View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by DCU Online Research Access Service

Privacy Enhanced Protocols using Pairing Based Cryptography

Caroline Sheedy

Bachelor of Science in Mathematical Science

Master of Science in Security and Forensic Computing

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisor: Dr. Stephen Blott

January, 2010

I, Caroline Sheedy, hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Candidate) ID No.:

Date:

Contents

Abstract

This thesis presents privacy enhanced cryptographic constructions, consisting of formal definitions, algorithms and motivating applications. The contributions are a step towards the development of cryptosystems which, from the design phase, incorporate privacy as a primary goal. Privacy offers a form of protection over personal and other sensitive data to individuals, and has been the subject of much study in recent years.

Our constructions are based on a special type of algebraic group called bilinear groups. We present existing cryptographic constructions which use bilinear pairings, namely Identity-Based Encryption (IBE). We define a desirable property of digital signatures, *blindness*, and present new IBE constructions which incorporate this property.

Blindness is a desirable feature from a privacy perspective as it allows an individual to obscure elements such as personal details in the data it presents to a third party. In IBE, blinding focuses on obscuring elements of the identity string which an individual presents to the key generation centre. This protects an individual's privacy in a direct manner by allowing her to blind sensitive elements of the identity string and also prevents a key generation centre from subsequently producing decryption keys using her full identity string. Using blinding techniques, the key generation centre does not learn the full identity string.

In this thesis, we study selected provably-secure cryptographic constructions. Our contribution is to reconsider the design of such constructions with a view to incorporating privacy. We present the new, privacy-enhanced cryptographic protocols using these constructions as primitives. We refine useful existing security notions and present feasible security definitions and proofs for these constructions.

Acknowledgements

First of all, I would like to thank my supervisor Dr. Stephen Blott for giving me the opportunity to undertake a Ph.D. I'd also like to thank my collaborators through the years, beginning with Dr. Steven Galbraith, Dr. Colm Ó hÉigeartaigh and Dr. Ponnurangam Kumaraguru. The main body of this work is a result of collaborations with Dr. Markulf Kohlweiss, Alfredo Rial Duran and Dr. Jan Camenisch, and Dr. David Gray. Thanks to Markulf for an enjoyable time spent as a visiting researcher in K.U. Leuven, as well as the opportunity to work with a genuinely inquisitive mind, and to Jan and Alfredo for the interesting discussions and work. Particular thanks is due to Dr. David Gray, whose interest in my work has made all the difference. I am extremely grateful to him for this, and for the opportunity he has afforded me to continue in the field.

Thanks to my examiners, Prof. Mike Scott and Prof. Kenny Patterson for their time and energy in directing me towards the final version of this thesis.

I would like to say a big cheers to the good people of School of Computing CAPG. From my original research chum Gavin and the lunchtime crew of Sara, Riona, the two Claires, Bert, Hego, Mark, Niall, Ronan, Neil, *et al.* to newer members in later years, this body of people offer a great source of support, understanding and general fun.

And now, to the most important people of all, my family. The Ph.D experience has taught me that tenacity, as it is politely called, or stubbornness as it is commonly known, is a strong trait of mine. My siblings, Mark, David and Susan have obviously been aware of this all along, as their support more often than not took the form of bafflement when I would express my concerns that I would fail to complete something I had started. My parents, Laura and John have always supported me fully. From my first read-it-yourself books to hours spent with my Dad over pints and at the kitchen table with my Mam over tea discussing the various frustrations and joys of the last few years, they have always been the best of parents.

Finally, thanks seems beyond inadequate to express the debt of gratitude I owe to my other half, Brian. He more than any other person has contributed to my time as a student being not only academically rewarding, but personally joyful. Having completed this venture side by side, I can't wait for our newest little adventure to arrive.

To my Nanny and Grandad;

I wish you could be here to see the blue book.

List of Figures

Chapter 1

Introduction

Electronic information technologies make the collection, storage and sharing of data a simpler task than previous paper based systems. It is feasible to store a myriad of information digitally, including photographs, certificates, signatures (both hand-written and digital) and other potentially sensitive data. Using electronic communications, collecting information is reduced to choosing what data to store. As the cost of storing data continues to drop, it is possible to store all collected data, and address the issue of processing that data at a later stage. Data stored electronically can be easily disseminated.

In practice, this means that it is possible to automate the collection, storage and sharing of all forms of electronic data. Certain data is more valuable than others. It can be used to profile a user, identify personal interests or leanings, identify financial status or for a host of other purposes. An individuals' right to have their privacy and private life respected emanates from many sources of law. While not explicitly enumerated in the Constitution of Ireland, the Supreme Court considers it a fundamental right:

> 'The right to privacy is one of the fundamental personal rights of the citizen which flow from the Christian and democratic nature of the State.... The nature of the right to privacy is such that it must ensure the dignity and freedom of the individual in a democratic society. This can not be insured if his private communications, whether written or telephonic, are deliberately and unjustifiably interfered with' [Gea87]

Article 8 of the European Convention on Human Rights [JWO75] specifies "Everyone has the right to respect for his private and family life, his home and his correspondence". The right is similarly espoused in Article 12 of the United Nations Declaration of Human Rights [Nat08], and

in many other sources of law internationally.

Additionally, organisations must be able to retain the trust of the individuals whose data they store and process by employing best-practice techniques when handling such data.

Privacy preserving technologies aim to protect a user's privacy preferences when they communicate their data electronically to third parties. Such technologies should facilitate the third party collecting and storing the data in a responsible and safe manner, while allowing them to access pertinent information on the user. If it is necessary for the third party to distribute the data, a privacy preserving technology should ensure this is done in accordance with the individual's stated privacy preferences and rights.

Cryptography has numerous aims in relation to information security, including confidentiality, data integrity, entity and data authentication and secret communication over an insecure channel. In the case of secret communication, the two communicating parties want to exchange messages, while preventing a listening third party from learning anything about the content of the messages. An encryption scheme, viewed at its most simplistic, is a triple of algorithms that facilitates secret communication. The first algorithm, *key generation*, produces a pair of encryption/decryption keys. The second algorithm, *encryption*, is applied by the sender to the message to create a *ciphertext*. The third algorithm, *decryption*, is applied by the recipient to the ciphertext to retrieve the message. In private-key (symmetric) cryptography, the encryption key is the same as the decryption key, which means the encryption key must be kept secret. Hereafter, we consider public-key (asymmetric) encryption schemes where the encryption algorithm is executed using a *public-key* and the decryption algorithm is executed using the corresponding *private-key*. In public-key cryptography, only the private key must be kept secret.

Cryptographic primitives have four basic goals [MVOV97]:

Confidentiality hides information from unauthorised users.

Integrity ensures any tampering with a message in transit is detected. Prevention of tampering is not generally possible, but it should always be detected.

Authentication ensures a message has originated from the entity it appears to have originated from and entity authentication ensures that an entity is who they claim to be.

Non-repudiation ensures an entity cannot deny his actions at a later stage.

Cryptographic primitives can also be used towards other objectives such as digital cash [Cha82]

and searchable encryption [ABC+08, BDCOP04, WBDS04]. We synthesise concepts from existing cryptographic primitives with identity-based encryption schemes to form new schemes that contribute to our goal of privacy. In identity-based encryption schemes, we focus on the key generation centre, and the level of trust that is required in it. By working towards the introduction of privacy from the key generation centre, our aim is to reduce the level of trust required in it by the key requesting user.

1.1 Motivation

Prior to the introduction of public-key (asymmetric-key) cryptography [DH76], sharing a common key for encryption and decryption (symmetric-key cryptography) was standard. Alice and Bob were required to agree upon a common key, perhaps by meeting in person, to protect their communications. This practice changed when Diffe and Hellman [DH76] proposed using *key pairs* consisting of a *public key* and a *private key*.

A *public key* is issued for Alice, who receives the corresponding *private key*. The public key is circulated, and Bob uses it to encrypt a message to Alice. Bob sends Alice the resultant *ciphertext*. Alice uses her private key to decrypt the ciphertext and retrieve the message. For this to work, the keys must be related in some way. It must also be infeasible to compute the private key from the corresponding public key. Examples of commonly used public-key cryptosystems include RSA [RSA78], ElGamal [Elg85] and Paillier [Pai99].

Public-key cryptography has many advantages over symmetric-key cryptography [MVOV97]. The most obvious advantage is that only the private key is required to be kept secret. The public key is made freely available, meaning Bob no longer needs to have an established relationship with Alice to send her encrypted messages. The public keys must be guaranteed authentic; that is, it should not be possible for an adversary to publish a public key and claim that it is Alice's key. This is necessary to prevent the adversary from being able to intercept and decrypt messages that Bob believes he has encrypted to Alice. Without any form of public key authentication, an adversary can claim to be Alice, trick Bob into encrypting messages for her using the impostor public key and use the corresponding private key to decrypt.

Public-key infrastructures (PKI) evolved to counter such attacks. In PKI, public keys are administered by a trusted third party, more commonly known as the *certificate authority* (CA). The CA is responsible for certifying a key-pair. A user is bound to the key-pair. This binding is carried

out by a *registration authority* (RA). A user is tied in an unforgeable manner to her identity, validated attributes, the associated public key, and other conditions by a *public-key certificate*. This infrastructure allows users who have had no prior contact to be authenticated to one another. However, PKI systems have practical limitations. These include the variability of the CA's conditions for certificate issuing, the propagation of certificate revocation and the complexities of the X.509 standard [HPFS02]. A major disadvantage of PKI schemes is the difficulty of finding the correct certificate corresponding to an individual.

Shamir [Sha85] proposed *identity-based encryption* (IBE) as an alternative to traditional public-key cryptography schemes. An identity-based scheme allows the public key of a receiver to be a known identity string. Identity strings can take the form of an e-mail address, a phone number or any set of terms or conditions such as a privacy policy or biometric data [SW05, BM05, BDMN06, SK08]. Identity-based schemes avoid the PKI issue of finding and authenticating a public key. The public key can be any known and available string. The need for an infrastructure to authenticate the public keys and provide assurances on their validity is thereby removed. Key revocation can be handled by including a validity period in the identity string. Administering user credentials is straightforward. Credentials such as security clearance, role or contract length can be added to the identity string. Only a receiver with the appropriate credentials can retrieve the private key and thus decrypt a ciphertext to retrieve the message.

Despite the advantages of identity-based schemes over PKI outlined above, they have not been embraced as the de facto choice for encryption. This can be attributed in part to the private key generation process. A user that wants to retrieve her private key presents her identity string to the key generating centre. This key generation centre has to be trusted by the user. Once a key generation centre knows an identity string, it is trivial for it to generate additional private keys. These extra keys may be held by the centre, or maliciously distributed to other parties, and subsequently used to decrypt ciphertexts intended for the user.

There are other contributing factors in the non adoption of identity-based schemes. The area of key revocation can require revoking a person's identity. It is not always possible or convenient to revoke an identifier, for example if biometric data has been used or an email address. Ensuring the authenticity of a claim to a particular identity is also a challenge.

Identity-Based Encryption Schemes

In an IBE scheme, the sender encrypts a message for the receiver using the identity string id and public parameters $params$. A trusted third party, the key generation centre ($\mathcal{KGC}$), generates the user's secret key sk_{id}. An IBE system consists of four polynomial time algorithms: Setup, Extract, Encrypt and Decrypt (see Figure 1.1).

Setup**:** Given as input a security parameter k, returns the public system parameters $params$ and a master secret msk.

Extract**:** Given as input $params$, msk and an identity id, returns the private key sk_{id} for id. The public key is the identity string id and the corresponding private key is sk_{id}.

Encrypt**:** Given as input $params$, id and a message m, returns a ciphertext ct.

Decrypt**:** Given as input $params$, ct and sk_{id}, returns m under the correctness condition for encryption that the decryption of an encryption under the correct key returns the message encrypted.

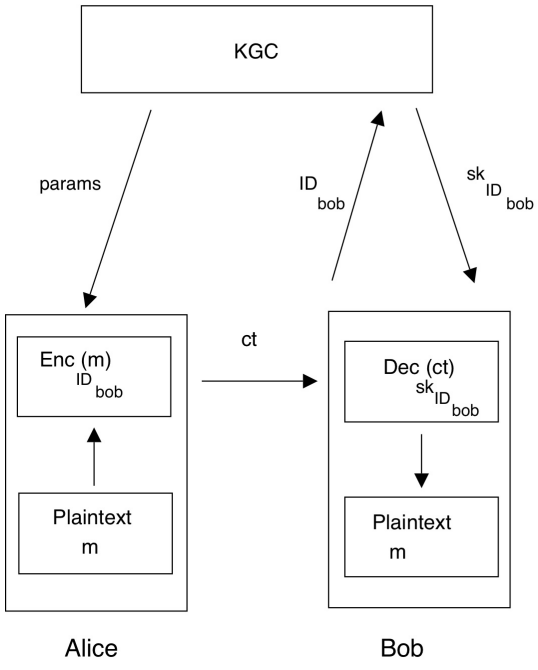

Figure 1.1: Identity-Based Encryption

IBE schemes can also be used to create *digital signature schemes*.

Signature Schemes

A digital signature scheme provides a method for verifying that a message has been approved by a specified party. Verification of the signature can be performed by anyone with access to the resulting digital signature, the public key of the signer and the message. Digital signatures aim to provide a digital equivalent of a hand-written signature. Practical digital signature schemes must make signatures efficient to generate, efficient to verify and make it infeasible to forge the signature of another user. A digital signature scheme consists of three algorithms: KeyGen, Sign and Verify.

KeyGen**:** Given as input a security parameter k, returns the private key sk used by an individual for signing messages and the public key pk used by others to verify the resulting signatures.

Sign**:** Given as input a message to be signed m and the private key sk of the signer, returns σ, the signature of m. Typically, this algorithm is deterministic.

Verify**:** Given as input the public key of the signer, the message m and the signature σ, $\mathsf{Verify}_{sk}(\sigma, m) = 1$ under the correctness condition for signatures that the verification of a signature under the correct key returns true.

Identity-Based Encryption implies Signatures

In Boneh and Franklin [BF01], the authors communicate Naor's observation that an IBE scheme can be converted into a public key signature scheme. In the resulting signature scheme, the private key sk is the master secret key msk for the IBE scheme, the public key pk is the global system parameters $params$ of the IBE scheme and the message m to be signed is the identity string id. The signature σ on the message m is the IBE private key corresponding to the identity string sk_{id}. To verify a signature, choose a random m' and encrypt m' using the public key of the IBE scheme id. Then attempt to decrypt using the signature σ. If the decryption occurs correctly, the signature is valid. Otherwise, reject the signature. This approach provides a randomised signature verification algorithm, which is unlike most signature schemes.

1.2 Summary and Main Contributions

In this thesis, we focus on identity-based encryption and examine incorporating privacy preserving properties. We distil privacy preserving properties of cryptographic primitives and protocols, and

combine them to present novel schemes.

The text of this thesis is subdivided into two main parts. The first part focuses on the construction of identity-based encryption schemes. We provide constructions that have the privacy enhancing property of *blindness*. These schemes are accompanied by relevant security definitions and proofs. The second part of this work treats the resulting blinded identity-based encryption schemes as cryptographic primitives and motivates application scenarios.

The results on constructing blind identity-based encryption schemes are joint work with Camenisch, Kohlweiss and Rial [CKRS09], and with Gray [SG09]. The results on applications of the schemes are joint work with Camenisch, Kohlweiss and Rial [CKRS09], and with Gray [SG09].

1.2.1 Blind Identity-Based Encryption Schemes and Extensions

The concept. Identity-based encryption schemes are limited by the level of trust required in the key generation centre. The key generation centre has the ability to decrypt or sign a message without authorisation if it learns the relevant identity string. Traditional schemes necessitate the requesting user presenting the key generation centre with the identity string. Thus, the key generation centre can create an *identity list* of all requested identity strings.

In Chapter 3, we propose reducing the level of trust required in the key generation entity by limiting the identity strings it is privy to. The approach taken is to merge the established property of *blindness* with existing identity-based encryption schemes to achieve a privacy enhanced identity-based encryption scheme.

Our contribution. The first blind identity-based encryption scheme is by Green and Hohenberger [GH07]. This scheme is selective-identity, chosen-plaintext secure in the standard model, meaning an adversary has to commit to the identity ahead of time and has access to encryptions, but is not anonymous. We present a blind identity-based encryption scheme that is adaptive-identity, chosen-ciphertext secure in the standard model, meaning an adversary can adaptively choose their identity and has conditional access to decryptions, as well as being anonymous in Section 3.4. An anonymous identity-based encryption scheme has the desirable property that the ciphertext does not leak any information about the identity used to create the ciphertext. Relevant definitions, as well as security proofs, are provided.

We extend the use of the blinding property to construct the first *partially-blind* construction of an identity-based encryption scheme in Section 3.5. Partially-blind IBE allows some elements of

the identity string be visible to the key generation centre, while the keeping the remaining elements obscured. We introduce the novel concept of *double-blind* key extraction in Section 3.6. Double-blind IBE allows some elements of the identity string be visible to the key generation centre, some to remain obscured from the key generation centre, and allows the key generation centre to introduce some elements to the identity string that remain obscured from the user. We construct an identity-based encryption scheme with this property. We provide security definitions and proofs for both schemes [SG09]. Finally, we present a transformation for our partially-blind and double-blind key extraction protocols that allows them to be used with our anonymous identity-based encryption scheme in Section 3.7.

1.2.2 Resulting Privacy Enhanced Protocols

The concept. We examine applications of blind signatures and identity-based encryption schemes to ascertain whether they can be improved using blind identity-based encryption. A scheme is considered improved if it affords a user a greater level of privacy over her identity string than was the case previously. By preventing her full or partial identity string being leaked to the key generation entity, the risk to the user of profiling, impersonation and other malicious behaviour by the key generation entity is reduced.

Our contributions. We present a variant of public-key encryption with keyword search in Section 4.3. This allows a user to provide a keyword and obtain a search result with the novel property that the keyword is not revealed. We call this public-key encryption with oblivious keyword search. We then proceed to construct an architecture for anonymous key issuing in Section 4.4, employing an integral property of blind identity-based encryption. It provides assurances to the key generation entity that the user is requesting an authorised and valid key. Finally, we describe a unique receipt issuing protocol that can be used to build online lotteries in Section 4.5.

1.3 Overview

The remainder of this thesis is structured as follows. We begin Chapter 2 with the requisite notation. We then provide an overview of cryptographic preliminaries, incorporating bilinear pairings, computational assumptions and cryptographic primitives. We explain relevant concepts of provable security. We describe identity-based encryption and present relevant related schemes.

Chapter 3 introduces the property of blinding, and its application to identity-based encryption schemes. Our results on the construction of a variety of such schemes are presented, along with the necessary security definitions and proofs.

Our applications of blind identity-based encryption are presented in Chapter 4. We examine some existing cryptosystems in detail, namely public-key encrypted keyword search and anonymous key issuing. We present new constructions of each using the novel blind identity-based encryption schemes detailed in Chapter 3. We conclude with a simple motivating application for double-blind identity-based encryption.

Chapter 5 concludes the thesis by providing a brief summary of our contributions. We also discuss directions for future work.

Chapter 2

Preliminaries

This chapter provides an overview of the foundations required for the remainder of this work and an introduction to the area of identity-based encryption. We begin with technical preliminaries, including the mathematical preliminaries and computational assumptions used in our constructions. We do not provide a comprehensive discussion of the foundations of cryptography, which can be found in detail in the *Handbook of Applied Cryptography* [MVOV97] and *The Foundations of Cryptography* [Gol01].

Section 2.1 begins with the relevant notation. Section 2.2 presents an overview of pairings. Section 2.3 introduces the requisite complexity assumptions and the foundations for the cryptographic constructs presented. Section 2.4 provides a brief discussion of cryptographic primitives. Section 2.5 details the security notions associated with encryption schemes. Finally, Section 2.6 surveys identity-based encryption schemes, and presents schemes relevant to our contribution.

2.1 Notation

Let $\{0,1\}$ be the set of individual bits, where $\{0,1\}^*$ denotes the space of finite binary strings and $\{0,1\}^\infty$ denotes the space of infinite binary strings. Let $k \in \mathbb{N}$. 1^k is the bit string of k ones and $\{0,1\}^k$ is the set of bit strings of length k, Strings are finite unless stated otherwise. The concatenation of strings a and b is denoted by $a|b$, or ab. The length of string a is given by $|a|$. The empty string is the string of length 0 and is denoted by $[]$.

A *linear time* algorithm is one in which the measure of computation $m(n)$, for example, execution time or memory space, is bounded by a linear function of the problem size n, i.e., $m(n)$ is $O(n)$. A *polynomial time* algorithm is one in which the measure of computation $m(n)$ is bounded

by a polynomial function of the problem size n, i.e., $m(n)$ is $O(n^k)$ for some k. An *exponential time* algorithm is one in which the measure of computation $m(n)$ is bounded by an exponential function of the problem size n, i.e., $m(n)$ is $O(c^n)$ where $c > 1$.

A function ν is *negligible* if, for every integer c there exists an integer K such that for all $k > K$, $|\nu(k)| < 1/k^c$. A function is said to be *non-negligible* if it is not negligible. A problem is said to be *hard* (or *intractable)* if there exists no algorithm whose running time is polynomial in the size of the input. Assigning a value a to a variable x is given by $x \leftarrow a$. For a non empty set A , $x \leftarrow A$ denotes x is a variable which has been chosen uniformly from A.

We write $P(\mathcal{A}(a), \mathcal{B}(b)) \rightarrow (c, d)$ to indicate that P is a protocol between parties $\mathcal{A}$ and $\mathcal{B}$, where a is $\mathcal{A}$'s input, b is $\mathcal{B}$'s input, c is $\mathcal{A}$'s output and d is $\mathcal{B}$'s output.

An *efficient computation* is polynomial-time in the security parameter. The polynomial bounding the running time is fixed, explicit and usually small. A *feasible computation* is also polynomial-time, but the polynomial is not specified a-priori. Thus, this polynomial is considered as arbitrarily large. An *infeasible computation* is anything beyond the class of polynomial time computations. An event that occurs with *noticeable probability* will almost surely (i.e., except with negligible probability) occur if the experiment is repeated a polynomial number of times.

2.2 Bilinear Pairings

Elliptic curve cryptosystems (ECC) have the advantage of shorter key size than their public-key counterparts [Oka06b]. In elliptic curve cryptography, bilinear pairings are functions that map a pair of elliptic curve points to an element of the multiplicative group of a finite field [Bla05]. Bilinear pairings have limitations that should be taken into account when designing cryptographic systems using pairings [GPS08]. For example, it is not possible to hash efficiently onto all groups, and a given pairing may not be efficiently computable.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two finite multiplicative abelian groups groups of prime order q. A bilinear map is called symmetric if $\mathbb{G}_1 = \mathbb{G}_2$, and asymmetric otherwise. $\mathbb{G}_3$ is a cyclic group of order q. A pairing is a function

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$$

satisfying the following properties:

Bilinearity: Given groups of prime order q and two generators $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$, $e(g^a, h^b) =$

$e(g,h)^{ab}$.

Non-degeneracy: $\forall g \in \mathbb{G}_1, g \neq 1$, and $h \in \mathbb{G}_2, h \neq 1$, $e(g,h) \neq 1$.

Efficiency: e should be easily computable. This property is required to ensure the map can be used to construct cryptographic schemes.

2.2.1 Construction of Bilinear Pairings

Bilinear maps are computed using either the Weil or Tate pairing over particular types of elliptic curves using Miller's algorithm. Miller's unpublished manuscript of 1986, which was later published in 2004 [Mil04], describes the efficient implementation of the Weil pairing using a double-and-add algorithm. In pairing-based cryptosystems, the computation of the Weil or Tate pairing is generally the most computationally intensive aspect of the system. Thus, it is has been the focus of a large volume of research in recent years.

