides a simple sufficient condition for an IND-CPA secure PKE scheme to have Key Privacy: given public keys pk_0 and pk_1 and the encryption of a random message under pk_b for a bit b chosen at random, even a computationally unbounded adversary should have negligible advantage in determining which public key was used. Abdalla *et al.* [2] extended this condition in the study of Recipient Anonymity of IND-CPA secure IBE schemes. We further extend these ideas to study multi-TA IBE schemes in the following sections.

Our m-RA-CCA security notion below captures the notion of Recipient Anonymity in the presence of chosen ciphertext attackers, in the multi-TA setting. The m-RA-CCA security notion is similar to the m-IND-CCA security notion except that in the challenge phase the adversary is given a ciphertext corresponding to a message, TA and one of two identities chosen by the adversary and the adversary's task is to determine which identity was used to create the ciphertext.

The single-TA version of the m-RA-CPA security notion was studied in detail in [2], where it was named IBE-ANO-CPA security. Here, we use "RA" in place of "ANO" because we wish to study two forms of anonymity, *viz* Recipient Anonymity (RA) and TA Anonymity (TAA).

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-RA-CCA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m^*, id_0, id_1, ta^*, state) \leftarrow$

$\mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK)$

$c^* \leftarrow \texttt{Enc}(mpk_{ta^*}, id_b, m^*)$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $ta^* \in TASet$

`Or` $id_0 \in IDSet_{ta^*}$

`Or` $id_1 \in IDSet_{ta^*}$

`Or` $\{(id_0, c^*)\} \in CSet_{ta^*}$

`Or` $\{(id_1, c^*)\} \in CSet_{ta^*}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Corrupt}(ta)$

$TASet \leftarrow TASet \cup \{ta\}$

`Return` msk_{ta}

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The m-RA-CCA Security Experiment.

Theorem 3.2. *Let $atk \in \{CPA, CCA\}$. Then for any m-RA-atk adversary $\mathcal{A}$ against a multi-TA IBE scheme with n TAs, having advantage ε and running in time t, there exists an RA-atk adversary $\mathcal{B}$ against the corresponding single-TA IBE scheme with advantage $\frac{\varepsilon}{n}$ and running in time $O(time(\mathcal{A}))$.*

The proof is similar to that of Theorem 3.1 and is omitted.

3.2.3 m-RA-RE-CCA Security

The m-RA-RE-CCA security notion is similar to the m-RA-CCA security notion except that while handling the challenge phase, the challenger encrypts a random message m' in place of the adversary's choice of message m, hence the choice "RE" in the nomenclature m-RA-RE-CCA to signify "randomized encryption".

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-RA-RE-CCA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m^*, id_0, id_1, ta^*, state) \leftarrow$

$\mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK)$

$m' \xleftarrow{\$} \mathsf{MsgSp}$

$c^* \leftarrow \texttt{Enc}(mpk_{ta^*}, id_b, m')$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $ta^* \in TASet$ `Or` $id_0 \in IDSet_{ta^*}$

`Or` $id_1 \in IDSet_{ta^*}$

`Or` $\{(id_0, c^*)\} \in CSet_{ta^*}$

`Or` $\{(id_1, c^*)\} \in CSet_{ta^*}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Corrupt}(ta)$

$TASet \leftarrow TASet \cup \{ta\}$

`Return` msk_{ta}

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The m-RA-RE-CCA Security Experiment.

In order to establish the m-RA-CPA/m-RA-CCA security of concrete schemes, it is helpful to work with these related m-RA-RE-CPA/m-RA-RE-CCA security notions. Our treatment here follows that of [2], with appropriate modifications for the multi-TA setting.

The following result relates the notions of m-RA-atk security and m-RA-RE-atk security; a single-TA version of this result for atk = CPA was given in [2].

Lemma 3.1. *Let* `m-IBE` *be a multi-TA IBE scheme that is m-IND-atk secure and m-RA-RE-atk secure. Then* `m-IBE` *is also m-RA-atk secure. Here atk* $\in \{CPA, CCA\}$.

Proof. Let $\mathcal{A}$ be a PTA attacking the m-RA-atk security of a scheme `m-IBE`. It is easy to construct PTAs $\mathcal{A}_1$, $\mathcal{A}_3$ attacking the m-IND-atk security of `m-IBE`, and a PTA $\mathcal{A}_2$ attacking m-RA-RE-atk security of `m-IBE` such that:

$$
\begin{aligned}
& \mathbf{Adv}^{\text{m-RA-atk}}_{\mathcal{A}}(k) \\
= \;& |\Pr[\mathbf{Exp}^{\text{m-RA-atk-1}}_{\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-atk-0}}_{\mathcal{A}}(k) = 1]| \\
= \;& |\Pr[\mathbf{Exp}^{\text{m-RA-atk-1}}_{\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-1}}_{\mathcal{A}}(k) = 1] \\
& + \Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-1}}_{\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-0}}_{\mathcal{A}}(k) = 1] \\
& + \Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-0}}_{\mathcal{A}}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-atk-0}}_{\mathcal{A}}(k) = 1]| \\
\leq \;& |\Pr[\mathbf{Exp}^{\text{m-RA-atk-1}}_{\mathcal{A}_1}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-1}}_{\mathcal{A}_1}(k) = 1]| \\
& + |\Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-1}}_{\mathcal{A}_2}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-0}}_{\mathcal{A}_2}(k) = 1]| \\
& + |\Pr[\mathbf{Exp}^{\text{m-RA-RE-atk-0}}_{\mathcal{A}_3}(k) = 1] - \Pr[\mathbf{Exp}^{\text{m-RA-atk-0}}_{\mathcal{A}_3}(k) = 1]| \\
\leq \;& \mathbf{Adv}^{\text{m-IND-atk}}_{\mathcal{A}_1}(k) + \mathbf{Adv}^{\text{m-RA-RE-atk}}_{\mathcal{A}_2}(k) + \mathbf{Adv}^{\text{m-IND-atk}}_{\mathcal{A}_3}(k)
\end{aligned}
$$

□

3.2.4 m-TAA-CCA Security

As discussed earlier, TA Anonymity is a necessary condition to achieve fully private communication thwarting adversarial activity like traffic analysis in the multi-TA setting. The m-TAA-CCA security notion formalizes TA Anonymity: a ciphertext should not leak which TA's master public key was used to compute the ciphertext. In this security experiment, the adversary has access to a key derivation oracle, a decryption oracle and in addition, a TA corruption oracle. In the challenge phase, the adversary is given a ciphertext corresponding to a message chosen by the adversary, corresponding to a specified identity and one of two TAs specified by the adversary. The adversary's task is to determine which TA's public parameters were used to create the ciphertext. Suitable restrictions are placed on the use of the oracles so as to prevent the adversary from winning the game by making a trivial oracle query.

We note here that Anonymous Hierarchical IBE (AHIBE) [2, 27] is related to, but different from, our notion of TA Anonymity for multi-TA IBE. In AHIBE, a single root TA generates public parameters and a master secret, using which the master secrets of all sub-TAs are produced. Ciphertexts are then anonymous, in that an adversary cannot distinguish which identity was used when producing a ciphertext, where now identities are comprised of a vector of strings identifying a hierarchy of TAs and a final user.

As already mentioned, in our multi-TA IBE setting, there is no single root TA, but rather a group of independent TAs (who may share some common parameters). The "right" generalization of our multi-TA IBE concept to the HIBE setting would then involve multiple, independent root TAs, each being the root of a tree of TAs and users. Thus we would have a forest of trees, and would then wish to study anonymity properties of ciphertexts in this multi-HIBE setting. We discuss this further in Chapter 9.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-TAA-CCA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m^*, id^*, ta_0, ta_1, state) \leftarrow$

$\mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK)$

$c^* \leftarrow \texttt{Enc}(mpk_{ta_b}, id^*, m^*)$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $ta_0 \in TASet$

`Or` $ta_1 \in TASet$

`Or` $id^* \in IDSet_{ta_0}$

`Or` $id^* \in IDSet_{ta_1}$

`Or` $\{(id^*, c^*)\} \in CSet_{ta_0}$

`Or` $\{(id^*, c^*)\} \in CSet_{ta_1}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Corrupt}(ta)$

$TASet \leftarrow TASet \cup \{ta\}$

`Return` msk_{ta}

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The m-TAA-CCA Security Experiment.

3.2.5 m-TAA-RE-CCA Security

In order to establish the m-TAA-CPA/m-TAA-CCA security of concrete schemes, it is helpful to work with the related m-TAA-RE-CPA/m-TAA-RE-CCA security notions which we describe next.

The m-TAA-RE-CCA security notion defined below is similar to the m-TAA-CCA security notion except that while handling the challenge phase, the challenger encrypts a random message m' in place of the adversary's choice of message m, hence the choice "RE" in the nomenclature m-RA-RE-CCA to signify "randomized encryption".

Experiment $\mathbf{Exp}^{\text{m-TAA-RE-CCA-}b}_{\mathcal{A}}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m^*, id^*, ta_0, ta_1, state) \leftarrow$

$\mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK)$

$m' \xleftarrow{\$} \mathsf{MsgSp}$

$c^* \leftarrow \texttt{Enc}(mpk_{ta_b}, id^*, m')$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $ta_0 \in TASet$

`Or` $ta_1 \in TASet$

`Or` $id^* \in IDSet_{ta_0}$

`Or` $id^* \in IDSet_{ta_1}$

`Or` $\{(id^*, c^*)\} \in CSet_{ta_0}$

`Or` $\{(id^*, c^*)\} \in CSet_{ta_1}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Corrupt}(ta)$

$TASet \leftarrow TASet \cup \{ta\}$

`Return` msk_{ta}

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The m-TAA-RE-CCA Security Experiment.

Lemma 3.2. *Let* `m-IBE` *be a multi-TA IBE scheme that is m-IND-atk secure and m-TAA-RE-atk secure. Then* `m-IBE` *is also m-TAA-atk secure. Here* $atk \in \{CPA, CCA\}$.

The proof is similar to that of Lemma 3.1 and is omitted.

3.2.6 A Combined Security Notion

We have so far given individual security experiments for ciphertext indistinguishability, Recipient Anonymity and TA Anonymity and investigated the relationships between them. Schemes that meet one or more of these individual security notions may be suitable for specific applications.

For the most general case, we define an m-IND-RA-TAA-CCA experiment that simultaneously captures ciphertext indistinguishability, Recipient Anonymity, and TA Anonymity in the multi-TA setting against chosen ciphertext adversaries. Here, the adversary has access to a key derivation oracle, a decryption oracle and in addition, a TA corruption oracle. In the challenge phase, the adversary specifies two message, identity and TA tuples and is given a ciphertext corresponding to one of the two tuples. The adversary's task is to determine which tuple was used to create the ciphertext. Suitable restrictions are placed on the use of the oracles so as to prevent the adversary from winning the game by making a trivial oracle query.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{m-IND-RA-TAA-CCA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m_0, m_1, id_0, id_1, ta_0, ta_1, state) \leftarrow$

$\mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK)$

$c^* \leftarrow \texttt{Enc}(mpk_{ta_b}, id_b, m_b)$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Corrupt},\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $ta_0 \in TASet$

`Or` $ta_1 \in TASet$

`Or` $id_0 \in IDSet_{ta_0}$

`Or` $id_1 \in IDSet_{ta_1}$

`Or` $\{(id_0, c^*)\} \in CSet_{ta_0}$

`Or` $\{(id_1, c^*)\} \in CSet_{ta_1}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Corrupt}(ta)$

$TASet \leftarrow TASet \cup \{ta\}$

`Return` msk_{ta}

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The m-IND-RA-TAA-CCA Security Experiment.

By placing suitable restrictions on the m-IND-RA-TAA-CCA security notion, we can define other, weaker security notions appropriate to the multi-TA IBE setting. For example, removing the decryption oracle gives the m-IND-RA-TAA-CPA security notion. This notion still allows the adversary access to the `Corrupt` oracle and may be too strong for some applications. Removing the adversary's access to the `Corrupt`

oracle gives a restricted version of the m-IND-RA-TAA-CCA notion, which we denote as r-m-IND-RA-TAA-CCA. Removing the adversary's access to both the decryption and `Corrupt` oracles gives the r-m-IND-RA-TAA-CPA security notion.

As discussed in Section 2.8.4, weaker models of security for IBE are the selective-id security models where the adversary commits to the identity that he will use in the challenge phase ahead of time, i.e. before he is given the TA's parameters. Selective-id versions of the multi-TA security notions that we have discussed in this chapter can be given in the usual straightforward manner by having the adversary commit to the identity or identities at the start of the security experiment.

Furthermore, setting $m_0 = m_1$, $id_0 = id_1$ or $ta_0 = ta_1$ gives security notions appropriate to specific circumstances. For example, we can define a security notion where Recipient Anonymity of the IBE scheme is not required by setting $id_0 = id_1$.

Lemma 3.3. *Let `m-IBE` be a multi-TA IBE scheme that is m-IND-atk secure, m-RA-atk secure and m-TAA-atk secure. Then `m-IBE` is also m-IND-RA-TAA-atk secure. Here atk* $\in \{CPA, CCA\}$.

Proof. The proof (informally) follows by noting that if `m-IBE` is m-TAA-atk secure, then the challenger may replace the triple (ta_0, id_0, m_0) with (ta_1, id_0, m_0) in its response to the challenge query without the adversary being able to detect the change. Likewise, using m-RA-atk security, the challenger may then replace (ta_1, id_0, m_0) with (ta_1, id_1, m_0). Finally, using m-IND-atk security, the challenger can replace (ta_1, id_1, m_0) with (ta_1, id_1, m_1), again, without the adversary being able to detect the change. This informal argument can be made rigourous using a sequence of games. □

A combined m-IND-RA-CCA security notion can also be defined and it is easy to show that m-IND-RA-CCA security holds for a scheme that has both m-IND-CCA

and m-RA-CCA security, using a similar strategy as above. In the single-TA setting, we obtain IND-RA-CCA and IND-RA-CPA security notions. The latter security notion for IBE was used to prove the security of PEKS schemes in [2]. Similarly, we can define combined m-IND-TAA-CPA and m-IND-TAA-CCA security notions.

In the following chapter we will study multi-TA versions of IBE scheme in the ROM with respect to the security notions we have introduced in this chapter.

Chapter 4

Multi-TA IBE in the Random Oracle Model

4.1 Introduction

In two separate but related strands of work [59, 58], Fujisaki and Okamoto studied the problem of building PKE schemes which are secure in a very strong sense from PKE schemes which are secure in a weaker sense, in the ROM.

In this chapter, we introduce a modified version of the Fujisaki-Okamoto transform in [58] for the multi-TA setting, proving that our modified transform lifts security and anonymity properties from the CPA to the CCA setting, in the ROM. We then apply these results to study the security and anonymity of the Boneh-Franklin [22] and the Sakai-Kasahara [102] IBE schemes in the multi-TA setting.

As well as formalizing the notion of TA Anonymity, our work also establishes new results concerning the Recipient Anonymity of important IBE schemes. For example, to the best of our knowledge, no CCA secure variant of the Boneh-Franklin IBE scheme was previously known to have Recipient Anonymity. Moreover, we show that the Sakai-Kasahara scheme (and a CCA secure variant of it) enjoys Recipient Anonymity, contradicting a claim of [26].

4.2 Background

In [59], Fujisaki and Okamoto gave a generic transform that takes any OW-CPA secure PKE scheme satisfying a mild technical condition (γ-uniformity) and outputs a PKE scheme that is IND-CCA secure in the ROM. Yang *et al.* [120] investigated how to adapt this particular Fujisaki-Okamoto (FO) technique to the identity based setting. They first showed that straightforward application of the technique does take an IBE scheme that is OW-CPA secure and gives an IBE scheme that is IND-CCA secure, but with a loose reduction. They then proposed a modified transform with the same security properties, but with a tighter reduction. Their modification entails the introduction of additional parameters, namely the identity of the recipient, as input to the hash function used in the scheme.

Similarly, in [58], Fujisaki and Okamoto gave a generic transform that takes any IND-CPA secure PKE scheme and outputs a PKE scheme that is IND-CCA secure in the ROM. The security analysis in [58] is significantly simpler than that of [59]. Kitagawa *et al.* [78] investigated how to modify this particular FO technique for the identity based setting and again showed that straightforward application of the technique does take an IBE scheme that is IND-CPA secure and gives an IBE scheme that is IND-CCA secure but with a loose reduction. They then proposed a modified transform with the same security properties but with a tighter reduction. Once again, their modification entails the introduction of additional parameters, namely the identity of the recipient, as input to the hash function used in the scheme.

4.3 Extending the Fujisaki-Okamoto Transform to Multi-TA IBE

Kitagawa *et al.* [78] modified the transform of [58], by introducing additional parameters, namely the identity of the recipient, as input to the hash function used in the scheme.

We now extend the ideas of [58, 78] to describe a modified FO transform for IBE in the multi-TA setting. We include further additional parameters in the input to the hash function used in the scheme, namely the public parameters of the TA. This allows us to efficiently respond to hash queries, simplifies book-keeping in the proof, and yields a simulation that has a reduced running time in comparison to an application of the unmodified Fujisaki-Okamoto transform.

We are able to show that in the multi-TA setting, we can apply this modified transform to build an IBE scheme that has m-IND-RA-TAA-CCA security, defined in Section 3.2.6, from an IBE scheme that is m-IND-RA-TAA-CPA secure and γ-uniform.

We begin by defining a suitable notion of γ-uniformity for the multi-TA setting.

Definition 4.1. *Let* Π *be a multi-TA IBE scheme with space of randomness* RSp*. Then* Π *is said to be* γ*-uniform if, for any fixed choice of* $c \in \mathsf{CtSp}$*,* $m \in \mathsf{MsgSp}$*,* $id \in \{0,1\}^*$ *and* $ta \in \mathcal{T}$*, we have:*

$$\Pr\left[c = \mathsf{Enc}(mpk_{ta}, id, m; r) : r \xleftarrow{\$} \mathsf{RSp}\right] \leq \gamma.$$

The Modified Fujisaki-Okamoto Transform for Multi-TA IBE

Let $\Pi' = \{\texttt{CommonSetup}', \texttt{TASetup}', \texttt{KeyDer}', \texttt{Enc}', \texttt{Dec}'\}$ be a multi-TA IBE scheme for messages of length $l_0 + l_1$ i.e. we assume that Π' has message space $\{0,1\}^{l_0+l_1}$. Let RSp be the space of randomness used by $\texttt{Enc}'$.

Then $\Pi = \{\texttt{CommonSetup}, \texttt{TASetup}, \texttt{KeyDer}, \texttt{Enc}, \texttt{Dec}\}$ denotes a new multi-TA IBE scheme for messages of length l_0 with algorithms defined as follows.

- `CommonSetup`: As in $\texttt{CommonSetup}'$, but in addition, we pick a hash function $H : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^{l_0} \times \{0,1\}^{l_1} \rightarrow \mathsf{RSp}$.
- `TASetup`: As in $\texttt{TASetup}'$.
- `KeyDer`: As in $\texttt{KeyDer}'$.
- `Enc`: This algorithm takes as input mpk_{ta} for $ta \in \mathcal{T}$, $id \in \{0,1\}^*$, and a message $m \in \{0,1\}^{l_0}$. Its output is:

$$\texttt{Enc}(mpk_{ta}, id, m; \sigma) = \texttt{Enc}'(mpk_{ta}, id, m||\sigma; H(mpk_{ta}, id, m, \sigma))$$

  where $\sigma \xleftarrow{\$} \{0,1\}^{l_1}$. So Π' has randomness space $\{0,1\}^{l_1}$.

- `Dec`: Let c denote a ciphertext to be decrypted using a private key $usk_{id,ta}$ issued by a trusted authority ta with master public key mpk_{ta} for identity id. This algorithm works as follows:

  1. Compute $m' = \texttt{Dec}'(mpk_{ta}, usk_{id,ta}, c)$.
  2. Let $m = [m']^{l_0}$ and $\sigma = [m']_{l_1}$ where $[a]^b$ and $[a]_b$ denote the first and last b bits of a string a respectively.
  3. If $\texttt{Enc}'(mpk_{ta}, id, m||\sigma; H(mpk_{ta}, id, m, \sigma)) = c$ then output m as the decryption of c. Else, output $\perp$.

Theorem 4.1. *Modelling H as a Random Oracle, if Π' is a multi-TA IBE scheme that is m-IND-RA-TAA-CPA secure and γ-uniform for some negligible γ, then Π is m-IND-RA-TAA-CCA secure.*

In more detail, suppose Π' is a γ-uniform multi-TA IBE scheme. Let $\mathcal{A}$ be an m-IND-RA-TAA-CCA adversary which has advantage ϵ against Π and which runs in time t. Suppose $\mathcal{A}$ makes at most q_H queries to H, at most q_E extraction queries, and at most q_D decryption queries. Suppose further that executing $\texttt{Enc}'$ once needs at most time τ. Then there is an m-IND-RA-TAA-CPA adversary $\mathcal{B}$ with running time t', which has advantage at least ϵ' against Π' such that

$$\epsilon' = (\epsilon + \frac{1}{2} - \frac{q_h}{2^{l_1} - 1})(1 - \gamma)^{q_d} - \frac{1}{2}$$

and

$$t' = t + \mathcal{O}(q_h \tau).$$

Proof. Suppose there is an m-IND-RA-TAA-CCA adversary $\mathcal{A}$ against Π with advantage ϵ and running in time t. We show how to construct an adversary $\mathcal{B}$ that uses $\mathcal{A}$ to break the m-IND-RA-TAA-CPA-security of Π'.

$\mathcal{B}$'s input is the set of all master public keys MPK. $\mathcal{B}$ gives $\mathcal{A}$ the set MPK. $\mathcal{A}$ also has access to Random Oracle H that is controlled by $\mathcal{B}$. $\mathcal{A}$ then makes a series of queries which $\mathcal{B}$ answers as follows.

- **H-queries:** $\mathcal{B}$ maintains a list of tuples $\langle mpk_i, id_i, m_i, \sigma_i, g_i, c_i \rangle$. We refer to this list as the H^{list}. The list is initially empty.

  When $\mathcal{A}$ makes a H query on (mpk, id, m, σ), $\mathcal{B}$ responds as follows:

  - If the query (mpk, id, m, σ) already appears in a tuple $\langle mpk_i, id_i, m_i, \sigma_i, g_i, c_i \rangle$ then $\mathcal{B}$ responds with $H(mpk, id, m, \sigma) = g_i$.

  - Otherwise $\mathcal{B}$ picks $g \xleftarrow{\$} \mathsf{RSp}$, generates $c = \mathtt{Enc}(mpk_{ta}, id, m||\sigma; g)$, adds the tuple $\langle mpk, id, m, \sigma, g, c\rangle$ to the H^{list} and responds to $\mathcal{A}$ with $H(mpk, id, m, \sigma) = g$.

- **Corrupt Queries:** If $\mathcal{A}$ issues a TA corrupt query on $ta \in \mathcal{T}$, then $\mathcal{B}$ simply relays this query to its challenger, which responds with the corresponding master secret key msk_{ta}. $\mathcal{B}$ then passes the resulting key to $\mathcal{A}$.

- **Extraction Queries:** If $\mathcal{A}$ issues an extraction query on (ta, id), then $\mathcal{B}$ forwards (ta, id) to its challenger, which responds with the private key $usk_{id,ta}$. $\mathcal{B}$ forwards this key to $\mathcal{A}$.

- **Decryption Queries:** If $\mathcal{A}$ issues a decryption query on (ta, id, c), $\mathcal{B}$ responds as follows:

  - Searches for a tuple $\langle mpk_i, id_i, m_i, \sigma_i, g_i, c_i\rangle$ from the H^{list} such that $mpk_{ta} = mpk_i$, $id = id_i$ and $c = c_i$.
  - If such a tuple exists, then outputs m, else outputs $\perp$.

- **Challenge:** $\mathcal{A}$ outputs data $(ta_0, ta_1, id_0, id_1, m_0, m_1)$ on which it wishes to be challenged. $\mathcal{B}$ chooses two l_1 bit strings σ_0 and σ_1 uniformly at random, subject to the condition that they be distinct and sends $(ta_0, ta_1, id_0, id_1, m_0||\sigma_0, m_1||\sigma_1)$ to its challenger. $\mathcal{B}$'s challenger picks a random bit b and sets

$$c^* = \mathtt{Enc}(mpk_{ta_b}, id_b, m_b||\sigma_b; r)$$

  where $r \in \mathsf{RSp}$. $\mathcal{B}$ forwards c^* to $\mathcal{A}$.

After the Challenge query has been issued, if the adversary $\mathcal{A}$ makes a hash oracle query on either $(ta_0, id_0, m_0, \sigma_0)$ or $(ta_1, id_1, m_1, \sigma_1)$ then the adversary $\mathcal{B}$ simply outputs $b' = 0$ or $b' = 1$, respectively, as its guess for the value of the bit b. If neither hash query is made then, at the end of $\mathcal{A}$'s attack, $\mathcal{B}$ simply outputs the same bit b' that $\mathcal{A}$ outputs. $\mathcal{B}$ wins if $b' = b$. This completes our description of the simulation.

Our analysis now follows closely the analysis in [58]. We define the following events and probabilities.

Let $\Pr[\mathtt{Succ}\mathcal{A}]$ be the probability that adversary $\mathcal{A}$ outputs a bit $b' = b$. Similarly, let $\Pr[\mathtt{Succ}\mathcal{B}]$ be the probability that adversary $\mathcal{B}$ outputs a bit $b' = b$. For notational convenience, we let ϵ denote $\mathcal{A}$'s advantage in the simulation.

Let $\mathtt{Ask}_b$ be the event that $\mathcal{A}$ asks a hash query that coincides with $(mpk_{ta_b}, id_b, m_b, \sigma_b)$ and $\mathtt{Ask}_{\bar{b}}$ be the event that $\mathcal{A}$ asks a hash query that coincides with $(mpk_{ta_{\bar{b}}}, id_{\bar{b}}, m_{\bar{b}}, \sigma_{\bar{b}})$. Notice that these two queries are distinct because $\sigma_0 \neq \sigma_1$.