In work with Stephen D. Galbraith and Colm Ó hÉigeartaigh [GÓhS07], we present one attempt to reduce the computational complexity of Miller's algorithm for the Tate pairing. By using a distortion map to include vertical line functions, it is possible to compute the Tate pairing without a final exponentiation step. This result does not improve the efficiency of the algorithm, but nevertheless is the first construction which avoids a final exponentiation for the Tate pairing.

Without the final exponentiation for specific curves, the pairing computation is simplified. This causes one to question whether the difficulty of the pairing inversion problem is affected for such curves.

Definition 1 (Generalized Pairing Inversion (GPI) Problem [Sat07]) *For given* $z \in \mathbb{G}_3$ *, find* $a \in \mathbb{G}_1$ *and* $b \in \mathbb{G}_2$ *such that* $e(a,b) = z$ *.*

This work [GÓhS07] further supports the belief that the pairing inversion problem is a hard computational problem as it presents several potential attacks on the pairing inversion problem, none of which lead to a practical attack.

2.2.2 Application of Pairings to Tripartite Key Agreement

It is important to note that bilinear pairings have been used to construct non-identity based cryptographic schemes. The "Tripartite Key Agreement" protocol proposed by Joux [Jou00] shows that it is possible to achieve key agreement between three parties in one round using bilinear pairings.

This is a marked improvement on the Diffe-Hellman [DH76] protocol that achieves key agreement between three participants in two rounds. It answers the outstanding question as to whether it is possible to achieve key agreement between three parties in only one round.

Suppose that Alice, Bob and Carol want to agree a key, and have public/private pairs $(g^a, a), (g^b, b)$ and (g^c, c) respectively where $a, b, c \in \mathbb{Z}_q^*$ are chosen at random and g is a generator of $\mathbb{G}_1$, thus $g^a, g^b, g^c \in \mathbb{G}_1$. To achieve key agreement in one round, Alice, Bob and Carol compute the pairings $e(g^b, g^c)^a$, $e(g^a, g^c)^b$ and $e(g^b, g^c)^a$. It is easily observable that $e(g^b, g^c)^a = e(g^a, g^c)^b = e(g^a, g^b)^c = e(g, g)^{abc}$.

This protocol is only resistant to passive attacks, that is, attacks in which the adversary cannot interact with any of the involved parties and is dependent solely on observed data. As with the basic Diffie-Hellman protocol, Joux's protocol does not authenticate the three communicating parties and is susceptible to the man-in-the-middle attack. It is possible to make it resistant to active attacks, that is, attacks in which the adversary attempts to add, delete, or otherwise alter the communication. Al-Riyami and Paterson [ARP03] presented several protocols which assure authenticity through use of certificates issued by a Certificate Authority (CA).

2.3 Computational Assumptions

Let $\mathbb{G}_1$, $\mathbb{G}_3$ be two groups of prime order q. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$ be an admissible bilinear map, and let g be a generator of $\mathbb{G}_1$. We present several computational assumptions on bilinear pairings. These computational assumptions underlie many pairing-based cryptosystems.

Bilinear Diffie-Hellman Assumption (BDH)

Given a tuple $g, g^a, g^b, g^c \in \mathbb{G}_1$ as input, output $= e(g, g)^{abc} \in \mathbb{G}_3$. The advantage of adversary $\mathcal{A}$ in solving the BDH problem is

$$\mathbb{P}\left[\mathcal{A}(g, g^a, g^b, g^c) = e(g, g)^{abc}\right].$$

Similarly, the advantage of adversary $\mathcal{A}$ for the Decisional BDH (DBDH) problem is

$$\mathbb{P}\left[\mathcal{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 0\right] - \mathbb{P}\left[\mathcal{A}(g, g^a, g^b, g^c, T) = 0\right]$$

where T is randomly selected from $\mathbb{G}_3$.

Assumption 1 *The (Decisional) BDH assumption is said to hold in $\mathbb{G}_1$ if any probabilistic polynomial time adversary has negligible advantage for the (Decisional) BDH problem.*

Decisional Linear (D-Linear)

Given a tuple $g, g^a, g^b, g^{ac}, g^{bd} \in \mathbb{G}_1$, $h, h^a, h^b \in \mathbb{G}_3$ and $Z \in \mathbb{G}_3$ for random exponents $a, b, c \in \mathbb{Z}_q$ as input, decide whether $Z = g^{c+d}$ or a random element in $\mathbb{G}_3$.

Assumption 2 *The Decisional Linear assumption is said to hold if any probabilistic polynomial time adversary has negligible advantage for the Decisional Linear problem.*

Computational Diffie-Hellman Assumption (CDH)

Given a tuple $g, g^a, g^b \in \mathbb{G}_1$, compute $Z = g^{ab}$. The CDH problem in $\mathbb{G}_1$ can be hard under certain mappings $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$ even though the DDH in $\mathbb{G}_1$ is easy and it may be easy in $\mathbb{G}_3$.

Assumption 3 *The CDH assumption is said to hold if any probabilistic polynomial time adversary has negligible advantage for the CDH problem.*

Bilinear Diffie-Hellman Inversion Assumption (BDHI)

Given the $(q+1)$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}) \in \mathbb{G}_1^{q+1}$ as input, compute $e(g,g)^{\frac{1}{x}} \in \mathbb{G}_3$. An adversary $\mathcal{A}$ has advantage in solving q-BDHI of

$$\mathbb{P}\left[\mathcal{A}(g, g^x, \ldots, g^{x^q}) = e(g,g)^{\frac{1}{x}}\right].$$

Similarly, an adversary $\mathcal{A}$ for the Decisional q-BDHI (q-DBDHI) problem has advantage in solving q-DBDHI of

$$\mathbb{P}\left[\mathcal{A}(g, g^x, \ldots, g^{x^q}, e(g,g)^{\frac{1}{x}}) = 0\right] - \mathbb{P}\left[(g, g^x, \ldots, g^{x^q}, T) = 0\right]$$

where T is randomly selected from $\mathbb{G}_3$.

Assumption 4 *The BDHI assumption holds if any probabilistic polynomial time adversary has negligible advantage in solving the $q-$BDHI problem.*

Strong Diffie-Hellman Assumption (SDH)

Given the $q+1$-tuple $(g, g^x, g^{x^2}, \ldots, g^{x^q}) \in \mathbb{G}_1^{q+1}$ as input, compute $(g^{\frac{a}{(x+c)}}, c) \in \mathbb{G}_1 \times \mathbb{N}$. The advantage of an adversary $\mathcal{A}$ for q-SDH is

$$\mathbb{P}\left[\mathcal{A}(g, g^x, g^{x^2}, \ldots, g^{x^q}) = (g^{\frac{1}{(x+c)}}, c)\right].$$

Assumption 5 *The q-SDH assumption holds if any probabilistic polynomial time adversary has negligible advantage for the q-SDH problem.*

Power Decision Diffie-Hellman Assumption (PDDH)

Assumption 6 *The PDDH problem is said to be (t, ϵ, ℓ)-hard if no algorithm running in time t can, given input tuple $(g, g^x, g^{x^2}, \ldots, g^{x^\ell}, H)$ where $g, g^x, g^{x^2}, \ldots, g^{x^\ell} \in \mathbb{G}_1$ and $H \in \mathbb{G}_3$, distinguish $T = (H^x, H^{x^2}, \ldots, H^{x^\ell})$ from a random vector in $\mathbb{G}_3^\ell$ with probability greater than $\frac{1}{2} + \frac{\epsilon}{2}$.*

Decisional q-Bilinear Diffie-Hellman Exponent Problem (q-BDHE)

Given $(q+2)$ elements of $\mathbb{G}_1$ $(g', g, g^\alpha, \ldots, g^{\alpha^q})$, and one $\mathbb{G}_3$ element $\hat{t}$, output 'yes' if $\hat{t} = e(g^{\alpha^{q+1}}, g')$ and 'no' otherwise.

Assumption 7 *The q-BDHE assumption holds if any probabilistic polynomial time adversary has negligible advantage for the q-BDHE problem.*

Augmented Diffie-Hellman Exponent Assumption (q-ABDHE)

The q-ABDHE problem is given a vector of $2q+2$ elements $(g', g'^{\alpha^{q+1}}, g, g^\alpha, g^{\alpha^q}, g^{\alpha^{q+2}}, \ldots, g^{\alpha^{2q}}) \in \mathbb{G}_1^{2q+2}$, output $e(g, g')^{\alpha^{q+1}} \in \mathbb{G}_3$.

In the truncated q-ABDHE problem, the terms $(g^{\alpha^{q+2}}, \ldots, g^{\alpha^{2q}})$ are omitted from the input vector. An algorithm $\mathcal{A}$ has advantage ϵ in solving the truncated q-ABDHE problem if

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

Assumption 8 *The q-ABDHE assumption holds if any probabilistic polynomial time adversary has negligible advantage for the q-ABDHE problem.*

2.4 Cryptographic Primitives

2.4.1 One-way Functions

One-way functions play a significant role in cryptography. A function $f : \{0,1\}^* \to \{0,1\}^*$ is called *one-way* if, on input x, there is an efficient algorithm that outputs $f(x)$ but it is infeasible to invert $f(x)$ to retrieve x. That is, any feasible algorithm that tries to find x given $f(x)$ may succeed with only negligible probability.

Definition 2 (one-way functions [Gol01]) *A function* $f : \{0,1\}^* \rightarrow \{0,1\}^*$ *is* one-way *if the following two conditions hold:*

easy to evaluate *there exists a polynomial-time algorithm* A *such that* $A(x) = f(x)$ *for every* $x \in \{0,1\}^*$.

hard to invert *for every probabilistic polynomial-time algorithm* A'*, every polynomial* p *and all sufficiently large* n

$$\mathbb{P}\left[A'(f(x), 1^n) \in f^{-1}(f(x))\right] < \frac{1}{p(n)}$$

where the probability is taken uniformly over all the possible choices of $x \in \{0,1\}^n$ *and all the possible outcomes of the internal coin tosses of algorithm* A'.

One way functions are often based on the presumed intractability of computational problems in number theory. Consider the problem of factoring a large integer. A function that takes as input two large, equal length primes and outputs their product is generally believed to be a one-way function. It is conjectured to be infeasible to find the factors of a large composite integer.

The definition given above requires that any feasible algorithm that succeeds in inverting the function does so with negligible probability. This can be weakened to a notion of *weakly inverting* which requires that any feasible algorithm fails to invert the function with *noticeable probability*.

2.4.2 Hash Functions

A cryptographic hash function produces a compact representation (*digital fingerprint* or *message digest or hash*) of the input string. Hash values are often used in place of or in addition to the actual message, for example in digital signatures and for data integrity.

Definition 3 (hash function [MVOV97]) *A* hash function *is a function* h *which has, as a minimum, the following two properties:*

compression h *maps an input* x *of arbitrary finite bit-length, to an output* $h(x)$ *of fixed bit-length* n.

ease of computation *given* h *and an input* x, $h(x)$ *is easy to compute.*

Hash functions can be split into two classes: *unkeyed hash functions* that have a single input parameter, a message; and *keyed hash functions* that have two input parameters, a message and a secret key. The definition above implies an unkeyed hash function with a single input parameter.

Practical unkeyed hash functions require three properties, in addition to compression and ease of computation. For an unkeyed hash function h with inputs x, x' and corresponding outputs y, y' [MVOV97]:

one-way (preimage resistance) for essentially all pre-specified outputs, is it computationally infeasible to find any input which hashes to that output, i.e., to find any preimage x' such that $h(x') = y$ when given any y for which a corresponding input is not known.

weak collision resistance (2nd-preimage resistance) it is computationally infeasible to find any second input which has the same output as any specified input, i.e., given x, to find a second preimage $x' \neq x$ such that $h(x) = h(x')$.

strong collision resistance it is computationally infeasible to find any two distinct inputs x, x' which hash to the same output, i.e., such that $h(x) = h(x')$.

Definition 4 (collision resistant hash function [MVOV97]) *A* collision resistant hash function *(CRHF) is a hash function* h *as per Definition 3 with the properties of one-wayness, weak collision resistance and strong collision resistance. Note that collision resistance implies weak collision resistance.*

2.4.3 Zero-Knowledge

A *proof of knowledge* is, generally speaking, an interactive protocol between two parties, a *prover* and a *verifier*. The prover wants to convince the verifier of the validity of an assertion, while revealing no information beyond the assertion itself. Such a proof requires two properties: *completeness* and *soundness*. Completeness dictates that a proof should allow a prover to convince the verifier of the validity of any true statement. Soundness dictates that a proof should not allow a prover to convince the verifier of the validity of any false statement.

Definition 5 (completeness property [MVOV97]) *An interactive proof (protocol) is* complete *if, given an honest prover and an honest verifier, the protocol succeeds with overwhelming probability (i.e., the verifier accepts the prover's claim). The definition of* overwhelming *depends on the application, but generally implies that the probability of failure is not of practical significance.*

Definition 6 (soundness property [MVOV97]) *An interactive proof (protocol) is* sound *if there exists an expected polynomial-time algorithm M with the following property: if a dishonest prover (impersonating honest prover A) can with non-negligble probability successfully execute the protocol with honest verifier B, then M can be used to extract from this prover knowledge essentially equivalent to A's secret which with overwhelming probability allows successful subsequent protocol executions.*

Zero-knowledge proofs, as introduced by Goldwasser *et al.* [GMR85], provide a means to prove the validity of an assertion without revealing anything else to the verifier. The verifier learns nothing from the zero-knowledge proof that she does not already learn from the assertion. Such proofs require the properties of completeness and soundness, in addition to the property of *zero knowledge*. Informally, an interactive proof system is zero-knowledge if whatever can be efficiently computed after interacting with the prover on input x can also be efficiently computed from x without any interaction. It is a property that captures a prover's robustness against attempts by a verifier to gain knowledge by interacting with it. Definitions of the zero-knowledge property have varying levels of rigor. We present computational zero-knowledge, as it is considered the most practicable [Gol01].

Definition 7 (Computational Zero-Knowledge [Gol01]) *An interactive protocol is* computational zero-knowledge *if for every probabilistic polynomial time verifier there exists a probabilistic polynomial-time simulator such that the following are computationally indistinguishable:*

- *The output of the verifier interacting with the prover on common input x*
- *The output of the simulator on input x*

2.4.4 Commitment Schemes

A commitment scheme allows a party to commit to a value while keeping the value secret. It is a two phase protocol, consisting of a *commit* phase and a *reveal* phase. The two parties in a commitment scheme are a *sender*, who commits to a value, and a *receiver*, who receives the commitment but cannot open it to reveal the value until the sender agrees.

A commitment scheme is required to have two properties [Gol01]:

concealing at the end of the first phase, the receiver does not gain any knowledge of the sender's value. This requirement has to be satisfied even if the receiver tries to cheat.

Input The public parameters are a cyclic group $\mathbb{G}_q$ of prime order q and two generators g, h of $\mathbb{G}_q$ such that $\log_g(h)$ is unknown by any party except the sender.

Commit Sender commits to message $\alpha \in \mathbb{Z}_q$ by choosing $\beta \in \mathbb{Z}_q$ and generating the commitment

$$\mathsf{Commit}(params, \alpha, \beta) = g^{\alpha} h^{\beta}.$$

Reveal The commitment is shown by Sender upon revealing α and β.

Figure 2.1: Pedersen's Commitment Scheme

binding given the transcript of the interaction in the first phase, there exists at most one value that the receiver can later (in the second phase) accept as a legal opening, or revealing, of the commitment.

In the *commit* phase, no information should be revealed to the receiver while at the same time binding the sender to a unique value. In the *reveal* phase, the commitment is opened in a manner which can return only the unique value committed to.

Pedersen's Commitment Scheme

Pedersen's scheme is based on the security of the discrete logarithm problem [Ped91]. The public parameters are a cyclic group $\mathbb{G}_q$ of prime order q and two generators g, h of $\mathbb{G}_q$ such that $\log_g(h)$ is unknown by any party except the sender.

The scheme is presented in Figure 2.1.

2.5 Security Notions

Intuitively, in public-key cryptography it is necessary to keep the private-key a secret. Retrieving the private-key from the public-key of an asymmetric cryptosystem is prevented by an underlying computationally difficult problem, or *hard problem*, encapsulated by one-way functions. Traditionally, the security of cryptosystems is evaluated by rigorous examination of such problems. One-way functions allow the construction of systems which are easy to use but hard to break. This gap between ease of use and difficulty to break is known as a *complexity gap* [Gol01]. Complexity gaps are believed to exist in hard problems such as those outlined in Section 2.3.

This reliance on complexity gaps to evaluate security is limited by a failure to consider other possible attacks. The underlying hard problem is used to build a *protocol*, and the construction

of such protocols may have weaknesses that can be exploited. Breaking a cryptosystem is not necessarily equivalent to solving the underlying mathematical hard problem on which the protocol is based. The Merkle-Hellman knapsack cryptosystem [MH78] is subject to such a break [Sha84].

For example, an *adversary* often attempts to manipulate the execution of a protocol. An adversary can be described as *passive* or *active*, in reference to whether the adversary has taken active steps to disrupt the execution of the protocol. Looking at a cryptographic scheme, one can identify *goals* of the scheme, as well as potential *attack models*.

The aim of security proofs for cryptographic algorithms and protocols (cryptosystems) is to provide security guarantees by modelling subversive behaviour [Den06].

Reductionist security claims are based on the premise that a party that can reveal messages from the ciphertext without the private key must be able to solve the underlying problem [KM07]. The 'reduction' comes from the strategy in such proofs that if one can show the hardness of one problem implies hardness of another problem, then there exists a reduction to a known hard problem, on which the security can be based.

Practice-oriented provable security investigates the details of the reduction presented in the security proof, which facilitates thinking about protocols and primitives in a systematic way. All security proofs are limited by the range of attacks considered in the model. Side channel attacks such as timing analysis, differential power and differential fault attacks are not usually encompassed by security arguments.

2.5.1 Random Oracle and Standard Model

Cryptographic schemes are proven secure to the required level using either *random oracles* or the *standard model*. Security proofs using random oracles were introduced [BR93] to address the gap between theory and practice. A random oracle is used in an idealised security game that captures the properties that real primitives appear to have. Random oracles are used to model cryptographic hash functions. All parties are given access to a (public) random oracle, and the protocol is proven correct in this model (*ideal game*). The random oracle is then replaced with a hash function (*real game*), providing an implementation of the protocol. The two models should be indistinguishable. Proof of security using the random oracles does not imply security in the real world. It is important to note that Canetti *et al.* [CGH04] have shown that some schemes proven secure in the random oracle are insecure when the random oracle is substituted with a function from a certain small

family of efficiently computable functions.

In contrast, the standard model models the situation whereby the adversary is only limited by the amount of time and computational power available. It is the difficulty of constructing a proof in the standard model that motivates proofs in the random oracle model.

2.5.2 Security Goals and Attack Models

We introduce the standard notions of security for public-key encryption schemes used to model *provable security*. Using this approach, a system designer first describes what is understood by the security of a scheme. This is followed by a proof that shows the scheme can be broken by either attacking an insecure cryptographic component of the scheme or by achieving a mathematical breakthrough.

Semantic security (SS)

Given the encryption of two messages of equal length, it is infeasible for an adversary with access to the encryption key to distinguish the encryption of one message from the other. The adversary is unable to obtain any information about the plaintext message that was encrypted to the ciphertext, other that its bitlength, even with access to a decryption oracle for any other ciphertexts. Semantic security is used when confidentiality is the desired end goal.

Indistinguishability (IND)

An adversary chooses two messages. One of the messages is chosen at random and encrypted. Given the resulting ciphertext the adversary cannot guess which message was encrypted with probability greater than $\frac{1}{2}$. This is a technical goal, which aims to capture a strong form of privacy and be easier to reason about than semantic security. Indistinguishability is used when the encryption is used as part of a cryptographic protocol, and is closely related to the notion of semantic security.

Non-malleability (NM)

It is computationally infeasible for an adversary given a ciphertext to generate a different ciphertext such that the respective plaintexts are related in a known manner. Non-malleabilty can be

considered as a form of indistinguishability.

Active attacks have been modeled into three modes, which are used in the analysis of cryptosystems [Mao03].

Chosen-plaintext attacks (CPA)

An adversary given access to the public key of a scheme can choose arbitrary plaintexts to encrypt and obtain the corresponding ciphertexts. A scheme is CPA-secure if it prevents an adversary from choosing a message and constructing the corresponding ciphertext in such a way as to leak any useful information.

Chosen-ciphertext attacks (CCA)

An adversary given access to the public key of a scheme can choose arbitrary plaintexts to encrypt and obtain the corresponding ciphertexts. Additionally, the adversary receives access to a decryption oracle, to which it can submit any ciphertext to be decrypted, except the challenge ciphertext. A scheme is CCA-secure if it prevents an adversary from constructing a ciphertext and obtaining the decryption in such a way as to leak any useful information after it no longer has access to a decryption oracle.

Adaptive chosen-ciphertext attacks (CCA2)

An adversary given access to the public key of a scheme can choose arbitrary plaintexts to encrypt and obtain the corresponding ciphertexts. Additionally, the adversary receives access to a decryption oracle, to which it adaptively submits ciphertexts chosen using the results of previous decryptions, to be decrypted, excluding the challenge ciphertext. A scheme is CCA2-secure if it prevents an adversary from adaptively constructing ciphertexts and obtaining the decryptions in such a way as to leak any useful information. The adversary can access a decryption oracle forever, with the restriction that the target ciphertext may never be queried.

2.5.3 Security Notions for Identity-Based Schemes

Security notions for identity-based schemes build upon existing public-key cryptography security notions. It is also necessary to elucidate the capabilities of an adversary. An adversary is given

access to an *extraction oracle* which, upon input of a public key id, outputs the corresponding private key sk_{id}. The attacker is capable of choosing the challenge id in two ways. Additionally, in some IBE schemes, it is possible to check if a given identity was used to create the ciphertext, using only the public parameters and ciphertext. Depending upon the application scenario, this can be an undesirable property.

Selective-identity (sID) An adversary commits to a chosen identity id ahead of time. That is, before the interactive security game is run.

Adaptive-identity (ID) An adversary may adaptively chose their id, depending upon information garnered thus far in the security game.

Recipient anonymity (ANON) An adversary is unable to distinguish the public key id used to generate a ciphertext ct given the ciphertext and the public parameters of the system $params$.

Security notions are combined to describe the level of security a scheme has. For example, a scheme may be described as IND-ID-CPA secure.

Chosen-Plaintext Attack

In IBE, security means privacy and it can be formalized in several ways, e.g., indistinguishability under chosen-plaintext attack (IBE-IND-CPA) (also semantic security) or indistinguishability under chosen-ciphertext attack (IBE-IND-CCA). We can describe IBE-IND-CPA by means of a game [BF01]:

Setup: The challenger takes a security parameter k and runs the Setup algorithm. It gives the adversary the resulting system parameters $params$, and keeps the master secret key msk to itself.

Phase 1: The adversary makes adaptive queries to $Oracle_{\mathsf{Extract}}(id)$ and gets back the secret keys for these identities.

Challenge: The adversary outputs two equal length plaintexts $m_0, m_1 \in m(k)$ and an identity id that has not been queried before. The challenger chooses a random bit b and runs $\mathsf{Encrypt}(params, id, m_b)$ to obtain a ciphertext C for message m_b under identity id. C is passed to the adversary.

Phase 2: The adversary continues making adaptive queries to $Oracle_{Extract}(id)$. Note that this oracle rejects answering if the identity is the one output by the adversary in the Challenge

step.

Guess: The adversary outputs a bit b' in order to guess which message was used to build the ciphertext.

The advantage of the adversary $\mathcal{A}$ has in the scheme $\mathcal{E}$ is a function of the security parameter k, and is given by:

$$Adv_{\mathcal{E},k}(A) = |Pr[b = b'] - 1/2|.$$

Definition 8 *An IBE system $\mathcal{E}$ is semantically secure against an adaptive chose plaintext attack if for any polynomial time IND-ID-CPA adversary $\mathcal{A}$ the function $Adv_{\mathcal{E},A}(k)$ is negligible.*

Chosen Ciphertext Attack

In IBE-IND-CCA, the adversary can also make queries to an oracle $Oracle_{\mathsf{Decrypt}}(id, C)$ and receive a decryption using the secret key corresponding to a given id_i for a ciphertext C. $Oracle_{\mathsf{Decrypt}}$ does not answer queries answering on the challenge identity / ciphertext pair (id, ct). The IBE scheme is said to be IBE-IND-CCA secure when the advantage of the adversary in the game is negligible.