We define $\mathcal{F}$ to be the event that $\mathcal{B}$ fails to answer a decryption query correctly at some point during the game so that $\Pr[\neg\mathcal{F}]$ is the probability that $\mathcal{B}$ answers all decryption queries correctly during the simulation. Now,

$$\begin{aligned}\Pr[\mathtt{Succ}\mathcal{A}] &= \Pr[\mathtt{Succ}\mathcal{A}|\mathtt{Ask}_b] \cdot \Pr[\mathtt{Ask}_b] \\ &\quad + \Pr[\mathtt{Succ}\mathcal{A}|(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}] \cdot \Pr[(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}})] \\ &\quad + \Pr[\mathtt{Succ}\mathcal{A}|(\neg\mathtt{Ask}_b) \wedge (\neg\mathtt{Ask}_{\bar{b}})] \cdot \Pr[(\neg\mathtt{Ask}_b) \wedge (\neg\mathtt{Ask}_{\bar{b}})].\end{aligned}$$

Similarly,

$$\begin{aligned}\Pr[\mathtt{Succ}\mathcal{B}] &= \Pr[\mathtt{Succ}\mathcal{B}|\mathtt{Ask}_b] \cdot \Pr[\mathtt{Ask}_b] \\ &\quad + \Pr[\mathtt{Succ}\mathcal{B}|(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}] \cdot \Pr[(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}})] \\ &\quad + \Pr[\mathtt{Succ}\mathcal{B}|(\neg\mathtt{Ask}_b) \wedge (\neg\mathtt{Ask}_{\bar{b}})] \cdot \Pr[(\neg\mathtt{Ask}_b) \wedge (\neg\mathtt{Ask}_{\bar{b}})].\end{aligned}$$

From the conditions of the simulation, we have the following:

$$\begin{aligned}
\Pr[\mathtt{Succ}\mathcal{B}|\mathtt{Ask}_b] &= 1, \\
\Pr[\mathtt{Succ}\mathcal{B}|(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}] &= 0, \\
\Pr[\mathtt{Succ}\mathcal{A}|(\neg\mathtt{Ask}_b) \wedge (\neg\mathtt{Ask}_{\bar{b}})] &= \Pr[\mathtt{Succ}\mathcal{B}|(\neg\mathtt{Ask}_b) \wedge (\neg\mathtt{Ask}_{\bar{b}})].
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pr[\mathtt{Succ}\mathcal{B}] - \Pr[\mathtt{Succ}\mathcal{A}] &= \Pr[\mathtt{Ask}_b](1 - \Pr[\mathtt{Succ}\mathcal{A}|\mathtt{Ask}_b]) \\
&\quad + \Pr[(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}](0 - \Pr[\mathtt{Succ}\mathcal{A}|(\neg\mathtt{Ask}_b \wedge \mathtt{Ask}_{\bar{b}})]) \\
&\geq -\Pr[(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}].
\end{aligned}$$

Since even a computationally unbounded adversary has no information about what the string $\sigma_{\bar{b}}$ is (except that it is distinct from σ_b and so is uniformly distributed on a set of size $2^{l_1} - 1$), and our adversary makes at most q_h queries to the oracle H, we infer that $\Pr[(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}] \leq \frac{q_h}{2^{l_1}-1}$. Hence,

$$\begin{aligned}
\Pr[\mathtt{Succ}\mathcal{B}] &\geq \Pr[\mathtt{Succ}\mathcal{A}] - \Pr[(\neg\mathtt{Ask}_b) \wedge \mathtt{Ask}_{\bar{b}}] \\
&\geq \epsilon + \tfrac{1}{2} - \tfrac{q_h}{2^{l_1}-1}.
\end{aligned}$$

The event $\mathcal{F}$ can occur only when $\mathcal{A}$ submits a decryption query (ta, id, c) such that

$$c = \mathtt{Enc}(mpk_{ta}, id, m||\sigma; H(mpk_{ta}, id, m, \sigma))$$

without first querying H on input $(mpk_{ta}, id, m, \sigma)$.

Given the values ta, id, c, there is at most one possible message $m' = m||\sigma$ that could result from decrypting ciphertext c under the private key $usk_{id,ta}$, namely $m' = \mathtt{Dec}(mpk_{ta}, usk_{id,ta}, c)$. Applying the definition of γ-uniformity, and noting that the randomness r that would be used to form c for the scheme Π' is still uniformly distributed whenever the relevant hash query has not been made, we see that $\mathcal{B}$ fails

to properly answer each decryption query with probability at most γ. Therefore $\Pr[\neg\mathcal{F}] \leq (1-\gamma)^{q_d}$.

Hence, we have

$$\mathbf{Adv}_{\mathcal{B}}(k) = \Pr[\mathtt{Succ}\mathcal{B}] \cdot \Pr[\neg\mathcal{F}] - \frac{1}{2} \geq (\epsilon + \frac{1}{2} - \frac{q_h}{2^{l_1}-1})(1-\gamma)^{q_d} - \frac{1}{2}.$$

For the running time analysis, note that in addition to the running time of $\mathcal{A}$, the adversary $\mathcal{B}$ has to run the encryption algorithm `Enc` at most q_h times. Therefore $t' = t + \mathcal{O}(q_h\tau)$. □

Notice that the above theorem as stated requires the initial scheme Π' to have all three security properties (IND, RA and TAA) in order to convert from CPA-security to CCA-security. In fact, it is easy to prove versions of Theorem 4.1 that convert IND-RA-CPA security to IND-RA-CCA security and IND-TAA-CPA security to IND-TAA-CCA security. However, the proof technique does not allow us to prove that the transform preserves either of our anonymity properties in isolation – we need the base scheme Π' to also be IND secure.

4.4 Applying the Modified FO Transform to the `BasicIdent` Scheme

We describe and analyse a multi-TA scheme `m-BasicIdent` that is based on the `BasicIdent` scheme from [22]. The `BasicIdent` scheme was described in Section 2.8.2 and the multi-TA analogue is defined as follows:

$\texttt{CommonSetup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow$
$\texttt{PairingGen}(1^k)$.

Output $params =$
$(\mathbb{G}, \mathbb{G}_T, e, p, g, H_1, H_2, l)$
where $H_1 : \{0,1\}^* \to \mathbb{G}$ and
$H_2 : \mathbb{G}_T \to \{0,1\}^l$ for some $l = l(k)$.
$\mathsf{MsgSp} = \{0,1\}^l$,
$\mathsf{CtSp} = \mathbb{G}_1 \times \{0,1\}^l$,
$\mathsf{RSp} = \mathbb{Z}_q^*$.

$\texttt{TASetup}(params)$

Set $s \xleftarrow{\$} \mathbb{Z}_p^*$, $h = g^s$.

Set $mpk = (params, h)$.

Set $msk = s$.

Output (mpk, msk).

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$:

Set
$usk_{id,ta} = H_1(id)^{msk_{ta}}$.

Output $usk_{id,ta}$.

$\texttt{Enc}(mpk_{ta}, id, m)$:

Parse
mpk_{ta} as $(params, h_{ta})$.

Set $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $T = e(H_1(id), h_{ta})^r$.

Output
$c = (g^r, m \oplus H_2(T))$.

$\texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$:

Parse c as
$(u, v) \in \mathbb{G} \times \{0,1\}^l$.

Set $T = e(usk_{id,ta}, u)$.

Output $m = v \oplus H_2(T)$.

The scheme `m-BasicIdent`.

We now show the scheme that results from applying our modified Fujisaki-Okamoto transform to the `m-BasicIdent` scheme above.

`CommonSetup`(1^k):

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow$ `PairingGen`(1^k).

Output $params =$
$(\mathbb{G}, \mathbb{G}_T, e, p, g, H_1, H_2, H_3, l_0, l_1, l)$
where $H_1 : \{0,1\}^* \to \mathbb{G}$,
$H_2 : \mathbb{G}_T \to \{0,1\}^n$
for some $l = l(k)$, $l_0 + l_1 = l$, and
$H_3 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^{l_0} \times \{0,1\}^{l_1} \to \mathbb{Z}_p^*$.
$\mathsf{MsgSp} = \{0,1\}^{l_0}$,
$\mathsf{CtSp} = \mathbb{G}_1 \times \{0,1\}^l$,
$\mathsf{RSp} = \{0,1\}^{l_1}$.

`TASetup`: As in `TASetup`

`KeyDer`: As in `KeyDer`

`Enc`(mpk_{ta}, id, m):

Parse mpk_{ta} as $(params, h_{ta})$.

Set $\sigma \xleftarrow{\$} \{0,1\}^{l_1}$.

Set $r = H_3(mpk_{ta}, id, m, \sigma)$.

Set $T = e(H_1(id), h_{ta})^r$.

Output $c = (g^r, (m||\sigma) \oplus H_2(T))$.

`Dec`$(mpk_{ta}, usk_{id,ta}, c)$:

Parse c as $(u, v) \in \mathbb{G} \times \{0,1\}^l$.

Set $T = e(usk_{id,ta}, u)$.

Set $m' = v \oplus H_2(T)$.

Set $m = [m']^{l_0}$ and $\sigma = [m']_{l_1}$.

If $g^r = g^{H_3(mpk_{ta}, id, m, \sigma)}$ output m as the decryption of c. Else, output $\perp$.

The scheme `FO-m-BasicIdent`.

In the original definition of the FO transform (see Section 4.3) a full re-encryption is performed by the decryption algorithm. The shortened comparison in the last step in the `FO-m-BasicIdent` scheme above is a scheme specific optimization.

Lemma 4.1. *The multi-TA scheme* ***m-BasicIdent*** *is m-IND-CPA secure, assuming the hardness of the BDH problem in groups output by* ***PairingGen****.*

Proof. The single-TA scheme corresponding to `m-BasicIdent` is nothing other than the Boneh-Franklin `BasicIdent` scheme, whose IND-CPA security is known to rest on the hardness of the BDH problem in groups output by `PairingGen` [22]. Applying Theorem 3.1 gives us the desired result. □

The following result is an extension of a result from [2] that showed that the `BasicIdent` scheme has Recipient Anonymity against CPA attackers.

Lemma 4.2. *The multi-TA scheme* ***m-BasicIdent*** *is m-RA-CPA secure and m-TAA-CPA secure, assuming the hardness of the BDH problem in groups output by* ***PairingGen****.*

Proof. Ciphertexts c in the `m-BasicIdent` scheme have two parts, namely $u = g^r$ and $v = m \oplus H_2(T)$. The value u is chosen uniformly at random from $\mathbb{G}$. If the message m is chosen uniformly at random from $\{0,1\}^l$ then v is also distributed uniformly in $\{0,1\}^l$ and is independent of $H_2(T)$. Thus, in both 0 and 1 versions of the m-RA-RE-CPA and m-TAA-RE-CPA experiments, the ciphertext c has exactly the same distribution and any adversary in the corresponding RE experiments will have zero advantage. By Lemma 4.1, `m-BasicIdent` is m-IND-CPA secure. Applying Lemmas 3.1 and 3.2 yields m-RA-CPA and m-TAA-CPA security for `m-BasicIdent`, assuming the hardness of the BDH problem in groups output by `PairingGen`. □

Lemma 4.3. *The* ***m-BasicIdent*** *scheme is γ-uniform for $\gamma = 1/p$.*

Proof. In the `m-BasicIdent` scheme, the first component of the ciphertext is $u = g^r$ where $r \xleftarrow{\$} \mathbb{Z}_p^*$. It is then immediate that `m-BasicIdent` is γ uniform with $\gamma =$

$1/p$. □

Theorem 4.2. *The scheme* `FO-m-BasicIdent` *obtained by applying the modified FO transform to the scheme* `m-BasicIdent` *is m-IND-RA-TAA-CCA secure, assuming the hardness of the BDH problem in groups output by* `PairingGen`*.*

Proof. We obtain the above result by combining Lemmas 4.1, 4.2 with Lemmas 3.3, Lemma 4.3 and Theorem 4.1. □

Thus we have obtained an efficient multi-TA IBE scheme enjoying indistinguishability, Recipient Anonymity and TA Anonymity for the CCA setting, in the ROM. We note as a corollary of our analysis that the single-TA version of our scheme offers Recipient Anonymity. To the best of our knowledge, this is the first such result for a CCA secure variant of `BasicIdent`.

4.5 Applying the Modified FO Transform to the Sakai-Kasahara Scheme

The Sakai-Kasahara IBE scheme [102] has an alternative (and attractive) private key extraction algorithm compared to the Boneh-Franklin scheme. We define `m-BasicSK`, a multi-TA version of this scheme using symmetric pairings, immediately below, and then provide a security analysis sketch.

$\texttt{CommonSetup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \texttt{PairingGen}(1^k)$.

Output $params =$

$(\mathbb{G}, \mathbb{G}_T, e, p, g, e(g,g), H_1, H_2, l)$

where $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$,

$H_2 : \mathbb{G}_T \rightarrow \{0,1\}^l$ for some $l = l(k)$.

$\mathsf{MsgSp} = \{0,1\}^l$, $\mathsf{CtSp} = \mathbb{G} \times \{0,1\}^l$,

$\mathsf{RSp} = \mathbb{Z}_p^*$.

$\texttt{TASetup}(params)$

Set $s \xleftarrow{\$} \mathbb{Z}_p^*, h = g^s$.

Set $mpk = (params, h)$.

Set $msk = s$.

Output (mpk,msk).

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$:

Output

$usk_{id,ta} = g^{1/(msk_{ta}+H_1(id))}$.

$\texttt{Enc}(mpk_{ta}, id, m)$:

Parse

mpk_{ta} as $(params, h_{ta})$.

Set $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $u = h_{ta}^r \cdot g^{r \cdot H_1(id)}$.

Output

$c = (u, m \oplus H_2(e(g,g)^r))$.

$\texttt{Dec}^{H_2}(mpk_{ta}, usk_{id,ta}, c)$:

Parse c as

$(u, v) \in \mathbb{G} \times \{0,1\}^l$.

Set $T = e(usk_{id,ta}, u)$.

Output $m = v \oplus H_2(T)$.

The scheme m-BasicSK.

Definition 4.2. *We define the advantage of an algorithm $\mathcal{A}$ in solving the q-Bilinear Diffie-Hellman Inversion (q-BDHI) problem in $(\mathbb{G}, \mathbb{G}_T)$ to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}BDHI}(k) = \Pr[\mathcal{A}(g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}) = e(g,g)^{1/\alpha}]$$

where $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. *Here, we implicitly assume that parameters* $(\mathbb{G}, \mathbb{G}_T, e, p, g)$ *as output by* **PairingGen** *on input* 1^k *are given to* $\mathcal{A}$ *as additional inputs. We say that the q-BDHI problem is* hard *in* $(\mathbb{G}, \mathbb{G}_T)$ *if no PTA that solves the q-BDHI problem in* $(\mathbb{G}, \mathbb{G}_T)$ *has a non-negligible advantage.*

The IND-CPA security of the single-TA scheme corresponding to `m-BasicSK` can be proved by making small modifications to the proof of [39, Theorem 2], which established the OW-CPA security of a closely related scheme based on the hardness of the q-BDHI problem (Definition 4.2) in groups output by `PairingGen` (for some value q related to the number of queries made by the adversary). The IND-CPA security of the single-TA scheme corresponding to `m-BasicSK` is also implicit in the proof of security for an IND-CCA variant of the Sakai-Kasahara scheme obtained by the application of the FO transform on the IND-CPA variant of the original Sakai-Kasahara scheme [40].

Using Theorem 3.1, we can deduce that `m-BasicSK` is m-IND-CPA secure under the same assumption. It is then easy to establish that `m-BasicSK` is m-RA-CPA secure and m-TAA-CPA secure; this requires a similar analysis as in Lemma 4.2. Moreover, `m-BasicSK` is γ-uniform for $\gamma = 1/p$. We may now apply Theorem 4.1 to deduce that the scheme `FO-m-BasicSK` that is obtained by applying our modified FO transform to `m-BasicSK` is m-IND-RA-TAA-CCA secure, assuming the hardness of the q-BDHI problem in groups output by `PairingGen`.

Thus we have obtained a second, efficient multi-TA IBE scheme enjoying indistinguishability, Recipient Anonymity and TA Anonymity for the CCA setting, in the ROM. Our CCA secure scheme has roughly the same performance as the KEM-DEM-derived scheme of [39], but offers stronger proven anonymity guarantees. We also note that even the Recipient Anonymity of the single-TA version of `m-BasicSK` was not previously known – indeed this is explicitly claimed *not* to hold in [26].

Having motivated TA Anonymity and studied the TA Anonymity properties of multi-TA versions of IBE schemes in the ROM, the natural question to ask is if there are multi-TA versions of any existing IBE scheme in the Standard Model that are TA Anonymous. We investigate this in the following chapter.

Chapter 5

Multi-TA IBE in the Standard Model

5.1 Introduction

The main focus of this chapter is on studying the TA Anonymity properties of multi-TA versions of well known Standard Model IBE schemes. We show that multi-TA versions of the two Standard Model schemes of Boneh and Boyen in [19], termed $\mathtt{BB}_1$ and $\mathtt{BB}_2$ in the literature, and multi-TA versions of the schemes related to the $\mathtt{BB}_1$ scheme, such as those of Waters [117] and Naccache [89], trivially do not meet the notion of TA Anonymity. The original (single-TA) versions of these schemes are not Recipient Anonymous. At an intuitive level, this is because additional ciphertext components in these schemes contain enough information to employ pairing based checks to find which identity or which TA's public key was used to construct them. The original (single-TA) version of the Gentry scheme [62] is Recipient Anonymous and we prove that a multi-TA version of the scheme is TA Anonymous.

5.2 Anonymity of Standard Model IBE Schemes

5.2.1 Boneh-Boyen $\mathtt{BB}_1$ IBE

$\mathtt{Setup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \mathtt{PairingGen}(1^k)$.

Pick $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. Set $g_1 = g^\alpha$.

Pick $h \xleftarrow{\$} \mathbb{G}$.

Pick a generator $g_2 \in \mathbb{G}^*$.

Define function

$F : \mathbb{Z}_p^* \to \mathbb{G}$ by $F(z) = g_1^z \cdot h$.

Set $mpk =$

$(\mathbb{G}, \mathbb{G}_T, e, p, g, g_1, g_2, e(g_1, g_2), h, F)$.

Set $msk = {g_2}^\alpha$.

Output (mpk, msk).

$\mathtt{KeyDer}(mpk, msk, id)$:

Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Output

$usk_{id} = (g_2^\alpha \cdot F(id)^r, g^r)$.

$\mathtt{Enc}(mpk, id, m)$:

Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $c =$

$(F(id)^s, g^s, e(g_1, g_2)^s \cdot m)$.

$\mathtt{Dec}(mpk, usk_{id}, c)$:

Parse c as

$(u, v, w) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$.

Parse usk_{id} as

$(d_0, d_1) \in \mathbb{G} \times \mathbb{G}$.

Output $m = w \cdot \frac{e(u,d_1)}{e(v,d_0)}$.

The $\mathtt{BB}_1$ scheme .

In [19] a HIBE scheme (see Section 2.8.5) is presented which is secure based on the DBDH assumption (Definition 2.13). This scheme is popularly referred to as the $\texttt{BB}_1$ scheme in the literature and is based on what has been termed the "Commutative Blinding" principle [26]. We have presented the IBE version (essentially a 1-level HIBE scheme) here. The $\texttt{BB}_1$ scheme is selective-id IND-CPA secure (see Section 2.8.4) if the DBDH assumption (Definition 2.13) holds in $(\mathbb{G}, \mathbb{G}_T)$ [19]. We assume identities are elements in $\mathbb{Z}_p^*$ and messages are elements in $\mathbb{G}_T$.

Recipient Anonymity of $\texttt{BB}_1$

We can show that the $\texttt{BB}_1$ scheme is not Recipient Anonymous. Consider an adversary that requests the encryption of a message m to either identity id_0 or id_1. The challenge ciphertext it receives is of the form

$$c^* = (u, v, w) = (F(id_b)^s, g^s, e(g_1, g_2)^s \cdot m),$$

where b is the bit selected by the Challenger. Note that,

$$e(u, g) = e(F(id_b)^s, g) = e(F(id_b), g^s) = e(F(id_b), v).$$

Since

$$params = (\mathbb{G}, \mathbb{G}_T, e, p, g, g_1, g_2, e(g_1, g_2), h, F)$$

and u is either equal to $F(id_0)^s$ or $F(id_1)^s$, and $v = g^s$, the adversary checks if

$$e(u, g) = e(F(id_0), v) \quad \text{or} \quad e(u, g) = e(F(id_1), v)$$

to find which identity was used to create the challenge ciphertext.

5.2.2 Multi-TA $\texttt{BB}_1$

We now sketch a multi-TA version of the $\texttt{BB}_1$ IBE scheme from [19]. Again, we assume identities are elements in $\mathbb{Z}_p^*$ and messages are elements in $\mathbb{G}_T$.

$\texttt{CommonSetup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \texttt{PairingGen}(1^k)$.

Output $params = (\mathbb{G}, \mathbb{G}_T, e, p, g)$.

$\texttt{TASetup}(params)$

Pick $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. Set $g_1 = g^\alpha$.

Pick a generator $g_2 \in \mathbb{G}^*$.

Pick $h \xleftarrow{\$} \mathbb{G}$.

Define function
$F : \mathbb{Z}_p^* \to \mathbb{G}$ by $F(z) = g_1^z \cdot h$.

Set $mpk =$
$(params, g_1, g_2, e(g_1, g_2), h, F)$.

Set $msk = {g_2}^\alpha$.

Output (mpk, msk).

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$:

Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Output
$usk_{id,ta} = (g_2^\alpha \cdot F(id)^r, g^r)$.

$\texttt{Enc}(mpk_{ta}, id, m)$:

Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $c =$
$(F(id)^s, g^s, e(g_1, g_2)^s \cdot m)$.

$\texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$:

Parse c as
$(u, v, w) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$.

Parse $usk_{id,ta}$ as
$(d_0, d_1) \in \mathbb{G} \times \mathbb{G}$.

Output $m = w \cdot \frac{e(u,d_1)}{e(v,d_0)}$.

The multi-TA $\texttt{BB}_1$ scheme.

Note that h can be included as part of *params* but to keep the presentation consistent we make h specific to each TA and include it as part of mpk. In the case of multi-TA $\texttt{BB}_1$, whether h is part of *params* or TA-specific, the following argument regarding (the absence of) TA Anonymity holds.

TA Anonymity of Multi-TA $\mathtt{BB}_1$

We can show that the multi-TA $\mathtt{BB}_1$ scheme sketched above is not TA Anonymous. Consider an adversary that requests the encryption of a message m for identity id in either ta_0 or ta_1. The adversary knows that

$$mpk_{ta_0} = (params, g_1, g_2, e(g_1, g_2), h, F)$$

and

$$mpk_{ta_1} = (params, g_1', g_2', e(g_1', g_2'), h', F').$$

The challenge ciphertext it receives is either of the form

$$c^* = (u, v, w) = (F(id)^s, g^s, e(g_1, g_2)^s \cdot m)$$

if the bit chosen by the Challenger is $b = 0$ or

$$c^* = (u, v, w) = (F'(id)^s, g^s, e({g_1}', g_2')^s \cdot m)$$

if $b = 1$. Now, the adversary simply checks if

$$e(u, g) = e(F(id), v) \quad \text{or} \quad e(u, g) = e(F'(id), v)$$

to find which TA's public parameters were used to create the challenge ciphertext.

5.2.3 Schemes related to $\mathtt{BB}_1$

Water's IBE scheme [117], the scheme of Naccache [89] and the scheme of Chatterjee and Sarkar [38] are all closely related to Boneh and Boyen's $\mathtt{BB}_1$ scheme [19]. The only difference is the way the function $F(id)$ is computed. The original (single-TA) versions are not Recipient Anonymous and the multi-TA versions are not TA Anonymous for the same reasons as for the original (single-TA) $\mathtt{BB}_1$ and multi-TA $\mathtt{BB}_1$ schemes respectively.

5.2.4 Boneh-Boyen $\mathtt{BB}_2$ IBE

The second IBE scheme from [19] has been termed the $\mathtt{BB}_2$ IBE scheme in the literature and is based on what has been termed the "exponent inversion" paradigm [26]. We assume identities are elements in $\mathbb{Z}_p^*$ and messages are elements in $\mathbb{G}_T$.

$\mathtt{Setup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \mathtt{PairingGen}(1^k)$.

Pick $x, y \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $X = g^x$ and $Y = g^y$.

Define function
$F : \mathbb{Z}_p^* \rightarrow \mathbb{G}$ by $F(z) = g^z \cdot X$.

Set $mpk =$
$(\mathbb{G}, \mathbb{G}_T, e, p, g, e(g,g), X, Y, F)$.

Set $msk = (x, y)$.

Output (mpk, msk).

$\mathtt{KeyDer}(mpk, msk, id)$:

Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $K = g^{\frac{1}{id+x+ry}}$.

Output $usk_{id} = (r, K)$.

$\mathtt{Enc}(mpk, id, m)$:

Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $c =$
$(F(id)^s, Y^s, e(g,g)^s \cdot m)$.

$\mathtt{Dec}(mpk, usk_{id}, c)$:

Parse c as
$(u, v, w) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$.

Parse usk_{id} as $(r, K) \in \mathbb{Z}_p^* \times \mathbb{G}$.

Output $m = \frac{w}{e(u \cdot v^r, K)}$.

The $\mathtt{BB}_2$ scheme.

Definition 5.1. *We define the advantage of algorithm $\mathcal{A}$ in solving the q-Decisional Bilinear Diffie-Hellman Inversion (q-DBDHI) problem in $(\mathbb{G}, \mathbb{G}_T)$ to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}DBDHI}(k) = |\Pr[\mathcal{A}(g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, e(g,g)^{1/\alpha}) = 1]$$

$$- \Pr[\mathcal{A}(g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, Z) = 1]|$$

where $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and $Z \xleftarrow{\$} \mathbb{G}_T$. We implicitly assume that parameters $(\mathbb{G}, \mathbb{G}_T, e, p, g)$ as output by `PairingGen` *on input 1^k are given to $\mathcal{A}$ as additional inputs. We say that the q-DBDHI problem is* hard *in $(\mathbb{G}, \mathbb{G}_T)$ if no PTA that solves the q-DBDHI problem in $(\mathbb{G}, \mathbb{G}_T)$ has a non-negligible advantage in the security parameter k.*

The $\mathtt{BB}_2$ scheme is selective-id IND-CPA secure if the q-DBDHI (Definition 5.1) assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ [19].

Recipient Anonymity of $\mathtt{BB}_2$

We can show that the $\mathtt{BB}_2$ scheme is not Recipient Anonymous. The reasoning is similar to that for the $\mathtt{BB}_1$ scheme. Consider an adversary that requests the encryption of a message m for identity id_0 or id_1. The ciphertext it receives is of the form

$$c^* = (u, v, w) = (F(id_b)^s, Y^s, e(g,g)^s \cdot m)$$

where b is the bit selected by the Challenger. Note that

$$e(u, Y) = e(F(id_b)^s, Y) = e(F(id_b), Y^s) = e(F(id_b), v).$$

Since

$$params = (\mathbb{G}, \mathbb{G}_T, e, p, g, e(g,g), X, Y, F),$$

and u is either equal to $F(id_0)^s$ or $F(id_1)^s$, and $v = Y^s$, the adversary checks if

$$e(u, Y) = e(F(id_0), v) \quad \text{or} \quad e(u, Y) = e(F(id_1), v)$$

to find which identity was used to create the challenge ciphertext.