Setup: The challenger takes a security parameter k and runs the Setup algorithm. It gives the adversary the resulting system parameters $params$, and keeps the master secret key msk to itself.

Phase 1: The adversary issues queries $q_1, \ldots, q_m$, where q_i is one of:

- Extraction query $\langle id_i \rangle$. The challenger responds by running algorithm Extract to generate the private key sk_{id_i} corresponding to the public key id_i. It then sends sk_{id_i} to the adversary.
- Decryption query$\langle id_i, ct_i \rangle$. The challenger responds by running algorithm Extract to generate the private key sk_{id_i} corresponding to the id_i. It then runs algorithm Decrypt to decrypt the ciphertext ct_i using the private key sk_{id_i}. It sends the resulting plaintext to the adversary.

Challenge: The adversary decides when Phase 1 is complete, and outputs two plaintexts $m_0, m_1 \in \mathcal{M}$ and an identity id on which it wishes to be challenged. The only constraint is

that id was not previously queried in Phase 1. The challenger picks a random bit $b \in \{0, 1\}$ and sets ct = Encrypt($params$, id, m_b).

Phase 2: The adversary issues more queries, $q_{m+1}, \ldots, q_n$ where query q_i is one of:

- Extraction query $\langle id_i \rangle$ where $id_i \neq id$. Challenger responds as in Phase 1.
- Decryption query $\langle id_i, ct_i \rangle \neq \langle id, ct \rangle$. Challenger responds as in Phase 1.

Guess: Finally the adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

Such an adversary $\mathcal{A}$ is referred to as an IND-ID-CCA adversary.

The advantage $\mathcal{A}$ has in the scheme $\mathcal{E}$ is a function of the security parameter k, and is given by: $\text{Adv}_{\mathcal{E},\mathcal{A}}(k) = \left|\mathbb{P}[b = b'] - \frac{1}{2}\right|$.

Definition 9 *An IBE system $\mathcal{E}$ is semantically secure against an adaptive chose ciphertext attack if for any polynomial time IND-ID-CCA adversary $\mathcal{A}$ the function $Adv_{\mathcal{E},A}(k)$ is negligible.*

Recipient Anonymity

Abdalla *et al.* [ABC$^+$08] define anonymity through a security game in which the adversary receives a ciphertext encrypted with an identity that is randomly picked from two identities of his choosing. The adversary has to guess the identity used to encrypt the ciphertext. As outlined by Gentry [Gen06], this game can be combined with either the standard chosen plaintext security game or the chosen ciphertext attack game for IBE. An IBE scheme can be proven anonymous by incorporating anonymity into the CCA or CPA security game as with the following modifications.

Setup: As per required CCA / CPA security game.

Phase 1: As per required CCA / CPA security game.

Challenge: The adversary outputs two identities id_0 and id_1 not queried in Phase 1 and two messages m_0 and m_1. The challenger picks two random bits $b, c \in \{0, 1\}$, uses id_b to encrypt m_c, and sends the resulting ciphertext ct to the adversary.

Phase 2: As per phase one, with the restriction that the adversary cannot request a private key for id_0 or id_1, or the decryption of ct under either identity.

Guess: Finally the adversary outputs a guess for each of $b', c' \in \{0, 1\}$ and wins the game if $b = b'$ and $c' = c$.

The advantage $\mathcal{A}$ has in the scheme $\mathcal{E}$ is a function of the security parameter k, and is given by:

$$\mathrm{Adv}_{\mathcal{E},A}(k) = \left| \mathbb{P}[b = b' \wedge c = c'] - \frac{1}{4} \right|.$$

Definition 10 ([Gen06]) *An IBE scheme $\mathcal{E}$ is $(t, q_{id}, d_{ct}, \epsilon)$ ANON-IND-ID-CCA secure if all t-time adversaries making at most q_{id} private key queries and at most q_{ct} chosen ciphertext queries have advantage of at most ϵ in the modified game. ANON-IND-ID-CPA security is defined similarly.*

2.6 Identity-Based Encryption Schemes

2.6.1 The Boneh-Franklin Identity-Based Encryption Scheme

Sakai *et al.* [SOK00] proposed using bilinear pairings to efficiently construct an identity-based, non-interactive key agreement scheme. Following this work, in 2001 Boneh and Franklin [BF01] further explored the application of pairings to cryptography and presented an identity-based encryption (IBE) scheme, along with a formalisation of the security, addressing a question that had beem open since 1984 when Shamir proposed the concept. Alternative approaches to identity-based schemes have been proposed [Coc01, TI89]; however, none are practical due to the conditions imposed on their application, including tamper-freeness and restrictions on users' colluding.

The scheme

The seminal scheme proposed by Boneh and Franklin [BF01] is described by the following four algorithms. The identity is represented as a bit-string.

Setup**:** Given input of security parameter k, generate a prime q, two groups $\mathbb{G}_1$, $\mathbb{G}_3$ of prime order q and an admissible bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$. Choose a random generator $g \in \mathbb{G}_1$, a random $s \in \mathbb{Z}_q^*$ and set $g_{pub} = g^s$. Choose cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_3 \rightarrow \{0,1\}^n$ for some value of n, $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$ and $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$.

Extract**:** Given input of identity string $id \in \{0,1\}^*$, compute $Q_{id} = H_1(id) \in \mathbb{G}_1^*$ and set the private key sk_{id} as $sk_{id} = Q_{id}^s$ where s is the master secret key.

Encrypt: Given input of message $m \in \{0,1\}^n$ and public key id, compute $Q_{id} = H_1(id) \in \mathbb{G}_1^*$, choose a random $\sigma \in \{0,1\}^n$, set $r = H_3(\sigma, m)$ and set the ciphertext ct to be

$$ct = \langle g^r, \sigma \oplus H_2(g_{id}^r), m \oplus H_4(\sigma) \rangle \text{ where } g_{id} = e(Q_{id}, g_{pub}) \in \mathbb{G}_3.$$

Decrypt: Let $ct = \langle U, V, W \rangle$ be a ciphertext encrypted with id as its public key. If $U \notin \mathbb{G}_1^*$, then reject the ciphertext. Otherwise decrypt the ciphertext using the private key sk_{id} by computing $v \oplus H_2(e(sk_{id}, U)) = \sigma$, $W \oplus H_4(\sigma) = m$. Set $r = H_3(\sigma, m)$ and reject the ciphertext if $U \neq g^r$. Output m, the decryption of ct.

Security of the scheme

The Boneh-Franklin scheme is adaptive-identity chosen ciphertext secure (IND-ID-CCA) using random oracles. It works with any efficiently-computable pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$ between two groups under the Bilinear Diffe-Hellman (BDH, Section 2.3) assumption. Suitable pairings can be derived using both the Weil and the Tate pairings. The construction of a chosen ciphertext secure system in the standard model remained an open problem at that time.

2.6.2 The Boneh-Boyen Efficient Selective Identity Identity-Based Encryption Scheme without Random Oracles

In response to the open problem at the time of an IBE construction secure in the standard model, Boneh-Boyen [BB04a] presented two IBE schemes which are secure in the standard model.

The Efficient Selective Identity IBE scheme

This first scheme extends to Hierarchical IBE (HIBE), as introduced by Horwitz and Lynn [HL02]. HIBE allows a root key generator (RKG) to generate the private key of any users in their hierarchy. Such a scheme is useful in large organisations to remove the dependence on a single key generator. The RKG can generate the private keys of the managers of the organisation, who in turn can generate the private keys for employees on their team, etc. To generate keys using the Boneh-Boyen method, a HIBE scheme must have a maximum depth of l. An identity is represented as a vector $ID = (id_1, \ldots, id_l)$, where id_j represents the identity at level $j \in \{1, 2, \ldots, l\}$.

The scheme

Setup**:** Given input of maximum depth l, select a random generator $g \in \mathbb{G}_1^*$, a random $\alpha \in \mathbb{Z}_q$ and set $g_1 = g^\alpha$. Choose random elements $h_1, \ldots, h_l \in \mathbb{G}_1$ and a random element $g_2 \in \mathbb{G}_1$. The public parameters $params$ are $g, g_1, g_2, h_1, \ldots, h_l$ and the master secret msk is g_2^α. For $j = 1, \ldots, l$, define a function $F_j : \mathbb{Z}_q \to \mathbb{G}_1$ as $F_j(x) = g_1^x h_j$.

Extract**:** Given input of $ID = (id_1, \ldots, id_j) \in \mathbb{Z}_q^j$ of depth $j \leq l$, choose random $r_1, \ldots, r_j \in \mathbb{Z}_q$ and output

$$d_{ID} = \big(g_2^\alpha \cdot \prod_{k=1}^{j} F_k(id_k)^{r_k}, g^{r_1}, \ldots, g^{r_j}\big).$$

Encrypt**:** Given input of message m and public key $ID = (id_1, \ldots, id_j) \in \mathbb{Z}_q^j$, choose a random $s \in \mathbb{Z}_q$ and output the ciphertext

$$ct = \big(e(g_1, g_2)^s \cdot m, g^s, F_1(id_1)^s, \ldots, F_j(id_j)^s\big).$$

Decrypt**:** Let ct = $(A, B, C_1, \ldots, C_j)$. Using the private key $d_{ID} = (d_0, d_1, \ldots, d_j)$, output

$$A \cdot \frac{\prod_{k=1}^{j} e(C_k, d_k)}{e(B, d_0)} = m.$$

Security of the scheme

The security of the HIBE scheme [HL02] extends chosen-ciphertext security, based on the assumption that an attacker can obtain private keys at any level excluding the master secret. The security game is as outlined for chosen-ciphertext above, except the adversary's queries in Phase 1 can use any prefix to a given identity at level l it chooses. The challenge identity N which the adversary wants to be queried on has the restriction that no prefix of N has been queried in Phase 1. Finally, in Phase 2 of the security game, neither the challenge ciphertext nor any prefix of N may be queried.

This scheme is secure in the selective-ID model, which is weaker than the adaptive-ID model. It requires that the adversary commits to an identity ahead of time and may only challenge using this identity. However, it is secure under the Decisional Bilinear Diffe-Hellman assumption (Decisional-BDH, Section 2.3) in the standard model.

The More Efficient Selective Identity IBE scheme

A second scheme has the advantage of a more efficient decryption algorithm. This advantage is based on the use of only one pairing computation in the decryption algorithm, while the encryption efficiency and the ciphertext size remain the same as in the first scheme. The pairing computation $e(g, g)$ required for the encryption can be pre-computed, meaning the encryption does not require any pairing computations.

The scheme

Setup**:** Select a random generator $g \in \mathbb{G}_1^*$, random elements $x, y \in \mathbb{Z}_q^*$ and set $X = g^x$ and $Y = g^y$. The public parameters $params$ are (g, X, Y) and the master secret msk is (x, y).

Extract**:** Given input of public key $id \in \mathbb{Z}_q^*$, choose random $r \in \mathbb{Z}_q$ and compute $K = g^{\frac{1}{(id+x+ry)}} \in \mathbb{G}_1$ and output the private key $sk_{id} = (r, K)$.

Encrypt**:** Given input of a message m and a public key id, choose at random $s \in \mathbb{Z}_q^*$ and output the ciphertext

$$ct = (g^{s \cdot id} X^s, Y^s, e(g, g)^s \cdot m).$$

Decrypt**:** Given input of ct and sk_{id}, let $ct = (A, B, C)$. Using the private key sk_{id}, output $\frac{C}{e(A \cdot B^r, K)} = e(\frac{C}{g^{s(id+x+ry)}}, \frac{1}{g^{id+x+ry}}) = \frac{C}{e(g,g)^s} = m$.

Security of the scheme

This scheme is IND-sID-CPA secure under the B-DHI assumption. As with the first scheme, it can be converted to a chosen ciphertext secure scheme using Canetti *et al.*'s transform [CHK07]. It is worth noting that this does not provide an efficient scheme, as the conversion relies on the use of non-interactive zero-knowledge constructions.

Furthermore, Boneh-Boyen [BB04a] shows that any selective-ID secure scheme is also an adaptive-ID secure scheme. The reduction they provide depends on having a sufficiently large identity space. This reduction is inefficient.

2.6.3 The Boneh-Boyen Secure Identity-Based Encryption Scheme without Random Oracles

This scheme [BB04b] is an adaptation of the first of the two Boneh-Boyen schemes [BB04a] presented above. The resulting scheme is impractical, but provides a proof of concept that a secure, adaptive-ID chosen-ciphertext scheme with a polynomial security reduction can exist in the standard model.

The scheme

Let $\mathbb{G}_1$ be a bilinear group of prime order q and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_3$. $\Sigma = \{1, \ldots, s\}$ is an alphabet of size s and let $\{H_k : \{0,1\}^w \rightarrow \Sigma^n\}_{k \in K}$ be a family of hash functions where $K \in \Sigma^{(n,m)}$ is a vector with $n \geq m > 0$. A public key id is an element of $\{0,1\}^w$.

Setup: Choose a random generator $g \in \mathbb{G}_1^*$, a random $\alpha \in \mathbb{Z}_q$ and set $g_1 = g^\alpha$. Choose a random element $g_2 \in \mathbb{G}_1$ and construct a random $n \times s$ matrix $U = (u_{i,j}) \in \mathbb{G}_1^{n \times s}$ where each $u_{i,j}$ is uniform in $\mathbb{G}_1$. Finally choose a random $k \in K$ as a hash function key. The system parameters $params$ are (g, g_1, g_2, U, k) and the master secret msk is g_2^α.

Extract: Given input of identity $id \in \{0,1\}^w$, let $\vec{a} = H_k(id) = a_1, \ldots, a_n \in \Sigma^n$ and pick random $r_1, \ldots, r_n \in \mathbb{Z}_q$. The private key is

$$sk_{id} = (g_2^\alpha \cdot \prod_{i=1}^{n} u_{i,a_i}^{r_i}, g^{r_1}, \ldots, g^{r_n}).$$

Encrypt: Given input of message m and public key identity id, set $\vec{a} = H_k(id) = a_1, \ldots, a_n \in \Sigma^n$, pick a random $t \in \mathbb{Z}_q$ and output the ciphertext

$$ct = (e(g_1, g_2)^t \cdot m, g^t, u_{1,a_1}^t, \ldots, u_{n,a_n}^t).$$

Decrypt: Let $ct = (A, B, C_1, \ldots, C_n)$, using the private key $sk_{id} = (d_0, d_1, \ldots, d_n)$, output

$$A \cdot \frac{\prod_{j=1}^{n} e(C_j, d_j)}{e(B, d_0)} = m.$$

Security of the scheme

The proof of security is in the standard model and is based on the Decisional-BDH assumption. This scheme is IND-ID-CPA provably secure. Adaptive-id security is achieved using the reduction in Section 7 of [BB04b] which requires hashing identities using a collision resistant function and introduces a 2^n factor in the security parameters.

2.6.4 The Waters Efficient Identity-Based Encryption Scheme without Random Oracles

The scheme proposed by Waters [Wat05] presents the first efficient IBE scheme provably secure in the standard model. It is based on the algebraic method used by Boneh-Boyen in the preceding scheme presented in Section 2.6.2. In the Boneh-Boyen scheme, the identity v is represented as $g_1^v h_j$. At level one hierarchy, and given h_1 and u' are random elements, the identity v can be evaluated as $u'g^v$. The Waters' scheme evaluates v as $u' \prod_{i \in \mathcal{V}} u_i$ where $\mathcal{V} \subseteq \{1, \ldots, n\}$ is the set of all i for which v_i = 1. It is this small modification that makes the scheme efficient and adaptively secure in the standard model.

The scheme

Let $\mathbb{G}_1$ be a group of prime order q, g be a random generator of $\mathbb{G}_1$ and e be a bilinear map from $\mathbb{G}_1$ to $\mathbb{G}_3$. Identities v are represented as bit-strings of length n.

Setup**:** Choose a secret random value $\alpha \in \mathbb{Z}_q$. Set $g_1 = g^\alpha$ and choose g_2 at random in $\mathbb{G}_1$. Choose a random $u' \in \mathbb{G}_1$ and a vector $U = (u_i)$, where each $u_i \in \mathbb{G}_1$. The public parameters are g, g_1, g_2, u' and U. The master secret of the $\mathcal{KGC}$ is α.

Extract**:** Given as input an identity v, choose random value $r \in \mathbb{Z}_q$. Let $\mathcal{V} \subseteq \{1, \ldots, n\}$ be the set of all i for which v_i = 1. The private key $d_v = (d_1, d_2)$ corresponding to identity v is constructed as

$$d_v = \big(g_2^\alpha (u' \prod_{i \in \mathcal{V}} u_i)^r, g^r\big).$$

Encrypt**:** A message m is encrypted for an identity string v. Choose a random value $t \in \mathbb{Z}_q$ and construct the ciphertext $ct = (c_1, c_2, c_3)$ as

$$ct = \big(e(g_1, g_2)^t \cdot m, g^t, \big(u' \prod_{i \in \mathcal{V}} u_i\big)^t\big).$$

Decrypt**:** A ciphertext that is the encryption of m under v can be decrypted using d_v as

$$\begin{aligned} c_1 \frac{e(d_2, c_3)}{e(c_2, d_1)} &= e(g_1, g_2)^t \cdot m \cdot \frac{e(g^r, (u' \prod_{i \in \mathcal{V}} u_i)^t)}{e(g^t, g_2^{\alpha}(u' \prod_{i \in \mathcal{V}} u_i)^r)} \\ &= e(g_1, g_2)^t \cdot m \cdot \frac{e(g, (u' \prod_{i \in \mathcal{V}} u_i^{v_i})^{rt})}{e(g_1, g_2)^t e(g, (u' \prod_{i \in \mathcal{V}} u_i)^{rt})} \\ &= \frac{e(g_1, g_2)^t}{e(g_1, g_2)^t} \cdot m \cdot \frac{e(g, (u' \prod_{i \in \mathcal{V}} u_i^{v_i})^{rt})}{e(g, (u' \prod_{i \in \mathcal{V}} u_i)^{rt})} \\ &= m. \end{aligned}$$

Security of the scheme

The security of this scheme is based on the Decisional-BDH (D-BDH) assumption, and is IND-ID-CPA secure in the standard model. The scheme as presented is limited to chosen-plaintext security; that is, it only guarantees security against adversaries that are prevented from choosing a message and constructing the corresponding ciphertext in such a way as to leak any useful information. Waters presents a modification to achieve CCA security.

2.6.5 The Naccache Secure and Practical Identity-Based Encryption Scheme

The Waters IBE scheme [Wat05] was the first practical and efficient IBE to have been proposed with its security assumptions in the standard model. The efficiency it achieves is a result of the need for fewer exponential and bilinear map computations. A remaining open problem at the time was to produce a scheme that did not have the very large public parameters of Waters' scheme. Naccache [Nac07] provides a solution to this problem with a scheme that is a variant of Waters' scheme with a smaller public-key size, resulting in the first practical and secure IBE scheme that is semantically secure against passive adversaries in the standard model.

The Chatterjee-Sarkar Scheme In concurrent independent work, Chatterjee and Sarkar [CS06] proposed a similar scheme. Their work contributes a concrete security analysis, in addition to the scheme and security proof. They provide a rigorous examination of the tightness of the security reduction. Additionally, they provide results on required group sizes for 80-bit security for identity lengths of 160 and 256, split into vectors of length l, where various values for l are considered.

The scheme

Let $\mathbb{G}_1$ be a group of prime order q, g be a random generator of $\mathbb{G}_1$ and e be a bilinear map from $\mathbb{G}_1$ to $\mathbb{G}_2$. Identities v are represented as n dimensional vectors $v = (v_1, \ldots, v_n)$ where each v_i is of length l.

Setup**:** Choose a secret α at random from $\mathbb{Z}_q$. Set $g_1 = g^\alpha$ and chose g_2 at random in $\mathbb{G}_1$. Choose a random $u' \in \mathbb{G}_1$ and a vector $U = (u_i)$, where each $u_i \in \mathbb{G}_1$. The public parameters are g, g_1, g_2, u' and U. The master secret of the $\mathcal{KGC}$ is α.

Extract**:** Let $v = (v_1, \ldots, v_n) \in (\{0,1\}^l)^n$ be an identity and choose a random value $r \in \mathbb{Z}_q$. The private key $d_v = (d_1, d_2)$ corresponding to identity v is constructed as

$$d_v = \big(g_2^\alpha \big(u' \prod_{i=1}^{n} u_i^{v_i}\big)^r, g^r\big).$$

Encrypt**:** A message m is encrypted for an identity string v. Choose a random value $t \in \mathbb{Z}_q$ and construct the ciphertext $ct = (c_1, c_2, c_3)$ as

$$ct = \big(e(g_1, g_2)^t \cdot m, g^t, \big(u' \prod_{i=1}^{n} u_i^{v_i}\big)^t\big).$$

Decrypt**:** A ciphertext that is the encryption of m under v can be decrypted using d_v as

$$\begin{aligned} c_1 \frac{e(d_2, c_3)}{e(c_2, d_1)} &= e(g_1, g_2)^t \cdot m \cdot \frac{e(g^r, (u' \prod_{i=1}^{n} u_i^{v_i})^t)}{e(g^t, g_2^\alpha (u' \prod_{i=1}^{n} u_i^{v_i})^r)} \\ &= e(g_1, g_2)^t \cdot m \cdot \frac{e(g, (u' \prod_{i=1}^{n} u_i^{v_i})^{rt})}{e(g_1, g_2)^t e(g, (u' \prod_{i=1}^{n} u_i^{v_i})^{rt})} \\ &= m. \end{aligned}$$

The security of the scheme

The scheme is IND-ID-CPA secure in the standard model under the Decisional-BDH assumption. Chatterjee and Sarkar [CS06] show that the conversion of security degradation into a trade-off between time and space is the most important feature of the generalisation of Waters scheme.

2.6.6 The Boyen-Waters Anonymous Identity-Based Encryption Scheme

This scheme is one of few schemes that achieve the anonymity property [BW06]. The first is that of Boneh and Franklin [BF01], although they did not explicitly state it. Their scheme is secure in

the random oracle model. Thus the Boyen-Waters scheme is the first anonymous scheme proven secure in the standard model.

The scheme

Setup**:** Given a group $\mathbb{G}_1$, choose a random generator $g \in \mathbb{G}_1$, random elements $g_0, g_1 \in \mathbb{G}_1$ and random exponents $\alpha, t_1, t_2, t_3, t_4 \in \mathbb{Z}_q$. The msk consists of the random exponents $\alpha, t_1, t_2, t_3, t_4$ and the system parameters $params$ are published as

$$params = \big[\omega = e(g,g)^{t_1 t_2 \alpha}, g, g_0, g_1, v_1 = g^{t_1}, v_2 = g^{t_2}, v_3 = g^{t_3}, v_4 = g^{t_4}\big].$$

Extract**:** Given input of msk and id, the $\mathcal{KGC}$ chooses two random exponents $r_1, r_2 \in \mathbb{Z}_q$ and computes the secret key $sk_{id} = [d_0, d_1, d_2, d_3, d_4]$ as

$$sk_{id} = \big[g^{r_1 t_1 t_2 + r_2 t_3 t_4}, g^{-\alpha t_2}(g_0 g_1^{id})^{-r_1 t_1}, g^{-\alpha t_2}(g_0 g_1^{id})^{-r_1 t_1}, (g_0 g_1^{id})^{-r_2 t_4}, (g_0 g_1^{id})^{-r_2 t_3}\big].$$

Encrypt**:** Given input of $params$, id and message $m \in \mathbb{G}_2$, to encrypt m for identity $id \in \mathbb{Z}_q \backslash \{0\}$ choose random exponents $s, s_1, s_2 \in \mathbb{Z}_q$ and create ciphertext as

$$ct = [C', C_0, C_1, C_2, C_3, C_4] = \big[\omega^s m, (g_0 g_1^{id})^s, v_1^{s-s_1}, v_2^{s_1}, v_3^{s-s_2}, v_4^{s_2}\big].$$

Decrypt**:** Given input of sk_{id}, ct, decrypt by computing

$$C' \cdot e(C_0, d_0) \cdot e(C_1, d_1) \cdot e(C_2, d_2) \cdot e(C_3, d_2) \cdot e(C_3, d_3) \cdot e(C_4, d_4) = m.$$

Security of the scheme

The scheme by Boyen-Waters is an anonymous scheme that is IND-sID-CPA secure in the standard model. It is selective-identity secure scheme under the Decisional BDH assumption, which is weaker than previous adaptive-identity secure schemes. Boyen and Waters [BW06] detailed transformations to achieve adaptive-ID and CCA security.