5.2.5 Multi-TA $\mathtt{BB}_2$

We now sketch a multi-TA version of the $\mathtt{BB}_2$ IBE scheme from [19]. Again, we assume identities are elements in $\mathbb{Z}_p^*$ and messages are elements in $\mathbb{G}_T$.

$\mathtt{CommonSetup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \mathtt{PairingGen}(1^k)$.

Output $params = (\mathbb{G}, \mathbb{G}_T, e, p, g)$.

$\mathtt{TASetup}(params)$

Pick $x, y \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $X = g^x$ and $Y = g^y$.

Define function
$F : \mathbb{Z}_p^* \to \mathbb{G}$ by $F(z) = g^z \cdot X$.

Set $mpk =$
$(params, e(g,g), X, Y, F)$.

Set $msk = (x, y)$.

Output (mpk, msk).

$\mathtt{KeyDer}(mpk_{ta}, msk_{ta}, id)$:

Pick $r \xleftarrow{\$} \mathbb{Z}_p^*$.

Set $K = g^{\frac{1}{id+x+ry}}$.

Output $usk_{id,ta} = (r, K)$.

$\mathtt{Enc}(mpk_{ta}, id, m)$:

Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $c =$
$(F(id)^s, Y^s, e(g,g)^s \cdot m)$.

$\mathtt{Dec}(mpk_{ta}, usk_{id,ta}, c)$:

Parse c as
$(u, v, w) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$.

Parse $usk_{id,ta}$ as
$(r, K) \in \mathbb{Z}_p^* \times \mathbb{G}$.

Output $m = \frac{w}{e(u \cdot v^r, K)}$.

The multi-TA $\mathtt{BB}_2$ scheme.

TA Anonymity of Multi-TA $\texttt{BB}_2$

We can show that the multi-TA $\texttt{BB}_2$ scheme sketched above is not TA Anonymous. The reasoning is similar to that for the multi-TA $\texttt{BB}_1$ scheme. Consider an adversary that requests the encryption of a message m for identity id in either ta_0 or ta_1. The adversary knows that

$$mpk_{ta_0} = (params, e(g,g), X, Y, F)$$

and

$$mpk_{ta_1} = (params, e(g,g), X', Y', F').$$

The challenge ciphertext it receives is either of the form

$$c^* = (u, v, w) = (F(id)^s, Y^s, e(g,g)^s \cdot m).$$

if the bit chosen by the Challenger is $b = 0$ or

$$c^* = (u, v, w) = (F'(id)^s, Y'^s, e(g,g)^s \cdot m)$$

if $b = 1$. Now, the adversary simply checks if

$$e(u, Y) = e(F(id), v) \quad \text{or} \quad e(u, Y') = e(F'(id), v)$$

to find which identity was used to create the challenge ciphertext.

5.2.6 The Gentry Scheme

Two IBE schemes are presented in [62]. The first scheme is proven IND-RA-CPA secure and we will prove TA Anonymity for a multi-TA version of this scheme in the following section. In the remainder of this thesis, when we refer to the Gentry scheme, we refer to this first IBE scheme. A second IBE scheme is also given in [62] and proved to be IND-RA-CCA secure. A multi-TA version of this second IBE scheme can also

be formulated. However, at present we do not know if TA Anonymity can be proved for a multi-TA version of this second IBE scheme.

$\texttt{Setup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \texttt{PairingGen}(1^k)$.

Pick $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. Set $g_1 = g^\alpha$.

Pick $h \xleftarrow{\$} \mathbb{G}$.

Define function

$F : \mathbb{Z}_p^* \to \mathbb{G}$ by $F(z) = g_1 \cdot g^{-z}$.

Set $mpk =$

$(\mathbb{G}, \mathbb{G}_T, e, p, g, e(g,g), g_1, h, e(g,h), F)$.

Set $msk = \alpha$.

Output (mpk, msk).

$\texttt{KeyDer}(mpk, msk, id)$:

Pick $r_{id} \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $d_{id} = (r_{id}, h_{id})$

where

$h_{id} = (h \cdot g^{-r_{id}})^{1/(\alpha - id)}$.

$\texttt{Enc}(mpk, id, m)$:

Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $c =$

$(F(id)^s, e(g,g)^s, e(g,h)^{-s} \cdot m)$.

$\texttt{Dec}(mpk, usk_{id}, c)$:

Parse c as

$(u, v, w) \in \mathbb{G} \times \mathbb{G}_T \times \mathbb{G}_T$.

Parse d_{id} as

$(r_{id}, h_{id}) \in \mathbb{Z}_p^* \times \mathbb{G}$.

Output

$m = w \cdot e(u, h_{id}) \cdot v^{r_{id}}$

The Gentry scheme.

The Gentry scheme [62] is based on the exponent inversion paradigm. This is similar to the Sakai-Kasahara IBE scheme [102], which has a proof of security in the ROM [102, 40], and the $\mathtt{BB}_2$ scheme which has a proof of security in the Standard Model [19]. We assume identities are elements in $\mathbb{Z}_p^*$ and messages are elements in $\mathbb{G}_T$.

We note that h need not be explicitly included in mpk. Precomputing and including the values $e(g,h)$ and $e(g,g)$ in mpk suffices. Then encryption requires no pairing computations.

Recipient Anonymity of the Gentry scheme

The Gentry scheme is IND-RA-CPA secure (Section 2.8.3) under the q-TDABDHE assumption. This is a rather involved assumption and we show how it arises. In all the following definitions in this section, we implicitly assume that parameters $(\mathbb{G}, \mathbb{G}_T, e, p, g)$ as output by `PairingGen` on input 1^k are given to the algorithms $\mathcal{A}$ as additional inputs. We start by defining the q-BDHE problem which was first introduced in [20].

Definition 5.2. *We define the advantage of an algorithm $\mathcal{A}$ in solving the q-Bilinear Diffie-Hellman Exponent (q-BDHE) problem in $(\mathbb{G}, \mathbb{G}_T)$ to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}BDHE}(k) = \Pr[\mathcal{A}(g', g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g^{(\alpha^{q+2})}, \ldots, g^{(\alpha^{2q})}) = e(g,g')^{(\alpha^{q+1})}]$$

where $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and $g' \xleftarrow{\$} \mathbb{G}$. We say that the q-BDHE problem is hard *in $(\mathbb{G}, \mathbb{G}_T)$ if no PTA that solves the q-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ has a non-negligible advantage in the security parameter k.*

Since the term $g^{(\alpha^{q+1})}$ is not given to the adversary, the bilinear map does not help to compute $e(g,g')^{(\alpha^{q+1})}$.

In [62] the q-Augmented Bilinear Diffie-Hellman Exponent (q-ABDHE) problem is introduced. Here, an additional term $g'^{(\alpha^{q+2})}$ is given as input to the adversary but this still does not facilitate the adversary, as it is not given the term $g^{(\alpha^{-1})}$.

Definition 5.3. *We define the advantage of algorithm $\mathcal{A}$ in solving the q-Augmented Bilinear Diffie-Hellman Exponent (q-ABDHE) problem in $(\mathbb{G}, \mathbb{G}_T)$ to be:*
$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}ABDHE}(k)$

$$= \Pr[\mathcal{A}(g', g'^{(\alpha^{q+2})}, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g^{(\alpha^{q+2})}, \ldots, g^{(\alpha^{2q})}) = e(g, g')^{(\alpha^{q+1})}]$$

where $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and $g' \xleftarrow{\$} \mathbb{G}$. We say that the q-ABDHE problem is hard *in $(\mathbb{G}, \mathbb{G}_T)$ if no PTA that solves the q-ABDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ has a non-negligible advantage in the security parameter k.*

The Truncated q-Augmented Bilinear Diffie-Hellman Exponent(q-TABDHE) problem is defined by removing the terms $(g^{(\alpha^{q+2})}, \ldots, g^{(\alpha^{2q})})$ from the inputs to the adversary in the q-ABDHE problem [62].

Definition 5.4. *We define the advantage of an algorithm $\mathcal{A}$ in solving the Truncated q-Augmented Bilinear Diffie-Hellman Exponent (q-TABDHE) problem in $(\mathbb{G}, \mathbb{G}_T)$ to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}TABDHE}(k) = \Pr[\mathcal{A}(g', g'^{(\alpha^{q+2})}, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}) = e(g, g')^{(\alpha^{q+1})}]$$

This can be written as

$$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}TABDHE}(k) = \Pr[\mathcal{A}(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q) = e(g_{(q+1)}, g')]$$

where $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$ and $g' \xleftarrow{\$} \mathbb{G}$ and $g_i = g^{(\alpha^i)}$ and $g'_i = g'^{(\alpha^i)}$. We say that the q-TABDHE problem is hard *in $(\mathbb{G}, \mathbb{G}_T)$ if no PTA that solves the q-TABDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ has a non-negligible advantage in the security parameter k.*

Finally we define the q-Truncated Decisional Augmented Bilinear Diffie-Hellman Exponent (q-TDABDHE) problem which is used to prove the security of the IBE schemes in [62].

Definition 5.5. *We define the advantage of an algorithm $\mathcal{A}$ in solving the q-Truncated Decisional Augmented Bilinear Diffie-Hellman Exponent (q-TDABDHE) problem in $(\mathbb{G}, \mathbb{G}_T)$ to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{q\text{-}TDABDHE}(k) = |\Pr[\mathcal{A}(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q, e(g_{(q+1)}, g')) = 1]$$

$$- \Pr[\mathcal{A}(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q, Z) = 1]|$$

where $\alpha \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, $Z \overset{\$}{\leftarrow} \mathbb{G}_T$, $g' \overset{\$}{\leftarrow} \mathbb{G}$, $g_i = g^{(\alpha^i)}$ and $g'_i = g'^{(\alpha^i)}$. We say that the q-TDABDHE problem is hard *in $(\mathbb{G}, \mathbb{G}_T)$ if no PTA that solves the q-TDABDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ has a non-negligible advantage in the security parameter k.*

For ease of presentation, the distribution of the first set of inputs to the adversary is referred to as P_{ABDHE} and the second set of inputs is referred to as R_{ABDHE}.

Result 5.1. ***[62, Theorem 1]*** *The Gentry scheme is IND-RA-CPA secure assuming the hardness of the q-TDABDHE problem in groups output by* `PairingGen`*.*

In more detail, let $\mathcal{A}$ be an IND-RA-CPA adversary against the Gentry scheme which has advantage ϵ', runs in time t' and which makes at most $(q-1)$ private key extraction queries. Then there exists an algorithm $\mathcal{B}$ that solves the q-TDABDHE problem in groups $(\mathbb{G}, \mathbb{G}_T)$ output by `PairingGen`*, running in time t and having advantage ϵ such that*

$$\epsilon' = \epsilon + (2/p) \quad \text{and} \quad t' = t - \mathcal{O}(t_{exp} \cdot q^2)$$

where t_{exp} is the time required to exponentiate in $\mathbb{G}$.

5.3 Multi-TA Gentry Scheme

We first sketch a multi-TA version of the Gentry scheme.

$\texttt{CommonSetup}(1^k)$:

$(\mathbb{G}, \mathbb{G}_T, e, p, g) \leftarrow \texttt{PairingGen}(1^k)$.

Output $params = (\mathbb{G}, \mathbb{G}_T, e, p, g)$.

$\texttt{TASetup}(params)$:

Pick $\alpha \xleftarrow{\$} \mathbb{Z}_p^*$. Set $g_1 = g^\alpha$.

Pick $h \xleftarrow{\$} \mathbb{G}$.

Define function

$F : \mathbb{Z}_p^* \to \mathbb{G}$ by $F(z) = g_1 \cdot g^{-z}$.

Set $mpk =$

$(params, e(g,g), g_1, h, e(g,h), F)$.

Set $msk = \alpha$.

Output (mpk, msk).

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$:

Pick $r_{id} \xleftarrow{\$} \mathbb{Z}_p^*$.

Output

$usk_{id,ta} = (r_{id}, h_{id})$

where

$h_{id} = (h \cdot g^{-r_{id}})^{\frac{1}{\alpha - id}}$.

$\texttt{Enc}(mpk_{ta}, id, m)$:

Pick $s \xleftarrow{\$} \mathbb{Z}_p^*$.

Output $c =$

$(F(id)^s, e(g,g)^s, e(g,h)^{-s} \cdot m)$.

$\texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$:

Parse c as

$(u, v, w) \in \mathbb{G} \times \mathbb{G}_T \times \mathbb{G}_T$.

Parse $usk_{id,ta}$ as

$(r_{id}, h_{id}) \in \mathbb{Z}_p^* \times \mathbb{G}$.

Output

$m = w \cdot e(u, h_{id}) \cdot v^{r_{id}}$.

The Multi-TA Gentry scheme.

We assume here that identities are elements in $\mathbb{Z}_p^*$ and messages are elements in $\mathbb{G}_T$. Later, we will need identities that are bit-strings of a fixed length; such identities can easily and securely be converted into elements of $\mathbb{Z}_p^*$ by applying a suitable collision-resistant hash function.

5.3.1 Anonymity of the Multi-TA Gentry Scheme

We will show that the multi-TA version of the Gentry scheme meets the r-m-IND-RA-TAA-CPA notion under the q-TDABDHE assumption. The r-m-IND-RA-TAA-CPA notion is similar to m-IND-RA-TAA-CPA security notion, except that the adversary cannot corrupt TAs.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{r-m-IND-RA-TAA-CPA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$\forall ta \in \mathcal{T}$,

$(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

$IDSet_{ta} \leftarrow \emptyset$

$(m_0, m_1, id_0, id_1, ta_0, ta_1, state) \leftarrow$

$\mathcal{A}^{\texttt{KeyDer}}(\texttt{find}, MPK)$

$c^* \leftarrow \texttt{Enc}(mpk_{ta_b}, id_b, m_b)$

$b' \leftarrow \mathcal{A}^{\texttt{KeyDer}}(\texttt{guess}, c^*, state)$

`If` $id_0 \in IDSet_{ta_0}$

`Or` $id_1 \in IDSet_{ta_1}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

The r-m-IND-RA-TAA-CPA Security Experiment.

We are able to obtain a reduction that has tightness similar to the original scheme [62] whose IND-RA-CPA security is also proven under the q-TDABDHE assumption. Our proof closely follows this latter proof with suitable modifications to reflect the multi-TA setting.

Theorem 5.1. *The multi-TA Gentry scheme is r-m-IND-RA-TAA-CPA secure assuming the hardness of the q-TDABDHE problem in groups output by* `PairingGen`*.*

In more detail, let $\mathcal{A}$ be an r-m-IND-RA-TAA-CPA adversary against the multi-TA Gentry scheme which has advantage ϵ', runs in time t' and which makes at most $(q-1)$ private key extraction queries per TA. Then there exists an algorithm $\mathcal{B}$ that solves the q-TDABDHE problem in groups $(\mathbb{G}, \mathbb{G}_T)$ output by `PairingGen`*, running in time t and having advantage ϵ such that*

$$\epsilon' = \epsilon + (4/p) \quad \text{and} \quad t' = t - \mathcal{O}(t_{exp} \cdot n \cdot q^2)$$

where t_{exp} is the time required to exponentiate in $\mathbb{G}$.

Proof. Let $\mathcal{A}$ be an adversary against the multi-TA Gentry scheme, running in time t' and making at most $(q-1)$ private key extraction queries per TA, with advantage at most ϵ'. We construct an algorithm $\mathcal{B}$ that solves the q-TDABDHE problem, as follows.

$\mathcal{B}$ takes as input a random q-TDABDHE challenge $(g', g'_{q+2}, g_1, g_2, \ldots, g_q, Z)$, where Z is either $e(g_{q+1}, g')$ or a random element of $\mathbb{G}_T$, along with expected additional inputs $(\mathbb{G}, \mathbb{G}_T, e, p, g)$. Recall that $g_i = g^{(\alpha^i)}$. Algorithm $\mathcal{B}$ proceeds as follows.

For an n TA system, $\mathcal{T} = \{ta_i : 1 \leq i \leq n\}$ represents the set of (labels of) TAs, where $n = n(k) \in \mathbb{N}$. $\mathcal{B}$ uses the input challenge to generate n related q-TDABDHE challenges, one for each TA in $\mathcal{T}$. Let CHAL_i denote the q-TDABDHE corresponding to $ta_i \in \mathcal{T}$.

For $ta_i \in \mathcal{T}$, $\mathcal{B}$ first sets $\beta_i \xleftarrow{\$} \mathbb{Z}_p^*$ and sets CHAL_i equal to:

$$(g', {g'_{q+2}}^{(\beta_i{}^{(q+2)})}, {g_1}^{(\beta_i)}, {g_2}^{(\beta_i{}^2)}, \ldots, {g_q}^{(\beta_i{}^q)}, Z^{(\beta_i{}^{(q+1)})})$$

or

$$(g', g'^{((\alpha\beta_i)^{q+2})}, g^{((\alpha\beta_i))}, g^{((\alpha\beta_i)^2)}, \ldots, g^{((\alpha\beta_i)^q)}, Z^{(\beta_i{}^{(q+1)})}).$$

We make a few important observations. Firstly, note that if $Z = e(g_{q+1}, g')$ i.e. the correct solution to the q-TABDHE problem corresponding to the original q-TDABDHE problem, then $Z^{(\beta_i{}^{(q+1)})}$ is the correct solution to the q-TABDHE problem corresponding to CHAL_i. That is, if the original input q-TDABDHE challenge is drawn from P_{ABDHE}, then so are all the CHAL_is. Similarly, if Z is random in $\mathbb{G}_T$ then so is $Z^{(\beta_i{}^{(q+1)})}$, i.e. if the original challenge is drawn from R_{ABDHE} then so are all the CHAL_is. Secondly, we note that the g' value is the same in all the n "related" challenges. This does not present a problem as g' is used only once to construct the challenge ciphertext.

- **CommonSetup**: $\mathcal{B}$ sets *params* equal to $(\mathbb{G}, \mathbb{G}_T, e, p, g)$.

- **Setup**: For each $ta_i \in \mathcal{T}$, $\mathcal{B}$ generates a random polynomial $f_i(x) \in \mathbb{Z}_p[x]$ of degree q. It sets $h_i = g^{f_i(\alpha\beta_i)}$, computing h_i from $g, {g_1}^{(\beta_i)}, {g_2}^{(\beta_i{}^2)}, \ldots, {g_q}^{(\beta_i{}^q)}$. It sets the public key for ta_i to $mpk_i = (params, {g_1}^{(\beta_i)}, h_i, e(g,g), e(g,h_i), F_i)$ where $F_i : \mathbb{Z}_p^* \to \mathbb{G}$ is such that $F_i(x) = {g_1}^{\beta_i} \cdot g^{-x}$, and sends all the n master public keys to $\mathcal{A}$. Since g, α are uniformly random and the β_i values and the polynomials $f_i(x)$ are chosen uniformly at random, the ${g_1}^{(\beta_i)}$ and h_i values are also uniformly random. Therefore the master public keys mpk_i are distributed exactly as they are when $\mathcal{A}$ interacts with a true r-m-IND-RA-TAA-CPA security experiment.

(At this stage, we have essentially succeeded in using the single q-TDABDHE challenge to set up n independent TAs.)

- **Phase 1**: $\mathcal{A}$ makes key derivation queries on (ta, id). $\mathcal{B}$ responds to a query on $ta = ta_i \in \mathcal{T}$ and $id \in \mathbb{Z}_p^*$ as follows.

  $\mathcal{B}$ checks if $g^{id} = g^{\alpha\beta_j}$ in each CHAL_j. If the equality holds, this implies that $id = \alpha\beta_j$ and $\mathcal{B}$ uses $\alpha\beta_j$ to solve the q-TDABDHE challenge immediately by computing the target response to the challenge itself.

  Otherwise, let $F_{id,i}(x)$ denote the $q-1$ degree polynomial

  $$F_{id,i}(x) = (f_i(x) - f_i(id))/(x - id).$$

  $\mathcal{B}$ sets the private key for id in ta_i to

  $$(r_{id,i}, h_{id,i}) = (f_i(id), g^{F_{id,i}(\alpha\beta_i)}).$$

  This is a valid private key since

  $$\begin{aligned} g^{F_{id,i}(\alpha\beta_i)} &= g^{(f_i(\alpha\beta_i) - f_i(id))/(\alpha\beta_i - id)} \\ &= (g^{f_i(\alpha\beta_i)} \cdot g^{-f_i(id)})^{1/(\alpha\beta_i - id)} \\ &= (h_i \cdot g^{-f_i(id)})^{1/(\alpha\beta_i - id)}. \end{aligned}$$

- **Challenge**: $\mathcal{A}$ outputs TAs ta_0, ta_1 (which correspond to $ta_x, ta_y \in \mathcal{T}$ respectively), identities id_0, id_1 and messages m_0, m_1.

  Again, as in Phase 1, $\mathcal{B}$ checks if either of g^{id_0} or g^{id_1} is equal to $g^{\alpha\beta_j}$ in each CHAL_j. If the equality holds, this implies that one of id_0 or id_1 is equal to $\alpha\beta_j$ and $\mathcal{B}$ uses $\alpha\beta_j$ to solve the q-TDABDHE challenge immediately by computing the target response to the challenge itself.

Otherwise, $\mathcal{B}$ generates a bit $b \in \{0,1\}$ and then computes a private key $d_{id_b,x} = (r_{id_b,x}, h_{id_b,x})$ for id_b in ta_x if $b = 0$ or $d_{id_b,y} = (r_{id_b,y}, h_{id_b,y})$ for id_b in ta_y if $b = 1$ as in Phase 1.

Now, let

$$g_2(x) = x^{(q+2)}$$

and

$$F_{2,id_b}(x) = (g_2(x) - g_2(id_b))/(x - id_b)$$

which is a $(q+1)$ degree polynomial. Then $F_{2,id_b}(x)$ can be written as

$$F_{2,id_b}(x) = \sum_{i=0}^{q+1} F_{2,id_b,i} \cdot x^i = x^{q+1} + \sum_{i=0}^{q} F_{2,id_b,i} \cdot x^i$$

where $F_{2,id_b,i}$ is the coefficient of x^i in $F_{2,id_b}(x)$.

$\mathcal{B}$ sets the challenge ciphertext $c^* = (u, v, w)$ as follows. In the following $\beta = \beta_x$ corresponding to ta_x if $b = 0$, or $\beta = \beta_y$ corresponding to ta_y if $b = 1$. $\mathcal{B}$ sets

$$u = g'^{(g_2(\alpha\beta) - g_2(id_b))}$$

$$v = Z^{(\beta^{(q+1)})} \cdot e(g', \prod_{i=0}^{q} g^{F_{2,id_b,i} \cdot (\alpha\beta)^i}).$$

and

$$w = m_0/(e(u, h_{id0,x}) \cdot v^{r_{id_0,x}})$$

if $b = 0$ and

$$w = m_1/(e(u, h_{id_1,y}) \cdot v^{r_{id_1,y}})$$

if $b = 1$.

To see that $c^* = (u, v, w)$ is a valid and appropriately distributed ciphertext when $Z = e(g_{q+1}, g')$, first let:

$$s = \log_g(g') \cdot F_{2,id_b}(\alpha\beta).$$

Note that s is uniformly random as $\log_g(g')$ is uniformly random and the β_i values are chosen uniformly at random. We will show that c^* is constructed using "implicit" randomness s. Now

$$g' = g^{s/(F_{2,id_b}(\alpha\beta))} = g^{(s(\alpha\beta - id_b))/(g_2(\alpha\beta) - g_2(id_b))}.$$

Therefore,

$$u = g^{s(\alpha\beta - id_b)}.$$

If Z is a random element in $\mathbb{G}_T$ then v is random in $\mathbb{G}_T$. On the other hand, if $Z = e(g_{q+1}, g')$, then $v = e(g,g)^s$ since

$$\begin{aligned}
& e(g', \textstyle\prod_{i=0}^{q} g^{F_{2,id_b,i} \cdot (\alpha\beta)^i}) \\
= \;& e(g', g^{\sum_{i=0}^{q} F_{2,id_b,i} \cdot (\alpha\beta)^i}) \\
= \;& e(g', g^{\sum_{i=0}^{q} F_{2,id_b,i} \cdot (\alpha\beta)^i + (\alpha\beta)^{q+1} - (\alpha\beta)^{q+1}}) \\
= \;& e(g', g^{\sum_{i=0}^{q+1} F_{2,id_b,i} \cdot (\alpha\beta)^i - (\alpha\beta)^{q+1}}) \\
= \;& e(g', g^{F_{2,id_b}(\alpha\beta) - (\alpha\beta)^{q+1}})
\end{aligned}$$

and therefore

$$\begin{aligned}
v &= Z^{(\beta^{(q+1)})} \cdot e(g', \textstyle\prod_{i=0}^{q} g^{F_{2,id_b,i} (\alpha\beta)^i}) \\
&= e(g_{q+1}, g')^{(\beta^{(q+1)})} \cdot e(g', g^{F_{2,id_b}(\alpha\beta) - (\alpha\beta)^{q+1}}) \\
&= e(g, g')^{(\alpha\beta)^{(q+1)}} \cdot e(g', g^{F_{2,id_b}(\alpha\beta) - (\alpha\beta)^{q+1}}) \\
&= e(g', g^{F_{2,id_b}(\alpha\beta)}) \\
&= e(g^{s/F_{2,id_b}(\alpha\beta)}, g^{F_{2,id_b}(\alpha\beta)}) \\
&= e(g,g)^s.
\end{aligned}$$

Finally, note that for any private key $d_{id_b,i} = (r_{id_b,i}, h_{id_b,i})$ corresponding to

$ta_i \in \mathcal{T}$, we have:

$$\begin{aligned}
& e(u, h_{id_b,i}) \cdot v^{r_{id_b,i}} \\
= \; & e(g^{s(\alpha\beta_i - id_b)}, [h_i \cdot g^{-f_i(id_b)}]^{1/(\alpha\beta_i - id_b)}) \cdot e(g,g)^{s \cdot f_i(id_b)} \\
= \; & e(g^{s(\alpha\beta_i - id_b)}, h_i^{1/\alpha\beta_i - id_b}) \cdot e(g^{s(\alpha\beta_i - id_b)}, g^{-f_i(id_b)/\alpha\beta_i - id_b}) \cdot e(g,g)^{s \cdot f_i(id_b)} \\
= \; & e(g, h_i)^s \cdot e(g,g)^{-s \cdot f_i(id_b)} \cdot e(g,g)^{s \cdot f_i(id_b)} \\
= \; & e(g, h_i)^s.
\end{aligned}$$

Therefore,

$$w = m_b \cdot e(g, h_x)^{-s}$$

if $b = 0$, and

$$w = m_b \cdot e(g, h_y)^{-s}$$

if $b = 1$.