2.6.7 Gentry's Practical Identity-Based Encryption Scheme without Random Oracles

Gentry provides an efficient IBE scheme with the anonymity property [Gen06]. The scheme has short public parameters. Although this scheme is anonymous, the $\mathcal{KGC}$ is required to keep a list of the random value $\tau_{id,i}$ associated with each id, and reuse it should further keys for the id be requested.

The scheme

Setup: The $\mathcal{KGC}$ selects g, h_1, h_2, h_3 randomly from $\mathbb{G}_1$, chooses a random exponent $\alpha \in \mathbb{Z}_q$, sets $g_1 = g^\alpha \in \mathbb{G}_1$, and chooses a hash function $H : \{0,1\} \rightarrow \mathbb{Z}_q$ from a family of universal one-way hash functions. The public parameters are $params = (g, g_1, h_1, h_2, h_3, H)$ and the master secret key is $msk = \alpha$.

Extract: Given input of $params$ and an identity id, the $\mathcal{KGC}$ chooses a random value $\tau_{id,i} \in \mathbb{Z}_q$ and computes $h_{id,i} = (h_i g^{-\tau_{id,i}})^{\frac{1}{\alpha - id}}$ for $i \in \{1,2,3\}$. $\mathcal{KGC}$ outputs $sk_{id} = \{\tau_{id,i}, h_{id,i}\}_{i \in \{1,2,3\}}$.

Encrypt: Given input of message m and identity id, choose random $r \in \mathbb{Z}_q$ and generate ciphertext

$$ct = (u, v, w, y) = \big((g_1 g^{-id})^r, e(g,g)^r, \frac{m}{e(g,h_1)^r}, e(g,h_2)^r e(g,h_3)^{r \cdot H(u,v,w)}\big).$$

Decrypt: Given input of sk_{id}, ct, first check the validity of ct by testing if $y = e(u, h_{id,2} h_{id,3}^{\beta}) v^{\tau_{id,2} + \tau_{id,3}\beta}$ where $\beta = H(u,v,w)$. If this does not hold, output $\perp$, else return $m = w \cdot e(u, h_{id,1}) v^{\tau_{id,1}}$.

Security of the scheme

The scheme is IND-ID-CCA secure in the standard model and achieves the anonymity property. The security reduction is to the strong truncated q-ABDHE complexity assumption.

2.7 Conclusion

In this chapter, the requisite background for the remainder of this work has been described. Sufficient mathematical foundations and computational assumptions have been presented. Identity-based encryption has been introduced, along with some notable constructions. In particular, we presented seminal constructions, along with those which will be used as a basis for later work.

Chapter 3

Identity-Based Schemes and the Blinding Property

3.1 Introduction

Privacy is the claim of owners, groups or organisations to autonomy, encompassing the right to control information about themselves and the right to limit access to that information [DL99]. It is a beneficial property that allows owners to exert control over their personal information and data [Bra00]. Privacy is also a design consideration for many systems, particularly in electronic communications where data is subject to contemporary concerns such as profiling of an owner or unauthorised disclosure of personal information [Can04]. One should assume that whatever personal data is collected by a third party will be stored indefinitely. Once such data is stored, it can be used for a myriad of purposes the owner may not have considered or consented to outside the original stated use. It is trivial to link data from a host of sources using common identifiers such as name, social security number, date of birth, student identification number. Linkability is an identified problem that results in a loss of privacy for the owner [Bra00]. The capacity to link an owner's actions and disclosures can culminate in profiling of the owner. The claim to privacy is supported by legislation, as detailed previously.

Identity-based encryption and trust in the KGC

The need to trust the $\mathcal{KGC}$ is considered a drawback of IBE. Known as the *key escrow problem*, the $\mathcal{KGC}$ in traditional IBE schemes learns the users identity when generating their private key. Thus,

a $\mathcal{KGC}$ can subsequently generate a private key corresponding to an identity and use it to decrypt ciphertexts for that identity. It is also possible for a $\mathcal{KGC}$ to generate keys for identity strings in advance of a user request. Many countermeasures to address the level of trust required in the $\mathcal{KGC}$ have been proposed, some of which are outlined below.

Distributed KGCs Boneh and Franklin [BF01] proposed distributing the master secret key of the $\mathcal{KGC}$ to multiple $\mathcal{KGC}$ entities, using Shamir's secret sharing technique [Sha79]. A user generates her key by interacting with at least k out of n of the possible $\mathcal{KGC}$ entities, presenting each with her identity. In turn, she receives a share of her private key. The user can check the validity of her key share, thus ensuring a misbehaving $\mathcal{KGC}$ can be detected. The user combines k of these key shares to retrieve her private key. This method imposes heavy loads on users, who have to authenticate themselves to multiple entities.

Accountable IBE Accountable IBE (A-IBE) schemes, introduced by Goyal [Goy07], aim to reduce the level of trust required by the user in the $\mathcal{KGC}$. In such schemes, if the $\mathcal{KGC}$ maliciously generates and distributes or uses a decryption key for an identity, then it may be caught using a *trace* algorithm. Should two keys for one identity be generated by the $\mathcal{KGC}$, this algorithm can identify which key was generated for the owner that requested it, and which was generated for potentially malicious use by the $\mathcal{KGC}$. Further to this, Goyal *et al.* [GSW08] provide a black-box A-IBE system. This scheme can be modified and used in conjunction with an IND-ID-CPA secure IBE scheme by splitting the message to be encrypted. Both schemes share the message using secret sharing methods [Sha79] and run as normal. Most recently, Libert and Vegnaud [LV09] proposed an efficient black-box A-IBE scheme.

Anonymous private key issuing Anonymous key issuing (AKI) [SCH$^+$05, Cho09] aims to prevent the identity of a user being leaked to the $\mathcal{KGC}$, while also facilitating an authenticated user retrieving the correct private key. In this system, the Setup algorithm is split into two parts. A trusted initialiser chooses the group and public elements, and passes them to the $\mathcal{KGC}$ who generates the msk. This prevents the $\mathcal{KGC}$ from maliciously choosing the system parameters. A user authenticates to an identity-certifying authority that issues the user a certificate on the authenticated identity. The user then presents this certificate to the $\mathcal{KGC}$, and they engage in an interactive protocol from which the user receives a private key as output and the $\mathcal{KGC}$ receives

nothing.

Contribution

In all IBE schemes, except those with a distributed approach, once an identity string is revealed to the $\mathcal{KGC}$ it can trivially generate a corresponding private key. To create IBE schemes which afford an owner some form of privacy from the $\mathcal{KGC}$, we focus on the identity string and how it is disclosed to the $\mathcal{KGC}$.

Our contribution is to present novel constructions of blind IBE cryptosystems. Blind IBE schemes prevent the $\mathcal{KGC}$ from learning the identity string of a user during the key extraction protocol. We begin by constructing an anonymous IBE scheme. As anonymity prevents the ciphertext from leaking the identity string, it is a beneficial property to have in conjunction with blindness. We present the associated blind extraction protocol for this scheme, resulting in the first construction of a blind, anonymous IBE scheme.

We then consider extensions to blind IBE. A natural extension is *partially-blind* IBE, which allows some of the identity string to be visible to the $\mathcal{KGC}$. We propose a new property, *double-blindness*. Constructions for both partially-blind and double-blind IBE schemes are presented. We also provide a transformation of these protocols for use with an underlying anonymous IBE. Security definitions and arguments are provided.

3.2 The Blinding Property

The *blinding property* was introduced by Chaum [Cha82]. Applying the blinding property to digital signatures, he proposed *blind signatures*. Blind signatures enable a receiver to obtain a signed message from a signer in such a manner that the message to be signed is not revealed to the signer. Chaum identified privacy as an important feature of any e-cash system, primarily to prevent an owner's spending habits from being observed. By blinding the message to be signed, the signer cannot link the message to the recipient. In terms of e-cash, this prevents a bank from linking an instance of e-cash to an honest owner and identifying how she spends her cash.

Chaum's carbon-envelope analogy provides an intuitive introduction to the concept of *blind signatures* [Cha83]. The sender wants to have a message signed to prove its validity to third parties. Following Chaum's untraceable payments scenario, the signer is a bank of which the sender is a customer. The objective is for the sender to receive a slip of paper (cash token) with a validating

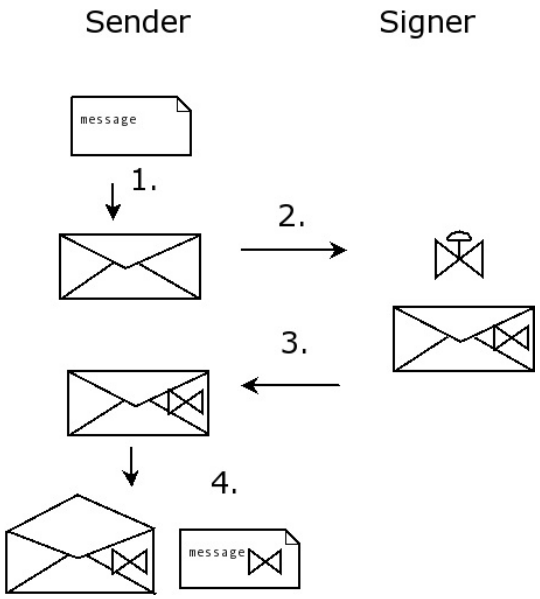

Figure 3.1: Blind signatures using carbon-lined envelopes

signature on it provided by the bank, that cannot be linked back to the sender.

The sender constructs her message and places it in a carbon lined envelope, as per step 1 of Figure 3.1. This message could be an invoice the sender needs to pay or a cash token. She then passes the sealed envelope to the signer, as per step 2. At this point, the signer may want some assurance that the message contained in the envelope is valid. The signer authenticates herself as a customer with the bank and requests to withdraw a certain amount. Once convinced, the bank deducts the amount from her account, signs the carbon-lined envelope and returns it to the sender, as per step 3. Upon receipt of the envelope, the sender opens it and takes out the signed message, as per step 4. The envelope is then cast aside and the sender can use the enclosed signed message as untraceable payment.

Partially blind signature schemes were proposed by Abe and Fujisaki [AF96]. Using these schemes, it is possible for the signer to see part of the message to be signed. Taking the untraceable payments scenario once again, this visible element could be some information such as the date or cash amount.

The sender constructs her message and places it in a carbon lined envelope, as per step 1 of Figure 3.2. This envelope has a window, through which some part of the message is visible. She then passes the sealed envelope to the signer, as per step 2. At this point, the signer may want some assurance that the message contained in the envelope is valid. The signer authenticates herself as a customer with the bank and requests to withdraw a certain amount, which is deducted from her

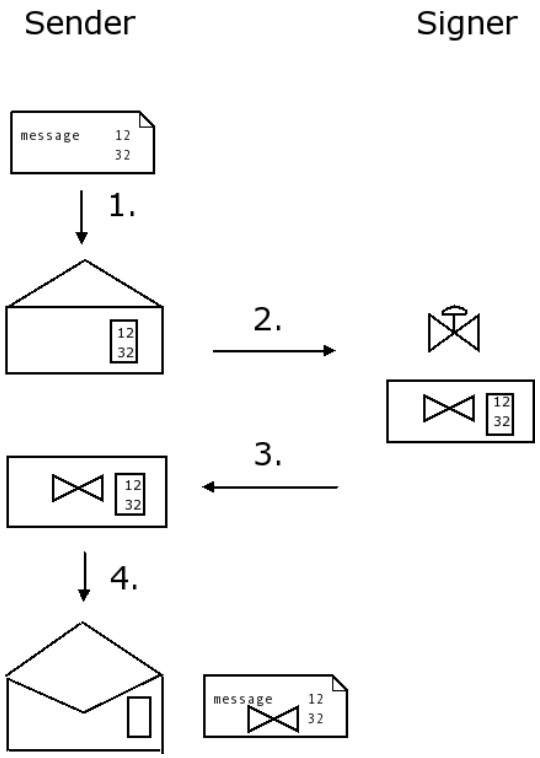

Figure 3.2: Partially-blind signatures using carbon-lined envelopes

account. In partially-blind schemes, the information visible through the window may also act to convince the signer of the validity of the message contained within. Once convinced, the bank signs the carbon-lined envelope for the appropriate amount and returns it to the sender, as per step 3. Upon receipt of the envelope, the sender opens it and takes out the signed message, as per step 4. The envelope is then cast aside and the sender can use the enclosed signed message as untraceable payment.

3.2.1 Blind Signatures

Digital signatures provide authentication, integrity and non-repudiation to communications, and as such are one of the most fundamental and widely applicable concepts in cryptography. Because digital signatures are so practicable in scenarios including e-cash, e-voting or e-auctions, they have been adapted to provide specific solutions to these applications. The original digital signature concept has been expanded to incorporate many additional properties, such as blind signatures, verifiably encrypted signatures and aggregated signatures [GHK06].

A basic blinding protocol, used to achieve the blind signature scheme outlined in Figure 3.1, is provided by Chaum. Let $n = pq$ be the product of two large random primes. A message m

is chosen as $0 \leq m \leq n-1$. In the case a larger message space is required, the message m is hashed using a suitable a hash function H and the resulting $H(m)$ is used as input to the signature scheme.

To begin, the bank publishes some *public value* b such that $\gcd(n, b) = 1$. Alice creates her message m and chooses a random value r, which correspond to the message and envelope respectively in the analogy outlined above. The bank's published value b can correspond to the value "worth x euro", or the relevant stamp as in Figure 3.1. Alice constructs the blind message to be signed as $m \cdot r^b \mod n$, and passes it to the Bank. The bank signs the message using the *private value* $\bar{b}$ corresponding to b such that $\bar{b} \cdot b = 1$.

Blind signature generation

$$\begin{aligned} \text{Sender} \rightarrow \text{Signer} &: m' = m \cdot r^b \mod n \\ \text{Signer} \rightarrow \text{Sender} &: \sigma' = (m')^{\bar{b}} \mod n \\ \text{Sender unblinds to retrieve} &: \sigma = \sigma' \cdot r^{-1} \mod n \end{aligned}$$

The signature needs to be verified to assure validity before a third party, for example a shop, will accept it. The verifier uses the public value b to verify a sender's claim that it has a signature $m^{\bar{b}}$ on a message m.

Blind signature verification

$$\begin{aligned} \text{Signature is correct } \sigma &= \sigma' \cdot r^{-1} = (m')^{\bar{b}} r^{-1} \\ &= m^{\bar{b}} r^{b\bar{b}} r^{-1} = m^{\bar{b}} \mod n \\ \text{Verifier checks that } \sigma^b &= m \mod n \end{aligned}$$

Blind signature scheme Formally, a blind signature scheme BS consists of three algorithms, BS = (KeyGen, Sign, Verify).

KeyGen**:** Given as input a security parameter k, returns a public / private key-pair (pk, sk).

Sign**:** Given as $Sender$ input (m, pk) where m is the message to be signed and $Signer$ input of the corresponding secret key sk, returns a signature on m, $\sigma(m)$ to $Sender$ and nothing to $Signer$.

Verify**:** Given as input pk, m, $\sigma(m)$, outputs $accept/reject$ to indicate if a valid signature has been presented.

Security of Blind Signatures

A blind signature scheme is secure if it satisfies two properties: *blindness* and *unforgeabilibity*. Formal definitions of blind signatures were given by Juels *et al.* [JLO97].

Blindness

The blinding property captures the notion of a signer attempting to obtain some information about the messages she is signing when these messages are obscured, or blinded, from her view [GHK06]. Blindness ensures the signer cannot learn the content of the message.

A blind digital signature scheme is considered as a four-tuple consisting of two Turing machines $(Signer, Sender)$ and two algorithms (KeyGen, Verify), with KeyGen output of a public / private key pair (pk, sk). The polynomially-bounded probabilistic interactive Turing machines $Signer(pk, sk)$ and $Sender(pk, m)$ have the following tapes: read-only input tape, write-only output tape, a read/write work tape, a read-only random tape, and two communication tapes, one read-only and one write-only. The $Signer$ machine is given input of the public and private keys pk, sk on it's input tape and the $Sender$ machine is given input of the public key and a message pk, m on it's input tape. The $Signer$ and $Sender$ engage in the interactive blind signing protocol, at the end of which $Signer$ outputs *complete* or *not-complete* and the $Sender$ outputs either *fail* or $\sigma(m)$. The adversary may interact either sequentially or in parallel during attacks; parallel attacks are considered stronger, as an adversary can initiate new interactions prior to the completion of previous ones [PS96].

Blindness security game Let $b \in \{0, 1\}$ be a random bit and (pk, sk) be a key pair generated using KeyGen. An adversary $\mathcal{A}$ controls the $Signer$ machine, but not the $Sender$, and executes the following steps:

1. $\mathcal{A}$ produces two messages $\{m_0, m_1\}$ which are polynomial in security parameter k and are by convention lexicographically ordered.

2. Denote the two messages $\{m_0, m_1\}$ as $\{m_b, m_{b-1}\}$, ordered according to the value of bit b which is kept secret from $\mathcal{A}$. $\mathcal{A}$ then engages in two parallel interactive protocols with $Sender(pk, m_b)$ and $Sender(pk, m_{b-1})$.

3. If neither of the $Sender$ protocols output fail, then their output is put on their private tapes,

$\sigma(m_b)$ and $\sigma(m_{b-1})$ respectively. $\mathcal{A}$ is given as input $\{\sigma(m_b), \sigma(m_{b-1})\}$, ordered according to the corresponding (m_0, m_1) order.

4. $\mathcal{A}$ outputs guess b'. $\mathcal{A}$ wins if $b' = b$.

Definition 11 (Blindness) *A signature scheme is blind if, for all probabilistic polynomial-time adversaries $\mathcal{A}$, $\mathcal{A}$ has advantage of at most $\frac{1}{2} + \frac{1}{k^c}$ in the blind security game for a sufficiently large security parameter k and some constant c. The probability is taken over coin flips of* KeyGen*, $Sender$, $Signer$ and $\mathcal{A}$.*

Unforgeability

The unforgeability property captures the notion that a sender may only obtain a valid signature from a signer if they execute the interactive blind signature protocol together [GHK06]. Any digital signature scheme allows one to sign documents or data in such a way that the signature can be universally verified, but no signatures can be forged on messages that have not been signed.

Unforgeability security game Let (pk, sk) be a key pair generated using KeyGen. An adversary $\mathcal{A}$ controls the $Sender$ machine, but not the $Signer$, and tries to get "one-more" signature. That is, given l valid signatures, $\mathcal{A}$ tries to forge more signatures using them. She succeeds if she holds $l + 1$ valid signatures after n iterations of the protocol.

1. $\mathcal{A}$ engages in polynomially many (in security parameter k) adaptive, parallel interactive protocols with polynomially many copies of $Signer(pk, sk)$. Let l be the number of executions, with $\mathcal{A}$ deciding adaptively when to stop, and outputting $completed$ when done.

2. $\mathcal{A}$ outputs a collection of messages and their corresponding signatures $\{(m_1, \sigma(m_1)), \ldots, (m_j, \sigma(m_j))\}$, subject to the constraint that all $(m_i, \sigma(m_i))$ for all $1 \leq i \leq j$ are $accepted$ by Verify$(pk, m_i, \sigma(m_i))$.

The probability that $j > l$ is at most $\frac{1}{k^c}$ where k is the security parameter and c is a constant [JLO97].

Definition 12 (Unforgeability) *A blind signature scheme is unforgeable if for any probabilistic polynomial-time algorithm $\mathcal{A}$ that plays the unforgeabililty security game, the probability that the output of $\mathcal{A}$ satisfies* Verify*(pk, m_i, σ_i) for $1 \leq i \leq l + 1$ is at most $\frac{1}{k^c}$ for a sufficiently large security parameter k and some constant c.*

3.2.2 Partially-blind Signatures

Partially-blind signature schemes, proposed by Abe and Fujisaki [AF96], are an extension to blind signatures. They allow a signer to produce a valid signature on a message for a recipient, and include in the message some pre-agreed, observable information while the rest of the message remains obscured. The motivation for partially blind signatures is to provide a signer with some control over the message to be signed by allowing her to explicitly choose or view part of the message.

The unblinded part of the message is subject to two constraints. Firstly, it should not allow a sender to cheat in any manner. An adversary should not be able to induce an failure in the scheme for example. Secondly, it should contain sufficiently generic information that the signer cannot then use it to distinguish the transaction at a later point.

Further work by Abe and Okamoto [AO00] formalised the concept of partially-blind signature schemes. A partially-blind signature scheme PBS consists of three algorithms: PBS = (KeyGen, Sign, Verify).

KeyGen**:** takes as input a security parameter k and returns a public / private key-pair (pk, sk).

Sign**:** is an interactive protocol between the sender $Sender$ and the signer $Signer$ who has a public key pk, with $Sender$ input of (m, pk) where m is the message to be signed containing some commonly agreed upon information $info$, the input of $Signer$ is the corresponding secret key sk. The output is a signature on m, $\sigma(m)$.

Verify**:** takes input pk, m, $\sigma(m)$ and outputs $accept/reject$.

Partial-blindness security game [AO00] Let $b \in \{0, 1\}$ be a random bit and (pk, sk) be a key pair generated using KeyGen. Let $Sender_0$ and $Sender_1$ be two honest senders following a blind signature issuing protocol. An adversary $\mathcal{A}$ controls the signer in the game.

On input 1^k, where k is a security parameter and secret key sk, $\mathcal{A}$ produces $m_0, m_1, info_{Sender_0}$, $info_{Sender_1}$.

The input tapes of $Sender_0, Sender_1$ are set up by selecting $b \in \{0, 1\}$ and putting m_b and m_{b-1} on the private input tapes of $Sender_0$ and $Sender_1$ respectively. The values $info_0$ and $info_1$ are put on the public input tapes of $Sender_0$ and $Sender_1$ respectively, along with pk. The contents of the private random tapes are randomly selected.

$\mathcal{A}$ engages in the signature issuing protocol with $Sender_0$ and $Sender_1$.

If $Sender_0$ and $Sender_1$ output $(info_0, m_0, \sigma_b)$ and $(info_1, m_1, \sigma_{b-1})$ respectively on their private tapes, and $info_0 = info_1$ holds, then the outputs are returned to $\mathcal{A}$. Otherwise $\perp$ is returned to $\mathcal{A}$.

$\mathcal{A}$ outputs guess $b' \in \{0, 1\}$. $\mathcal{A}$ wins if $b' = b$.

Definition 13 (Partial Blindness) *A signature scheme is partially blind if, for all probabilistic polynomial-time adversaries $\mathcal{A}$, $\mathcal{A}$ has advantage of at most $\frac{1}{2}+\frac{1}{k^c}$ in the partial-blindness security game for a sufficiently large security parameter k and some constant c. The probability is taken over coin flips of* KeyGen*, $Sender_0, Sender_1$ and $\mathcal{A}$.*

Unforgeability

As with blind signature schemes, the unforgeability property captures the notion that a sender may only obtain a valid signature from a signer if they execute the interactive partially-blind signature protocol.

3.3 Blind Identity-Based Encryption

There exists an innate relationship between IBE schemes and digital signatures, as detailed in Section 1.1, and so it is natural to consider the applicability of signature scheme extensions to IBE schemes.

Blind-IBE schemes are the result of merging traditional IBE schemes with a desirable property of digital signatures: blindness. In standard IBE, the $\mathcal{KGC}$ executes the *key extraction* algorithm Extract that returns the secret key corresponding to input identity id. Green and Hohenberger [GH07] propose extracting the secret key in a *blinded* manner, thus keeping the identity completely obscured from the $\mathcal{KGC}$. The BlindExtract($\mathcal{U}$ ($params$, id), $\mathcal{KGC}(msk)$) $\rightarrow$ (sk_{id}, *nothing*) protocol has user $\mathcal{U}$ input of system parameters and the identity string, and $\mathcal{KGC}$ input of the system master secret; it returns output of the private key to $\mathcal{U}$ and output of nothing or an error message to the $\mathcal{KGC}$. Two efficient BlindExtract protocols are proposed [GH07].

BlindExtract$(\mathcal{U}(params, id), \mathcal{KGC}(msk) \rightarrow (sk_{id}, nothing))$

generates the secret decryption key sk_{id} for $\mathcal{U}$'s identity id in an interactive key issuing

protocol between $\mathcal{U}$ and the $\mathcal{KGC}$. $\mathcal{U}$'s output is a decryption key sk_{id} or an error message, and the output of the $\mathcal{KGC}$ is empty or an error message.

Green and Hohenberger also formalise the *blind* execution of the Extract protocol. The $\mathcal{KGC}$ does not learn anything about the identity, nor can she cause any failures in the protocol that are dependent on the identity. The user requesting the key learns nothing more than she would if the standard Extract protocol were used. The resulting blind-IBE scheme is used to build an efficient oblivious transfer protocol.

3.3.1 Security Notions for Blind Identity-Based Encryption

A definition of security for blind IBE is given by Green and Hohenberger [GH07]. It focuses on defining secure blindness for the BlindExtract protocol. Informally, given an IBE scheme with a BlindExtract protocol, secure blindness is achieved by satisfying two properties: *leak-freeness* and *selective-failure blindness*.

Leak-freeness requires that BlindExtract be a secure two-party computation that does not leak any more information to $\mathcal{U}$ than Extract.

Selective-failure blindness requires that a potentially malicious authority does not learn anything about the user's identity during the BlindExtract protocol. Also, the authority cannot cause the algorithm BlindExtract to selectively fail depending on the user's choice of identity.

A BlindExtract protocol is a blind signature scheme, when one considers sk_{id} as a signature by the $\mathcal{KGC}$ on a message id.

Definition 14 (Leak Freeness [GH07])

A BlindExtract *protocol of an IBE scheme is leak free if, for all efficient adversaries $\mathcal{A}$, there exists an efficient simulator $\mathcal{S}$ such that no efficient distinguisher $\mathcal{D}$ can determine whether it is playing Game Real or Game Ideal with non-negligible advantage, where*

Game Real: *Run* Setup. *As many times as $\mathcal{D}$ wants he picks $\mathcal{A}$'s input $state$. $\mathcal{A}$ executes* BlindExtract *with the $\mathcal{KGC}$. $\mathcal{A}$ returns the resulting $view$ to $\mathcal{D}$.*

Game Ideal: *Run* Setup. *As many times as $\mathcal{D}$ wants he picks $\mathcal{A}$'s initial input $state$. $\mathcal{S}$ obtains $(params, state)$ and may choose id to query an oracle $\mathcal{O}_{\mathsf{Extract}}$ that knows msk. The oracle*

returns key $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$*; otherwise it returns* $\perp$*.* $\mathcal{S}$ *returns a simulated* $view$ *to* $\mathcal{D}$*.*

Definition 15 (Selective-failure Blindness [CNS07])

A BlindExtract *protocol is said to be selective-failure blind if every adversary* $\mathcal{A}$ *has a negligible advantage in the following game:* $\mathcal{A}$ *outputs* $params$ *and a pair of identities* id_0, id_1*. A random bit* $b \in \{0, 1\}$ *is chosen, and* $\mathcal{A}$ *is given black-box access to two oracles:* $\mathcal{O}(params, id_b)$ *and* $\mathcal{O}(params, id_{1-b})$*. The* $\mathcal{O}$ *algorithms produce local output* sk_b, sk_{1-b} *respectively. If both* $sk_b \neq \perp$*,* $\mathcal{A}$ *receives* (sk_0, sk_1)*; if only* $sk_{1-b} = \perp$*,* $\mathcal{A}$ *receives* $(\epsilon, \perp)$*; if only* $sk_b = \perp$*,* $\mathcal{A}$ *receives* $(\perp, \epsilon)$*; and if* $sk_b = sk_{1-b} = \perp$*,* $\mathcal{A}$ *receives* $(\perp, \perp)$*. Finally,* $\mathcal{A}$ *outputs its guess* b'*. The advantage of* $\mathcal{A}$ *in this game is* $|Pr[b' = b] - \frac{1}{2}|$*.*

3.3.2 The BlindExtract Protocol for an IND-sID-CPA secure scheme

The first scheme by Green and Hohenberger [GH07] is based on the IND-sID-CPA secure scheme by Boneh-Boyen [BB04a]. The IND-sID-CPA security definition is presented in Section 2.5.3.

The underlying Boneh-Boyen IBE scheme is presented in Section 2.6.2. The Setup, Encrypt and Decrypt algorithms remain as in the original scheme. In this scheme, the identity set $\mathcal{I} \subseteq \mathbb{Z}_q$ and the function $F : \mathcal{I} \rightarrow \mathbb{G}_1$ is defined as $F(id) = h \cdot g_1^{id}$. The secret key for an identity id is of the form

$$sk_{id} = (d_0, d_1) = (g_2^{\alpha} \cdot F(id)^r, g^r) = (g_2^{\alpha} \cdot (h \cdot g_1^{id})^r, g^r)$$

where $r \in \mathbb{Z}_q$ is a random value.

The protocol BlindExtract is detailed in Figure 3.3. The security of the protocol is defined as follows, with security proofs provided in appendix A of [GH07].

Definition 16 *Under the DBDH assumption, the blind IBE* $\Pi = (\mathsf{Setup}, \mathsf{BlindExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is IND-sID-CPA secure if and only if: (1)* Π *is IND-sID-CPA secure and (2)* BlindExtract *is leak-free and selective-failure blind.*

3.3.3 The BlindExtract Protocol for an IND-ID-CPA secure scheme

The second scheme Green and Hohenberger [GH07] present is based on the IND-ID-CPA secure Naccache scheme [Nac07], a generalised version of Waters' scheme [Wat05]. The underlying IBE scheme is presented in Section 2.6.5. The Setup, Encrypt and Decrypt algorithms remain

$\mathcal{KGC}(params, msk)$	$\mathcal{U}(params, id)$
	1. Choose a random value $y \leftarrow \mathbb{Z}_q$.
	2. Compute $h' = g^y g_1^{id}$ and send h' to the $\mathcal{KGC}$.
	3. Execute $PoK\{(y, id) : h' = g^y g_1^{id}\}$.
4. If the proof fails to verify, abort,	
5. Choose a random value $r \in \mathbb{Z}_q$.	
6. Compute $d_0' = g_2^{\alpha} \cdot (h'h)^r$.	
7. Compute $d_1 = g^r$.	
8. Send (d_0', d_1') to $\mathcal{U}$.	
	9. Check that $e(g_1, g_2) \cdot e(d_1', h'h) = e(d_0', g)$.
	10. If the check passes, choose random value $z \leftarrow \mathbb{Z}_q$; otherwise, output $\perp$ and abort.
	11. Compute $d_0 = (d_0'/(d_1')^y) \cdot F(id)^z$ and $d_1 = d_1' \cdot g^z$.
	12. Output sk_{id} = (d_0, d_1).

Figure 3.3: BlindExtract protocol for Boneh-Boyen's IND-sID-CPA IBE

as in the original scheme. In this scheme, $\mathcal{I}$ is the set of bit strings of length N, where N is polynomial in κ, represented as n blocks of length l. The function $F : \{0,1\}^N \rightarrow \mathbb{G}_1$ is defined as $F(id) = h \cdot \Pi_{i=1}^n u_i^{a_i}$ where each $u_i \in \mathbb{G}_1$ is randomly selected by the $\mathcal{KGC}$ and each a_i is an l-bit segment of id. The secret key for an identity id is of the form:

$$sk_{id} = (d_0, d_1) = (g_2^{\alpha} \cdot F(id)^r, g^r) = (g_2^{\alpha} \cdot (h \cdot \Pi_{i=1}^n u_i^{a_i})^r, g^r).$$

where $r \in \mathbb{Z}_q$ is a random value.

$\mathcal{KGC}(params, msk)$	$\mathcal{U}(params, id)$
	1. Choose a random value $y \leftarrow \mathbb{Z}_q$.
	2. Parse identity as $id = (a_1, \ldots, a_n)$, $\lvert a_i \rvert = l$. Compute $h' = g^y \cdot \Pi_{i=1}^n u_i^{a_i}$ and send h' to $\mathcal{KGC}$.
	3. Execute $PoK\{(y, a_1, \ldots, a_n) : h' = g^y \cdot \Pi_{i=1}^n u_i^{a_i} \wedge 0 \leq a_i < 2^l\}$.
4. If the proof fails to verify, abort.	
5. Choose a random value $r \in \mathbb{Z}_q$.	
6. Compute $d_0' = g_2^{\alpha} \cdot (h'h)^r$.	
7. Compute $d_1 = g^r$.	
8. Send (d_0', d_1') to $\mathcal{U}$.	
	9. Check that $e(g_1, g_2) \cdot e(d_1', h \cdot h) = e(d_0', g)$.
	10. If the check passes, choose random value $z \leftarrow \mathbb{Z}_q$; otherwise, output $\perp$ and abort.
	11. Compute $d_0 = (d_0'/(d_1')^y) \cdot F(id)^z$ and $d_1 = d_1' \cdot g^z$.
	12. Output sk_{id} = (d_0, d_1).

Figure 3.4: BlindExtract protocol for Naccache's IND-ID-CPA IBE

The protocol BlindExtract is detailed in Figure 3.4. The security of the protocol is defined as follows, with security proofs provided in appendix A of [GH07].

Definition 17 *Under the DBDH assumption, the blind IBE scheme* $\Pi = (\mathsf{Setup}, \mathsf{BlindExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is IND-ID-CPA secure if and only if: (1)* Π *is IND-ID-CPA secure and (2)* BlindExtract *is leak-free and selective-failure blind.*

3.4 Anonymous Blind Identity-Based Encryption

An IBE scheme is recipient-anonymous [ABC$^+$08, Gen06] if it is not possible to check if an identity has been used to encrypt a message, given a ciphertext and corresponding public parameters, unless the identity string is stored with the ciphertext. In the context of public key encryption this is also known as key privacy [BBDP01]. Recent work [IP08] strengthens this notion of recipient-anonymity to those schemes which retain the anonymity property even if the adversary has access to the master secret key. This does not address the issue of a $\mathcal{KGC}$ generating keys maliciously; rather, it means that a $\mathcal{KGC}$ presented with a ciphertext does not have the advantage of learning the encrypting identity using the master secret.

In work with Camenisch *et al.* [CKRS09], we extend Green and Hohenberger's scheme [GH07] by proposing a *blind anonymous* IBE scheme. We construct a *committed blind anonymous* IBE scheme, consisting of the algorithms of a novel construction of an anonymous IBE scheme (labelled Π hereafter), a secure commitment scheme Commit and the protocol BlindExtract. This construction of an ANON-IND-ID-CCA scheme is motivated by the ibe-2-peks transform presented by abdalla [ABC$^+$08], which we use in our application scenario detailed in Section 4.3. We identified blindness as a desirable feature for this transform as it facilitates searchable encryption without revealing the search terms chosen by the user.

3.4.1 The Underlying Anonymous IBE Scheme

Several anonymous IBE schemes have been proposed [BW06, BF01, Gen06]. Our scheme is based on the anonymous selective identity secure IBE scheme presented by Boyen-Waters [BW06]. A transformation due to Naccache [Nac07], a variant of that of Waters [Wat05], is employed to achieve the desirable property of adaptive identity security. The use of such a transformation to achieve anonymous adaptive identity IBE secure schemes was proposed by Boyen-Waters.

The Anonymous Adaptive-ID IBE Scheme

This scheme supports asymmetric bilinear pairings, allowing for the use of a wider range of potentially more efficient implementations using different pairing types [GPS08].

Let identity $id \in \{0,1\}^{\ell \times n}$ and let $id_1 | \ldots | id_n = id$ be the separation of id into n ℓ bit integers id_i, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a bilinear pairing. Let $H_1(id) = g_0 \prod_{i=1}^{n} g_i^{id_i}$ and $H_2(id) = h_0 \prod_{i=1}^{n} h_i^{id_i}$ where $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ are random generators and $g_1, \ldots g_n \in \mathbb{G}_1$, $h_1, \ldots, h_n \in \mathbb{G}_2$ are random group elements. The anonymous IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ consists of the following algorithms :

$\mathsf{Setup}(1^k)$: Run $\mathsf{Setup}(1^k)$ to obtain a bilinear map setup $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$. Choose values $\alpha, z_0, z_1, \ldots, z_n, t_1, t_2, t_3, t_4 \leftarrow \mathbb{Z}_q^*$ and keep $msk = (\alpha, t_1, t_2, t_3, t_4)$ as the master key. Compute the system parameters as

$$params = \Big(\Omega = e(g,h)^{t_1 t_2 \alpha}, g, h, g_0 = g^{z_0}, \ldots, g_n = g^{z_n}, v_1 = g^{t_1}, \ldots, v_4 = g^{t_4}, \\ h_0 = h^{z_0}, \ldots, h_n = h^{z_n}\Big).$$

$\mathsf{Extract}(params, msk, id)$: Choose two random values $\tilde{r}_1, \tilde{r}_2 \leftarrow \mathbb{Z}_q^*$ and compute the key

$$sk_{id} = \Big(h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4}, h^{-\alpha t_2} H_2(id)^{-\tilde{r}_1 t_2}, h^{-\alpha t_1} H_2(id)^{-\tilde{r}_1 t_1}, H_2(id)^{-\tilde{r}_2 t_4}, \\ H_2(id)^{-\tilde{r}_2 t_3}\Big) .$$

$\mathsf{Encrypt}(params, id, m)$: To encrypt a message $m \in \mathbb{G}_T$, choose $s, s_1, s_2 \leftarrow \mathbb{Z}_q$, and generate the ciphertext

$$ct = \Big(\Omega^s \cdot m, H_1(id)^s, v_1^{s-s_1}, v_2^{s_1}, v_3^{s-s_2}, v_4^{s_2}\Big).$$

$\mathsf{Decrypt}(params, sk_{id}, ct)$: Parse sk_{id} as $(d_0, d_1, d_2, d_3, d_4)$ and ct as $(c', c_0, c_1, c_2, c_3, c_4)$ and return

$$m = c' \cdot e(c_0, d_0) \cdot e(c_1, d_1) \cdot e(c_2, d_2) \cdot e(c_3, d_3) \cdot e(c_4, d_4).$$

Theorem 1 *The scheme Π is a secure anonymous IBE under the DBDH and D-Linear assump-*

tions.

Proof. The proof of security uses hybrid games. Let $ct = (ct_i{}', ct_{i0}, ct_{i1}, ct_{i2}, ct_{i3}, ct_{i4})$ be the challenge ciphertext given to the adversary during a real attack. Let R be a random element of $\mathbb{G}_T$ and R', R'' be random elements of $\mathbb{G}_1$. The following games differ as to which challenge ciphertext is given by the challenger to the adversary in the security game:

Game 1. The challenge is $ct = (ct_i{}', ct_{i0}, ct_{i1}, ct_{i2}, ct_{i3}, ct_{i4})$.

Game 2. The challenge is $ct = (R, ct_{i0}, ct_{i1}, ct_{i2}, ct_{i3}, ct_{i4})$.

Game 3. The challenge is $ct = (R, ct_{i0}, R', ct_{i2}, ct_{i3}, ct_{i4})$.

Game 4. The challenge is $ct = (R, ct_{i0}, R', ct_{i2}, R'', ct_{i4})$.

Note that the ciphertext in Game 2 leaks no information about the message. Indistinguishability between Game 1 and 2 thus corresponds to chosen plaintext attack security. Similarly, Game 4 leaks no information about the identity since it is composed of six random group elements. We show that the transition from Game 1 to Game 2 (Lemma 1) and from Game 2 to Game 3 and Game 3 to Game 4 (Lemma 2 and Lemma 3 respectively) are all computationally indistinguishable.

An adversary playing the recipient anonymity game in Section 2.5.3 reacts the same upon receiving real challenge ciphertexts or random ciphertexts. The probability of this not occurring is negligible (otherwise he could act as a distinguisher which would contradict the above results about Games 1 to 4). For random ciphertexts his success probability is $\frac{1}{4}$, consequently in the real game his success probability is at most $\frac{1}{4} + \nu(k)$.

□

Lemma 1 (semantic security) *Under the DBDH assumption, no p.p.t. adversary can distinguish Games 1 and 2 with non-negligible advantage.*

Proof. Begin by assuming that such an adversary $\mathcal{A}$ exists. We construct a reduction $\mathcal{B}$ that solves the DBDH problem with non-negligible advantage. Let q be a security parameter. Algorithm $\mathcal{B}$ receives a DBDH challenge $(\hat{g}, \tilde{A} = \hat{g}^a, \tilde{B} = \hat{g}^b, C = \hat{g}^c, \hat{h}, A = \hat{h}^a, B = \hat{h}^b, z)$ as input and outputs a guess β' as to whether $z = e(\hat{g}, \hat{h})^{abc}$, return guess $\beta' = 1$, or z is a random element in $\mathbb{G}_T$, return guess $\beta' = 0$. We first describe a simulator that does not quite work, and then modify it so that it does work, an approach that has been used previously [Nac07, Wat05].

Setup: The simulator sets an integer $m = 2q$ and chooses a random integer $k \in \{0, \ldots, n\}$, a random n-length vector $\vec{x} = (x_1, \ldots, x_n)$, where $x_i \in \{0, \ldots, m-1\}$ and an integer $x' \in \{1, \ldots, m-1\}$. Let X^* denote the pair $(x', \vec{x})$. The simulator also chooses a random $y' \leftarrow \mathbb{Z}_q$ and an n-length vector $\vec{y} = (y_i)$, where $y_i \leftarrow \mathbb{Z}_q$. Let Y^* denote the pair $(y', \vec{y})$.

For a given identity $id = (id_1, \ldots, id_n)$, define three functions for ease of analysis:

$$F(id) = x' + \sum_{i=1}^{n} id_i x_i - km \ ,$$
$$J(id) = y' + \sum_{i=1}^{n} id_i y_i \ ,$$
$$K(id) = \begin{cases} 0 \text{ if } x' + \sum_{i=1}^{n} id_i x_i \equiv 0 \pmod{m} \\ 1 \text{ otherwise} \end{cases} .$$

The simulator then generates the public parameters $h = \hat{h}$, $h_0 = B^{x'-mk} h^{y'}$ and $h_i = B^{x_i} h^{y_i}$, as well as $g = \hat{g}$, $g_0 = \tilde{B}^{x'-mk} g^{y'}$ and $g_i = \tilde{B}^{x_i} g^{y_i}$, where $1 \leq i \leq n$. It also chooses random $t_1, \ldots, t_4 \leftarrow \mathbb{Z}_q$ and publishes the parameters $(\Omega = e(\tilde{A}, B)^{t_1 t_2}, g, h, g_0, \ldots g_n, h_0, \ldots h_n, g^{t_1}, \ldots, g^{t_4})$, where the distribution is identical to the real construction and the master secret is $(\alpha, t_1, t_2, t_3, t_4)$. Note that α is implicitly set to ab.

Phase 1: The simulator must answer the private key queries of $\mathcal{A}$, who issues a query for an identity id. If $K(id) = 0$, the simulator aborts and randomly chooses its guess β' of the challenger's value β.

Otherwise the simulator chooses random values $r_1, r_2 \in \mathbb{Z}_q^*$ and constructs the private key d as $d = \left(d_0, d_1, d_2, d_3, d_4 \right)$ where

$$d_0 = (A^{\frac{-1}{F(id)}} h^{r_1})^{t_1 t_2} h^{r_2 t_3 t_4} \ ,$$
$$d_1 = (A^{\frac{-J(id)}{F(id)}} (h_0 \prod_{i=1}^{n} h_i^{id_i})^{r_1})^{-t_2} \ ,$$
$$d_2 = (A^{\frac{-J(id)}{F(id)}} (h_0 \prod_{i=1}^{n} h_i^{id_i})^{r_1})^{-t_1} \ ,$$
$$d_3 = (h_0 \prod_{i=1}^{n} h_i^{id_i})^{-r_2 t_4} \ ,$$
$$d_4 = (h_0 \prod_{i=1}^{n} h_i^{id_i})^{-r_2 t_3} \ .$$

Let $\tilde{r}_1 = r_1 - \frac{a}{F(id)}$ and $\tilde{r}_2 = r_2$. Then

$$\begin{aligned} d_0 &= (A^{\frac{-1}{F(id)}} h^{r_1})^{-t_1 t_2} h^{r_2 t_3 t_4} \\ &= (h^{\frac{-a}{F(id)}} h^{r_1})^{t_1 t_2} h^{r_2 t_3 t_4} \\ &= h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4}. \end{aligned}$$

Using the fact that $(g_0 \prod_{i=1}^{n} g_i^{id_i}) = \tilde{B}^{F(id)} \hat{h}^{J(id)}$ and $(g_0 \prod_{i=1}^{n} g_i^{id_i})^{a/F(id)} = B^a A^{J(id)/F(id)}$ we obtain

$$\begin{aligned} d_1 &= (A^{\frac{-J(id)}{F(id)}} (h_0 \prod_{i=1}^{n} h_i^{id_i})^{r_1})^{-t_2} \\ &= (A^{\frac{-J(id)}{F(id)}} (B^{F(id)} h^{J(id)})^{r_1})^{-t_2} \\ &= (B^{s_1} (B^{F(id)} h^{J(id)})^{\frac{-a}{F(id)}} (B^{F(id)} h^{J(id)})^{r_1})^{-t_2} \\ &= (B^a (h_0 \prod_{i=1}^{n} h_i^{id_i})^{r_1 - \frac{a}{F(id)}})^{-t_2} \\ &= h^{-\alpha t_2} (h_0 \prod_{i=1}^{n} h_i^{id_i})^{-\tilde{r}_1 t_2}. \end{aligned}$$

Similarly,

$$d_2 = (A^{\frac{-J(id)}{F(id)}} (h_0 \prod_{i=1}^{n} h_i^{id_i})^{r_1})^{-t_1} = h^{-\alpha t_1} (h_0 \prod_{i=1}^{n} h_i^{id_i})^{-\tilde{r}_1 t_1}$$

As $\tilde{r}_2 = r_2$, d_3 and d_4 are easily seen to be correct. The reduction is feasible as $F(id) \neq 0 \pmod{p}$ is implied by $K(id) \neq 0$.

Challenge: Adversary $\mathcal{A}$ submits a message m and identity $id = id_1 | \ldots | id_n$. If $x' + \sum_{i=1}^{n} id_i x_i \neq km$ the simulator aborts and answers with a random guess. Otherwise, the simulator constructs the ciphertext

$$ct = \left(z^{t_1 t_2} m, C^{J(id)}, C^{t_1} g^{-s_1 t_1}, g^{s_1 t_2}, C^{t_3} g^{-s_2 t_3}, g^{s_2 t_4} \right).$$

If $z = e(g,h)^{abc}$, then for $\alpha = ab$ and $s = c$, the above is a correctly formed ciphertext for an identity $id = id_1 | \ldots | id_n$ that fulfills the equation $x' + \sum_{i=1}^{n} id_i x_i = km$.

$$\begin{aligned} \left(c', c_0, c_1, c_2, c_3, c_4 \right) &= \left(z^{t_1 t_2} m, C^{J(id)}, C^{t_1} g^{-s_1 t_1}, g^{s_1 t_2}, C^{t_3} g^{-s_2 t_3}, g^{s_2 t_4} \right) \\ &= \left(e(g,h)^{\alpha t_1 t_2 s} m, H_1(id)^s, g^{t_1 (s - s_1)}, g^{t_2 s_1}, g^{t_3 (s - s_2)}, g^{t_4 s_2} \right). \end{aligned}$$

If z is a random element, then c' is also a random element in $\mathbb{G}_T$.