- **Phase 2**: $\mathcal{A}$ continues to make key extraction queries and $\mathcal{B}$ responds as in Phase 1.

- **Guess**: Finally, the adversary outputs a guess $b' \in \{0, 1\}$. If $b = b'$ then $\mathcal{B}$ outputs 0 indicating that $Z = e(g_{q+1}, g')$; otherwise, it outputs 1.

We have already shown that the public keys and ciphertexts are appropriately distributed. We now show that the private keys issued by $\mathcal{B}$ are appropriately distributed as well. If I_i denotes the set consisting of $\alpha\beta_i, id_b$ and all the identities queried by $\mathcal{A}$ for ta_i then $|I_i| \leq (q+1)$. Then, from $\mathcal{A}$'s view the values $\{f_i(a) : a \in I\}$ are uniformly random and independent and this follows from the fact that $f_i(x)$ is a uniformly random polynomial of degree q.

Probability Analysis:

As we have already seen, if $Z = e(g_{(q+1)}, g')$, then the simulation is perfect and $\mathcal{A}$ will guess the bit b with probability $(1/2) + \epsilon'$. On the other hand, if Z is a random element in $\mathbb{G}_T$ then, u, v are uniformly random and independent elements in $\mathbb{G}, \mathbb{G}_T$ respectively. It remains to reason about w.

Now, $w = m_b/(e(u, h_{id_b,i}) \cdot v^{r_{id_b,i}})$ where i is x or y corresponding to ta_x or ta_y. The value in the denominator can be expressed as follows:

$$\begin{aligned} & e(u, h_{id_b,i}) \cdot v^{r_{id_b,i}} \\ = \; & e(u, (h_i \cdot g^{-f_i(id_b)}))^{1/(\alpha\beta_i - id_b)} \cdot v^{f_i(id_b)} \\ = \; & e(u, h_i)^{1/(\alpha\beta_i - id_b)}) \cdot e(u, g)^{-f_i(id_b)/\alpha\beta_i - id_b} \cdot v^{f_i(id_b)} \\ = \; & e(u, h_i)^{1/(\alpha\beta_i - id_b)} \cdot (v/e(u, g)^{1/(\alpha\beta_i - id_b)})^{f_i(id_b)}. \end{aligned}$$

Now $f_i(id_b)$ is independent of $\mathcal{A}$'s view (recall that for each $ta_i \in \mathcal{T}$, $\mathcal{B}$ generates a random polynomial $f_i(x) \in \mathbb{Z}_p[x]$ of degree q). Therefore as long as the inequalities $v \neq e(u, g)^{1/(\alpha\beta_x - id_0)}$, $v \neq e(u, g)^{1/(\alpha\beta_x - id_1)}$ and $v \neq e(u, g)^{1/(\alpha\beta_y - id_0)}$, $v \neq e(u, g)^{1/(\alpha\beta_y - id_1)}$ hold (and they hold with probability $(1 - 4/p)$), the value $e(u, h_{id_b,i}) \cdot v^{r_{id_b,i}}$ is random and independent of $\mathcal{A}$'s view. Consequently the value w is random and independent of $\mathcal{A}$'s view. This implies that if Z is a random element then $c^* = (u, v, w)$ can impart no information regarding the bit b.

Assuming that no queried identity equals $\alpha\beta_j$ such that $g^{\alpha\beta_j}$ is in one of the challenges CHAL_j (which would only increase $\mathcal{B}$'s success probability), we can see that:

$$\left|\Pr[\mathcal{B}(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q, e(g_{(q+1)}, g'), Z) = 1] - 1/2\right| \leq (4/p)$$

when $(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q, e(g_{(q+1)}, g'), Z)$ is sampled from R_{ABDHE}.

However,

$$\left|\Pr[\mathcal{B}(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q, e(g_{(q+1)}, g'), Z) = 1] - 1/2\right| \geq \epsilon'$$

when $(g', g'_{(q+2)}, g_1, g_2, \ldots, g_q, e(g_{(q+1)}, g'), Z)$ is sampled from P_{ABDHE}. Thus, for uniformly random g, g', α, Z we have:

$$|\Pr[\mathcal{B}(g', g'_{(l+2)}, g_1, g_2, \ldots, g_l, e(g_{(l+1)}, g') = 1]$$

$$-\Pr[\mathcal{B}(g', g'_{(l+2)}, g_1, g_2, \ldots, g_l, Z) = 1]| \geq \epsilon' - (4/p).$$

Time-Complexity:

In the simulation, $\mathcal{B}$'s overhead is dominated by computing $g^{F_{id,i}(\alpha\beta_i)}$ in response to $\mathcal{A}$'s key generation query on identity id for $ta_i \in \mathcal{T}$, where $F_{id,i}(x)$ is a polynomial of degree $(q-1)$. Each such computation requires $O(q)$ exponentiations in $\mathbb{G}$. Since $\mathcal{A}$ makes at most $(q-1)$ such queries for each TA, $t = t' + O(t_{exp} \cdot n \cdot q^2)$.

□

The above proof can be modified slightly to enable $\mathcal{B}$ to respond to `Corrupt` queries as well, thereby giving us a proof of security for the m-IND-RA-TAA-CPA security for the multi-TA version of the Gentry scheme, under the same assumptions. The m-IND-RA-TAA-CPA security notion is obtained by removing the adversary's access to the decryption oracle in the m-IND-RA-TAA-CCA security experiment (Section 3.2.6).

Theorem 5.2. *The multi-TA Gentry scheme is m-IND-RA-TAA-CPA secure assuming the hardness of the q-TDABDHE problem in groups output by* `PairingGen`*.*

In more detail, let $\mathcal{A}$ be an m-IND-RA-TAA-CPA adversary against the multi-TA Gentry scheme which has advantage ϵ', runs in time t' and which makes at most

$(q-1)$ private key extraction queries per TA. Then there exists an algorithm $\mathcal{B}$ that solves the q-TDABDHE problem in groups $(\mathbb{G}, \mathbb{G}_T)$ output by `PairingGen`*, running in time t and having advantage ϵ such that*

$$\epsilon' = \epsilon + (4/p) \cdot \binom{n}{2} \quad \text{and} \quad t' = t - O(t_{exp} \cdot n \cdot q^2)$$

where t_{exp} is the time required to exponentiate in $\mathbb{G}$.

Proof. $\mathcal{B}$ simply generates two related q-TDABDHE challenges from the original input challenge and uses these to respond to private key extraction queries for two specific TAs indexed by $ta_x, ta_y \in \mathcal{T}$. It cannot respond to `Corrupt` queries on these two TAs and the success of the proof relies on $\mathcal{A}$ choosing these two TAs in its Challenge query (thereby reducing the tightness of the reduction). For all other TAs $\{ta_i \in T : i \neq x, i \neq y\}$, $\mathcal{B}$ simply generates the master public keys and master secret keys itself and can therefore respond to private key extraction and TA corruption queries on these TAs. Further details are similar to the proof of Theorem 5.1.

□

Chapter 6

Building Key Private PKE from Multi-TA IBE

6.1 Introduction

The notion of semantic security for PKE was first defined by Goldwasser and Micali [66]. Semantic security requires that the adversary should be able to tell nothing of the underlying message by observing the ciphertext. Semantic security is shown to be equivalent to the IND-CPA security notion [66, 87] which requires that an adversary should be unable to determine to which of two chosen messages a ciphertext corresponds. In practice, security against chosen ciphertext attacks (IND-CCA security) [90, 98, 52, 12] is the preferred notion of security [106]. This requires that an adversary should be unable to determine to which of two chosen messages a challenge ciphertext corresponds, even when given access to a decryption oracle, except of course that the adversary may not use the oracle to perform decryption of the challenge ciphertext.

Building PKE schemes that are secure in a very strong sense, satisfying indistinguishability against chosen ciphertext attacks or IND-CCA secure, remains a very active area of research. Only a handful of approaches [90, 46, 33] are known for constructing IND-CCA secure PKE schemes without resorting to the ROM [14]. In

the usual public key setting, the security notion termed Key Privacy has also gained increasing importance in recent years, in the context of anonymous communications [9]. While specific schemes such as ElGamal, Cramer-Shoup and RSA based schemes are known to be Key Private [9], no generic method is known for constructing Key Private PKE schemes.

Following the results of Cocks [44], and the pairing based solutions of Sakai, Ohigishi and Kasahara [101] and Boneh and Franklin [22], Identity Based Cryptography [104] has become one of the most active areas of cryptographic research and research in the field of IBE has influenced research in the public key world. Canetti *et al.* [33] give a generic construction, now called the CHK transform, which obtains an IND-CCA secure PKE scheme from an IBE scheme that is selective-id IND-CPA secure, and a strongly secure on-time signature scheme. No mention is made in [33] of the Key Privacy of the PKE schemes arising from the CHK transform. The transform of Boneh and Katz [24] builds on the ideas of [33] to give a more efficient construction of PKE from IBE. (The results from both [33, 24] appear in [21].)

We consider the CHK transform in the setting of multiple public keys that is needed when studying Key Privacy. This quite naturally gives rise to a multi-TA IBE setting of the type considered in this thesis. We show how to modify the CHK construction to reflect this setting. We then prove that the Key Privacy of the PKE scheme resulting from our modified CHK transform follows from a weak form of TA Anonymity for the underlying multi-TA IBE scheme. Our result gives us the first generic method of constructing a PKE scheme in the Standard Model that is Key Private, as well as being IND-CCA secure. We also prove similar results for the Boneh-Katz transform [24].

We proved in Chapter 5 that a multi-TA version of the Gentry scheme [62] meets a TA Anonymity security notion suitable for application in our modified transforms. Instantiating the CHK transform with this multi-TA Gentry scheme gives us a concrete

and reasonably efficient Key Private, IND-CCA secure PKE scheme in the Standard Model, but with large ciphertexts. Instantiating the Boneh-Katz transform with the multi-TA Gentry scheme gives us a concrete and efficient Key Private, IND-CCA secure PKE scheme in the Standard Model with shorter ciphertexts.

Based on our current results, we argue that the relatively new notion of TA Anonymity is not only of interest in the area of anonymous communications, for example in thwarting traffic analysis as mentioned earlier, but also has rather subtle cryptographic implications for schemes that use IBE as a building block.

6.2 Background

We first present some of the relevant background information on Key Privacy for PKE and TA Anonymity for IBE so as to facilitate the discussions that follow. We will also formally define signature schemes, encapsulation schemes and MAC schemes and present appropriate security definitions for them.

6.2.1 Key Privacy for PKE

Key Privacy captures the security requirement that ciphertexts do not leak information about the public keys used to create them [9]. Key Privacy has surfaced as a desirable property in a number of applications [103, 30, 67, 1]. Specific schemes such as ElGamal, Cramer-Shoup and RSA based schemes are known to be Key Private [9].

Bellare *et al.* define two notions of security which they term IK-CPA and IK-CCA, that capture the notions of indistinguishability of keys, or Key Privacy, under chosen plaintext attacks and chosen ciphertext attacks, respectively [9]. We define via a security experiment, a combined security notion which simultaneously captures

both message indistinguishability and Key Privacy, which we term IND-IK-CCA security. As in the IND-CCA security experiment for a regular PKE scheme which we introduced in Section 2.9.1, the `CommonSetup` algorithm outputs a set of common parameters $\mathcal{I}$ which are used as input to the `KeyGen` algorithm.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{IND-IK-CCA-}b}(k)$

$\mathcal{I} \xleftarrow{\$} \texttt{CommonSetup}(1^k)$

$(PK_0, SK_0) \xleftarrow{\$} \texttt{KeyGen}(\mathcal{I})$

$(PK_1, SK_1) \xleftarrow{\$} \texttt{KeyGen}(\mathcal{I})$

$CSet_0 \leftarrow \emptyset,\ CSet_1 \leftarrow \emptyset$

$(m_0, m_1, state) \leftarrow \mathcal{A}^{\texttt{Dec}}(\texttt{find}, PK_0, PK_1)$

$c^* \leftarrow \texttt{Enc}(PK_b, m_b)$

$CSet_0 \leftarrow \emptyset,\ CSet_1 \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $c^* \in CSet_0$

`Or` $c^* \in CSet_1$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{Dec}(PK_b, c)$

$CSet_b \leftarrow CSet_b \cup \{c\}$

$m \leftarrow \texttt{Dec}(SK_b, c)$

`Return` m

The IND-IK-CCA Security Experiment.

The advantage of an adversary $\mathcal{A}$ in the IND-IK-CCA security experiment is defined as

$$\begin{aligned} & \mathbf{Adv}_{\mathcal{A}}^{\text{IND-IK-CCA}}(k) \\ = \ & |\Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{IND-IK-CCA-1}}(k) = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{IND-IK-CCA-0}}(k) = 1]|. \end{aligned}$$

We say that a PKE scheme is IND-IK-CCA secure if the advantage of all PPT adversaries is negligible as a function of the security parameter k.

6.2.2 A TA Anonymity Security Notion for Multi-TA IBE

We first define the selective-id r-m-IND-TAA-CPA security notion (abbreviated to s-id r-m-IND-TAA-CPA) for multi-TA IBE, this being a weakened version of the m-IND-RA-TAA-CCA notion defined in Section 3.2.6. A single identity is used in the challenge phase in this model, i.e. we set $id_0 = id_1$. Furthermore, the adversary commits to this identity at the start of the game. The adversary is not allowed to make decryption or TA corruption queries. Again, we define this security notion via an experiment, which we describe next.

Experiment $\mathbf{Exp}^{\text{s-id r-m-IND-TAA-CPA-}b}_{\mathcal{A}}(k)$

$id^* \leftarrow \mathcal{A}(1^k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$\forall ta \in \mathcal{T}$,

  $(mpk_{ta}, msk_{ta}) \leftarrow \texttt{TASetup}(params)$

  $IDSet_{ta} \leftarrow \emptyset$

$(m_0, m_1, ta_0, ta_1, state)$

  $\leftarrow \mathcal{A}^{\texttt{KeyDer}}(\texttt{find}, MPK)$

$c^* \leftarrow \texttt{Enc}(mpk_{ta_b}, id^*, m_b)$

$b' \leftarrow \mathcal{A}^{\texttt{KeyDer}}(\texttt{guess}, c^*, state)$

`If` $ta_0 \in TASet$ `Or` $ta_1 \in TASet$

`Or` $id^* \in IDSet_{ta_0}$ `Or` $id^* \in IDSet_{ta_1}$

`Then Return` 0

`Else Return` b'

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta}$

  $\leftarrow \texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

The selective-id r-m-IND-TAA-CPA Security Experiment.

The advantage of an adversary $\mathcal{A}$ in the s-id r-m-IND-TAA-CPA security game is defined in the usual fashion.

6.2.3 Signature Schemes

Definition 6.1. *A signature scheme is defined by a triple of algorithms:*

- `Gen`: A randomized key generation algorithm that takes as input a security parameter 1^k and outputs a signing key sk and a matching verification key vk.
- `Sgn`: A signing algorithm that takes as input a signing key sk and a message m from some message space MsgSp and outputs a signature σ. We write $\sigma = \texttt{Sgn}(sk, m)$. This algorithm may be randomized or deterministic.
- `Vrfy`: A verification algorithm that takes as input a verification key vk, a message m and a signature σ and outputs a bit $b \in \{0, 1\}$ (where $b = 1$ signifies acceptance and $b = 0$ signifies rejection). We write $b = \texttt{Vrfy}(vk, m, \sigma)$.

We require that for all (sk, vk) output by `Gen`, $\forall m \in \mathsf{MsgSp}$ and all σ output by $\texttt{Sgn}(sk, m)$, we have $\texttt{Vrfy}(vk, m, \sigma) = 1$.

Security of Signature Schemes

We formally define strong one-time security for signature schemes via the following experiment.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{strong-OT-Sig}}(k)$
$(sk, vk) \leftarrow \texttt{Gen}(1^k)$
$(m, state) \leftarrow \mathcal{A}(vk)$
$\sigma \leftarrow \texttt{Sgn}(sk, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}(vk, \sigma, state)$
`If` $(m^*, \sigma^*) \neq (m, \sigma)$
`And` $\texttt{Vrfy}(vk, m^*, \sigma^*) = 1$
`Then Return` 1
`Else Return` 0

We note that the adversary may output (m^*, σ^*) without outputting m, in which case (m, σ) are undefined and the win condition is simply if $\texttt{Vrfy}(vk, m^*, \sigma^*) = 1$. We also note that the adversary can succeed even if $m^* = m$, but then only if $\sigma^* \neq \sigma$.

A signature scheme is a strongly secure one-time signature scheme if an adversary in the following strong-OT-Sig security experiment has success probability that is negligible in the security parameter k.

6.2.4 Encapsulation Schemes

Definition 6.2. *An encapsulation scheme is defined by a triple of PPT algorithms:*

- `Init`: Takes as input a security parameter 1^k and outputs a string pub.
- `Encap`: Takes as input 1^k and pub and outputs (r, com, dec) where $r \in \{0,1\}^k$. Here, com is called the commitment string and dec the decommitment string. We assume that $|com| = |dec|$ for a given value of the security parameter.
- `Decap`: Takes as input (pub, com, dec) and outputs $r \in \{0,1\}^k$ or $\perp$.

We require that for all pub output by `Init`, and for all (r, com, dec) output by $\texttt{Encap}(1^k, pub)$, $\texttt{Decap}(pub, com, dec) = r$.

Security of Encapsulation Schemes

An encapsulation scheme is secure if it has both the hiding and binding properties as defined below.

Hiding

This security property implies that it should be hard for an adversary against the encapsulation scheme that is given *pub* and *com* (but not *dec*), to tell whether or not a string $r \in \{0,1\}^k$ is from a valid tuple (r, com, dec). It can formally be defined by the following security experiment.

$$
\begin{aligned}
&\text{Experiment } \mathbf{Exp}_{\mathcal{A}}^{\text{Hiding}}(k) \\
&pub \leftarrow \texttt{Init}(1^k) \\
&r_0 \leftarrow \{0,1\}^k \\
&(r_1, com, dec) \leftarrow \texttt{Encap}(1^k, pub) \\
&b \xleftarrow{\$} \{0,1\} \\
&b' \leftarrow \mathcal{A}(1^k, pub, com, r_b)
\end{aligned}
$$

We define the advantage of an adversary in the above Hiding Experiment to be

$$\mathbf{Adv}_{\mathcal{A}}^{\text{Hiding}}(k) = \left| \Pr[b' = b] - \frac{1}{2} \right|.$$

The encapsulation scheme is said to be hiding if the advantage of all PPT adversaries $\mathcal{A}$ in the Hiding Experiment above, is negligible in the security parameter k.

Binding

This security property implies that it should be hard for an adversary that is given a valid tuple (pub, com, dec) with $r = \texttt{Decap}(pub, com, dec)$, to produce a second

valid decommitment dec', such that $\texttt{Decap}(pub, com, dec')$ does not fail, and outputs a string $r' \neq r$. More formally, we have the following security experiment.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{Binding}}(k)$
$pub \leftarrow \texttt{Init}(1^k)$
$(r, com, dec) \leftarrow \texttt{Encap}(1^k, pub)$
$dec' \leftarrow \mathcal{A}(1^k, pub, com, dec)$
`If` $\texttt{Decap}(pub, com, dec') \notin \{\perp, r\}$
`Then Return` 1
`Else Return` 0.

We define the advantage of an adversary in the Binding experiment as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{Binding}}(k) = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\text{Binding}}(k) = 1].$$

The encapsulation scheme is said to be binding if the advantage of all PPT adversaries $\mathcal{A}$ in the Binding Experiment above, is negligible in the security parameter k.

6.2.5 MAC Schemes

Definition 6.3. *A Message Authentication Code (MAC) is defined by a pair of algorithms:*

- `MAC`: A tagging algorithm that takes as input a key $sk \in \{0,1\}^k$ (where k is the security parameter) and a message m in an appropriate message space, and outputs a string tag. We write $tag = \texttt{MAC}(sk, m)$. This algorithm may or may not be randomized.

- `Vrfy`: A verification algorithm that takes as input a key sk, a message m and a string tag and outputs a bit $b \in \{0,1\}$ where $b = 1$ signifies an accept and $b = 0$ signifies a reject. We write this as $\texttt{Vrfy}(sk, m, tag) = b$.

We require that for all sk, all m in the message space and all strings tag output by $\texttt{MAC}(sk, m)$, we have $\texttt{Vrfy}(sk, m, tag) = 1$.

Security of MAC schemes

We formally define strong one-time security for MAC schemes via the following experiment.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{strong-OT-MAC}}(k)$
$sk \xleftarrow{\$} \{0,1\}^k$
$(m, state) \leftarrow \mathcal{A}(1^k)$
$tag \leftarrow \texttt{MAC}(sk, m)$
$(m^*, tag^*) \leftarrow \mathcal{A}(tag, state)$
`If` $(m^*, tag^*) \neq (m, tag)$
`And` $\texttt{Vrfy}(sk, m^*, tag^*) = 1$
`Then Return` 1
`Else Return` 0

We note that the adversary may output (m^*, tag^*) before outputting m, in which case (m, tag) are undefined, and the win condition is simply that $\texttt{Vrfy}(sk, m^*, tag^*) = 1$. We also note that the adversary can succeed even if $m^* = m$.

A MAC scheme $(\texttt{MAC}, \texttt{Vrfy})$ is a strong one-time MAC if the success probability of any PPT adversary in the strong-OT-MAC security experiment is negligible in the security parameter k.

6.3 Key Privacy of the CHK Transform

Canetti *et al.* [33] give a construction that builds an IND-CCA secure PKE scheme from a selective-id IND-CPA secure IBE scheme and a strongly secure one-time signature scheme. We first give an informal description of this construction. The public key of the PKE scheme is the master public key of the underlying IBE scheme and the secret key is the corresponding master secret key. To encrypt a message, the sender first generates a key pair for the underlying signature scheme. The sender uses the encryption algorithm of the underlying IBE scheme to encrypt the message, using the public parameters obtained from the public key of the recipient, using the verification key as the identity. The resulting IBE ciphertext is signed using the signing key corresponding to the verification key. The final ciphertext consists of three components, the verification key, the IBE ciphertext and the signature. To decrypt a ciphertext, the receiver first verifies that the signature on the IBE ciphertext with respect to the verification key is valid. If so, the receiver derives the secret key corresponding to the verification key (it can do this as it knows the master secret key of the IBE scheme) and decrypts the IBE ciphertext using the decryption algorithm of the underlying IBE scheme.

The original construction is presented for the case of a single user who generates the master public key and master secret key of the underlying IBE scheme. We will formally modify the original CHK construction [33] to reflect the setting of multiple users. In our modified construction, we have multiple users who share common parameters as part of their public keys i.e. as part of the master public keys of the underlying IBE scheme. They will of course generate the master public keys and master secret keys independently.

Finally, we will show that the IND-IK-CCA security of the PKE scheme built using the (modified) CHK transform follows from the selective-id r-m-IND-TAA-CPA

security of the underlying IBE scheme and the one-time security of the underlying signature scheme. We note that we do not require Recipient Anonymity of the IBE scheme to obtain our result. Rather, the security property needed from the IBE scheme is the form of TA Anonymity which is captured in our selective-id r-m-IND-TAA-CPA security notion. In Section 5.3.1 we proved that a multi-TA version of the Gentry scheme meets the stronger m-IND-RA-TAA-CPA security notion. This is sufficient for the application of our result. Instantiating the CHK transform with the multi-TA Gentry scheme and any strongly secure one-time signature scheme will give us a concrete construction of a Key Private and IND-CCA secure PKE scheme. We discuss this further in Section 6.3.1.

6.3.1 The Modified CHK Transform

Let $\Pi' = \{\texttt{CommonSetup}', \texttt{TASetup}', \texttt{KeyDer}', \texttt{Enc}', \texttt{Dec}'\}$ be a multi-TA IBE scheme for identities of length l.

Let $\{\texttt{Gen}, \texttt{Sgn}, \texttt{Vrfy}\}$ be a signature scheme in which the verification keys output by `Gen` have length l.

Define $\Pi = \{\texttt{CommonSetup}, \texttt{KeyGen}, \texttt{Enc}, \texttt{Dec}\}$ as follows

- `CommonSetup`: On input 1^k, this algorithm runs $\texttt{CommonSetup}'$ on input 1^k, to obtain *params*. It sets $\mathcal{I} = params$.

- `KeyGen`: On input $\mathcal{I} = params$, this algorithm runs $\texttt{TASetup}'$ on input *params* to obtain mpk, msk. The public key PK is set to mpk (PK includes *params*, as mpk by definition includes *params*) and the secret key is $SK = msk$.

- `Enc`: To encrypt a message m using public key PK, the sender first runs `Gen` to obtain a verification key vk and the corresponding signing key sk (with $|vk| = n$). Then, the sender computes $c = \texttt{Enc}'(mpk, vk, m)$ (i.e. the sender

encrypts the message m with respect to identity vk under TA, whose mpk is obtained from PK) and $\sigma = \texttt{Sgn}(sk, c)$. The final ciphertext is (vk, c, σ).

- `Dec`: To decrypt (vk, c, σ) using the secret key $SK = msk$, corresponding to the public key $PK = mpk$, the recipient first checks whether $\texttt{Vrfy}(vk, c, \sigma) \stackrel{?}{=} 1$. If not, the receiver outputs $\perp$. Otherwise, the receiver computes $usk_{vk} = \texttt{KeyDer}'(msk, vk)$ and outputs $m = \texttt{Dec}'(mpk, usk_{vk}, c)$.

Theorem 6.1. *If Π' is an IBE scheme which is selective-id r-m-IND-TAA-CPA secure and $\{\texttt{Gen}, \texttt{Sgn}, \texttt{Vrfy}\}$ is a strongly secure one-time signature scheme, then Π is an IND-IK-CCA secure PKE scheme.*

Proof. Our proof follows closely the proof of [33] with suitable modifications to reflect the setting of multiple users. Let $\mathcal{A}$ be an IND-IK-CCA adversary against Π. We say a ciphertext (vk, c, σ) is valid if $\texttt{Vrfy}(vk, c, \sigma) = 1$. Let (vk^*, c^*, σ^*) denote the challenge ciphertext received by $\mathcal{A}$ during a particular run of the IND-IK-CCA experiment and let `Forge` denote the event that $\mathcal{A}$ submits a valid ciphertext (vk^*, c, σ) to its decryption oracle in this experiment.

In the following, expressions of the form $\Pr_{A,S}[\texttt{Event}]$ denote the probability that an `Event` occurs when an adversary $\mathcal{A}$ interacts with a scheme S in a specified security experiment.

Claim 1 : $\Pr_{A,\Pi}[\texttt{Forge}]$ is negligible.

Proof of Claim 1

$\mathcal{A}$ is an IND-IK-CCA adversary against the PKE scheme Π. We use $\mathcal{A}$ to construct an adversary $\mathcal{F}$ that forges a signature with respect to the one-time signature scheme, with probability $\Pr_{A,\Pi}[\texttt{Forge}]$.

$\mathcal{F}$ is given a verification key vk^*. $\mathcal{F}$ first runs $\texttt{KeyGen}$ to obtain (PK_0, SK_0) and (PK_1, SK_1). It gives $\mathcal{A}$ the two public keys PK_0 and PK_1. Note that $\mathcal{F}$ can answer any decryption queries of $\mathcal{A}$.

If $\mathcal{A}$ happens to submit a valid ciphertext (vk^*, c, σ) to its decryption oracle before requesting the challenge ciphertext then $\mathcal{F}$ simply outputs the forgery (c, σ) and stops. Otherwise, when $\mathcal{A}$ outputs messages m_0 and m_1, $\mathcal{F}$ chooses a random bit b and computes $c^* = \texttt{Enc}'(mpk_b, vk^*, m_b)$ and obtains from its signing oracle a signature σ^* on the message c^*, i.e. $\sigma^* = \texttt{Sgn}(sk^*, c^*)$ where sk^* is the signing key corresponding to vk^*. $\mathcal{F}$ gives $\mathcal{A}$ the challenge ciphertext (vk^*, c^*, σ^*). Subsequently, if $\mathcal{A}$ submits a valid ciphertext (vk^*, c, σ) to its decryption oracle (note that we must have $(c, \sigma) \neq (c^*, \sigma^*)$), $\mathcal{F}$ simply outputs (c, σ) as its forgery. It is easy to see that $\mathcal{F}$'s success probability is exactly $\Pr_{\mathcal{A},\Pi}[\texttt{Forge}]$.

Claim 2 : $|\Pr_{\mathcal{A},\Pi}[\texttt{Succ} \wedge \overline{\texttt{Forge}}] + \frac{1}{2}\Pr_{\mathcal{A},\Pi}[\texttt{Forge}] - \frac{1}{2}|$ is negligible.

Proof of Claim 2

We now use $\mathcal{A}$ to construct a selective-id r-m-IND-TAA-CPA attacker $\mathcal{B}$ against the IBE scheme Π'. Adversary $\mathcal{B}$ acts as a Challenger for $\mathcal{A}$ as follows.

$\mathcal{B}$ runs $\texttt{Gen}(1^k)$ to obtain (sk^*, vk^*) and outputs a target identity $id^* = vk^*$ to its Challenger $\mathcal{C}$. The challenger $\mathcal{C}$ gives $\mathcal{B}$ MPK, the set of all master public keys in the multi-TA IBE scheme. Adversary $\mathcal{B}$ gives $\mathcal{A}$ the two public keys $PK_0 = mpk_{ta_0}$ and $PK_1 = mpk_{ta_1}$.

$\mathcal{A}$ is an IND-IK-CCA attacker against the public key scheme. When $\mathcal{A}$ makes decryption queries on ciphertexts of the form (vk, c, σ), it specifies whether it wants the decryption corresponding to PK_0 or PK_1. $\mathcal{B}$ answers decryption queries as follows.

- If $vk = vk^*$, then $\mathcal{B}$ checks whether $\texttt{Vrfy}(vk^*, c, \sigma) = 1$. In this case, $\mathcal{B}$ does not know the corresponding IBE secret key corresponding to the identity vk^* and it is not allowed to make this query to its Challenger $\mathcal{C}$. Consequently, $\mathcal{B}$ aborts and outputs a random bit. If $\texttt{Vrfy}(vk^*, c, \sigma) \neq 1$ then $\mathcal{B}$ responds with $\perp$.
- If $vk \neq vk^*$ and $\texttt{Vrfy}(vk, c, \sigma) \neq 1$ then $\mathcal{B}$ responds with $\perp$.
- If $vk \neq vk^*$ and $\texttt{Vrfy}(vk, c, \sigma) = 1$ then $\mathcal{B}$:
  - makes the oracle query $\texttt{KeyDer}(mpk_{ta_i}, vk)$ where PK_i is specified in the query and obtains usk_{vk,ta_i},
  - computes $m = \texttt{Dec}'(mpk_{ta_i}, usk_{vk,ta_i}, c)$, and
  - responds with m.

At some point during the simulation $\mathcal{A}$ outputs two equal length messages m_0 and m_1. $\mathcal{B}$ forwards (ta_0, m_0) and (ta_1, m_1) to its Challenger. $\mathcal{B}$ is given the challenge ciphertext $c^* = \texttt{Enc}'(mpk_{ta_b}, id^*, m_b)$. $\mathcal{B}$ computes $\sigma^* = \texttt{Sgn}(sk^*, c^*)$ and gives $\mathcal{A}$ (vk^*, c^*, σ^*).

$\mathcal{A}$ continues to make decryption oracle queries which are answered by $\mathcal{B}$ as before. Finally $\mathcal{A}$ outputs a guess b' and this same guess is output by $\mathcal{B}$. We note that $\mathcal{B}$ provides a perfect simulation for $\mathcal{A}$ as well as a legal strategy for attacking the IBE scheme, provided that $\mathcal{B}$ is not forced to abort (which occurs only when the event `Forge` occurs). In particular it never requests the secret key corresponding to the target identity vk^* for either of the target TAs.

We note that $\mathcal{B}$ wins if $\mathcal{A}$ does, and this can only happen when the event `Forge` does not happen. When the event `Forge` happens, $\mathcal{B}$ is forced to abort and output a

random bit. Therefore we have

$$|\Pr_{\mathcal{B},\Pi'}[\mathtt{Succ}] - \tfrac{1}{2}| \;=\; |\Pr_{\mathcal{A},\Pi}[\mathtt{Succ} \wedge \overline{\mathtt{Forge}}] + \tfrac{1}{2} \cdot \Pr_{\mathcal{A},\Pi}[\mathtt{Forge}] - \tfrac{1}{2}|.$$

Claim 2 then follows as we know the left hand side of the above equation is negligible by the assumed security of the IBE scheme.

Finally, we have

$$\begin{aligned}
& |\Pr_{\mathcal{A},\Pi}[\mathtt{Succ}] - \tfrac{1}{2}| \\
= \;& |\Pr_{\mathcal{A},\Pi}[\mathtt{Succ} \wedge \mathtt{Forge}] + \Pr_{\mathcal{A},\Pi}[\mathtt{Succ} \wedge \overline{\mathtt{Forge}}] \\
& -\tfrac{1}{2} \cdot \Pr_{A,\Pi}[\mathtt{Forge}] + \tfrac{1}{2} \Pr_{\mathcal{A},\Pi}[\mathtt{Forge}] - \tfrac{1}{2}| \\
\leq \;& |\Pr_{\mathcal{A},\Pi}[\mathtt{Succ} \wedge \mathtt{Forge}] - \tfrac{1}{2} \cdot \Pr_{A,\Pi}[\mathtt{Forge}]| \\
& +|\Pr_{\mathcal{A},\Pi}[\mathtt{Succ} \wedge \overline{\mathtt{Forge}}] + \tfrac{1}{2} \Pr_{\mathcal{A},\Pi}[\mathtt{Forge}] - \tfrac{1}{2}| \\
\leq \;& \tfrac{1}{2} \cdot \Pr_{\mathcal{A},\Pi}[\mathtt{Forge}] + |\Pr_{\mathcal{A},\Pi}[\mathtt{Succ} \wedge \overline{\mathtt{Forge}}] + \tfrac{1}{2} \Pr_{\mathcal{A},\Pi}[\mathtt{Forge}] - \tfrac{1}{2}|.
\end{aligned}$$

The proof of the result follows from the proofs of claims 1 and 2.

□

Final Observations

We showed in Theorems 5.1 and 5.2 that the multi-TA version of the Gentry scheme meets stronger notions of security than those required for the application of Theorem 6.1. We can therefore instantiate the modified CHK transform with the multi-TA version of the Gentry scheme and a strongly secure one-time signature scheme [83, 54] to obtain an IND-IK-CCA PKE scheme. The known one-time signature schemes, while allowing for very efficient key generation and verification, have very long public keys and signatures. So while the scheme is quite efficient, with encryption and decryption costing roughly the same as encryption and decryption in the original

Gentry scheme, the verification key and the signature that have to be included as ciphertext components add significant overhead resulting in long ciphertexts.

6.4 Key Privacy of the Boneh-Katz Transform

Boneh and Katz [24] give a construction for obtaining an IND-CCA secure PKE scheme with better efficiency than the construction of Canetti *et al.* [33]. The basic ideas are similar, however, instead of using a signature scheme as in [33], the construction of [24] uses an encapsulation scheme and a MAC scheme, in addition to the selective-id IND-CPA secure IBE scheme.

In this section we present a modified version of the Boneh-Katz transform from [24] to reflect the setting of multiple users and show that the IND-IK-CCA security of the PKE scheme built using the (modified) BK transform follows from the selective-id r-m-IND-TAA-CPA security of the underlying IBE scheme.

6.4.1 The Modified Boneh-Katz Transform

Let $\Pi' = \{\texttt{CommonSetup}', \texttt{TASetup}', \texttt{KeyDer}', \texttt{Enc}', \texttt{Dec}'\}$ be a multi-TA IBE scheme supporting identities of length l'.

Let $\{\texttt{Init}, \texttt{Encap}, \texttt{Decap}\}$ be an encapsulation scheme where commitments *com* output by `Encap` have length l'.

Let $\{\texttt{MAC}, \texttt{Vrfy}\}$ be a MAC scheme.

We construct a PKE scheme $\Pi = \{\texttt{CommonSetup}, \texttt{KeyGen}, \texttt{Enc}, \texttt{Dec}\}$ with algorithms defined below.

The message space for the PKE scheme will be bit strings of length l (say), such that the message space of the multi-TA IBE scheme is the set of bit strings of length $(l + l')$. In practice, the message space of the underlying IBE scheme may be elements

of a specific group rather than bit strings and so the key mask used to perform the encryption may need to be suitably converted to bit strings. (See Section 7.3 and Section 7.4 in [21] for details.)

- `CommonSetup`: On input 1^k, this algorithm runs $\texttt{CommonSetup}'$ on input 1^k, to obtain *params*. It runs $\texttt{Init}(1^k)$ to generate *pub*. It sets $\mathcal{I} = (params, pub)$.
- `KeyGen`: On input $\mathcal{I}$, this algorithm runs $\texttt{TASetup}'$ on input *params* to generate *mpk* and *msk*. The public key *PK* is set to be (mpk, pub) (*PK* includes *params*, as *mpk* by definition includes *params*). The secret key *SK* is set to be (msk, pub).
- `Enc`: To encrypt a message m using public key $PK = (mpk, pub)$, the sender first runs $\texttt{Encap}(1^k, pub)$ to obtain (r, com, dec). Then, the sender computes $c = \texttt{Enc}'(mpk, com, (m||dec))$. Finally, the sender computes $tag = \texttt{MAC}(r, c)$. The final ciphertext is (com, c, tag).
- `Dec`: To decrypt a ciphertext (com, c, tag) using secret key $SK = (msk, pub)$, the receiver computes $usk_{com} = \texttt{KeyDer}'(msk, com)$ followed by $(m||dec) = \texttt{Dec}'(mpk, usk_{com}, c)$. Next the recipient runs $\texttt{Decap}(pub, com, dec)$ to obtain a string r. If this does not fail and $\texttt{Vrfy}(r, c, tag) = 1$, then the recipient outputs m. Otherwise, the algorithm outputs $\perp$.

Theorem 6.2. *If* Π' *is a selective-id r-m-IND-TAA-CPA secure IBE scheme, the encapsulation scheme is both hiding and binding and the MAC scheme is a strong one-time MAC, then* Π *is an IND-IK-CCA secure PKE scheme.*

Proof. Let $\mathcal{A}$ be an IND-IK-CCA adversary against the scheme Π. We say that a ciphertext (com, c, tag) is valid if decryption of the ciphertext does not result in $\perp$.

Let (com^*, c^*, tag^*) denote the challenge ciphertext received by $\mathcal{A}$. The proof is based on showing that:

- $\mathcal{A}$ submits to its decryption oracle a valid ciphertext (com^*, c, tag), with $(c, tag) \neq (c^*, tag^*)$, only with negligible probability. Proving this is complicated by the fact that ciphertext validity cannot be checked publicly (as in the CHK construction) without knowledge of the master secret keys.
- Assuming that the previous event does not occur, the decryption queries made by $\mathcal{A}$ do not help it learn the underlying plaintext. This is proved based on the security of the underlying IBE scheme Π'.

The proof is structured as a sequence of games. Let $\Pr_i[\cdot]$ denote the probability of a particular event occurring in Game i.

Game 0:

This is the original game in which $\mathcal{A}$ is an IND-IK-CCA attacker against scheme Π. Let (r^*, com^*, dec^*) denote the values that are used to compute the challenge ciphertext. These values can be assumed to have been generated at the start of the game as they are generated independently of $\mathcal{A}$'s actions. Let $\texttt{Succ}$ denote the event that $\mathcal{A}$'s output bit b' is identical to the bit b used in constructing the challenge ciphertext. We have, $\mathbf{Adv}^{\text{IND-CCA}}_{\mathcal{A},\Pi}(k) = |\Pr_0[\texttt{Succ}] - 1/2|$.

Game 1:

This game is slightly modified as follows. On input a ciphertext (com^*, c, tag), the decryption oracle simply outputs $\perp$.

Let $\texttt{Valid}$ denote the event that $\mathcal{A}$ submits a valid ciphertext (com^*, c, tag) to its decryption oracle. We note that $|\Pr_1[\texttt{Succ}] - \Pr_0[\texttt{Succ}]| \leq \Pr_0[\texttt{Valid}] = \Pr_1[\texttt{Valid}]$, which holds since Game 0 and Game 1 are identical until the event $\texttt{Valid}$ occurs. Let $\texttt{NoBind}$ denote the event that $\mathcal{A}$ at some point submits a ciphertext (com^*, c, tag) such that:

- c decrypts to $(m||dec)$ using usk_{com^*} derived from the secret key as specified in the query, and
- $\texttt{Decap}(pub, com^*, dec) \notin \{r^*, \perp\}$.

Let $\texttt{Forge}$ denote the event that $\mathcal{A}$ submits a ciphertext (com^*, c, tag) to its decryption oracle such that $\texttt{Vrfy}(r^*, c, tag) = 1$.

Then,

$$\begin{aligned} |\Pr_1[\texttt{Succ}] - \Pr_0[\texttt{Succ}]| &\leq \Pr_0[\texttt{Valid}] \\ &= \Pr_1[\texttt{Valid}] \\ &\leq \Pr_1[\texttt{NoBind}] + \Pr_1[\texttt{Forge}]. \end{aligned}$$

Claim 1A: $\Pr_1[\texttt{NoBind}]$ is negligible.

Proof of Claim 1A: This follows from the Binding property of the encapsulation scheme. Consider an adversary $\mathcal{B}$ against the Binding property of the encapsulation scheme that is given input $(1^k, pub, com^*, dec^*)$.

- $\mathcal{B}$ runs $\texttt{CommonSetup}'(1^k)$ to obtain $params$. It then runs $\texttt{TASetup}(params)$ twice, to obtain (mpk_{ta_0}, msk_{ta_0}) and (mpk_{ta_1}, msk_{ta_1}) and then runs $\mathcal{A}$ on inputs $PK_0 = (mpk_{ta_0}, pub)$ and $PK_1 = (mpk_{ta_1}, pub)$.
- When $\mathcal{A}$ makes a decryption query $\mathcal{B}$ responds as required in Game 1, i.e. with

$\perp$ to decryption queries of the form (com^*, c, tag) and to other queries using its knowledge of msk_{ta_1} and msk_{ta_2}.

- When $\mathcal{A}$ submits two messages m_0 and m_1, $\mathcal{B}$ chooses a bit b at random and encrypts m_b with PK_b to produce a challenge ciphertext (com^*, c^*, tag^*). Note that $\mathcal{B}$ has (pub, com^*, dec^*) and can use this to calculate r^* and consequently tag^* on the ciphertext component c^*.

At the end of the experiment, $\mathcal{B}$ decrypts queries of the form (com^*, c, tag) to see if the event `NoBind` occurred. If so, it can learn a value dec such that $\texttt{Decap}(pub, com^*, dec) \notin \{r^*, \perp\}$, violating the Binding property of the encapsulation scheme. Therefore, we conclude that $\Pr[\texttt{NoBind}]$ must be negligible.

Game 2:

In Game 2 the way the challenge ciphertext is computed is modified. $\mathcal{B}$ generates an additional public key PK_2 by running `TASetup` on input *params* to obtain (mpk_{ta_2}, msk_{ta_2}) and setting $PK_2 = (mpk_{ta_2}, pub)$.

When $\mathcal{A}$ submits its two messages m_0 and m_1, the ciphertext is computed as $c^* = \texttt{Enc}'(mpk_{ta_2}, com^*, (0^{l+l'}))$, followed by $tag^* = \texttt{MAC}(r^*, c^*)$. The challenge ciphertext is (com^*, c^*, tag^*). The bit b is only used to define the event `Succ` and since the challenge ciphertext is independent of the bit b, it follows that $\Pr_2[\texttt{Succ}] = 1/2$. i.e. $\Pr_2[\texttt{Succ}] - 1/2 = 0$.

Claim 2A: $|\Pr_2[\texttt{Succ}] - \Pr_1[\texttt{Succ}]|$ is negligible.

Proof of Claim 2A: This follows from the security of Π'. Consider an adversary $\mathcal{A}'$ attacking the IBE scheme Π' in a selective-id r-m-IND-TAA-CPA game.

- Adversary $\mathcal{A}'$ first runs $\texttt{Init}(1^k)$ to obtain pub. It then runs $\texttt{Encap}(1^k, pub)$ to obtain (r^*, com^*, dec^*). It outputs com^* as the target identity and is then given $MPK = \{mpk_{ta} : ta \in \mathcal{T}\}$ for $\mathcal{T} = \{ta_i : 0 \leq i \leq n\}$. Finally, $\mathcal{A}'$ runs $\mathcal{A}$ on inputs 1^k and $PK_0 = (mpk_{ta_0}, pub)$ and $PK_1 = (mpk_{ta_1}, pub)$.

- $\mathcal{A}$'s decryption queries are answered by $\mathcal{A}'$ as follows:

  - Queries of the form $((com^*, c, tag), PK_x)$ with $x \in \{0, 1\}$ are answered with $\perp$.

  - Queries of the form $((com, c, tag), PK_x)$ with $com \neq com^*$ and $x \in \{0, 1\}$ are answered by first querying $\mathcal{A}'$'s Challenger for the private key corresponding to com in ta_x and using it to decrypt the ciphertext.

- When $\mathcal{A}$ submits two equal length messages m_0 and m_1, $\mathcal{A}'$ first selects a bit b at random. Then, it sends $((m_b||dec^*), mpk_{ta_b})$ and $((0^{l+l'}), mpk_{ta_2})$ to its Challenger.

- $\mathcal{A}'$ receives c^* and then computes $tag^* = \texttt{MAC}(r^*, c^*)$. It sends (com^*, c^*, tag^*) to $\mathcal{A}$.

- Further decryption queries are answered as above.

- Finally $\mathcal{A}$ outputs a bit b'. If $b' = b$, then $\mathcal{A}'$ outputs 0; otherwise $\mathcal{A}'$ outputs 1.

When the encryption query of $\mathcal{A}'$ is answered with an encryption corresponding to $((m_b||dec^*), mpk_{ta_b})$ then $\mathcal{A}$'s view is exactly as in Game 1. On the other hand, when the encryption query of $\mathcal{A}'$ is answered with the encryption corresponding to $((0^{l+l'}), mpk_{ta_2})$ then $\mathcal{A}$'s view is exactly as in Game 2.

Therefore we have,

$$\begin{aligned}\mathbf{Adv}^{\text{s-id r-m-IND-TAA-CPA}}_{\mathcal{A}',\Pi'}(k) &= |\tfrac{1}{2}\Pr_1[\texttt{Succ}] + \tfrac{1}{2}\Pr_2[\texttt{Succ}] - \tfrac{1}{2}| \\ &= \tfrac{1}{2}|\Pr_1[\texttt{Succ}] + \Pr_2[\overline{\texttt{Succ}}] - 1| \\ &= \tfrac{1}{2}|\Pr_1[\texttt{Succ}] - \Pr_2[\texttt{Succ}]|\end{aligned}$$

and the claim follows from the security of Π'.

Claim 2B: $|\Pr_2[\texttt{Forge}] - \Pr_1[\texttt{Forge}]|$ is negligible.

Proof of Claim 2B: The proof is the same as that for Claim 2A except that now $\mathcal{A}'$ runs $\mathcal{A}$ to completion and then for decryption queries of the form (com^*, c, tag) it checks if $\texttt{Vrfy}(r^*, c, tag) = 1$. If such a query exists then $\mathcal{A}'$ outputs 1, else it outputs 0. The claim follows from the security of the MAC scheme.

Game 3:

In this game components com^* and c^* of the ciphertext are computed as in Game 2. However, tag^* is now computed with a random key $\hat{r} \in \{0,1\}^k$ as $tag^* = \texttt{MAC}(\hat{r}, c^*)$. Event `Forge` is defined as earlier, except using the key $\hat{r}$, i.e. `Forge` is now the event that $\mathcal{A}$ makes a decryption query with (com^*, c, tag) and $\texttt{Vrfy}(\hat{r}, c, tag) = 1$.

Claim 3: $|\Pr_3[\texttt{Forge}] - \Pr_2[\texttt{Forge}]|$ is negligible.

Proof of Claim 3: This follows from the hiding property of the encapsulation scheme. Consider an adversary $\mathcal{B}$ against the hiding property of the encapsulation scheme, that is given input $(1^k, pub, com^*, \hat{r})$.

$\mathcal{B}$ runs $\texttt{CommonSetup}(1^k)$ to obtain $params$, runs $\texttt{TASetup}(params)$ twice to obtain (mpk_{ta_0}, msk_{ta_0}) and (mpk_{ta_1}, msk_{ta_1}) and then runs $\mathcal{A}$ on inputs $PK_0 = (mpk_{ta_0}, pub)$ and $PK_1 = (mpk_{ta_1}, pub)$. $\mathcal{B}$ also runs $\texttt{TASetup}(params)$ and obtains (mpk_{ta_2}, msk_{ta_2}) and sets $PK_2 = (mpk_{ta_2}, pub)$.

- Decryption queries are answered as usual.
- When $\mathcal{A}$ submits two messages m_0 and m_1, the adversary $\mathcal{B}$ computes the ciphertext $c^* = \texttt{Enc}'(mpk_{ta_2}, com^*, (0^{l+l'}))$ and then computes $tag^* = \texttt{MAC}(\hat{r}, c^*)$ and returns the challenge ciphertext (com^*, c^*, tag^*) to $\mathcal{A}$.
- At the end of the experiment, $\mathcal{B}$ decrypts queries of the form (com^*, c, tag), using the secret keys msk_{ta_0} and msk_{ta_1}, to see if $\texttt{Vrfy}(\hat{r}, c, tag) = 1$. If so, $\mathcal{B}$ outputs 1, else it outputs 0.

If $\hat{r}$ is chosen such that $(\hat{r}, com^*, dec^*)$ is output by $\texttt{Encap}(1^k, pub)$ then $\mathcal{A}$'s view is identical to that in Game 2 and $\mathcal{B}$ outputs 1 with probability $\Pr_2[\texttt{Forge}]$. If on the other hand, $\hat{r}$ is chosen independently of com^* then the view of $\mathcal{A}$ is identical to that in Game 3 and $\mathcal{B}$ outputs 1 with probability $\Pr_3[\texttt{Forge}]$. The claim follows from the hiding property of the encapsulation scheme.

Claim 4: $\Pr_3[\texttt{Forge}]$ is negligible.

Proof of Claim 4: This follows from the security of the MAC scheme. Let $q = q(k)$ be an upper bound on the number of the number of decryption queries made by $\mathcal{A}$. Let us consider a forging algorithm $\mathcal{F}$. $\mathcal{F}$ chooses a random index $j \leftarrow \{1, \ldots, q\}$. It begins simulating Game 3 for $\mathcal{A}$, by running $\texttt{Init}(1^k)$ to obtain pub, running $\texttt{CommonSetup}(1^k)$ to obtain $params$, then running $\texttt{TASetup}(params)$ thrice to obtain (mpk_{ta_0}, msk_{ta_0}), (mpk_{ta_1}, msk_{ta_1}) and (mpk_{ta_2}, msk_{ta_2}). Finally, the forger $\mathcal{F}$ runs $\mathcal{A}$ on inputs $PK_0 = (mpk_{ta_0}, pub)$ and $PK_1 = (mpk_{ta_1}, pub)$. If the j^{th} decryption query (com_j, c_j, tag_j) occurs before the Challenge query than $\mathcal{F}$ outputs (c_j, tag_j) and stops. Otherwise, in response to the Challenge query on (m_0, m_1), the forger $\mathcal{F}$ computes $\texttt{Encap}(1^k, pub)$ to obtain (r^*, com^*, dec^*) followed by $c^* =$

$\texttt{Enc}'(mpk_{ta_2}, com^*, (0^{l+l'}))$. Next, it submits c^* to its MAC oracle and receives tag^*. $\mathcal{F}$ then gives the challenge ciphertext (com^*, c^*, tag^*) to $\mathcal{A}$ and continues running $\mathcal{A}$ till its submits its j^{th} decryption query (com_j, c_j, tag_j) . At this point, $\mathcal{F}$ outputs (c_j, tag_j) and stops. $\mathcal{F}$'s success probability in outputting a valid forgery is at least $\Pr_3[\texttt{Forge}]/q$. The claim follows from the security of the MAC scheme.