Phase 2: The simulator repeats Phase 1.

Guess: The adversary outputs a bit γ to guess which hybrid game it is playing. The reduction forwards γ as its educated guess for the solution to the DBDH problem.

Artificial Aborts As in [Nac07, Wat05], this simulation has an issue. That is, it aborts with a probability that is a function of the identities id and id^*. A solution is to artificially abort the simulator at the end of the guess phase. The idea is to make the overall probability of the simulator aborting consistent.

Using the probabilistic analysis of Naccache [Nac07], we have the following. Beginning at the challenge phase, fix the random variables that are visible to the adversary, fix the DBDH tuple and the public parameters. Also fix the random values $\tilde{r}_1, \tilde{r}_2$ in phase one. These fixed parameters are remembered by $\mathcal{B}$. This fixes the queried identities $id^j, 1 \leq j \leq q$ and the challenge identity id^*. The adversary can now be seen as a deterministic algorithm.

Using the random variables x' and x_i, the list of private key queries $\vec{ID} = (id^1, \ldots, id^q)$ and $\vec{X} = (x', x_1, \ldots, x_n)$, define the function

$$\tau(\vec{X}, \vec{id}, id^*, k) = \begin{cases} 0, \text{if } F(id^*) = 0 \text{ and } F(id^j) \neq 0 \mod m & \text{for all } 1 \leq j \leq q \\ 1, \text{ otherwise .} \end{cases}$$

The reduction does not abort iff $\tau(\vec{X}, \vec{id}, id^*, k) = 0$. The lower bound for the probability that $\mathcal{B}$ does not abort is

$$\Pr_{\vec{X},k}[\tau(\vec{X}, \vec{id}, id^*, k) = 0] \geq \lambda = \frac{1}{4 \cdot q \cdot 2^l \cdot n} .$$

The simulator is modified so that it will always abort with probability approaching λ. In the guess phase, the new simulator $\mathcal{B}'$ samples an estimate η' of the probability $Pr_{\vec{X},k}[\tau(\vec{X}, \vec{id}, id^*, k) = 0]$, which is a function of id and id^*. Following from the analysis of Naccache [Nac07], the probability of breaking the IBE scheme is less than $q \cdot 2^{l+4} \cdot n$ times the probability of solving the DBDH problem. For the analysis of an optimum value for l, the reader is referred to [Nac07].

□

Lemma 2 (Anonymity part 1) *Under the Decision Linear assumption, no p.p.t adversary can distinguish between the Games 2 and 3 with non-negligible probability.*

Proof. Suppose the existence of an adversary $\mathcal{A}$ that distinguishes between the two games, Game 2 and Game 3, with advantage ϵ. We construct a simulator that wins the Decisional Linear game as follows.

The simulator takes in a D-Linear instance $(g, g^a, g^b, g^{ac}, g^{bd}, Z, h, h^a, h^b)$ where Z is either g^{c+d} or random in $\mathbb{G}_1$ with equal probability. For convenience, we rewrite this as $[g, g^a, g^b, g^{ac}, Y, g^s, h, h^a, h^b]$ for s such that $g^s = Z$ (that is, s is either $c + d$ or random). Consider the task of deciding if $Y = g^{b(s-c)}$. The simulator plays the following game:

Setup: The simulator first chooses random exponents α, t_3, t_4. It lets g and h in the simulation be as in the instance and sets $v_1 = g^b, v_2 = g^a$. If we posit that $t_1 = b$ and $t_2 = a$, we note that the parameters are distributed as in the real scheme. The simulator sets an integer $m = 2q$ and chooses a random integer $k \in \{0, \ldots, n\}$, a random n-length vector $\vec{x} = (x_1, \ldots, x_n)$, where $x_i \in \{0, \ldots, m\}$ and $x' \in \{1, \ldots, m-1\}$. Let X^* denote the pair $(x', \vec{x})$. The simulator also chooses a random $y' \leftarrow Z_p$ and an n-length vector $\vec{y} = (y_i)$, where $y_i \leftarrow Z_p$. Let Y^* denote the pair $(y', \vec{y})$. For a given identity $id = (id_1, \ldots, id_n)$, define three functions $F(id)$, $J(id)$ and $K(id)$ as above for ease of analysis.

The simulator generates the public parameters $h_0 = h^{b(x'-mk)}h^{y'}$ and $h_i = h^{bx_i}h^{y_i}$, as well as $g_0 = g^{b(x'-mk)}g^{y'}$ and $g_i = g^{bx_i}g^{y_i}$, where $1 \leq i \leq n$. The public parameters are published as

$$\left(\Omega = e(g^a, h^b)^\alpha, g, h, g_0, \ldots, g_n, h_0, \ldots, h_n, v_1 = g^b, v_2 = g^a, v_3 = g^{t_3}, v_4 = g^{t_4}\right) .$$

Phase 1: To answer a private key extraction query for identity $id = id_1|\ldots|id_n$ the simulator chooses random exponents $r_1, r_2 \in Z_p$ and outputs a private key $d = (d_0, d_1, d_2, d_3, d_4)$ where

$$\begin{aligned}
d_0 &= h^{ar_1}h^{r_2t_3t_4} , \\
d_1 &= (h^b)^{-\alpha - F(id)r_1} , \\
d_2 &= (h^a)^{-\alpha - F(id)r_1} , \\
d_3 &= (h^a)^{-\frac{r_1 J(id)}{t_3}} H_2(id)^{-r_2t_4} , \\
d_4 &= (h^a)^{-\frac{r_1 J(id)}{t_4}} H_2(id)^{-r_2t_3} .
\end{aligned}$$

This is a well formed private key sk_{id} for

$$sk_{id} = \left(h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4}, h^{-\alpha t_2} H_2(id)^{-\tilde{r}_1 t_2}, h^{-\alpha t_1} H_2(id)^{-\tilde{r}_1 t_1}, \quad H_2(id)^{-\tilde{r}_2 t_4}, H_2(id)^{-\tilde{r}_2 t_3}\right),$$

with $\tilde{r}_1 = \frac{r_1 F(id)}{F(id)b + J(id)}$, and $\tilde{r}_2 = r_2 + \frac{J(id) a r_1}{(t_3 t_4)(F(id)b + J(id))}$.

Challenge: The simulator gets from the adversary a message m which it can discard, and responds with a challenge ciphertext for the identity id^*. Posit that $s_1 = c$. To proceed, the simulator picks a random exponent $s_2 \in Z_p$ and a random element $R \in G_T$, and outputs the ciphertext as:

$$ct = (R, (g^s)^{J(id)}, Y, (g^{ac}), (g^s)^{t_3} g^{-s_2 t_3}, g^{s_2 t_4}) .$$

If $Y = g^{b(s-c)}$, i.e. $g^s = Z = g^{c+d}$ then $ct_{i1} = v_1^{s-s_1}$ and $ct_{i2} = v_2^{s_1}$. All elements of the challenge but $ct_i{}'$ are thus well formed and the simulator behaved as in Game 2. If instead Y is independent of a, b, s, s_1, s_2, which happens when Z is random, then the simulator responds as in Game 3.

Phase 2: The simulator answers the queries as in Phase 1.

Output: The adversary outputs a bit γ to guess which hybrid game the simulator has been playing. To conclude, the simulator forwards γ as its own answer in the Decision-Linear game.

Artificial Aborts are handled analogously to Lemma 1.

□

Lemma 3 (Anonymity part 2) *Under the Decision Linear assumption, no p.p.t. adversary can distinguish between the Games 3 and 4 with non-negligible advantage.*

Proof. Proof follows from that of anonymity part 1, except the simulation is done over the parameters v_3 and v_4 in place of v_1 and v_2. □

3.4.2 The Committed BlindExtract Protocol for the Anonymous IBE scheme

The underlying anonymous adaptive identity IBE scheme is presented in Section 3.4.1. The Setup, Encrypt and Decrypt algorithms remain as detailed. We present the protocol using commitments which are required for the application presented in Section 4.3.

Intuition behind the construction Generating a randomly distributed secret key in the BlindExtract protocol requires the values $\tilde{r}_1, \tilde{r}_2$ to be jointly chosen by the user and the key issuer in a manner which prohibits either party from learning anything about the other's randomness. This prevents a user that learns the issuer's randomness from potentially decrypting messages of other users and an issuer that learns a user's randomness from potentially breaking the blindness of the key issued.

The key issuer, $\mathcal{KGC}$, chooses random values $\hat{r}_1, \hat{r}_2 \leftarrow \mathbb{Z}_q^*$, and the user U picks random values $r'_1, r'_2 \leftarrow \mathbb{Z}_q^*$. The key generation protocol may be implemented using standard secure two-party computation techniques [Yao82], as a protocol in which the user inputs r'_1, r'_2 and the $\mathcal{KGC}$ inputs $\alpha, t_1, t_2, t_3, t_4, \hat{r}_1, \hat{r}_2$. The user's output in the protocol is a secret key

$$sk_{id} = (h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4}, h^{-\alpha t_2} H_2(id)^{-\tilde{r}_1 t_2}, h^{-\alpha t_1} H_2(id)^{-\tilde{r}_1 t_1}, H_2(id)^{-\tilde{r}_2 t_4}, H_2(id)^{-\tilde{r}_2 t_3}) \ ,$$

with $\tilde{r}_1 = \hat{r}_1 r'_1 \mod p$ and $\tilde{r}_2 = \hat{r}_2 r'_2 \mod p$. The $\mathcal{KGC}$ learns nothing further, and outputs nothing. By decomposing this protocol into sub-protocols whose results only require simple arithmetic operations (addition and multiplication), we achieve an efficient protocol.

Recall that $H_1(id) = g_0 \prod_{i=1}^{n} g_i^{id_i}$ and $H_2(id) = h_0 \prod_{i=1}^{n} h_i^{id_i}$. The committed blind anonymous IBE scheme consists of the algorithms Π of the underlying IBE scheme, the Pedersen commitment scheme Commit presented in Figure 2.1, and the following BlindExtract protocol presented in Figure 3.5.

BlindExtract($\mathcal{U}(params, id, open_{id}) \leftrightarrow \mathcal{KGC}(params, msk, C_{id})$).

1. The $\mathcal{KGC}$ chooses at random $\hat{r}_1, \hat{r}_2 \leftarrow \mathbb{Z}_q^*$, and the user $\mathcal{U}$ chooses at random $u_0, u_1, u_2 \leftarrow \mathbb{Z}_q$ and $u_3, r_1', r_2' \leftarrow \mathbb{Z}_q^*$. Implicitly, $\tilde{r}_1 = \hat{r}_1 r_1'$ and $\tilde{r}_2 = \hat{r}_2 r_2'$. $\mathcal{KGC}$ and $\mathcal{U}$ are required to run a two-party computation protocol for simple arithmetic computations. The input of $\mathcal{U}$ is r_1', r_2' and the blinding values u_0, u_1, u_2, u_3 and the input of $\mathcal{KGC}$ is $\alpha, t_1, t_2, t_3, t_4, \hat{r}_1, \hat{r}_2$.

   Additionally, $\mathcal{U}$ provides a commitment C_{u_3} to u_3, and $\mathcal{KGC}$ provides commitments $C_{\hat{r}_1}$ and $C_{\hat{r}_2}$ to $\hat{r}_1, \hat{r}_2$ respectively. They prove that their input corresponds to the committed values. $\mathcal{KGC}$ additionally proves that $\alpha, t_1, t_2, t_3, t_4$ corresponds to the master secret key.

   $\mathcal{KGC}$ obtains

$$\begin{aligned} x_0 &= (\hat{r}_1 r_1' t_1 t_2 + \hat{r}_2 r_2' t_3 t_4) + u_0 \pmod p , \\ x_1 &= -(u_3 / r_1' \cdot \alpha t_2) + u_1 \pmod p , \\ x_2 &= -(u_3 / r_1' \cdot \alpha t_1) + u_2 \pmod p . \end{aligned}$$

   Provided that $\mathcal{KGC}$ does not abort at that moment, $\mathcal{U}$ obtains commitments C_{x_0}, C_{x_1} and C_{x_2} to these values. Otherwise, both parties output $\perp$. In Section 3.4.3 we show how to efficiently realise such a protocol.

2. $\mathcal{U}$ computes $ID' = H_2(id)^{u_3}$ using the blinding value u_3. $\mathcal{U}$ executes a proof of knowledge with $\mathcal{KGC}$ to show that ID' is correctly constructed and that the identity id in ID' corresponds to the id committed to in C. Details about this proof of knowledge can be found in Section 3.4.3.

3. $\mathcal{KGC}$ returns $\perp$ if the proof fails, otherwise it computes

$$\begin{aligned} {sk_{id}}' =& (d_0', d_1', d_2', d_3', d_4') \\ & (h^{x_0}, h^{x_1} ID'^{-\hat{r}_1 t_2}, h^{x_2} ID'^{-\hat{r}_1 t_1}, ID'^{-\hat{r}_2 t_4}, ID'^{-\hat{r}_2 t_3}). \end{aligned}$$

4. $\mathcal{KGC}$ sends the blinded key ${sk_{id}}'$ to $\mathcal{U}$, and engages in a proof of knowledge that it is correctly constructed. Details about this proof of knowledge can be found in Section 3.4.3.

5. If the proof fails, $\mathcal{U}$ returns $\perp$. Otherwise, she computes

$$sk_{id} = (d_0, d_1, d_2, d_3, d_4) = (d_0' h^{-u_0}, (d_1' h^{-u_1})^{r_1'/u_3}, (d_2' h^{-u_2})^{r_1'/u_3}, {d_3'}^{r_2'/u_3}, {d_4'}^{r_2'/u_3}) .$$

Figure 3.5: BlindExtract protocol for Committed Anonymous IBE

Security

Theorem 2 *Under the DBDH and D-Linear assumptions, the blind IBE scheme* Π *is secure. That is, the interactive* BlindExtract *protocol provides a* leak-free *and* selective-failure blind *committed blind extraction protocol for the adapted anonymous IBE scheme* Π.

Proof. Leak freeness: Note that the simulator S can rewind an instance of the adversary $\mathcal{A}$ that he runs internally. He simulates the communication between the distinguisher D and $\mathcal{A}$ by passing D's input to $\mathcal{A}$ and $\mathcal{A}$'s output to D.

In the two party protocol S can provide random input. Using rewinding techniques, S extracts adversary $\mathcal{A}$'s input r'_1, r'_2, and u_0, u_1, u_2, u_3 to the two party computation protocol. In the next step of the blind issuing protocol $\mathcal{A}$ must send $ID' = H_2(id)^{u_3}$ together with a proof of knowledge of a correct representation of ID' and C_{id}. S uses its rewinding access to $\mathcal{A}$ in order to also extract id, and $open_{id}$.

Next S submits id, $open_{id}$ to $\mathcal{O}_{\mathsf{Extract}}$ to obtain a valid secret key $sk_{id} = (d_0, d_1, d_2, d_3, d_4)$. S returns $(d_0 \cdot h^{u_0}, d_1^{u_3/r'_1} h^{u_1}, d_2^{u_3/r'_1} h^{u_2}, d_3^{u_3/r'_2}, d_4^{u_3/r'_2})$ to $\mathcal{A}$. These values are distributed in the same way as in BlindExtract.

Selective-failure blindness: The adversary $\mathcal{A}$ provides $params$ and two identities id_0, id_1. The game chooses a random bit b $\mathcal{A}$ has blackbox access to two oracles $U(params, id_{1-b})$ and $U(params, id_b)$.

Note that once an oracle U is activated, $\mathcal{A}$ can run a two-party protocol with the oracle, the result of which are three randomly distributed values in $\mathbb{Z}_q$ (x_0, x_1, x_2). In the next step, the oracle provides a randomly distributed value $ID' \in \mathbb{G}_2$ to $\mathcal{A}$. Then the oracle performs a zero-knowledge proof with $\mathcal{A}$.

Suppose that $\mathcal{A}$ runs one or both of the oracles up to this point. Up to now the distributions of the two oracles are computationally indistinguishable. (Otherwise we could break the security of the two party computation, the hiding property of the commitment scheme or the witness indistinguishability of the zero-knowledge proof. The latter is implied by the zero-knowledge property of the proof system.)

$\mathcal{A}$ must provide values $(d'_0, d'_1, d'_2, d'_3, d'_4)$ and a proof that these values were correctly computed. We can assume that $\mathcal{A}$ chooses these values using an arbitrary complex strategy. We show that any adversary $\mathcal{A}$ can predict the output sk_i of U without further interaction with the oracles:

1. $\mathcal{A}$ does the proof of Step 4 internally with itself. If the proof fails, it records $sk_0 = \perp$. Otherwise, the adversary temporarily records $sk_0 = \mathsf{Extract}(params, msk, id_0)$.

2. In turn, $\mathcal{A}$ generates different $(d'_0, d'_1, d'_2, d'_3, d'_4)$ and executes a second proof of knowledge (again internally), now for the second oracle. It performs the same checks and recordings for sk_1 and id_1.

3. Finally the adversary $\mathcal{A}$ predicts (sk_0, sk_1) if both $sk_0 \neq \perp$ and $sk_1 \neq \perp$. $\mathcal{A}$ predicts $(\epsilon, \perp)$if only $sk_1 = \perp$. $\mathcal{A}$ predicts $(\perp, \epsilon)$ if only $sk_0 = \perp$. Finally, $\mathcal{A}$ predicts $(\perp, \perp)$, if $sk_0 = sk_1 = \perp$.

These predictions result in the same distributions as that returned by the oracle, as the same checks are performed. Moreover, note that for the case that keys are returned by the game they are in both cases uniformly distributed random keys because of the random values r'_1 and r'_2 contributed by the oracles. □

3.4.3 Subprotocols for Blind Key Derivation

Figure 3.5 outlines the need for subprotocols in the construction of the blind key derivation protocol presented therein. Here, we outline the protocols required for each step sequentially, beginning with the two-party protocols required for Step 1 and then detailing the proofs of knowledge required to be run in Step 2 and Step 4.

Two-Party Protocol for Simple Arithmetics The two-party protocols required use an additive homomorphic encryption scheme. Such a scheme has encryption and decryption functions Enc and Dec such that $Enc(x) \otimes y = Enc(xy)$ and $Enc(x) \oplus Enc(y) = Enc(x + y)$. A key pair is generated by $\mathcal{KGC}$ and is made available to $\mathcal{U}$. We provide two protocols, one to compute x_1, shown in Figure 3.6 and a second to compute x_2, shown in Figure 3.7. To compute x_3, follow the protocol for x_2 using u_3 and t_2 in place of u_2 and t_1.

Proofs of Knowledge of Correct Key Derivation

Proof for Step 2 The $\mathcal{KGC}$ has commitment C_{u_3} to u_3, and C_{id} to the user's choice of id. In Step 2 of the BlindExtract protocol the user executes the following proof of knowledge to convince the

User($r'_1, r'_2, u_1, v_1, \ldots, v_4, C_{\hat{r}_1}, C_{\hat{r}_2}$)		KGC($\hat{r}_1, \hat{r}_2, t_1, t_2, t_3, t_4$)
		$e_1 = Enc(\hat{r}_1 t_1 t_2)$
	$\xleftarrow{e_1, e_2}$	$e_2 = Enc(\hat{r}_2 t_3 t_4)$
	$\xleftrightarrow{PoK_1}$	
$e_{x_1} = (e_1 \otimes r'_1) \oplus (e_2 \otimes r'_2) \oplus Enc(u_1)$	$\xrightarrow{e_{x_1}}$	
	$\xleftrightarrow{PoK_2}$	
		$x_1 = Dec(e_{x_1}) =$
		$\hat{r}_1 r'_1 t_1 t_2 + \hat{r}_2 r'_2 t_3 t_4 + u_1$
		$open_{x_1} \leftarrow \mathbb{Z}_p$
	$\xleftarrow{C_{x_1}}$	$C_{x_1} = h_0^{x_1} h_1^{open_{x_1}}$
	$\xleftrightarrow{PoK_3}$	
$PoK_1 = PoK\{(\rho_1, \rho_2, \tau_1, \tau_2, \tau_3, \tau_3) :$		
$v_1 = g^{\tau_1} \wedge v_2 = g^{\tau_2} \wedge v_3 = g^{\tau_3} \wedge v_4 = g^{\tau_4} \wedge$		
$C_{\hat{r}_1} = Commit(\rho_1, open_{\rho_1}) \wedge C_{\hat{r}_2} = Commit(\rho_2, open_{\rho_2}) \wedge$		
$e_1 = Enc(\rho_1 \tau_1 \tau_2) \wedge e_2 = Enc(\rho_2 \tau_3 \tau_4)\}$		
$PoK_2 = PoK\{(\rho'_1, \rho'_2, \mu_1) : e_{x_1} = (e_1 \otimes \rho'_1) \oplus (e_2 \otimes \rho'_2) \oplus \mu_1\}$		
$PoK_3 = PoK\{(\chi, open_\chi) : e_{x_1} = Enc(\chi) \wedge C_{x_1} = h_0^\chi h_1^{open_\chi}\}$		

Figure 3.6: Protocol for deriving x_1 in Step 1

User($r'_1, u_0, u_2, v_1, v_2, \Omega$)		KGC(ω, t_1, t_2)
	$\xleftarrow{e}$	$e = Enc(\omega t_2)$
	$\xleftrightarrow{PoK_1}$	
$e_{x_2} = ((e \otimes u_0 / r'_1) \oplus Enc(u_2)) \otimes -1$	$\xrightarrow{e_{x_2}}$	
	$\xleftrightarrow{PoK_2}$	
		$x_2 = Dec(e_{x_2}) =$
		$-((u_0/r'_1)\omega t_2 + u_2)$
		$open_{x_2} \leftarrow \mathbb{Z}_p$
	$\xleftarrow{C_{x_2}}$	$C_{x_2} = h_0^{x_2} h_1^{open_{x_2}}$
	$\xleftrightarrow{PoK_3}$	
$PoK_1 = PoK\{(\omega', \tau_1, \tau_2) :$		
$v_1 = g^{\tau_1} \wedge v_2 = g^{\tau_2} \wedge \Omega = e(g,h)^{t_1 t_2 \omega'} \wedge e = Enc(\omega' t_2)\}$		
$PoK_2 = PoK\{(\rho'_1, \mu_0, \mu_2) : e_{x_2} = -((\mu_0/\rho'_1 \otimes e) \oplus \mu_2)\}$		
$PoK_3 = PoK\{(\chi, open_\chi) : e_{x_2} = Enc(\chi) \wedge C_{x_2} = h_0^\chi h_1^{open_\chi}\}$		

Figure 3.7: Protocol for deriving x_2 in Step 1

$\mathcal{KGC}$ that her message ID' is well formed:

$$
\begin{aligned}
&PoK\{(id_1,\ldots,id_n,u_3,id_1\cdot u_3,\ldots,id_n\cdot u_3,open_{id},open_{u_3},open_{id}\cdot u_3):\\
&C_{id}=(\prod_{i=1}^{n}(h_0^{2^{l(i-1)}})^{id_i})h_1^{open_{id}}\wedge\bigwedge_{i=1}^{n}0\leq id_i<2^l\wedge C_{u_3}=h_0^{u_3}h_1^{open_{u_3}}\wedge 1\\
&=C_{id}^{u_3}(\prod_{i=1}^{n}((1/h_0)^{2^{l(i-1)}})^{id_i\cdot u_3})(1/h_1)^{open_{id}\cdot u_3}\wedge ID'=h_0^{u_3}\prod_{i=1}^{n}h_i^{id_i\cdot u_3}\}\ .
\end{aligned}
$$

The user proves that id is correctly encoded in ID'. This is the step during which $\mathcal{U}$ proves that the identity that she submits to $\mathcal{KGC}$ is the identity that is contained in the commitment.