Finally, putting everything together,

$$
\begin{aligned}
& \mathbf{Adv}_{\mathcal{A}}^{\text{IND-IK-CCA}}(k) \\
= \;& |\Pr_0[\texttt{Succ}] - 1/2| \\
= \;& |\Pr_0[\texttt{Succ}] - \Pr_1[\texttt{Succ}] + \Pr_1[\texttt{Succ}] - 1/2| \\
\leq \;& |\Pr_0[\texttt{Succ}] - \Pr_1[\texttt{Succ}]| + |\Pr_1[\texttt{Succ}] - 1/2| \\
= \;& \Pr_1[\texttt{Valid}] + |\Pr_1[\texttt{Succ}] - 1/2| \\
\leq \;& \Pr_1[\texttt{NoBind}] + \Pr_1[\texttt{Forge}] + |\Pr_1[\texttt{Succ}] - \Pr_2[\texttt{Succ}] + \Pr_2[\texttt{Succ}] - 1/2| \\
\leq \;& \Pr_1[\texttt{NoBind}] + \Pr_1[\texttt{Forge}] + |\Pr_1[\texttt{Succ}] - \Pr_2[\texttt{Succ}]| + |\Pr_2[\texttt{Succ}] - 1/2| \\
= \;& \Pr_1[\texttt{NoBind}] + \Pr_1[\texttt{Forge}] + |\Pr_1[\texttt{Succ}] - \Pr_2[\texttt{Succ}]| \\
= \;& \Pr_1[\texttt{NoBind}] + \Pr_1[\texttt{Forge}] + \Pr_2[\texttt{Forge}] - \Pr_2[\texttt{Forge}] \\
& + \Pr_3[\texttt{Forge}] - \Pr_3[\texttt{Forge}] + |\Pr_1[\texttt{Succ}] - \Pr_2[\texttt{Succ}]| \\
\leq \;& \Pr_1[\texttt{NoBind}] + \Pr_3[\texttt{Forge}] + |\Pr_2[\texttt{Forge}] - \Pr_3[\texttt{Forge}]| \\
& + |\Pr_1[\texttt{Forge}] - \Pr_2[\texttt{Forge}]| + |\Pr_1[\texttt{Succ}] - \Pr_2[\texttt{Succ}]|
\end{aligned}
$$

The proof follows, as all terms are negligible.

□

Final Observations

The multi-TA version of the Gentry scheme (Section 5.2) meets stronger notions of security than those required for the application of Theorem 6.2. Boneh and Katz suggest an encapsulation scheme suitable for application in their original transform [24] and this is also suitable for our modified construction. We can therefore instantiate the modified BK transform with the multi-TA version of the Gentry scheme, the hiding and blinding encapsulation scheme proposed in [24] and any suitable MAC (for example [119]) to obtain an IND-IK-CCA PKE scheme. The use of the encapsulation and MAC scheme in the place of the signature scheme in the CHK transform results in improved efficiency and shorter ciphertexts.

Chapter 7

IBE for Coalition Environments

7.1 Introduction

Complex environments involving cooperation between entities formed under distinct roots of trust are very relevant in Mobile Ad-Hoc Networks (MANETs). In this chapter, we consider the fundamental and important question of how to enable secure communications between disparate entities within such heterogeneous, potentially resource-constrained environments.

Traditional public key cryptography is not well-suited for such networks since it utilizes large amounts of energy and bandwidth and requires a constant connection to a public key infrastructure (PKI) to look up public keys, certificates and revocation data. In addition, transmitting, storing and verifying certificates puts extra strain on already limited resources. IBE is an attractive choice for such resource-constrained environments since it eliminates the need for public key lookups, does not need certificates, and allows revocation to be simplified by using time-based identifiers.

A number of authors have studied the applicability of IBE to MANETs [77, 35, 85, 70, 6]. However, almost all prior work in this area has been limited to the case of a single TA. In more complex settings, such as those encountered in dynamic coalition forming, we may have multiple TAs (from different administrative domains) and hence

multiple roots of trust. As an example, in the application scenario envisioned in [99], coalition forces controlled by one coalition member need the ability to communicate securely with individuals and entities associated with other members of the coalition.

In this type of scenario, it is necessary to find methods that allow entities that are under different roots of trust to securely communicate with each other. A particular challenge then arises when coalitions are formed *dynamically*. In this situation pre-configuration of devices with all of the required static security data (such as system-wide public parameters for each of the coalition TAs) before a mission commences will not, in general, be possible: the exact set of coalition members may be unpredictable in advance, and their security data may need to be updated during the course of a mission. Issuing a fresh set of system-wide public parameters for each new coalition, along with private keys for all of the coalition members, is an unattractive solution since it involves high communication costs and needs secure channels for distribution of the new private keys. This is particularly true when coalitions are short-lived, forming and re-forming rapidly. Another approach would be for coalition members to distribute the required security data amongst themselves as and when necessary, in an *ad hoc* fashion. However, this would require each member to maintain a complete set of data, would involve significant storage overhead, and could result in problems if a secure communication facility was urgently needed but the required security data had not yet been received.

We present an alternative solution to the problem of enabling secure communications in dynamic coalitions. Our solution enables any entity A to securely communicate with any other coalition entity B, even if B is associated with a different TA and A is unable to obtain authentic public parameters for B's TA (for example, if that TA is simply unknown). The cost of our solution is that when a coalition forms, the TAs must broadcast a small amount of additional information to each coalition member. However, this broadcast need not itself be encrypted. Now A can use its

own TA's public parameters (along with B's identifier) to perform the encryption to B. In fact, A can use a set of authentic public parameters from any coalition TA. Meanwhile, B, upon receipt of the broadcast information from the TAs, is able to perform key translations: B can convert its existing private key issued by its TA into private keys that are valid for the same identifier under each and every one of the other TAs in the coalition. This means that B is able to decrypt A's message no matter which public parameters were selected by A during encryption. On the other hand, no other entity is able to translate its private key to enable decryption of the message intended for B.

We provide security models that are appropriate to this kind of multi-TA application scenario. The security models we consider for these scenarios are similar to the multi-TA security notions we have discussed in earlier chapters, with suitable modifications to reflect the complexities arising from the additional public parameters that are broadcast.

We also give a specific instantiation of our solution that adapts the Boneh-Franklin `BasicIdent` IBE scheme [22]. This results in a highly efficient encryption scheme in which the size of the coalition-enabling broadcast is linear in the number of coalition partners.

In comparison to the approach based on distributing a fresh set of system-wide public parameters for each new coalition, our approach eliminates the need for the secure channel to distribute new private keys, as well as the need for bespoke communication between TAs and individual nodes. Compared with the *ad hoc* approach, our method may be more reliable (since nodes in receipt of the single broadcast will immediately have all the information needed to enable secure communications). In the specific instantiation we provide in this chapter, our approach also requires less communication. In general, our approach does make use of a network-wide authentic broadcast, and assumes that entities do know one another's identifiers even if the

relevant TA public parameters are not available. It also requires the various TAs to share some common cryptographic parameters which may reduce its flexibility.

7.2 Related Work

To the best of our knowledge, multi-TA IBE schemes where the TAs collaborate to enable secure communications in dynamic coalitions have not been investigated in the literature prior to this work.

We note that HIBE schemes [74, 63] do employ multiple TAs but the scenario we consider cannot be satisfied using HIBE. In HIBE schemes, the setup of the lower level TAs is closely bound to the upper level TAs, whereas we envisage a scenario where existing single-TA deployments can be made to interact dynamically with minimum overhead, i.e. without reissuing private keys or similar infrastructure overhaul.

In the situation where there are no prior relationships between the different TAs except that they may share a subset of common public parameters, if the sender of a message can obtain authentic public parameters for all the TAs, then cross-TA communication can be enabled by encrypting the same message for the identifier in question using the public parameters of each of the TAs. However, such schemes typically lead to significant ciphertext expansion and additional cryptographic computation, which may render them unsuitable in resource-constrained environments. This kind of scenario has been addressed in [112], where a scheme is given that only needs a single pairing operation during encryption. However, the security model used in [112] is the usual one for the single TA setting, and no consideration is given as to how security may be affected by encrypting the same message using multiple, different sets of public parameters. In addition, the schemes of [112] reuse randomness to enhance efficiency, and this is not formally addressed in the security analysis. Barbosa and Farshim [7] do consider the security of multi-recipient IBE with randomness

re-use, but only in the single-TA setting.

7.3 IBE for Coalition Environments

In our approach, each user obtains a single private key corresponding to its identifier from its respective TA, and later uses additional information broadcast by the TAs to translate its private key. The translation allows a user to convert an existing private key into one that would have been issued by any of the other TAs. This in turn allows a sender to encrypt messages to a recipient using the recipient's identifier and the public parameters of any one of the TAs. The recipient can then use translation to obtain a private key that allows it to decrypt the ciphertext.

We call an identity based encryption scheme satisfying these properties a Key-Translating IBE scheme.

Definition 7.1. *A Key-Translating IBE scheme is defined in terms of the following algorithms:*

- `CommonSetup`: On input 1^k, outputs *params*, a set of system parameters shared by all TAs. We assume that *params* includes a description of the message space MsgSp and the ciphertext space CtSp.
- `Setup`: Takes as input *params* and an integer $n = n(k)$. Here $n \geq 2$ denotes the number of TAs. To facilitate presentation, we let $\mathcal{T} = \{ta_i : 1 \leq i \leq n\}$ denote the set of labels of all the TAs. Then,
  - For each $ta \in \mathcal{T}$, generates a unique master public key mpk_{ta} (which includes *params*) and a master secret key msk_{ta}. We let *MPK* and *MSK* denote the set of all master public keys and master secret keys respectively.

  - For every pair of TAs, $ta_i, ta_j \in \mathcal{T}$, uses $params$, msk_{ta_i}, msk_{ta_j} and returns a relation rel_{ta_i,ta_j} from a space $\mathcal{R}$ of relations. Let $\mathcal{R}_{ta}$ denote the set of all relations pertaining to $ta \in \mathcal{T}$.

- `KeyDer`: On input mpk_{ta}, msk_{ta} for $ta \in \mathcal{T}$ and an identifier $id \in \{0,1\}^*$, returns a private key $usk_{id,ta}$ corresponding to id for the TA with label ta. This algorithm may or may not be randomized.

- `TranslateKey`: On input usk_{id,ta_i} and $rel_{ta_i,ta_j}, \in \mathcal{R}_{ta_i}$ returns a private key usk_{id,ta_j} corresponding to id for the TA with label ta_j.

- `Enc`: On input mpk_{ta} for $ta \in \mathcal{T}$, $id \in \{0,1\}^*$, $m \in \mathsf{MsgSp}$, returns a ciphertext $c \in \mathsf{CtSp}$.

- `Dec`: On input mpk_{ta} for $ta \in \mathcal{T}$, $c \in \mathsf{CtSp}$ and private key $usk_{id,ta}$, returns $m \in \mathsf{MsgSp}$ or a failure symbol $\perp$.

These algorithms must satisfy the standard consistency requirement that $\forall m \in \mathsf{MsgSp}$, $\forall id \in \{0,1\}^*$ and $\forall ta \in \mathcal{T}$,

$$\mathtt{Dec}(c, usk_{id,ta}) = m \text{ where } c = \mathtt{Enc}(mpk_{ta}, id, m).$$

In addition, decryption with an appropriately translated private key must also be consistent i.e. $\forall m \in \mathsf{MsgSp}$, $\forall id \in \{0,1\}^*$ and $\forall ta \in \mathcal{T}$,

$$\mathtt{Dec}(c, \mathtt{TranslateKey}(usk_{id,ta_i}, rel_{ta_i,ta_j})) = m \text{ where } c = \mathtt{Enc}(mpk_{ta_j}, id, m).$$

We assume here that the relations $\mathcal{R}_{ta}$ pertaining to each $ta \in \mathcal{T}$ are broadcast to all users in the system (for convenience, we let $\mathcal{R}_{TA}$ denote all the broadcast relations). After this is done, a user with identifier id and private key usk_{id,ta_i} may use the relation $rel_{ta_i,ta_j}, \in \mathcal{R}_{ta_i}$ in Algorithm `TranslateKey` to obtain usk_{id,ta_j}. This enables that user to decrypt ciphertexts computed using identifier id and the master public key of ta_j.

7.3.1 Security Notions

We introduce security models appropriate to the above cryptographic primitive. We define the KT-m-IND-RA-CCA security experiment to capture an indistinguishability and anonymity notion of security in this setting. This security notion is an extension of the m-IND-RA-CCA security experiment. The adversary has access to the usual key derivation and decryption oracles and specifies two messages/identity pairs and a single TA in its challenge. The adversary is given the ciphertext corresponding to one of the message/identity pairs corresponding to the specified TA and its task is to determine which message/identity pair was used to compute the ciphertext. However, now the adversary also has access to $\mathcal{R}_{TA}$, the set of relations that are broadcast. This means that once the adversary performs a key derivation query for an identity corresponding to a particular TA, it has effectively obtained the private key for that identity corresponding to every TA for which relations are available in $\mathcal{R}_{TA}$. The win conditions must therefore be appropriately strengthened. Removing the adversary's access to the decryption oracle gives us the KT-m-IND-RA-CPA security notion.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{KT-m-IND-RA-CCA-}b}(k)$

$params \leftarrow \texttt{CommonSetup}(1^k)$

$TASet \leftarrow \emptyset$

$(MPK, MSK, \mathcal{R}_{TA}) \leftarrow \texttt{Setup}(params, n)$

$\forall ta \in \mathcal{T}$,

$IDSet_{ta} \leftarrow \emptyset,\ CSet_{ta} \leftarrow \emptyset$

$(m_0, m_1, id_0, id_1, ta, state) \leftarrow$

$\mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{find}, MPK, \mathcal{R}_{TA})$

$c^* \leftarrow \texttt{Enc}(mpk_{ta}, id_b, m_b)$

$\forall ta \in \mathcal{T},\ CSet_{ta} \leftarrow \emptyset$

$b' \leftarrow \mathcal{A}^{\texttt{KeyDer},\texttt{Dec}}(\texttt{guess}, c^*, state)$

`If` $id_0 \in IDSet_{ta_i}$ for any $1 \leq i \leq n$

`Or` $id_1 \in IDSet_{ta_i}$ for any $1 \leq i \leq n$

`Or` $\{(id_0, c^*)\} \in CSet_{ta}$

`Or` $\{(id_1, c^*)\} \in CSet_{ta}$

`Then Return` 0

`Else Return` b'.

Oracle $\texttt{KeyDer}(ta, id)$

$IDSet_{ta} \leftarrow IDSet_{ta} \cup \{id\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

`Return` $usk_{id,ta}$

Oracle $\texttt{Dec}(ta, id, c)$

$CSet_{ta} \leftarrow CSet_{ta} \cup \{(id, c)\}$

$usk_{id,ta} \leftarrow$

$\texttt{KeyDer}(mpk_{ta}, msk_{ta}, id)$

$m \leftarrow \texttt{Dec}(mpk_{ta}, usk_{id,ta}, c)$

`Return` m

The KT-m-IND-RA-CCA Security Experiment.

7.3.2 An Instantiation Based on the `BasicIdent` Scheme

We give an instantiation of a Key-Translating IBE scheme based on the `BasicIdent` scheme of [22]. We will show that this scheme is KT-m-IND-RA-CPA secure.

- `CommonSetup`: On input a security parameter 1^k, this algorithm:
  - Runs `PairingGen` on input 1^k to obtain $(\mathbb{G}, \mathbb{G}_T, e, p, g)$.

- Chooses cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}^*$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^l$ for some $l = l(k)$.
- Outputs $params = (\mathbb{G}, \mathbb{G}_T, e, p, g, H_1, H_2, k, l)$.

The message space is $\mathsf{MsgSp} = \{0,1\}^{l-k}$, where k is the length of padding employed. The ciphertext space is $\mathsf{CtSp} = \mathbb{G}^* \times \{0,1\}^l$.

- `Setup`: On input $params$ and $n = n(k), n \geq 2$, where n denotes the number of TAs and $\mathcal{T} = \{ta_i : 1 \leq i \leq n\}$ denote the set of labels of all TAs, this algorithm:
  - Sets $s_i \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $msk_i = s_i$ and $mpk_i = (params, g^{s_i})$ for $1 \leq i \leq n$.
  - For each i with $1 \leq i \leq n$, sets

$$\mathcal{R}_i = \{rel_{ta_i,ta_j} : 1 \leq j \leq n\}$$

    where $rel_{ta_i,ta_j} = s_j \cdot {s_i}^{-1} \bmod p$.
  - Outputs $\mathcal{R}_i$ for $1 \leq i \leq n$.
- `KeyDer`: On input mpk_i, msk_i and an identifier $id \in \{0,1\}^*$, this algorithm outputs $usk_{id,ta_i} = H_1(id)^{s_i}$.
- `Enc`: On input mpk_i for $ta_i \in \mathcal{T}$, to encrypt a message $m \in \mathsf{MsgSp}$, under $id \in \{0,1\}^*$, this algorithm:
  - Chooses $r \xleftarrow{\$} \mathbb{Z}_p^*$.
  - Outputs the ciphertext

$$c = (u, v) = (g^r, (m||0^k) \oplus H_2(e(H_1(id), g^{s_i})^r))$$

- `TranslateKey`: On input usk_{id,ta_i}, the private key corresponding to $id \in \{0,1\}^*$ for $ta_i \in \mathcal{T}$ and $rel_{ta_i,ta_j} \in \mathcal{R}_i$, returns $(usk_{id,ta_i})^{rel_{ta_i,ta_j}}$.

  Note that

$$\begin{aligned}(usk_{id,ta_i})^{rel_{ta_i,ta_j}} &= H_1(id)^{s_i s_j \cdot s_i{}^{-1}} \\ &= H_1(id)^{s_j} \\ &= usk_{id,ta_j}.\end{aligned}$$

  so that algorithm `TranslateKey` does indeed convert private keys correctly.

- `Dec`: On input $c = (u, v)$, mpk_i and private key usk_{id,ta_i} this algorithm:

  - Runs `TranslateKey` to obtain usk_{id,ta_j} for all $1 \leq j \leq n, j \neq i$. (Note that this step needs to be performed just once if the user can store all the keys usk_{id,ta_j}.)
  - Computes $m_j^{'} = v \oplus H_2(e(u, usk_{id,ta_j}))$ for all $1 \leq j \leq n$.
  - For each $m_j^{'}$ checks if the last k bits are zero. (Parameter k is selected such that with overwhelming probability only the decryption of c using the correct private key yields this padding format.) If it is, it sets m to the left $l - k$ bits of $m_j^{'}$ and outputs m. If no $m_j^{'}$ obtained in this way has its last k bits equal to 0, then this algorithm outputs $\perp$.

We note that if we assume that ciphertexts contain a label indicating which TA's master public key was used to perform the encryption, then the decryption operation can be done in a single step using the correct private key. Note too that, for this particular scheme, the complete set of n^2 values rel_{ta_i,ta_j} can be computed upon receipt of a single broadcast containing the values rel_{ta_1,ta_j} for $j = 2, \ldots, n$. This means that this specific scheme needs a broadcast whose size is linear in the number of TAs, rather than quadratic.

Theorem 7.1. *The multi-TA scheme based on* ***BasicIdent*** *is KT-m-IND-RA-CPA secure in the Random Oracle model, assuming the hardness of the BDH problem in groups output by* ***PairinGen****.*

Proof. Suppose there is an KT-m-IND-RA-CPA adversary $\mathcal{A}$ against the multi-TA IBE scheme with advantage ε and running in time t. We show how to construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the IND-RA-CPA property of the `BasicIdent` scheme.

$\mathcal{B}$'s inputs are the parameters $mpk = (\mathbb{G}, \mathbb{G}_T, e, p, g, P_{pub}, H_1, H_2, l)$ of the `BasicIdent` scheme. $\mathcal{B}$'s task is to break the IND-RA-CPA security of the `BasicIdent` scheme and it does this by acting as a challenger for $\mathcal{A}$.

$\mathcal{B}$ generates the parameters of the multi-TA IBE scheme. It sets $params = (\mathbb{G}, \mathbb{G}_T, e, p, g, H_1, H_2, l)$. Let $\mathcal{T} = \{ta_i : 1 \leq i \leq n\}$ denote the set of labels of all TAs where n is the number of TAs . Then $\mathcal{B}$ sets $mpk_I = (params, P_{Pub})$ for $ta_I \in \mathcal{T}$ with $I \xleftarrow{\$} \{1 \ldots n\}$. Note that $P_{Pub} = g^s$ for $s \in \mathbb{Z}_p^*$ but the value of s is not known to $\mathcal{B}$. It sets $mpk_j = (params, P_{Pub}^{\lambda_j})$ for ta_j, for each $1 \leq j \leq n, j \neq I$, where each λ_j is drawn uniformly at random from $\mathbb{Z}_p^*$. It also sets $\lambda_I = 1$. The master secret key for ta_j is $msk_j = \lambda_j \cdot s$ which, again, $\mathcal{B}$ does not know. However, with the knowledge of the λ_j values it is able to generate, for every $ta_i \in \mathcal{T}$, the appropriate set $\mathcal{R}_i = \{rel_{ta_i,ta_j} : 1 \leq j \leq n\}$. $\mathcal{B}$ does this by setting $rel_{ta_i,ta_j} = \lambda_j \cdot {\lambda_i}^{-1} \bmod p$. $\mathcal{B}$ then gives these public keys to $\mathcal{A}$, along with the sets of relations $\mathcal{R}_i$ for $1 \leq i \leq n$.

$\mathcal{A}$ makes a series of queries which $\mathcal{B}$ answers as follows.

- `Phase 1:` $\mathcal{A}$ makes a series of key derivation queries on (ta, id) combinations. $\mathcal{B}$ asks the corresponding key derivation query on id to its challenger which responds with the private key usk_{id,ta_i}. If $ta = ta_i$, $\mathcal{B}$ relays the response to $\mathcal{A}$.

Otherwise, B uses the appropriate λ_j value to compute $usk_{id,ta_j} = usk_{id,ta_i}^{\lambda_j}$ which corresponds to the private key of id for ta_j and sends this to $\mathcal{A}$.

- `Challenge:` When $\mathcal{A}$ decides to end Phase 1 of the attack it outputs two messages/identifier tuples (m_0, id_0) and (m_1, id_1) and the TA, ta, on which it wishes to be challenged, subject to the condition that no key derivation query was asked on id_0 or id_1 in Phase 1. $\mathcal{B}$ pads the two messages m_0 and m_1 with k zero bits to obtain $m_0' = m_0||0^k$ and $m_1' = m_1||0^k$ and sends the tuples (m_0', id_0) and (m_1', id_1) to its challenger and receives the `BasicIdent` encryption $c^* = (u, v)$ corresponding to m_b' and id_b for ta_i, for a bit b chosen uniformly at random. $\mathcal{B}$ sets $c^{*\prime} = (u^{{\lambda_j}^{-1}}, v)$ which corresponds to the encryption of m_b' to identifier id_b for ta_j and sends this to $\mathcal{A}$.

- `Phase 2:` Phase 2 of the attack proceeds as in Phase 1 with the restriction that no key derivation query is allowed on id_0 or id_1.

This completes our description of $\mathcal{B}$'s simulation. Note that $\mathcal{A}$'s view of the simulation is identical to its view in a real attack. All the queries are responded to correctly. When $\mathcal{A}$ terminates by outputting a bit b', $\mathcal{B}$ simply relays this bit to its challenger. Clearly $\mathcal{B}$'s advantage in breaking the IND-RA-CPA security of `BasicIdent` is equal to $\mathcal{A}$'s advantage against the KT-m-IND-RA-CPA security of the multi-TA IBE scheme. We know that $\mathcal{B}$'s advantage against `BasicIdent` is negligible and hence $\mathcal{A}$ must have negligible advantage against the multi-TA scheme.

□

7.4 Open Problems

The scheme we have presented in this chapter is secure in a rather restricted sense, our main aim being to demonstrate an example of a Key-Translating IBE scheme. We pose a number of interesting problems as a consequence of this work.

The Key-Translating IBE scheme we have considered in this chapter involves multiple TAs. It would be interesting to consider extended security notions in this setting, similar to the TA Anonymity security notions we have considered in the previous chapters. We note that the proof of security for the scheme we have presented in this chapter does not allow us to prove TA Anonymity. However, the ciphertexts are standard Boneh-Franklin ciphertexts and we can expect the scheme to be TA anonymous. We may also be able to adapt standard techniques such as the Fujisaki-Okamoto transform to achieve security against CCA attackers.

It will be worth exploring other existing IBE schemes to see whether we can perform key translation of the type considered in this chapter. In addition, it will be interesting to produce schemes where the relations can be derived without the TAs having to share their master secret keys or to prove that this is impossible.

Schemes that are secure in stronger models, for example where a subset of TAs can be corrupted, or which tolerate the corruption of private keys corresponding to an identifier for a subset of TAs pose additional challenges.

Chapter 8

Relations between ID-NIKD and IBE

8.1 Introduction

In this chapter, we investigate the relationship between Identity Based Non Interactive Key Distribution (ID-NIKD) and IBE. We provide a new security model for ID-NIKD, and a generic construction that converts a secure ID-NIKD scheme into a secure IBE scheme. This conversion is used to explain the relationship between the ID-NIKD scheme of Sakai, Ohgishi and Kasahara [101] and the `BasicIdent` scheme of Boneh and Franklin [22].

We begin by establishing a new security model for ID-NIKD that more accurately captures the security desired from this primitive. Our new model is stronger than the existing model of Dupont and Enge [53] and is inspired by earlier models for (interactive) authenticated key exchange [13, 17, 11, 34]. Since ID-NIKD is itself a very useful cryptographic primitive with a wide range of applications in cryptography (see for example [111, 25, 5]), our new model should be of independent interest. We show that the well-known ID-NIKD scheme of Sakai *et al.* [101] is secure in our model, thus strengthening the main result of [53].

We then explore the relationship between IBE and ID-NIKD. We give a generic conversion that takes any ID-NIKD scheme that is secure in our model and that satisfies some mild technical conditions, and produces from it an IBE scheme that is secure in the IND-ID-CPA security model of [22]. The conversion itself works in the Standard Model. Chosen-ciphertext security in the ROM can easily be obtained by applying a secondary conversion, for example the transforms of [78, 120]. Our ID-NIKD-to-IBE conversion provides a framework within which existing ID-NIKD and IBE schemes can be related. For example, our conversion allows us to show how the Boneh-Franklin IBE scheme [22] arises from the ID-NIKD scheme of Sakai *et al.* [101].

We are not aware of previous work pointing out the relationship between ID-NIKD and IBE. Indeed, these two different primitives are sometimes confused in the literature [86]. In the particular case of the ID-NIKD scheme of Sakai *et al.* [101] and the Boneh-Franklin IBE scheme [22], it has been noted by many authors that the same algebraic setting and keying method is used. But, as far as we are aware, their exact relationship has not been clarified until now.

8.2 Background and Definitions

Definition 8.1. *Consider a family of functions* $f = (f_k)_{k\in\mathbb{N}}$ *where* $f_k : \{0,1\}^* \to \mathcal{T}_k$ *for some set* $\mathcal{T}_k$*. We define the advantage of an algorithm* $\mathcal{A}$ *in breaking the one-wayness of function* f_k *to be:*

$$\boldsymbol{Adv}_{\mathcal{A}}^{OW}(k) = \Pr[f_k(x) = y : x' \leftarrow \{0,1\}^*, y = f_k(x'), x \leftarrow \mathcal{A}(y)].$$

Here, $\mathcal{A}$ *is given a description of* f_k *as part of its input.*

The family $(f_k)_{k\in\mathbb{N}}$ *is said to be one-way if* $\boldsymbol{Adv}_{\mathcal{A}}^{OW}(k)$ *is a negligible function for all algorithms* $\mathcal{A}$ *running in time polynomial in* k*.*

8.3 Identity Based Non Interactive Key Distribution

We formally define an Identity Based Non Interactive Key Distribution (ID-NIKD) scheme by three distinct algorithms: `Setup`, `Extract` and `SharedKey`. Algorithms `Setup` and `Extract` are executed by the Trusted Authority (TA), while `SharedKey` can be executed by any entity in possession of its private key and the identifier of any other entity with which it wishes to generate a shared key.

- `Setup`: On input 1^k, outputs a master public key (or system parameters) mpk and master secret key msk.
- `Extract`: On input mpk, msk and identifier $id \in \{0,1\}^*$, returns a private key usk_{id} from some space of private keys $\mathcal{SK}$.
- `SharedKey`: On input mpk, a private key usk_{id_A} and an identifier $id_B \in \{0,1\}^*$, where $id_B \neq id_A$, this algorithm returns a key $K_{A,B}$ from some space of shared keys $\mathcal{SHK}$ specified in mpk.

We require that, for any pair of identities id_A, id_B, and corresponding private keys usk_{id_A}, usk_{id_B}, `SharedKey` satisfies the constraint:

$$\texttt{SharedKey}(mpk, usk_{id_A}, id_B) = \texttt{SharedKey}(mpk, usk_{id_B}, id_A).$$

This ensures that entities A and B can indeed generate a shared key without any interaction. We will normally assume that $\mathcal{SHK}$, the space of shared keys, is $\{0,1\}^{n(k)}$ for some function $n(k)$. In practice, this can be arranged by hashing a "raw" key.

8.3.1 Definition of Security for ID-NIKD

Dupont and Enge [53] introduced the first formal security model for ID-NIKD. We present our new model, discuss a variant of it, and then explain how it strengthens

the model of [53].

Our model is stated in terms of a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The challenger $\mathcal{C}$ takes as input the security parameter 1^k, runs algorithm `Setup` of the ID-NIKD scheme and gives $\mathcal{A}$ mpk. It keeps msk to itself. $\mathcal{A}$ then makes queries of the following three types:

- `Extract`(id): $\mathcal{C}$ responds by running algorithm `Extract` of the ID-NIKD scheme with input (mpk, msk, id) to generate a private key usk_{id}. $\mathcal{C}$ gives usk_{id} to $\mathcal{A}$.
- `Reveal`(id_A, id_B): $\mathcal{C}$ responds by running `Extract`(mpk, msk, id_A) to obtain a private key usk_{id_A} and then `SharedKey`(mpk, usk_{id_A}, id_B) to generate a shared key $K_{A,B}$. $\mathcal{C}$ gives $K_{A,B}$ to $\mathcal{A}$.
- `Test`(id_A, id_B): $\mathcal{C}$ responds by calculating $K_{A,B}$ as above. $\mathcal{C}$ then selects $b \stackrel{\$}{\leftarrow} \{0,1\}$. If $b = 0$ then $\mathcal{C}$ gives $K_{A,B}$ to $\mathcal{A}$; if $b = 1$, then $\mathcal{C}$ gives $\mathcal{A}$ a random element from $\mathcal{SHK}$.

$\mathcal{A}$'s queries may be made adaptively and are arbitrary in number, except that $\mathcal{A}$ is allowed to make only one **Test** query. A straightforward hybrid argument can be used to relate our model with a single **Test** query to a model allowing multiple **Test** queries for a fixed bit b. In our model, no query to the **Reveal** oracle is allowed on the pair of identities selected for the **Test** query (in either order), and no **Extract** query is allowed on either of the identities involved in the **Test** query. These last two conditions are necessary to prevent the adversary from trivially winning the security game.

Finally, $\mathcal{A}$ outputs a bit b', and wins the game if $b' = b$. $\mathcal{A}$'s advantage in this IND-SK (indistinguishability of shared key) security game is defined to be

$$\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SK}}(k) = \left|\Pr[b = b'] - 1/2\right|.$$

We say that an ID-NIKD scheme is IND-SK secure if for any polynomial time adversary $\mathcal{A}$, the function $\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SK}}(k)$ is negligible.

A weaker "computational" version of this model can be obtained by removing the **Test** oracle and changing the win condition for the game to require that $\mathcal{A}$ output the actual shared key $K_{A,B}$ for two identities id_A, id_B neither of which is the subject of an **Extract** query, and that are not together the subject of a **Reveal** query. We refer to OW-SK security in this case.

Comparison to the Model of Dupont and Enge

We detail a number of differences between our security model for ID-NIKD and that of Dupont and Enge [53]. The model of [53] gives the adversary access to the **Extract** oracle, but not to a **Reveal** oracle. Thus it captures collusion attacks, but does not give the adversary any direct oracle access to shared keys. This is somewhat analogous to a "No Reveals" adversary that is sometimes considered in weak security models for (interactive) key exchange. This restriction means that the model of [53] does not capture the natural requirement that the adversary, even after capturing keys shared between some pairs of entities, should still not be able to compute further shared keys. An adversary can, of course, compute any given shared key after extracting the private key of one of the relevant entities. But the different key types (private keys and shared keys) may be afforded different levels of protection in practice, so differentiating between the different types of compromise that are possible gives us a more refined model that may better represent real applications. For example, private keys may be stored and used only in tamper-resistant hardware, while shared keys may be used as transport keys and so be exposed to cryptanalysis or other forms of attack.

We note that, in our model, queries to the **Reveal** oracle can be *simulated* by making appropriate access to the **Extract** oracle. This leads to a reduction from our

security model to a model in which the adversary does not have access to a **Reveal** oracle. However, this reduction requires a correct guess as to which identities will be involved in the **Test** query, implying that it is not tight.

The model of [53] requires the adversary to compute the shared key held between two entities in order to be judged successful, whereas an indistinguishability based definition is stronger, and more closely aligned with existing models for key exchange. Thus the model of [53] is analogous to our OW-SK security model, though it is still weaker than even that model since we provide access to a **Reveal** oracle.

Finally, for the avoidance of confusion, we note that no non interactive key distribution scheme can meet the notion of *forward security* that is enjoyed by many interactive key distribution protocols.

8.3.2 Security of the ID-NIKD Scheme of Sakai *et al.*

In this section, we prove the security of the Sakai-Ohgishi-Kasahara (SOK) ID-NIKD scheme [101] in our extended security model. This strengthens the main result of [53].

The SOK ID-NIKD scheme makes use of a pairing-friendly group generator `PairingGen` and has the following algorithms:

- `Setup`: On input 1^k, this algorithm runs `PairingGen` to obtain a tuple $(\mathbb{G}, \mathbb{G}_T, e, p, g)$ with the usual properties. It selects $s \xleftarrow{\$} \mathbb{Z}_p^*$ and outputs $mpk = (\mathbb{G}, \mathbb{G}_T, e, p, g, g^s, H_1, H_2, n)$ where $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n$ are hash functions, and $\mathcal{SHK} = \{0,1\}^n$. It also outputs $msk = s$.

- `Extract`: On input mpk, msk and identifier $id \in \{0,1\}^*$, this algorithm outputs $usk_{id} = H_1(id)^s$.

- `SharedKey`: On input mpk, a private key usk_{id_A} and an identifier $id_B \in \{0,1\}^*$, where $id_B \neq id_A$, this algorithm outputs $H_2(e(usk_{id_A}, H_1(id_B))) \in \{0,1\}^n$.

It is clear from bilinearity of the map e that `SharedKey` defined in this way satisfies the requirement that entities A and B are able to compute a common key. It also should be pointed out that Dupont and Enge [53] analyse a slight generalization of the SOK ID-NIKD scheme that operates in the more general setting of asymmetric pairings. Our analysis can also be transferred to this setting. Note too that, strictly speaking, there is no need to include the value g^s in the public parameters of the scheme. However, including it simplifies our later presentation.

Theorem 8.1. *The SOK ID-NIKD scheme is secure assuming the hardness of the BDH problem in groups output by* ***PairingGen****. In more detail, for any IND-SK adversary $\mathcal{A}$ against the SOK ID-NIKD scheme that makes q_i queries to hash function H_i for $i = 1, 2$, there is an algorithm $\mathcal{B}$ that solves the BDH problem in groups (G, G_T) output by* ***PairingGen*** *with*

$$\boldsymbol{Adv}_{\mathcal{B}}^{BDH}(k) \geq \boldsymbol{Adv}_{\mathcal{A}}^{IND\text{-}SK}(k)/q_1^2 q_2.$$

Moreover, $\mathcal{B}$ runs in time $O(time(\mathcal{A}))$. The proof is in the random oracle model, i.e. H_1, H_2 are modelled as random oracles.

Proof: Suppose there is an adversary $\mathcal{A}$ against the SOK ID-NIKD scheme with advantage ϵ and running time t. We show how to construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve the BDH problem in groups $(\mathbb{G}, \mathbb{G}_T)$ output by `PairingGen`.

$\mathcal{B}$'s input is $(\mathbb{G}, \mathbb{G}_T, e, p, g, g^a, g^b, g^c)$ where $\mathbb{G}$, $\mathbb{G}_T$ are cyclic groups of prime order p, g generates $\mathbb{G}$, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map, and (g^a, g^b, g^c) is an instance of the BDH problem in $\mathbb{G}$. $\mathcal{B}$'s task is to compute $e(g, g)^{abc}$, and it does this by acting as a challenger for $\mathcal{A}$.

$\mathcal{B}$ gives $\mathcal{A}$ $mpk = (\mathbb{G}, \mathbb{G}_T, e, p, g, g^c, H_1, H_2, n)$ where H_1 and H_2 are Random Oracles simulated by $\mathcal{B}$. Let q_1 be a bound on the number of queries made to H_1 by $\mathcal{A}$ in the course of its attack; similarly let q_2 be a bound on the number of queries

made to H_2. $\mathcal{B}$ chooses two distinct indices I and J uniformly at random from $\{1, 2, \ldots, q_1\}$ and a third index L uniformly at random from $\{1, 2, \ldots, q_2\}$. $\mathcal{A}$ makes a series of queries which $\mathcal{B}$ answers as follows:

- H_1 **queries**: $\mathcal{B}$ maintains a table to handle $\mathcal{A}$'s H_1 queries. If id already appears in an entry of the form (id, d, h) in the table, then $\mathcal{B}$ returns h in response to $\mathcal{A}$'s query. Otherwise, if $\mathcal{A}$'s i-th distinct query to H_1 is on id_i, then $\mathcal{B}$ proceeds as follows:

  1. If $i = I$, then $\mathcal{B}$ adds $(id_I, \perp, g^a)$ to the H_1 table and returns g^a.
  2. If $i = J$, then $\mathcal{B}$ adds $(id_J, \perp, g^b)$ to the H_1 table and returns g^b.
  3. Otherwise, $\mathcal{B}$ chooses d_i uniformly at random from $\mathbb{Z}_q$, adds (id_i, d_i, g^{d_i}) to the H_1 table, and returns g^{d_i}.

  $\mathcal{B}$'s responses to H_1 queries are uniformly and independently generated.

- H_2 **queries**: $\mathcal{B}$ maintains a table to handle $\mathcal{A}$'s H_2 queries. If the query s already appears in an entry of the form (s, K) in the table, then $\mathcal{B}$ returns K in response to $\mathcal{A}$'s query. Otherwise, if $\mathcal{A}$'s i-th distinct query to H_2 is on s_i, then $\mathcal{B}$ selects a random element K_i from $\mathbb{G}_T$, adds (s_i, K_i) to the table, and returns K_i to $\mathcal{A}$. Again, $\mathcal{B}$'s responses to H_2 queries are uniformly and independently generated.

- **Extract** queries: If $\mathcal{A}$ makes an **Extract** query on an identifier id, $\mathcal{B}$ first makes an H_1 query on id if this has not already been done. If $id \in \{id_I, id_J\}$ then $\mathcal{B}$ aborts the simulation. Otherwise, $\mathcal{B}$ finds an entry (id, d, h) in the H_1 table and outputs g^{cd}.

- **Reveal** queries: When $\mathcal{A}$ makes a **Reveal** query on a pair of identities $\{id_i, id_j\}$, $\mathcal{B}$ first makes H_1 queries on id_i and id_j if this has not already been done.

If $\{id_i, id_j\} = \{id_I, id_J\}$ then $\mathcal{B}$ aborts the simulation. Otherwise, suppose $|\{id_i, id_j\} \cap \{id_I, id_J\}| \leq 1$. Then $\mathcal{B}$ can obtain from the H_1 table two entries (id_i, d_i, h_i) and (id_j, d_j, h_j), where either $d_i \neq \perp$ or $d_j \neq \perp$. If $d_i \neq \perp$, then $\mathcal{B}$ responds with the value $H_2(e(h_j, g^{cd_i}))$, first making the H_2 query if necessary. If $d_j \neq \perp$, then $\mathcal{B}$ responds with the value $H_2(e(h_i, g^{cd_j}))$.

- **Test** query: At some point during the simulation $\mathcal{A}$ makes a single **Test** query on a pair of identities. If $\mathcal{A}$ does not choose id_I and id_J as the identities in this query, then $\mathcal{B}$ aborts its interaction with $\mathcal{A}$ and fails. Otherwise, $\mathcal{B}$ outputs a randomly generated value $K \in \{0,1\}^n$. Notice that because of the way in which the simulation is set up, the "correct" key that would be computed by $\mathcal{B}$ in responding to this query is equal to $H_2(e(g,g)^{abc})$.

This completes our description of $\mathcal{B}$'s simulation. When $\mathcal{A}$ terminates by outputting a bit b' (or if $\mathcal{A}$ exceeds its normal running time), then $\mathcal{B}$ outputs the value s_L held in the L-th entry of the H_2 list.

We now assess $\mathcal{B}$'s success probability. Let $\mathcal{F}$ denote the event that $\mathcal{B}$ is not forced to abort during its simulation and let $\mathcal{G}$ denote the event that a query to H_2 on input $e(g,g)^{abc}$ is made at some point during $\mathcal{B}$'s simulation. Then, $\Pr[\mathcal{F}] \geq 1/q_1^2$, and conditioned on event $\mathcal{F}$ occurring, then up to the point where $\mathcal{G}$ occurs, $\mathcal{B}$'s simulation is indistinguishable from that seen when $\mathcal{A}$ interacts with a true challenger. Moreover, if $\mathcal{F}$ occurs, but $\mathcal{G}$ does not occur, then, $\mathcal{A}$'s advantage is zero. For then the shared key that should be held between the two identities involved in the **Test** query (namely id_I and id_J) is equal to $H_2(e(g,g)^{abc})$, and this is independent of $\mathcal{A}$'s view unless event $\mathcal{G}$ occurs. Hence, assuming event $\mathcal{F}$ occurs, we have:

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{A}}^{\text{IND-SK}}(k) &= \left|\Pr[b = b'] - \frac{1}{2}\right| \\
&= \left|\Pr[b = b'|\mathcal{G}]\Pr[\mathcal{G}] + \Pr[b = b'|\neg\mathcal{G}]\Pr[\neg\mathcal{G}] - \frac{1}{2}\right| \\
&\leq |\Pr[\mathcal{G}]| + \left|\Pr[b = b'|\neg\mathcal{G}] - \frac{1}{2}\right| \\
&= \Pr[\mathcal{G}].
\end{aligned}$$

On the other hand, when events $\mathcal{F}$ and $\mathcal{G}$ do occur, $\mathcal{B}$ is successful in outputting $e(g,g)^{abc}$ with probability at least $1/q_2$, since in this case we know that $e(g,g)^{abc}$ is on the H_2 list in some position, and $\mathcal{B}$ selects its output from the list in a random position. Combining these facts, we see that $\mathbf{Adv}_{\mathcal{B}}^{\text{BDH}}(k) \geq \epsilon/q_1^2 q_2$. This completes the proof.

A variant scheme whose security is based on the hardness of the DBDH problem is obtained simply by omitting the hash function H_2, so that the shared key is defined to be the "raw" value $e(usk_{id_A}, H_1(id_B)) \in \mathbb{G}_T$. We can also obtain OW-SK security based on the hardness of the BDH problem for this variant scheme.

8.4 From ID-NIKD to IBE

We show how to construct an IBE scheme from any ID-NIKD scheme that meets two additional technical conditions. We will then show that if the ID-NIKD scheme is IND-SK secure, then the IBE scheme that results from our conversion is IND-ID-CPA secure in the sense of [22].

A KEM formulation of our construction is also possible. In some ways this would be a more natural approach to take, since both ID-NIKD schemes and KEMs are concerned with keys, and one can generically obtain IBE from suitable KEMs using results of [15]. However, we are interested in exploring the relationship between

existing ID-NIKD and IBE schemes, and so have focussed here on an IBE formulation instead.

8.4.1 The Generic Conversion

We are now ready to present our generic conversion from ID-NIKD to IBE. Informally, the main idea of the conversion is that an encrypting party A can generate an "on-the-fly" key-pair (S, U) for each encryption to a party B. A uses the public parameters of the scheme, the identifier of the recipient B and the one-time private key S to compute a shared key, which is used to protect the message. Sending the one-time public key U as part of the ciphertext to B allows B to compute the same shared key and recover the message.

We require the ID-NIKD scheme to satisfy the following requirements:

1. The `Extract` and `SharedKey` algorithms of the ID-NIKD scheme should, as a first step, hash the input identifier id using a member $h = h_k$ of a one-way hash function family $(h_k)_{k\in\mathbb{N}}$ to produce an element U_{id} in some set $\mathcal{P}K$, with all further computations in the `Extract` and `SharedKey` algorithms depending only on U_{id} and not on id. We assume that h is described in the public parameters of the scheme. We may think of U_{id} as being the public key corresponding to the string id. If this condition is satisfied, then, from the algorithm `SharedKey` with inputs mpk, usk_{id_A}, id_B we can construct a new algorithm `SharedKey`$'$ with inputs $mpk, usk_{id_A}, U_{id_B} = h(id_B)$ that has the same output as `SharedKey`.

2. There should exist a randomized algorithm `Sample` that on input mpk, outputs pairs $(S, U) \in \{\mathcal{S}K\}\times\{\mathcal{P}K\}$ with S being a private key corresponding to public key U and $U \xleftarrow{\$} \{\mathcal{P}K\}$. Note that an identifier corresponding to U will not be obtainable from U when it is generated in this way without inverting the one-way hash function h.

These conditions are certainly satisfied for the SOK ID-NIKD scheme considered in Section 8.3.2. In this scheme, a hash function H_1 is first used to convert identifiers into group elements before any further processing. Modelling H_1 as a Random Oracle in the security analysis is a stronger assumption than H_1 being one-way. For the SOK scheme, we can define `Sample` as setting $b \xleftarrow{\$} \mathbb{Z}_p^*$, $S = g^{sb}$ and $U = g^b$. Here we see the need for including g and g^s in the public parameters of the ID-NIKD scheme.

Assuming these two conditions are met for an ID-NIKD scheme $\mathcal{S}$, we construct an IBE scheme $\mathcal{IBE}(\mathcal{S})$ as follows:

- `Setup`: On input 1^k, this algorithm runs the `Setup` of $\mathcal{S}$ (with the same input) to obtain mpk, msk. The master public key of $\mathcal{IBE}(\mathcal{S})$ is set to mpk and the master secret key is set to msk. We assume that mpk contains a description of the shared key space $\mathcal{SHK}$, and that $\mathcal{SHK} = \{0,1\}^n$; this will also define the message space of $\mathcal{IBE}(\mathcal{S})$.

- `Extract`: On input (mpk, msk, id), this algorithm runs the `Extract` algorithm of the ID-NIKD scheme $\mathcal{S}$ with the same input to obtain a private key usk_{id}. The output is usk_{id}.

- `Encrypt`: Let the input be $mpk, id, M \in \{0,1\}^n$. This algorithm runs `Sample` to obtain a pair (S, U). It then runs `SharedKey` of scheme $\mathcal{S}$ on input (mpk, S, id) to obtain a key $K \in \{0,1\}^n$. The output is $C = (U, V)$ where $V = M \oplus K$.

- `Decrypt`: Let the input be (mpk, usk_{id}, C) with $C = (U, V)$. This algorithm runs `SharedKey`$'$ (derived from `SharedKey` of $\mathcal{S}$) on input (mpk, usk_{id}, U) to obtain a key K. The output is $M = V \oplus K$.

Note that the `Setup` and `Extract` algorithms of the IBE scheme $\mathcal{IBE}(\mathcal{S})$ are almost identical to those of the ID-NIKD scheme $\mathcal{S}$. As discussed earlier, the main idea of the construction is that the encrypting party A can generate an "on-the-fly"

key-pair (S, U) for each encryption to B. Running $\texttt{SharedKey}(mpk, S, id_B)$ allows A to generate a shared key with B having identifier id_B, and sending the public key U to B as part of the ciphertext allows B to compute the same shared key by running $\texttt{SharedKey}'(mpk, usk_{id_B}, U)$. This shared key is then used to protect the message.

The scheme can be generalized to handle an ID-NIKD scheme whose shared key space $\mathcal{SHK}$ is any set equipped with a group operation; messages are then constrained to also lie in $\mathcal{SHK}$. Alternatively, the key K produced during `Encrypt` can be used to derive a key for a symmetric encryption scheme. We have the following security result:

Theorem 8.2. *Suppose the ID-NIKD scheme $\mathcal{S}$ is IND-SK secure and satisfies the two conditions above. Suppose also that $(h_k)_{k\in\mathbb{N}}$ used in the construction of $\mathcal{S}$ is a one-way function family. Then the IBE scheme $\mathcal{IBE}(\mathcal{S})$ is IND-ID-CPA secure. More precisely, for any adversary $\mathcal{A}$ against $\mathcal{IBE}(\mathcal{S})$, there are adversaries $\mathcal{B}_1$ against the one-wayness of (h_k) and $\mathcal{B}_2$ against the IND-SK security of $\mathcal{S}$ such that:*

$$Adv_{\mathcal{A}}^{\textit{IND-ID-CPA}}(k) \leq Adv_{\mathcal{B}_1}^{OW}(k) + 2 \cdot Adv_{\mathcal{B}_2}^{\textit{IND-SK}}(k).$$

Here, $\mathcal{B}_1$, $\mathcal{B}_2$ have running time roughly the same as $\mathcal{A}$.

Proof: Let $\mathcal{A}$ be an adversary against $\mathcal{IBE}(\mathcal{S})$ and let $\mathbf{Adv}_{\mathcal{A}}^{\text{IND-ID-CPA}}(k)$ denote its advantage. We construct from $\mathcal{A}$ two distinct algorithms $\mathcal{B}_1, \mathcal{B}_2$. Algorithm $\mathcal{B}_1$ attempts to break the one-wayness of hash function h, while algorithm $\mathcal{B}_2$ attempts to break the ID-NIKD scheme. The subsequent joint analysis of these two algorithms will then give us our result.

$\mathcal{B}_1$ receives from its challenger $\mathcal{C}_1$ a value $U \in \mathcal{PK}$ and uses $\mathcal{A}$ in an attempt to find a string $id' \in \{0,1\}^*$ such that $h(id') = U$. Clearly, if $\mathcal{B}_1$ is successful, then it breaks the one-wayness of h. $\mathcal{B}_1$ runs `Setup` of the scheme $\mathcal{IBE}(\mathcal{S})$, obtaining msk, mpk. Note that h is assumed to be described in mpk. Now $\mathcal{B}_1$ gives mpk to $\mathcal{A}$.

$\mathcal{B}_1$ handles $\mathcal{A}$'s **Extract** queries using its knowledge of msk to run algorithm `Extract` of $\mathcal{IBE}(\mathcal{S})$. When $\mathcal{A}$ submits challenge messages M_0, M_1 and a challenge identifier id^*, $\mathcal{B}_1$ sets $C^* = (U, V)$ where $V = M_b \oplus K$ and $K = \texttt{SharedKey}'(mpk, usk_{id^*}, U)$. Here, usk_{id^*} is obtained by running `Extract` and $b \xleftarrow{\$} \{0,1\}$. Eventually $\mathcal{A}$ outputs its bit b'. If at any point in the game, $\mathcal{A}$ makes a query to its **Extract** oracle on an identifier id satisfying $h(id) = U$, then $\mathcal{B}_1$ now outputs id. If id^* satisfies $h(id^*) = U$, then $\mathcal{B}_1$ outputs id^*. Otherwise, $\mathcal{B}_1$ fails when $\mathcal{A}$ outputs its bit. It is clear that the attack environment provided by $\mathcal{B}_1$ to $\mathcal{A}$ is indistinguishable from that provided by a real challenger. Moreover, if $\mathcal{A}$ does at any point make a query (either an **Extract** query or during the challenge phase) involving an identifier $id \in \{0,1\}^*$ such that $h(id) = U$, then $\mathcal{B}_1$ breaks the one-wayness of h.