Proof for Step 4 The user has commitments $C_{\hat{r}_1}$, $C_{\hat{r}_2}$ and C_{x_0}, C_{x_1}, and C_{x_2}. In Step 4 of the BlindExtract protocol, the $\mathcal{KGC}$ performs the following proof of knowledge to convince the user that the blinded key $(d_0',d_1',d_2',d_3',d_4')$ it returns is well formed:

$$
\begin{aligned}
&PoK\{(\hat{r}_1,\hat{r}_2,open_{\hat{r}_1},open_{\hat{r}_2},t_1,t_2,t_3,t_4,x_0,x_1,x_2,open_{x_0},\\
&\quad open_{x_1},open_{x_2},-\hat{r}_1t_1,-\hat{r}_1t_2,-\hat{r}_2t_3,-\hat{r}_2t_4):\\
&\quad C_{\hat{r}_1}=h_0^{\hat{r}_1}h_1^{open_{\hat{r}_1}}\wedge C_{\hat{r}_2}=h_0^{\hat{r}_2}h_1^{open_{\hat{r}_2}}\wedge v_1=g^{t_1}\wedge\\
&\quad v_2=g^{t_2}\wedge v_3=g^{t_3}\wedge v_4=g^{t_4}\wedge C_{x_0}=h_0^{x_0}h_1^{open_{x_0}}\wedge\\
&\quad C_{x_1}=h_0^{x_1}h_1^{open_{x_1}}\wedge C_{x_2}=h_0^{x_2}h_1^{open_{x_2}}\wedge\\
&\quad 1=(1/v_1)^{\hat{r}_1}(1/g)^{-\hat{r}_1t_1}\wedge 1=(1/v_2)^{\hat{r}_1}(1/g)^{-\hat{r}_1t_2}\wedge\\
&\quad 1=(1/v_3)^{\hat{r}_2}(1/g)^{-\hat{r}_2t_3}\wedge 1=(1/v_4)^{\hat{r}_2}(1/g)^{-\hat{r}_2t_4}\wedge\\
&\quad d_0'=h^{x_0}\wedge d_1'=h^{x_1}ID'^{-\hat{r}_1t_2}\wedge d_2'=h^{x_2}ID'^{-\hat{r}_1t_1}\wedge\\
&\quad d_3'=ID'^{-\hat{r}_2t_4}\wedge d_4'=ID'^{-\hat{r}_2t_3}\}\ .
\end{aligned}
$$

By means of this proof the $\mathcal{KGC}$ demonstrates to the user that it uses the correct values for $x_0,x_1,x_2,t_1,t_2,t_3,t_4,\hat{r}_1,\hat{r}_2$ when it computes $(d_0',d_1',d_2',d_3',d_4')$. The proof involves proving the multiplicative relations $-\hat{r}_1t_1,-\hat{r}_1t_2,-\hat{r}_2t_3,-\hat{r}_2t_4$ between $t_1,t_2,t_3,t_4,\hat{r}_1,\hat{r}_2$.

3.5 Partially-Blind Identity-Based Encryption

We extend blind IBE to incorporate the property of *partial-blindness*. In a partially-blind IBE scheme, elements of the identity string are visible to the $\mathcal{KGC}$. Such elements could include va-

lidity date, security clearance and other generic data that proves adherence to a set of regulations without either revealing personally identifying information about the user or incurring the overhead of proofs of knowledge.

In work with Gray [SG09], we construct a partially-blind IBE scheme consisting of an IBE scheme Π where the Extract algorithm is replaced with an interactive protocol PartialBlindExtract. A partially-blind IBE scheme is a generalisation of blind IBE; that is a fully blind IBE scheme is merely an instance of a partially blind IBE scheme where the set of non-blinded elements is empty. A fully blind scheme occurs when all identity string elements are not disclosed to the $\mathcal{KGC}$, a partially blind scheme occurs when some of the identity string elements are not disclosed to the $\mathcal{KGC}$, and a standard IBE scheme occurs when all of the identity string elements are disclosed to the $\mathcal{KGC}$.

PartialBlindExtract($\mathcal{U}(params, id, info), \mathcal{KGC}(params, msk)) \rightarrow (sk_{id}, info)$ returns a private decryption key sk_{id} to $\mathcal{U}$ corresponding to identity string id and the commonly agreed public information $info$ to $\mathcal{KGC}$. The identity string consists of a partially blinded identity set containing blinded id and non blinded $info$ elements, in an interactive key issuing protocol between $\mathcal{U}$ and the $\mathcal{KGC}$.

3.5.1 The PartialBlindExtract Protocol for Waters' IBE Scheme

This scheme is a modification of Naccache's scheme [Nac07] to produce a partially-blind IBE scheme. As with fully blind-IBE, the Extract stage of the IBE scheme is altered from a polynomial algorithm to an interactive protocol - the Setup, Encrypt and Decrypt algorithms remain unchanged. To produce a partially-blind scheme, the $\mathcal{KGC}$ must be able to generate a private key d_v for a given public key v such that some or all of the v_i remain unknown to it.

The scheme

A user $\mathcal{U}$ wants to retrieve a well formed key from $\mathcal{KGC}$ without $\mathcal{KGC}$ being able to associate the full identity $v = (v_1, \cdots, v_n)$ with the extraction instance. Two random values are introduced by $\mathcal{U}$ which prevent $\mathcal{KGC}$ matching the value it generated with a particular instance of the extract protocol. This is achieved by $\mathcal{KGC}$ producing a blinded version of the key required, and $\mathcal{U}$ retrieving the actual key by unblinding.

Begin by assuming the user $\mathcal{U}$ knows the full identity $v = (v_1, \ldots, v_n)$ for which she requires

a private key, a necessary condition for leak-freeness. $\mathcal{U}$ wants to retrieve a well formed private key from $\mathcal{KGC}$ without $\mathcal{KGC}$ being able to associate the full identity v with the extraction instance. The PartialBlindExtract protocol is presented in Figure 3.8.

The security of the scheme

The private key resulting from PartialBlindExtract has the same form as the private key generated by Extract in the original scheme; as such, the Encrypt and Decrypt algorithms remain correct. $\mathcal{U}$ is in possession of a private key d_v corresponding to v, and $\mathcal{KGC}$ has learnt nothing other than the non-blinded elements $\hat{v}$ of v. As with the Naccache scheme [Nac07], multiple decryption keys corresponding to the identity v can be generated. Naccache's scheme is proven semantically secure against passive adversaries (IND-ID-CPA) in the standard model under the DBDH assumption; our scheme holds the same security as the key is of the same form.

The definition of leak-freeness, Definition 14, is applicable to partially-blind IBE. However, the definition of selective-failure blindness, Definition 15, requires that an authority learn nothing about an identity; in partially-blind IBE the $\mathcal{KGC}$ does learn something about the identity. To allow for partial-blindness, construct the pair of identities id_0, id_1 to include the non blinded information $info$, but restrict the non-blinded information in each identity id_0, id_1 to be precisely $info$. An adversary viewing id_0, id_1 should be unable to distinguish them, as his view will consist of ($info$, $random$) for both identities. To incorporate this requirement, the following definition is proposed for partial-blindness.

Definition 18 (Selective-failure Partial-Blindness)

A PartialBlindExtract *protocol is said to be selective-failure blind if every adversary $\mathcal{A}$ has a negligible advantage in the following game: $\mathcal{A}$ outputs $params$, $info$ and a pair of identities id_0, id_1 which both contain $info$ visible to $\mathcal{A}$. A random bit $b \in \{0, 1\}$ is chosen, and $\mathcal{A}$ is given black-box access to two oracles: $U(params, id_b)$ and $U(params, id_{1-b})$. The U algorithms produce sk_b, sk_{1-b} respectively. If both $sk_b \neq \perp$, $\mathcal{A}$ receives (sk_0, sk_1); if only $sk_{1-b} = \perp$, $\mathcal{A}$ receives $(\epsilon, \perp)$; if only $sk_b = \perp$, $\mathcal{A}$ receives $(\perp, \epsilon)$; and if $sk_b = sk_{1-b} = \perp$, $\mathcal{A}$ receives $(\perp, \perp)$. Finally, $\mathcal{A}$ outputs its guess b'. The advantage of $\mathcal{A}$ in this game is $|\mathbb{P}[b' = b] - 1/2|$.*

Definition 19 (Secure Partially-Blinded IBE)

An IBE scheme Π is secure if and only if: (1) the underlying Π is a secure IBE scheme and (2) PartialBlindExtract *is leak-free and selective-failure partially-blind.*

PartialBlindExtract

1. Assume $\mathcal{U}$ knows the identity $v = (v_1, \ldots, v_n)$ and a vector $\gamma = (\gamma_1, \ldots, \gamma_n) \in (\{0,1\})^n$ such that if $\gamma_i = 1$ then v_i is to be blinded. Define $\hat{v} = (\hat{v}_i)$ where $\hat{v}_i = v_i$ if $\gamma_i = 0$ and $\perp$ otherwise. Also define $\overline{v} = (\overline{v}_i)$ where $\overline{v}_i = v_i$ if $\gamma_i = 1$ and $\perp$ otherwise. Note that $v = \hat{v}|\overline{v}$, where $|$ represents the merger of the non-$\perp$ components of $\hat{v}$ and $\overline{v}$. $\mathcal{U}$ chooses random values $\beta, y \in \mathbb{Z}_q$ and computes $\vec{v} = (\vec{v}_1, \ldots, \vec{v}_n)$ where

$$\vec{v_i} = \begin{cases} (u_i^{\beta}, v_i) & \text{if } \gamma_i = 0 \\ \perp & \text{if } \gamma_i = 1 \end{cases}$$

   $\mathcal{U}$ computes $X \leftarrow (g^{\beta y} u'^{\beta} \prod_{i=1,\gamma_i=1}^{n} u_i^{\beta v_i})$ and sends $(X, \vec{v}, g^{\beta}, u'^{\beta})$ to $\mathcal{KGC}$. $\mathcal{U}$ can prove to $\mathcal{KGC}$ that it knows y, β, v_i where $\gamma_i = 1$ using zero-knowledge proofs as outlined in [Oka06a].

2. $\mathcal{KGC}$ chooses random $r \in \mathbb{Z}_q$ and constructs $d'_v = (d'_0, d'_1)$ as

$$\begin{aligned} d'_v &= \big(g_2^{\alpha}(u'^{\beta} \prod_{i=1,\gamma_i \neq 1}^{n} u_i^{\beta v_i})^r X^r, g^{\beta r}\big) \\ &= \big(g_2^{\alpha}(u'^{\beta} \prod_{i=1,\gamma_i \neq 1}^{n} u_i^{\beta v_i})^r (g^{\beta y} u'^{\beta} \prod_{i=1,\gamma_i=1}^{n} u_i^{\beta v_i})^r, g^{\beta r}\big) \\ &= \big(g_2^{\alpha} g^{\beta y r} (u' \prod_{i=1}^{n} u_i^{v_i})^{\beta r}, g^{\beta r}\big) \end{aligned}$$

   and passes d'_v to $\mathcal{U}$.

   $\mathcal{U}$ then tests that $e(g_1, g_2) \cdot e(d'_1, g^{y \prod_{i=1}^{n} u_i^{v_i}}) = e(g, d'_1)$.

3. If the test passes, $\mathcal{U}$ chooses random $z \in \mathbb{Z}_q$ and computes

$$\begin{aligned} d_v &= \big(d'_0/(d'_1)^y \cdot (u'^{\beta} \prod_{i=1}^{n} u_i^{\beta v_i})^z, d'_1 \cdot g^z\big) \\ &= \big(g_2^{\alpha}(u' \prod_{i=1}^{n} u_i^{v_i})^{\beta r + z}, g^{\beta r + z}\big). \end{aligned}$$

   If the test fails, $\mathcal{U}$ outputs $\perp$ and aborts. Note that $\mathcal{KGC}$ does not know d_0 or d_1.

Figure 3.8: PartialBlindExtract protocol for Waters' IBE

Lemma 4 *The scheme* Π *is leak-free.*

Proof. The proof follows the same form as that presented in [GH07]. In the Real Game an adversary $\mathcal{A}$ interacts with an honest $\mathcal{KGC}$ executing the PartialBlindExtract protocol.

Construct a simulator $\mathcal{S}$ such that no efficient distinguisher $\mathcal{D}$ can distinguish the Real Game from the Ideal Game. In the ideal game, an adversary $\mathcal{S}$ given access to a trusted party executing Extract is described as:

1. On input $params$ from the $\mathcal{KGC}$, S passes $params$ to a copy of $\mathcal{A}$ that it runs internally.

2. Each time $\mathcal{A}$ engages S in a PartialBlindExtract protocol, $\mathcal{S}$ behaves in a predetermined manner. In the first message of the protocol $\mathcal{A}$ must send $\mathcal{S}$ a value of the form $(X, \vec{v})$. $v = (v_1, \ldots, v_n)$ is an identity and $\gamma = (\gamma_1, \ldots, \gamma_n) \in (\{0,1\})^n$ is a vector such that if $\gamma_i = 1$, v_i is to be blinded. Blinded values are sent as $X = (u'^{\beta} \prod_{i=1,\gamma_i=1}^{n} u_i^{v_i\beta})$; non-blinded values are sent as the vector $\vec{v} = (u_i^{\beta}, v_i)$. $\mathcal{A}$ constructs proofs of knowledge of the values β, v_i for $\gamma_i = 1$. If the proof fails to verify, $\mathcal{S}$ aborts. Otherwise, using extraction techniques, $\mathcal{S}$ can extract $v = (v_1, \ldots, v_n)$, β and y.

3. Next, $\mathcal{S}$ submits identity v to the $\mathcal{KGC}$, who returns the valid private key $d_v = (g_2^{\alpha} \cdot (u' \prod_{i=1}^{n} u_i^{v'_i})^r, g^r)$, where $r = r'\beta$.

4. $\mathcal{S}$ computes $d'_v = (d'_1, d'_2)$ using the blinding value β, y, and returns these values to $\mathcal{A}$.

The responses of $\mathcal{S}$ are always well formed, which $\mathcal{A}$ can easily verify, and are drawn from the same distribution as those of the $\mathcal{KGC}$. Thus the games Real and Ideal are indistinguishable to $\mathcal{A}$ and $\mathcal{D}$. □

Lemma 5 *The* PartialBlindExtract *protocol is selective-failure partially-blind.*

Proof. The proof follows the same form that presented in [GH07].

Adversary $\mathcal{A}$ outputs $params$, $info$ and two identities id_0, id_1, both of which contain $info$. A random bit b is chosen, and $\mathcal{A}$ is given black-box access to two oracles $\mathcal{U}$ ($params$, id_b, $info$) and $\mathcal{U}$ ($params$, id_{b-1}, $info$). The $\mathcal{U}$ algorithms produce local output sk_b and sk_{b-1} respectively. If $sk_b \neq \perp$ and $sk_{b-1} \neq \perp$ then $\mathcal{A}$ receives sk_0, sk_1; if $sk_b = \perp$ and $sk_{b-1} \neq \perp$, $\mathcal{A}$ receives $(\perp, \epsilon)$; if $sk_b \neq \perp$ and $sk_{b-1} = \perp$, $\mathcal{A}$ receives $(\epsilon, \perp)$; if $sk_b = \perp$ and $sk_{b-1} = \perp$, $\mathcal{A}$ receives $(\perp, \perp)$. $\mathcal{A}$ then tries to predict b, and he is able to do so only with a negligible advantage over guessing.

In the PartialBlindExtract protocol, both parties agree in advance some $info$ which is contained in the identities id_0, id_1. $\mathcal{U}$ then constructs the value $X = (u' \prod_{i=1}^{n} (u_i^{v_i})^{b_i})^{\beta}$, which is the blinded remainder of the id_b identity string. $\mathcal{U}$ performs a proof of knowledge $PoK(u', u_i, v_i, \beta) : X$. $\mathcal{U}$ also passes $(u'^{\beta}, (u_i^{\beta}, v_i))$ to $\mathcal{A}$, who can then construct $(u' \prod_{i=1}^{n} (u_i^{v_i}))^{\beta}$, where $b_i = 0$ for the non blinded $info$ values.

Suppose that $\mathcal{A}$ runs one or both of his oracles up to this point. $\mathcal{A}$ must respond to $\mathcal{U}$, and thus far his views are computationally indistinguishable. $\mathcal{A}$ must now return two values d_1', d_2' to the first oracle. $\mathcal{A}$ chooses this pair in any manner he wants, and once he chooses d_1', d_2' he is able to predict the output sk_{id} of the oracle $\mathcal{U}$ ($params$, id_i, $info$) as follows:

1. $\mathcal{A}$ checks that (d_1', d_2') are correctly constructed, i.e., of the form $(g_2^{\alpha}(u' \prod_{i=1}^{n} (u_i^{v_i})^{\beta r}, g^{\beta r}))$. If they are not, record $sk_0 = \perp$. If they are, $\mathcal{A}$ temporarily records $sk_0 = \mathsf{PartialBlindExtract}(msk, id_0, info)$.

2. $\mathcal{A}$ chooses any two values (d_0', d_1') for the second oracle, performs the same check and records for sk_1, id_1.

3. Finally, if both tests failed or both tests succeeded, output (sk_0, sk_1). If $sk_0 = \perp$ and $sk_1 \neq \perp$ output $(\epsilon, \perp)$.

The prediction is correct, because $\mathcal{A}$ is performing the same check as the honest $\mathcal{U}$, and when both succeed $\mathcal{A}$ outputs a valid secret key from $\mathsf{PartialBlindExtract}(msk,\ id,\ info)$, as does $\mathcal{U}$. Note that if $\mathcal{A}$ is able to predict the final output of its oracles, then its advantage in distinguishing $\mathcal{U}$ ($params$, id_0, $info$) and $\mathcal{U}$ ($params$, id_1, $info$) is the same without the output from the predictions. Thus, all of $\mathcal{A}$'s advantage must come from distinguishing the earlier output of the oracles. We know from the underlying proof of security that the actions the oracles undertake mean their output is indistinguishable to $\mathcal{A}$.

□

3.6 Double-Blind Identity-Based Encryption

We propose the novel concept of *double blinding* [SG09] to allow a $\mathcal{KGC}$ to add elements of its choosing to an identity string without revealing them to the user $\mathcal{U}$. The resulting identity string comprises three types elements; those disclosed to both parties, those not disclosed by $\mathcal{U}$ to $\mathcal{KGC}$ and those not disclosed by $\mathcal{KGC}$ to $\mathcal{U}$.

A *double-blind* IBE scheme consists of an IBE scheme Π, where the Extract algorithm is replaced with an interactive protocol DoubleBlindExtract. A double-blind IBE scheme is also a generalisation of blind IBE; that is, a fully blind IBE scheme is an instance of a double-blind IBE scheme where the set of non-blinded elements is empty. In DoubleBlindExtract the identity string id consists of non-blinded elements $\hat{id}$ known to both $\mathcal{U}$ and $\mathcal{KGC}$, blinded elements $\overline{id}$ known only to $\mathcal{U}$, and double-blind elements $\overline{\overline{id}}$ known only to the $\mathcal{KGC}$, such that $id = \hat{id}|\overline{id}|\overline{\overline{id}}$.

The interactive key issuing protocol between $\mathcal{U}$ and $\mathcal{KGC}$ is described as follows:

DoubleBlindExtract$(\mathcal{U}(params, \hat{id}|\overline{id}), \mathcal{KGC}(params, msk, \overline{\overline{id}})) \rightarrow (sk_{\hat{id}|\overline{id}|\overline{\overline{id}}}, \hat{id})$ returns a private key $sk_{\hat{id}|\overline{id}|\overline{\overline{id}}}$ to $\mathcal{U}$ that corresponds to the identity string $\hat{id}|\overline{id}|\overline{\overline{id}}$ provided; the non-blinded elements $\hat{id}$ of the identity are returned to $\mathcal{KGC}$.

3.6.1 The DoubleBlindExtract Protocol for Waters' IBE Scheme

To construct a double-blind IBE scheme, we use the approach of existing blind IBE extraction protocols and alter the Extract stage of the IBE scheme. A double-blind scheme does not follow the standard assumption that $\mathcal{U}$ knows the full identity $v = (v_1, \ldots, v_n)$ for which she requires a private key. Instead, assume an identity $v = (v_1, \ldots, v_n)$ such that both parties know elements $\hat{v} = v_i$ where $i \in \{1, m\}$ and $\gamma_i = 0$, only $\mathcal{U}$ knows elements $\overline{v} = v_i$ where $i \in \{1, m\}$ and $\gamma_i = 1$, and only $\mathcal{KGC}$ knows elements $\overline{\overline{v}} = (v_{m+1}, \ldots, v_n)$. The DoubleBlindExtract protocol is presented in Figure 3.9.

As with the PartialBlindExtract scheme, the resulting private key has the same form as the private key generated by Extract in the Naccache scheme [Nac07], so the Encrypt and Decrypt algorithms remain correct.

DoubleBlindExtract($\mathcal{U}(params, \hat{id}|\overline{id}), \mathcal{KGC}(params, msk, \overline{\overline{id}})$)

1. Given a vector $\gamma = (\gamma_1, \ldots, \gamma_n) \in (\{0,1\})^n$, if $\gamma_i = 1$ then v_i is to be blinded by $\mathcal{U}$. We define a vector $\overrightarrow{v}$ of length n such that

$$\overrightarrow{v} = \begin{cases} (u_i^{\beta}, v_i) & \text{if } i \in \{0, m\} \wedge \gamma_i = 0 \\ \perp & \text{if } i \in \{0, m\} \wedge \gamma_i = 1 \\ u_i^{\beta} & \text{if } m < i < n \end{cases}$$

   $\mathcal{U}$ computes $X \leftarrow (g^{\beta y} u'^{\beta} \prod_{i=1,\gamma_i=1}^{m} u_i^{\beta v_i})$ and sends $(X, \overrightarrow{v}, g^{\beta}, u'^{\beta})$ to $\mathcal{KGC}$. $\mathcal{U}$ can prove to $\mathcal{KGC}$ that it knows y, β and v_i where $\gamma_i = 1$ using zero-knowledge proofs as outlined in [Oka06a].

2. $\mathcal{KGC}$ chooses random $r \in \mathbb{Z}_q$ and constructs $d'_v = (d'_1, d'_2)$ as

$$\begin{aligned} d'_v &= \big(g_2^{\alpha} (\prod_{i=1,\gamma_i \neq 1}^{m} u_i^{\beta v_i})^r X^r (\prod_{i=m+1}^{n} u_i^{\beta v_i})^r, g^{\beta r}\big) \\ &= \big(g_2^{\alpha} g^{\beta y r} (u' \prod_{i=1}^{n} u_i^{v_i})^{\beta r}, g^{\beta r}\big) \end{aligned}$$

   computes $f = (\prod_{i=m+1}^{n} u_i^{v_i})^r$ and passes d'_v, f to $\mathcal{U}$. This f value is required to allow $\mathcal{U}$ check that the key is correctly constructed. In order to prevent f from potentially leaking values v_i where $i = m+1, \ldots, n$, $\mathcal{KGC}$ blinds it using r.

   $\mathcal{U}$ then tests that

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{m} u_i^{v_i}) \cdot e(g^{\beta}, (\prod_{i=m+1}^{n} u_i^{v_i})^r) = e(g, d'_1).$$

3. If the test passes, $\mathcal{U}$ chooses random $z \in \mathbb{Z}_q$ and computes

$$\begin{aligned} d_v &= \big(d'_1/(d'_2)^y \cdot (u' \prod_{i=1}^{n} u_i^{v_i})^z, d'_1 \cdot g^z\big) \\ &= \big(g_2^{\alpha} (u' \prod_{i=1}^{n} u_i^{v_i})^{\beta r + z}, g^{\beta r + z}\big). \end{aligned}$$

   If the test fails, $\mathcal{U}$ outputs $\perp$ and aborts. Note that the $\mathcal{KGC}$ does not know d_1 or d_2, where $d_v = (d_1, d_2)$.