$\mathcal{B}_2$ receives from its challenger $\mathcal{C}_2$ the public parameters mpk of the ID-NIKD scheme $\mathcal{S}$ and uses $\mathcal{A}$ in an attempt to win against $\mathcal{C}_2$ in the IND-SK security game for $\mathcal{S}$. Let d denote the hidden bit used by $\mathcal{C}_2$ in responding to $\mathcal{B}_2$'s **Test** query. $\mathcal{B}_2$ begins by selecting $id' \xleftarrow{\$} \{0,1\}^*$ and computing $U = h(id')$. $\mathcal{B}_2$ then passes mpk to $\mathcal{A}$. $\mathcal{A}$'s **Extract** queries on identifiers id are handled by $\mathcal{B}_2$ by passing them to $\mathcal{C}_2$ as **Extract** queries in the IND-SK game. However, if $h(id) = U$ for any of these queries, then $\mathcal{B}_2$ aborts. When $\mathcal{A}$ submits challenge messages M_0, M_1 and a challenge identifier id^*, $\mathcal{B}_2$ makes its **Test** query to $\mathcal{C}_2$ on input (id', id^*), receiving a value K in return. $\mathcal{B}_2$ then selects $b \xleftarrow{\$} \{0,1\}$ and sets $C^* = (U, V)$ where $V = M_b \oplus K$. However, if $h(id^*) = U$, then $\mathcal{B}_2$ aborts. Eventually $\mathcal{A}$ outputs its bit b'. If $b' = b$ then $\mathcal{B}_2$ outputs bit $d' = 0$; otherwise $\mathcal{B}_2$ outputs $d' = 1$. The attack environment provided by $\mathcal{B}_2$ to $\mathcal{A}$ is indistinguishable from that provided by a real challenger, provided that $\mathcal{B}_2$ does not abort. Moreover, when $\mathcal{B}_2$ does not abort, $\mathcal{B}_2$ makes only legal queries to its challenger $\mathcal{C}_2$ – we can be sure that id' is distinct from id^* and that id' is not involved in any **Extract** query. In this situation, $\mathcal{B}_2$'s advantage can,

via a standard argument, be expressed as:

$$\mathbf{Adv}_{\mathcal{B}_2}^{\text{IND-SK}}(k) = \frac{1}{2}\left|\Pr[d'=0|d=0] - \Pr[d'=0|d=1]\right|.$$

When $d = 1$, the key K returned by $\mathcal{C}_2$ to B_2 as a result of the **Test** query is random in $\mathcal{SHK} = \{0,1\}^n$, and then the ciphertext C^* received by $\mathcal{A}$ is independent of the bit b. Hence in this case, $\mathcal{A}$ has zero advantage and $\Pr[b = b'] = 1/2$. So $\Pr[d' = 0|d = 1] = 1/2$. On the other hand, when $d = 0$, the key K returned by $\mathcal{C}_2$ to $\mathcal{B}_2$ is equal to the shared key for identifiers id', id^*, and hence C^* is a proper encryption of M_b. Then $\Pr[d' = 0|d = 0] = \Pr[b = b'|\neg\mathcal{F}]$, where $\mathcal{F}$ denotes the event that $\mathcal{B}_2$ aborts and $\neg\mathcal{F}$ its complement. Combining this information, we obtain

$$\mathbf{Adv}_{\mathcal{B}_2}^{\text{IND-SK}}(k) = \frac{1}{2}\cdot\left|\Pr[b=b'|\neg\mathcal{F}] - 1/2\right|.$$

Notice that $\mathcal{A}$'s view is identical when playing against either $\mathcal{B}_1$ or $\mathcal{B}_2$, unless $\mathcal{B}_2$ aborts. So the occurrence of $\mathcal{F}$ is independent of which algorithm $\mathcal{A}$ plays against. Notice too that if $\mathcal{F}$ occurs, then $\mathcal{B}_1$ successfully inverts h. Then we have

$$\begin{aligned}
\mathbf{Adv}_{\mathcal{A}}^{\text{IND-ID-CPA}}(k) &= \left|\Pr[b=b'] - 1/2\right| \\
&= \left|\Pr[b=b'|\mathcal{F}]\cdot\Pr[\mathcal{F}] + \Pr[b=b'|\neg\mathcal{F}]\cdot(1-\Pr[\mathcal{F}]) - 1/2\right| \\
&\leq \Pr[\mathcal{F}] + \left|\Pr[b=b'|\neg\mathcal{F}] - 1/2\right| \\
&= \mathbf{Adv}_{\mathcal{B}_1}^{\text{OW}}(k) + 2\cdot\mathbf{Adv}_{\mathcal{B}_2}^{\text{IND-SK}}(k).
\end{aligned}$$

This, together with the observation that the running times of $\mathcal{B}_1$ and $\mathcal{B}_2$ are roughly the same as that of $\mathcal{A}$, completes the proof.

The above construction provides IND-ID-CPA security for the scheme $\mathcal{IBE}(\mathcal{S})$. In the ROM, we may apply the generic transform of [120] to obtain an IBE scheme with IND-ID-CCA security, provided $\mathcal{IBE}(\mathcal{S})$ satisfies a mild technical condition (γ-uniformity). The resulting scheme (using either conversion) will be only a little less efficient than $\mathcal{IBE}(\mathcal{S})$. In turn, $\mathcal{IBE}(\mathcal{S})$ has roughly the same performance

characteristics as the ID-NIKD scheme that it is built from. In fact, the generic transform of [120] only requires a one-way security notion for the starting IBE scheme in order to obtain IND-ID-CCA security. We can achieve this notion of security for our ID-NIKD-to-IBE conversion under the weaker requirement that the ID-NIKD scheme be OW-SK secure. This can allow a slightly more efficient overall construction for an IND-ID-CCA secure IBE scheme, since we can typically obtain OW-SK security using one less hash function than is needed for IND-SK security, and with a slightly tighter reduction. Moreover, it can be seen from the proof of Theorem 8.2 that no access to the **Reveal** oracle is needed during the simulation. This means that security in the weaker model for ID-NIKD originally proposed in [53] is actually sufficient for our application to IBE.

8.4.2 Applying the Conversion

We have already explained how the SOK ID-NIKD scheme meets the requirements for our conversion to be applicable. In this case, we obtain an IBE scheme with the following algorithms:

- `Setup`: On input 1^k, this algorithm runs `Setup` of the SOK scheme and outputs $mpk = (\mathbb{G}, \mathbb{G}_T, e, p, g, g^s, H_1, H_2, n)$ where $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_2 : \mathbb{G} \rightarrow \{0,1\}^n$ are hash functions. It also outputs $msk = s$.

- `Extract`: On input mpk, msk and identifier $id \in \{0,1\}^*$, this algorithm runs algorithm `Extract` of the SOK scheme and outputs a value $usk_{id} = H_1(id)^s$.

- `Encrypt`: On input mpk, identifier $id \in \{0,1\}^*$ and message m, the algorithm firstly sets $b \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $S = g^{sb}$ and $u = g^b$ (i.e. runs `Sample` corresponding to the SOK scheme). It then computes `SharedKey` of the SOK-NIKD scheme on inputs mpk, S and id to obtain a key $K = H_2(e(S, H_1(id)))$. Finally, it

computes $v = m \oplus K$. The algorithm returns the ciphertext $c = (u, v)$.

- `Decrypt`: On input mpk, a private key usk_{id} and a ciphertext $c = (u, v)$, this algorithm first computes `SharedKey` of the SOK-NIKD scheme on inputs mpk, usk_{id} and u to obtain a key K' and outputs $m = v \oplus K'$.

The IND-ID-CPA security of this IBE scheme is guaranteed by Theorems 8.1 and 8.2, assuming the hardness of the BDH problem in groups output by `PairingGen`.

It is easy to see that this IBE scheme is nothing other than the `BasicIdent` IBE scheme of Boneh and Franklin [22]. Thus our approach provides a "new" proof of security for this scheme (and since the IND-ID-CCA secure scheme FullIdent of [22] is effectively obtained via the later generic transform of [58], a proof for FullIdent too).

8.5 Concluding Remarks

In [96] the authors show how to build ID-NIKD schemes that are secure in the models presented in this chapter, from any Trapdoor Discrete Logarithm (TDL) group. In a TDL group, the discrete logarithm problem is easy if some trapdoor information is known. They apply the generic conversion presented in this chapter to obtain IBE schemes whose security is consequently not based on the hardness assumptions in pairing-friendly groups. In keeping with the theme of this thesis, we have only discussed pairing based schemes in this chapter. However, our conversion demonstrates how seemingly quite different IBE schemes can be seen as arising in a uniform way from a common underlying primitive, namely ID-NIKD. Specifically, it explains in a new way the relationship between the SOK ID-NIKD scheme and the IBE schemes of Boneh and Franklin [22]. It would also be interesting to explore if further known IBE schemes can be related to (presumably as yet unknown) ID-NIKD schemes.

Chapter 9

Concluding Remarks

The main focus of this thesis has been on considering IBE in the setting of multiple TAs. We provided motivations, formal definitions and security notions for such multi-TA IBE schemes. We concentrated on the scenario where multiple TAs share some common system parameters and considered how we could achieve the notion of TA Anonymity in this setting. We discussed appropriate multi-TA versions of well known IBE schemes in the Random Oracle Model and the Standard Model and studied their security properties. We also showed how the TA Anonymity security notion may have rather subtle implications for schemes that use IBE as a building block. We have already identified a number of avenues for future research in this thesis and list a few additional open problems here.

9.1 Robustness for Multi-TA IBE

Abdalla *et al.* [3, 4] define robustness as the property of a PKE scheme such that decrypting a ciphertext created with a public key, should fail when decrypting with a private key corresponding to another public key. In practice such a functionality is achieved using appropriate padding. However, there is nothing inherent in the regular security definitions to encompass such a functionality. To highlight that strong

security notions such as IND-CCA security do not imply robustness, the authors show in [3] that the Fujisaki-Okamoto transform and the CHK transform do not in general impart robustness (although they may for specific instantiations).

In the multi-TA IBE setting, we may require a similar additional robustness condition – decrypting a ciphertext created using an identity and the master public key of one TA should fail to decrypt using a private key for that (or any other) identity issued by another TA. This seems to be an interesting area meriting specific attention and we leave this for future work.

9.2 Multi-TA HIBE

HIBE schemes were introduced in Section 2.8.5. We stress that multi-TA IBE is different from HIBE. In a HIBE scheme, a single root TA generates public parameters and a master secret, using which the master secrets of all sub-TAs are produced. On the other hand, in the multi-TA IBE setting, there is no single root TA, but rather a group of independent TAs (who have unique master public keys and master secret keys, but may share some common system parameters)

In the same vein, as discussed in Section 3.2.4, Anonymous HIBE (AHIBE) [2, 27] is related to, but different from TA Anonymity for IBE. In AHIBE, ciphertexts are anonymous in that an adversary cannot distinguish which identity was used when producing a ciphertext, where now identities are comprised of a vector of strings identifying a hierarchy of TAs and a final user. By contrast, in the multi-TA IBE setting TA Anonymity captures the security property that ciphertexts do not leak the public parameters of the TA.

In [2] the authors give security definitions for AHIBE, parameterized by level. They note that the HIBE schemes of [63] and [20] are not anonymous, even at level 1. They present a modified construction of the HIBE scheme from [63] and show

that it is anonymous at level 1. Subsequently, an AHIBE scheme is presented in the Standard Model in [27] where ciphertexts are anonymous at all levels of the hierarchy.

The "right" generalization of the multi-TA IBE concept to the HIBE setting would involve multiple, independent root TAs, each being the root of a tree of TAs and users. Thus we would have a forest of trees, and would then wish to study anonymity properties of ciphertexts in this multi-HIBE setting. A multi-TA HIBE scheme can be defined in a manner similar to multi-TA IBE scheme, by having a `CommonSetup` algorithm output a set of parameters shared by (independent) root TAs. In such a multi-TA HIBE setting, a TA Anonymity notion would capture the requirement that ciphertexts, even for the same message and identity vector, but composed using the public parameters of different root TAs, do not leak the public parameters of the root TA.

We note that Canetti *et al.* [33] give a construction that builds an h-level HIBE scheme that is selective-id IND-CCA secure, from an $(h+1)$-level HIBE scheme that is selective-id IND-CPA secure, and a strongly secure one-time signature scheme. In fact the result in which they build a IND-CCA secure PKE scheme from a selective-id IND-CPA secure IBE scheme is a special case of this more general result (to see this, note that an IBE scheme is nothing other a 1-level HIBE scheme and that a PKE scheme can in some senses be viewed as a 0-level IBE scheme, i.e. a scheme where messages are encrypted to the root TA only). We note that with rigourous definitions and security notions for multi-TA HIBE schemes we should also be able to extend the more general result from [33] to the setting of multi-TA HIBE scheme. However, at present we do not know how to prove TA Anonymity for any HIBE scheme.

Bibliography

[1] Martín Abadi and Cédric Fournet. Private authentication. *Theor. Comput. Sci.*, 322(3):427–476, 2004.

[2] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Shoup [107], pages 205–222.

[3] Michel Abdalla, Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Robust public-key and identity-based encryption. *Unpublished manuscript*, 2008.

[4] Michel Abdalla, Mihir Bellare, and Gregory Neven. A provable-security treatment of robust encryption. Cryptology ePrint Archive, Report 2008/440, 2008. `http://eprint.iacr.org/`.

[5] Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana K. Smetters, Jessica Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *IEEE Symposium on Security and Privacy*, pages 180–196. IEEE Computer Society, 2003.

[6] Shane Balfe, Kent D. Boklan, Zev Klagsbrun, and Kenneth G. Paterson. Key refreshing in identity-based cryptography and its applications in MANETs. *IEEE Military Communications Conference, MILCOM*, 2007.

[7] Manuel Barbosa and Pooya Farshim. Efficient identity-based key encapsulation to multiple parties. In Smart [109], pages 428–441.

[8] Mihir Bellare. Practice-oriented provable security. In Ivan Damgård, editor, *Lectures on Data Security*, volume 1561 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1998.

[9] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer, 2001.

[10] Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. In Cachin and Camenisch [29], pages 171–188.

[11] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In *STOC*, pages 419–428, 1998.

[12] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Krawczyk [82], pages 26–45.

[13] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 1993.

[14] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, New York, NY, USA, 1993. ACM.

[15] Kamel Bentahar, Pooya Farshim, John Malone-Lee, and Nigel P. Smart. Generic constructions of identity-based and certificateless KEMs. *J. Cryptology*, 21(2):178–199, 2008.

[16] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby. Collisions of SHA-0 and reduced SHA-1. In Cramer [45], pages 36–57.

[17] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key agreement protocols and their security analysis. In Michael Darnell, editor, *IMA Int. Conf.*, volume 1355 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 1997.

[18] Kent D. Boklan, Zev Klagsbrun, Kenneth G. Paterson, and Sriramkrishnan Srinivasan. Flexible and secure communications in an identity-based coalition environment. *IEEE Military Communications Conference, MILCOM*, 2008.

[19] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In Cachin and Camenisch [29], pages 223–238.

[20] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Cramer [45], pages 440–456.

[21] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

[22] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[23] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *FOCS*, pages 647–657. IEEE Computer Society, 2007.

[24] Dan Boneh and Jonathan Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 87–103. Springer, 2005.

[25] Colin Boyd, Wenbo Mao, and Kenneth G. Paterson. Key agreement using statically keyed authenticators. In Markus Jakobsson, Moti Yung, and Jianying Zhou, editors, *ACNS*, volume 3089 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 2004.

[26] Xavier Boyen. The $\mathtt{BB}_1$ identity-based cryptosystem: A standard for encryption and key encapsulation. *Submission to IEEE P1363.3*, 2006. `http://grouper.ieee.org/groups/1363/IBC/submissions/Boyen-bb1_ieee.pdf`.

[27] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.

[28] Robert W. Bradshaw, Jason E. Holt, and Kent E. Seamons. Concealing complex policies with hidden credentials. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *ACM Conference on Computer and Communications Security*, pages 146–157. ACM, 2004.

[29] Christian Cachin and Jan Camenisch, editors. *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*. Springer, 2004.

[30] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Pfitzmann [97], pages 93–118.

[31] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[32] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2003.

[33] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Cachin and Camenisch [29], pages 207–222.

[34] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Pfitzmann [97], pages 453–474.

[35] D.W. Carman. New directions in sensor network key management. *International Journal of Distributed Sensor Networks*, 1(1):3–15, 2005.

[36] Claude Castelluccia, Stanislaw Jarecki, and Gene Tsudik. Secret handshakes from CA-oblivious encryption. In Pil Joong Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 293–307. Springer, 2004.

[37] Melissa Chase. Multi-authority attribute based encryption. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534. Springer, 2007.

[38] Sanjit Chatterjee and Palash Sarkar. Trading time for space: Towards an efficient ibe scheme with short(er) public parameters in the standard model. In Dongho Won and Seungjoo Kim, editors, *ICISC*, volume 3935 of *Lecture Notes in Computer Science*, pages 424–440. Springer, 2005.

[39] Liqun Chen, Z. Cheng, John Malone-Lee, and Nigel P. Smart. An efficient ID-KEM based on the Sakai-Kasahara key construction. Cryptology ePrint Archive, Report 2005/224, 2005. `http://eprint.iacr.org/`.

[40] Liqun Chen and Zhaohui Cheng. Security proof of Sakai-Kasahara's identity-based encryption scheme. In Smart [109], pages 442–459.

[41] Liqun Chen and Keith Harrison. Multiple trusted authorities in identifier based cryptography from pairings on elliptic curves. *Trusted Systems Laboratory, HP*, 2003.

[42] Liqun Chen, Keith Harrison, Andrew Moss, David Soldera, and Nigel P. Smart. Certification of public keys within an identity based system. In Agnes Hui Chan and Virgil D. Gligor, editors, *ISC*, volume 2433 of *Lecture Notes in Computer Science*, pages 322–333. Springer, 2002.

[43] Liqun Chen, Keith Harrison, David Soldera, and Nigel P. Smart. Applications of multiple trust authorities in pairing based cryptosystems. In George I. Davida, Yair Frankel, and Owen Rees, editors, *InfraSec*, volume 2437 of *Lecture Notes in Computer Science*, pages 260–275. Springer, 2002.

[44] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In Bahram Honary, editor, *IMA Int. Conf.*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.

[45] Ronald Cramer, editor. *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*. Springer, 2005.

[46] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Krawczyk [82], pages 13–25.

[47] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2004.

[48] Alexander W. Dent. Fundamental problems in provable security and cryptography. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1849):3215–3230, 2006.

[49] Alexander W. Dent. A brief history of provably-secure public-key encryption. In Serge Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pages 357–370. Springer, 2008.

[50] Alexander W. Dent and Chris J. Mitchell. *User's guide to cryptography and standards*. Artech House, 2005.

[51] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on information Theory*, 22(6):644–654, 1976.

[52] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *STOC*, pages 542–552. ACM, 1991.

[53] Régis Dupont and Andreas Enge. Provably secure non-interactive key distribution based on pairings. *Discrete Applied Mathematics*, 154(2):270–276, 2006.

[54] Shimon Even, Oded Goldreich, and Silvio Micali. On-line/off-line digital signatures. *J. Cryptology*, 9(1):35–67, 1996.

[55] International Organization for Standardization. Information processing systems – open systems interconnection – basic reference model – part 2: Security architecture. 1988.

[56] Marc P. C. Fossorier, Hideki Imai, Shu Lin, and Alain Poli, editors. *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 16th International Symposium, AAECC-16, Las Vegas, NV, USA, February 20-24, 2006, Proceedings*, volume 3857 of *Lecture Notes in Computer Science*. Springer, 2006.

[57] Gerhard Frey, Michael Müller, and Hans-Georg Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1719, 1999.

[58] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 1999.

[59] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999.

[60] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[61] Dan Geer. Risk management is where the money is. *Forum on Risks to the Public in Computers and Related Systems, ACM Committee on Computers and Public Policy, Risk-Forum Digest*, 20(6):12, 1998.

[62] Craig Gentry. Practical identity-based encryption without random oracles. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.

[63] Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In Yuliang Zheng, editor, *ASIACRYPT*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

[64] Oded Goldreich. On post-modern cryptography. Cryptology ePrint Archive, Report 2006/461, 2006. `http://eprint.iacr.org/`.

[65] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. *Electronic Colloquium on Computational Complexity*, 10(015), 2003.

[66] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[67] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul F. Syverson. Universal re-encryption for mixnets. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 163–178. Springer, 2004.

[68] Shai Halevi. A sufficient condition for key-privacy. Cryptology ePrint Archive, Report 2005/005, 2005. `http://eprint.iacr.org/`.

[69] I.N. Herstein. *Abstract algebra*. Prentice-Hall, 1996.

[70] Katrin Hoeper and Guang Gong. Key revocation for identity-based schemes in mobile ad hoc networks. In Thomas Kunz and S. S. Ravi, editors, *ADHOC-NOW*, volume 4104 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 2006.

[71] Jason E. Holt. Key privacy for identity based encryption. Cryptology ePrint Archive, Report 2006/120, 2006. `http://eprint.iacr.org/`.

[72] Jason E. Holt, Robert W. Bradshaw, Kent E. Seamons, and Hilarie K. Orman. Hidden credentials. In Sushil Jajodia, Pierangela Samarati, and Paul F. Syverson, editors, *WPES*, pages 1–8. ACM, 2003.

[73] Jason E. Holt and Kent E. Seamons. Reconciling CA-oblivious encryption, hidden credentials, OSBE, and secret handshakes. *Internet Security Research Lab Technical Report, Brigham Young University*, June 2006.

[74] Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer, 2002.

[75] Malika Izabachène and David Pointcheval. New anonymity notions for identity-based encryption. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *SCN*, volume 5229 of *Lecture Notes in Computer Science*, pages 375–391. Springer, 2008.

[76] David Kahn. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Simon & Schuster, 1967.

[77] Aram Khalili, Jonathan Katz, and William A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *SAINT Workshops*, pages 342–346. IEEE Computer Society, 2003.

[78] Takashi Kitagawa, Peng Yang, Goichiro Hanaoka, Rui Zhang, Hajime Watanabe, Kanta Matsuura, and Hideki Imai. Generic transforms to acquire CCA-security for identity based encryption: The cases of FOpkc and REACT. In

Lynn Margaret Batten and Reihaneh Safavi-Naini, editors, *ACISP*, volume 4058 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006.

[79] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, pages 203–209, 1987.

[80] Neal Koblitz and Alfred Menezes. Another look at "Provable Security". II. In Rana Barua and Tanja Lange, editors, *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 148–175. Springer, 2006.

[81] Neal Koblitz and Alfred Menezes. Another look at "Provable Security". *J. Cryptology*, 20(1):3–37, 2007.

[82] Hugo Krawczyk, editor. *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*. Springer, 1998.

[83] Leslie Lamport. Constructing digital signatures from a one-way function. Technical report, Technical Report CSL-98, SRI International, Palo Alto, 1979.

[84] Ninghui Li, Wenliang Du, and Dan Boneh. Oblivious signature-based envelope. *Distributed Computing*, 17(4):293–302, 2005.

[85] B.J. Matt. Toward hierarchical identity-based cryptography for tactical networks. *IEEE Military Communications Conference, MILCOM*, 2004.

[86] Ueli M. Maurer and Yacov Yacobi. Non-interactive public-key cryptography. In Donald W. Davies, editor, *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 498–507. Springer, 1991.

[87] Silvio Micali, Charles Rackoff, and Bob Sloan. The notion of security for probabilistic cryptosystems. *SIAM J. Comput.*, 17(2):412–426, 1988.

[88] Victor S. Miller. Use of elliptic curves in cryptography. In Hugh C. Williams, editor, *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.

[89] David Naccache. Secure and practical identity-based encryption. *IET*, 1(2):59–64, 2007.

[90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437. ACM, 1990.

[91] Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2002.

[92] Tatsuaki Okamoto. Cryptography based on bilinear maps. In Fossorier et al. [56], pages 35–50.

[93] Tatsuaki Okamoto. On pairing-based cryptosystems. In Phong Q. Nguyen, editor, *VIETCRYPT*, volume 4341 of *Lecture Notes in Computer Science*, pages 50–66. Springer, 2006.

[94] Kenneth G. Paterson and Sriramkrishnan Srinivasan. Security and anonymity of identity-based encryption with multiple trusted authorities. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 354–375. Springer, 2008.

[95] Kenneth G. Paterson and Sriramkrishnan Srinivasan. Building key-private public-key encryption schemes. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP*, volume 5594 of *Lecture Notes in Computer Science*, pages 276–292. Springer, 2009.

[96] Kenneth G. Paterson and Sriramkrishnan Srinivasan. On the relations between non-interactive key distribution, identity-based encryption and trapdoor discrete log groups. *Des. Codes Cryptography*, 52(2):219–241, 2009.

[97] Birgit Pfitzmann, editor. *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, volume 2045 of *Lecture Notes in Computer Science*. Springer, 2001.

[98] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[99] David Roberts, Gavin Lock, and Dinesh C. Verma. Holistan: A futuristic scenario for international coalition operations. *International Conference on Integration of Knowledge Intensive Multi-Agent Systems.*, pages 423–427, 2007.

[100] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2004.

[101] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. *The 2000 Sympoium on Cryptography and Information Security, Okinawa, Japan, January*, pages 26–28, 2000. (In Japanese).

[102] Ryuichi Sakai and Masao Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054, 2003. `http://eprint.iacr.org/`.

[103] Kazue Sako. An auction protocol which hides bids of losers. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 422–432. Springer, 2000.

[104] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[105] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[106] Victor Shoup. Why chosen ciphertext security matters. *IBM Research Report, RZ 3076*, Nov 1998. `http://www.zurich.ibm.com/security/ace/expo.pdf`.

[107] Victor Shoup, editor. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.

[108] Michael Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1996.

[109] Nigel P. Smart, editor. *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*, volume 3796 of *Lecture Notes in Computer Science*. Springer, 2005.

[110] John Talbot and Dominic Welsh. *Cryptography and Complexity*. Cambridge University Press, Cambridge, 2005.

[111] Yuh-Min Tseng and Jinn-ke Jan. ID-based cryptographic schemes using a non-interactive public-key distribution system. In *ACSAC*, pages 237–243. IEEE Computer Society, 1998.

[112] Shengbao Wang and Zhenfu Cao. Practical identity-based encryption (IBE) in multiple PKG environments and its applications. Cryptology ePrint Archive, Report 2007/100, 2007. `http://eprint.iacr.org/`.

[113] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the hash functions MD4 and RIPEMD. In Cramer [45], pages 1–18.

[114] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Shoup [107], pages 17–36.

[115] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Cramer [45], pages 19–35.

[116] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Efficient collision search attacks on SHA-0. In Shoup [107], pages 1–16.

[117] Brent Waters. Efficient identity-based encryption without random oracles. In Cramer [45], pages 114–127.

[118] Brent R. Waters, Edward W. Felten, and Amit Sahai. Receiver anonymity via incomparable public keys. In Sushil Jajodia, Vijayalakshmi Atluri, and Trent

Jaeger, editors, *ACM Conference on Computer and Communications Security*, pages 112–121. ACM, 2003.

[119] Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[120] Peng Yang, Takashi Kitagawa, Goichiro Hanaoka, Rui Zhang, Kanta Matsuura, and Hideki Imai. Applying Fujisaki-Okamoto to identity-based encryption. In Fossorier et al. [56], pages 183–192.