Figure 3.9: DoubleBlindExtract protocol for Waters' IBE

$\mathcal{U}$ checks key correctness in step 2. This check reduces to $e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{n} u_i^{v_i}) = e(g, d'_1)$, the check for previous schemes by:

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{m} u_i^{v_i}) \cdot e(g^{\beta}, (\prod_{i=m+1}^{n} u_i^{v_i})^r) = e(g, d'_1)$$

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{m} u_i^{v_i}) \cdot e(g^{\beta \frac{r}{r}}, (\prod_{i=m+1}^{n} u_i^{v_i})^r) = e(g, d'_1)$$

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{m} u_i^{v_i}) \cdot e(g^{\beta r}, (\prod_{i=m+1}^{n} u_i^{v_i})^{\frac{r}{r}}) = e(g, d'_1)$$

$$e(g_1, g_2) \cdot e(g^{\beta r}, g^y u' \prod_{i=1}^{m} u_i^{v_i}) \cdot e(g^{\beta r}, (\prod_{i=m+1}^{n} u_i^{v_i})) = e(g, d'_1)$$

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{n} u_i^{v_i}) = e(g, d'_1).$$

The blinding of the f value by $\mathcal{KGC}$ in this manner means that all communications can be observed in this protocol by a third party, as it is not possible for it to perform an exhaustive search for the v_i values contained therein.

The security of the scheme

It is possible for $\mathcal{U}$ to perform an exhaustive search for $v_{m+1}, \ldots, v_n$ by constructing a ciphertext and then checking if her d_v decrypts it correctly. To prevent this, it is necessary for $\mathcal{KGC}$ to set a minimum required level of bit security. For example, if 128-bit security is required, then $\mathcal{KGC}$ is required to add $n - m + 1 = \frac{128}{l}$ double blinded v_i values.

Unusually, double-blind IBE schemes require that the user applying for a private key does not know the full corresponding public key $id = \hat{id}|\overline{id}|\overline{\overline{id}}$. We capture this requirement by introducing the concept of *Ciphertext Awareness* (CTA). Informally, CTA is a requirement that a decrypting entity can only produce a valid plaintext by applying the decryption algorithm to his private key and a ciphertext encrypted using the corresponding public key. This concept mirrors that of *Plaintext Awareness* (PA) [BDPR98, BR95, BD08, TO08], which models an adversary's inability to produce a ciphertext without knowledge of the underlying plaintext. That is, if a scheme is PA, then the only way an adversary can produce a valid ciphertext is to apply the encryption algorithm to the public key. Similarly, CTA models an adversary's inability to produce a plaintext / ciphertext pair x, y without knowing the ciphertext y.

Consider an adversary $\mathcal{A}$ for ciphertext awareness, given the secret key sk and access to an

Real Game	**Ideal Game**
$(pk, sk) \leftarrow \mathsf{KeyGen}(params, msk, pk)$ $x_{Real} \leftarrow \mathcal{A}^{\mathsf{Encrypt}(pk,\cdot),\mathsf{Decrypt}(sk,C(\cdot))}(sk)$	$(pk, sk) \leftarrow \mathsf{KeyGen}(params, msk, pk)$ $x_{Ideal} \leftarrow \mathcal{A}^{\mathcal{A}^*(\mathsf{Encrypt}(pk,\cdot,R,\mathcal{O}_{list}),\mathsf{Decrypt}(sk,C(\cdot))}(sk)$

Figure 3.10: Security game for Ciphertext Awareness

encryption oracle $\mathcal{O}$. $\mathcal{A}$ is also given access to a second oracle $\mathcal{D}_{sk}^{\mathcal{O}}$. This second oracle is used to model the ability of an oracle to access valid plaintexts without the corresponding ciphertexts that it would get using queries to $\mathcal{O}$, where such plaintexts are denoted $R[\mathcal{A}]$. By querying $\mathcal{O}$, $\mathcal{A}$ has access to a decryption oracle that will, on input ciphertext C, extract the corresponding plaintext P, add P and C to a list of queried plaintexts $\mathcal{O}_{list}$ and return P.

In the real game, $\mathcal{A}$ can query an encryption oracle on any plaintext $P \notin \mathcal{O}_{list}$, and the oracle will return $\mathsf{Encrypt}(pk, P)$. In the ideal game, $\mathcal{A}$ can query an encryption oracle on any plaintext $P \notin \mathcal{O}_{list}$ and the oracle will execute the ideal simulation $\mathcal{A}^*(pk, P, R[\mathcal{A}], \mathcal{O}_{list})$ and return the result. The two games are summarised in Figure 3.10.

Definition 20 (Ciphertext Awareness) *A double-blind IBE scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is ciphertext aware if for all polynomial-time plaintext extractors* $\mathcal{A}$*, there exists a polynomial-time ciphertext creator*$\mathcal{A}^*$ *such that for an efficient distinguisher* D*, the advantage*

$$Adv_{\mathcal{A},\mathcal{A}^*,C,\mathsf{Encrypt}} = |\Pr[D(x_{Real})] - \Pr[D(x_{Ideal})]|$$

is negligible.

The definitions of security for blind and partial-blind IBE can be extended to double-blinded IBE schemes by constructing the pair of identities $\hat{id}|\overline{id}_0|\overline{\overline{id}}$, $\hat{id}|\overline{id}_1|\overline{\overline{id}}$. It is necessary to restrict the non-blinded information for each identity to be a common value for $\hat{id}$ and the double-blind information to be a common value $\overline{\overline{id}}$ also. Thus, as with partially-blind IBE, the elements visible to $\mathcal{KGC}$ are identical in both identity strings.

Definition 21 (Secure Double-Blinded IBE)

An IBE scheme Π *is secure if and only if: (1) the underlying* Π *is a secure IBE scheme, (2)* $\mathsf{DoubleBlindExtract}$ *is leak-free and selective-failure double-blind and (3) the resulting scheme is ciphertext aware.*

The private key resulting from the DoubleBlindExtract protocol has the same form as the private key generated by the Extract algorithm in the original scheme. As such, the Encrypt and Decrypt algorithms remain correct. Security arguments pertaining to leak-freeness and selective failure for double-blindness follow from those outlined previously for partial-blindness and thus are not provided. For leak-freeness, an additional restriction on the identities $id_0 = id_0|\overline{id_0}|\overline{\overline{id_0}}$ $id_1 = id_1|\overline{id_1}|\overline{\overline{id_1}}$ such that $id_0 = id_1$ and $\overline{\overline{id_0}} = \overline{\overline{id_1}}$ is required.

Lemma 6 *The scheme* $\Pi =$ Setup, DoubleBlindExtract, Encrypt, Decrypt *is ciphertext aware.*

Proof. Consider the two games outlined in Figure 3.10, the Real game and the Ideal game that the adversary $\mathcal{A}$ interacts with, and receives output x_{Real} , x_{Ideal}. The intuition for the ciphertext awareness of the scheme can be described as follows. The underlying IBE scheme Π is IND-CPA secure. Given two messages, their resulting ciphertexts are indistinguishable.

To prove ciphertext awareness, begin with the contradiction that there exists a polynomial distinguisher D which can computationally distinguish the output from game Ideal and game Real. This is a contradiction as constructing such a D that can distinguish pairs $\{(m_b, ct_b), (m_{1-b}, ct_{1-b})\}$ with non-negligible probability is not possible as it would mean that D can be used to construct an IND-CPA adversary against the scheme. Thus, distinguishing the distributions of x_{Real}, x_{Ideal} with a non-neglible advantage is not possible.

□

3.7 Transformation for Anonymous Partially-Blind and Double-Blind Identity-Based Encryption

Above, we present the first blind extract protocol for an anonymous IBE scheme (Section 3.4). Anonymity is a desirable feature for IBE schemes, as it prevents a ciphertext from being associated with the identity used to encrypt it. We adapt the extract algorithm of the anonymous IBE scheme for use with our partially and fully blind extraction protocols. The output of the extract protocol of the anonymous scheme is required to be of the form

$$sk_{id} = \Big(h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4}, h^{-\alpha t_2} H_2(id)^{-\tilde{r}_1 t_2}, h^{-\alpha t_1} H_2(id)^{-\tilde{r}_1 t_1}, H_2(id)^{-\tilde{r}_2 t_4}, H_2(id)^{-\tilde{r}_2 t_3}\Big).$$

Using step 1 of the PartialBlindExtract and DoubleBlindExtract protocols outlined above, the $\mathcal{KGC}$ obtains the hash of the identity. We modify these protocols using the parameters of the

anonymous IBE scheme, taking $v = (v_1, \ldots, v_n)$ as the identity, to show they are suitable for use with the scheme to achieve partially-blind and double-blind anonymous IBE schemes.

In our extract protocols, $\mathcal{U}$ sends the tuple $(X, \overrightarrow{v}, h^\beta, h_0^\beta)$ required to compute this hash. We modify this tuple to be suitable for use with the anon-IBE scheme. $\mathcal{U}$ chooses random values $u_0, u_1, u_2 \in \mathbb{Z}_q$ and $\beta, r_1', r_2' \in \mathbb{Z}_q^*$ and constructs the following, where $\overrightarrow{v}$ is determined by the use of either PartialBlindExtract or DoubleBlindExtract:

$$X = h_0^\beta \prod_{i=1,\gamma=1}^{n} h_i^{\beta v_i}$$

$$\overrightarrow{v} = \begin{cases} (u_i^\beta, v_i) & \text{if } i \in \{0, m\} \wedge \gamma_i = 0 \\ \perp & \text{if } i \in \{0, m\} \wedge \gamma_i = 1 \\ u_i^\beta & \text{if } i > m \end{cases}$$

$$h_0^\beta; h^{r_1'}; h^{r_2'}; h^{u_0}; h^{u_1}; h^{u_2}; h^{-\beta/r_1'}$$

The tuple $\mathcal{U}$ sends to $\mathcal{KGC}$ is $(X, \overrightarrow{v}, h_0^\beta; h^{r_1'}; h^{r_2'}; h^{u_0}; h^{u_1}; h^{u_2}; h^{-\beta/r_1'})$. The $\mathcal{KGC}$ chooses random values $\hat{r_1}, \hat{r_2} \in \mathbb{Z}_q^*$ and sends $h^{\hat{r}_1}, h^{\hat{r}_2}$ to $\mathcal{U}$. Implicitly, $\tilde{r}_1 = r_1'\hat{r}_1$, $\tilde{r}_2 = r_2'\hat{r}_2$. The $\mathcal{KGC}$ constructs the key as $(d_v' = d_0', d_1', d_2', d_3', d_4')$ where $ID' = h_0^\beta \prod_{i=1}^n h_i^{\beta v_i}$.

$$\begin{aligned} d_0' &= (h^{r_1'})^{\hat{r}_1 t_1 t_2} (h^{r_2'})^{\hat{r}_2 t_3 t_4} (h^{u_0}) \\ &= h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4 + u_0} = h^{x_0} \\ d_1' &= (h^{-\beta/r_1'})^{\alpha t_2} (h^{u_1}) (ID')^{\hat{r}_1 t_2} \\ &= h^{-\beta/r_1' \alpha t_2 + u_1} ID'^{-\hat{r}_1 t_2} = h^{x_1} ID'^{-\hat{r}_1 t_2} \\ d_2' &= (h^{-\beta/r_1'})^{\alpha t_1} (h^{u_2}) (ID')^{\hat{r}_1 t_1} \\ &= h^{-\beta/r_1' \alpha t_1 + u_2} ID'^{\hat{r}_1 t_1} = h^{x_2} ID'^{-\hat{r_1} t_1} \\ d_3' &= ID'^{-\hat{r}_2 t_4} \\ d_4' &= ID'^{-\hat{r}_2 t_3} \end{aligned}$$

The key d'_v is sent to $\mathcal{U}$, who unblinds it to retrieve her secret key,

$$\begin{aligned} sk_{id} &= (d_0, d_1, d_2, d_3, d_4) \\ &= (d'_0 h^{-u_0}, (d'_1 h^{-u_1})^{r'_1/\beta}, (d'_2 h^{-u_2})^{r'_1/\beta}, d'^{r'_2}_3, d'^{r'_2/\beta}_4) \\ &= \Big(h^{\tilde{r}_1 t_1 t_2 + \tilde{r}_2 t_3 t_4}, h^{-\alpha t_2} H_2(id)^{-\tilde{r}_1 t_2}, h^{-\alpha t_1} H_2(id)^{-\tilde{r}_1 t_1}, H_2(id)^{-\tilde{r}_2 t_4}, H_2(id)^{-\tilde{r}_2 t_3}\Big). \end{aligned}$$

The resulting key is in the correct form, which $\mathcal{U}$ can check by testing

$$e(g_1, g_2) \cdot e(d'_2, g^y u' \prod_{i=1}^{n} u_i^{v_i}) = e(g, d'_1).$$

3.8 Conclusion

In this chapter, we have motivated the amalgamation of the blinding property of digital signatures and identity-based encryption. We have presented constructions for new blind identity-based encryption schemes. We first extended the scope of blind identity-based encryption schemes to incorporate the property of anonymity by constructing a suitable underlying anonymous IBE scheme and providing the BlindExtract protocol in work with Camenisch *et al.* [CKRS09]. Security definitions and arguments were provided.

We then presented novel extensions to the area of identity-based encryption: partially-blind and double-blind key extraction protocols. These constructions are the result of work with Gray [SG09]. We presented constructions of each, together with the requisite security definitions and proofs. We showed these protocols to be applicable to anonymous identity-based encryption by providing a transformation for the novel blinding protocols of PartialBlindExtract and DoubleBlindExtract to achieve this desirable property.

Chapter 4

Constructions using Blind Identity-Based Encryption

4.1 Introduction

In Chapter 3, we introduced the concept of blind identity-based encryption, along with constructions and extensions in the form of partially-blind and double-blind identity-based encryption. In this chapter, applications of all forms of blind identity-based encryption are presented, illustrating how they can be used as a primitive in cryptographic protocols.

The first application of blind identity-based encryption presented is *simulatable oblivious transfer* by Green and Hohenberger [GH07]. The use of blind IBE provides advantages in that the complexity problem required is less restrictive than that of equivalent schemes, while maintaining the efficiency and properties of recent adaptive oblivious transfer schemes.

The second application of blind identity-based encryption is *public key encryption with oblivious keyword search*, which is the result of work with Camenisch *et al.* [CKRS09]. We begin by motivating the construction of a public-key encryption scheme with oblivious keyword search. We identify the requirements on a blind identity-based encryption scheme necessary to realise such a construction. Our contribution is to construct such a scheme.

The third application is anonymous key issuing, which is the result of work with Gray [SG09]. Anonymous key issuing has the objective of preventing a $\mathcal{KGC}$ from learning anything about the identity used to request a private key. We present a framework using blind identity-based encryption to do so. We show that this framework is also suitable for use with partially-blind and

double-blind identity-based encryption.

Finally, we present a *unique receipt issuing scheme*, with an application scenario. We use the double-blind IBE scheme as a primitive in an online lottery protocol. This lottery generates a unique type of ticket, with input from both the ticket seller and the ticket buyer. It provides a challenge-reponse ticket validation protocol, which ensures only a valid ticket can be used to claim a prize.

4.2 Simulatable Oblivious Transfer using Blind Identity-Based Encryption

The first construction of blind IBE, outlined in Section 3.3, was motivated by the challenge of constructing a *simulatable oblivious transfer* with standard complexity assumptions. We begin by explaining what is required of such a scheme.

4.2.1 Oblivious Transfer

Oblivious Transfer (OT) is a generalisation of the secret sharing protocol introduced by Rabin [Rab81]. Rabin describes oblivious transfer as the transferral of information where the sender does not know if the recipient actually received the information. OT is a cryptographic primitive that allows a receiver to choose one message from a set of messages sent by a sender without revealing to the sender which message was chosen by the receiver. Additionally, the sender is guaranteed that the receiver does not learn anything about the rest of the messages in the set. In this manner, it protects the privacy of the receiver by not revealing the message chosen, and the privacy of the sender, by not revealing any of the other data sent [NP99, NP01]. It is a method for private information retrieval (PIR). PIR allows a user to retrieve an item without revealing which item she is retrieving

Rabin's exchange of secrets by oblivious transfer. In Rabin's protocol [Rab81], Alice and Bob wish to exchange secret values, S_A, S_B respectively. The values of S_A, S_B could be sensitive data such as passwords. The challenge is to exchange the data without using a trusted third party or a secure simultaneous exchange mechanism. In order to prevent either party cheating by sending invalid messages, each party generates a commitment to the validity of their S_i. This gives Alice recourse to prove that Bob has cheated should he send a value $S_{B'} \neq S_B$, i.e., in the case that he

sends an invalid password.

This does not address the exchange of secrets, however, as Alice can still send an invalid password to Bob, while receiving a valid one. While she may be forced to send the valid one at a later stage, she still has the advantage of accessing Bob's message first. To address this problem, Rabin constructed a protocol such that given the fact that Bob has learnt S_A, Alice learns S_B. The exchange of secrets protocol is based on the hardness of the factoring problem. It involves the first use of oblivious transfers. It is limited in that it only works for honest parties.

Forms of oblivious transfer. In a 1-out-of-2 oblivious transfer scheme, denoted as OT_1^2, the receiver chooses one message from the two constructed by the sender. It follows that a 1-out-of-N oblivious transfer is represented as OT_1^N, where the receiver chooses one message from the N constructed by the sender, and a k-out-of-N is represented as OT_k^N, where the receiver chooses k messages from the N constructed by the sender. Finally, adaptive k-out-of-N oblivious transfer is represented as $\mathsf{OT}_{k\times 1}^N$, where the receiver chooses k messages from the N constructed by the sender.

OT_k^N involves a Commit phase and a Transfer phase. In the Commit phase, the sender commits to N messages, and sends the commitments to the receiver. The Transfer phase is interactive between the sender and receiver. It allows the receiver to obtain k messages of her choice by using the commitments. In non-adaptive OT_k^N, at the beginning of the Transfer phase the receiver states the messages she wishes to obtain. In adaptive $\mathsf{OT}_{k\times 1}^N$ the receiver may choose messages in the transfer subphase $i \in \{1, \ldots, k\}$ after $i-1$ subphases, where her choice may depend on the messages obtained previously.

Applications of oblivious transfer. Oblivious transfer can be used to construct oblivious circuit evaluation, priced oblivious transfer and interactive zero knowledge proof systems [AIR01, Kil88]. In priced oblivious transfer, for example, a buyer may purchase goods without revealing what she is buying, or even when she buys it, on the condition her pre-payment (credit) balance is sufficient to cover the cost of the item she wishes to purchase. OT can be used as a building block for other protocols.

Definition 22 (k-out-of-N Oblivious Transfer ($\mathsf{OT}_{k\times 1}^N, \mathsf{OT}_k^N$) [CNS07]) *We generalise an OT scheme as a tuple of algorithms ($\mathsf{S_I}$, $\mathsf{R_I}$, $\mathsf{S_T}$, $\mathsf{R_T}$). These algorithms are used in matched pairs. During the initialisation phase the sender runs $\mathsf{S_I}(m_1, \ldots, m_N)$ to obtain state value S_0, and the*

receiver runs $\mathsf{R}_\mathsf{I}()$ *to obtain state value* R_0*. The sender and receiver execute* S_T, R_T k *times as described below.*

Non-adaptive OT *In the non-adaptive* OT_k^N *case the parties execute the protocol as above; however, for round* $i < k$ *the algorithm* $\mathsf{R}_\mathsf{T}(R_{i-1}, \sigma_i)$ *does not output a message. At the end of the* k^{th} *transfer* $\mathsf{R}_\mathsf{T}(R_{k-1}, \sigma_k)$ *outputs the messages* $(m'_{\sigma_1}, \ldots, m'_{\sigma_k})$ *where for* $j = 1, \ldots, N$ *each* m'_{σ_j} *is* m_{σ_j} *or* $\perp$*. (Note that in a non-adaptive scheme, the initialisation and* k *transfers do not necessarily require a corresponding number of communication rounds.)*

Adaptive OT *In the adaptive* $\mathsf{OT}_{k\times 1}^N$ *case, for* $1 \leq u \leq k$*, the* i^{th} *transfer proceeds as follows: the sender runs* $\mathsf{S}_\mathsf{T}(S_{i-1})$ *to obtain state value* S_i*, and the receiver runs* $\mathsf{R}_\mathsf{T}(R_{i-1}, \sigma_i)$ *where* $1 \leq \sigma_i \leq N$ *is the index of the message to be received. This produces state information* R_i *and the message* m_{σ_i} *or* $\perp$ *indicating failure.*

This definition requires for correctness that at the end of an honest, successful execution of the protocol, the receiver should obtain M_{σ_i}.

4.2.2 Security of Oblivious Transfer Protocols

The notion of security in OT has evolved from an honest-but-curious-model, to a half-simulation model, and most recently to a full-simulation model.

Intuitively, the *honest-but-curious* model has all parties behaving honestly while running the protocol but examining the transcript of the protocol after it has run. This security model guarantees that nothing further can be learnt from examining the transcript than is already known from running the protocol.

The *half-simulatation* model [NP05] views the security of senders and receivers separately. The security requirement of the receiver is *indistinguishability*. It implies that the sender should be unable to distinguish from its views of the protocol the iteration in which the receiver retrieved M_σ from the iteration in which M'_σ was retrieved. The security of the sender is based on a comparison with the *ideal-world model*. An ideal-world counterpart for every real-world malicious receiver is constructed such that an adversary in the real world gains no more information than in an ideal world implemented by a trusted third party. The half-simulation model is vulnerable to selective-failure attacks. A sender may cause a transfer to fail by sending invalid messages during the

initialisation phase. In such an attack, the sender learns nothing but the receiver cannot complain about the message received without loss of privacy.

This flaw motivates the *full-simulation* model [CNS07], in which both sender and receiver security follow the ideal-world / real-world model. In the real-world, both parties are active in the protocol. In the ideal-world, the sender's role is implemented by a trusted third party. This model further requires that the combined outputs of the sender and receiver are indistinguishable.

Security for the simultable, adaptive $\mathsf{OT}^N_{k\times 1}$ scheme can be defined as follows:

Definition 23 (Security for $\mathsf{OT}^N_{k\times 1}$ [CNS07].) *The security of $\mathsf{OT}^N_{k\times 1}$ is described by the following real world/ideal world game:*

Real experiment. *The experiment is for arbitrary sender and receiver algorithms $\hat{\mathsf{S}}$ and $\hat{\mathsf{R}}$. The experiment $\mathbf{Real}_{\hat{\mathsf{S}},\hat{\mathsf{R}}}(N, k, m_1, \ldots, m_N, \Sigma)$ proceeds as follows. $\hat{\mathsf{S}}$ is given messages $(m_1, \ldots, m_N)$ as input and interacts with $\hat{\mathsf{R}}(\Sigma)$, where Σ is an adaptive selection algorithm that, on input messages $(m_{\sigma_1}, \ldots, m_{\sigma_N})$, outputs the index σ_i of the next message to be queried. In their first run, $\hat{\mathsf{S}}$ and $\hat{\mathsf{R}}$ produce initial states S_0 and R_0 respectively. Next, the sender and receiver engage in k interactions. In the i^{th} interaction for $1 \leq i \leq k$ the sender and receiver interact by running $S_i \leftarrow \hat{\mathsf{S}}(S_{i-1})$ and $(R_i, m_i^*) \leftarrow \hat{\mathsf{R}}(R_{i-1})$, and update their states to S_i and R_i respectively. Note that m_i^* may be different from m_{σ_i} when either participant cheats. At the end of the k^{th} interaction, sender and receiver output strings S_k and R_k respectively. The output of the $\mathbf{Real}_{\hat{\mathsf{S}},\hat{\mathsf{R}}}$ experiment is the tuple (S_k, R_k).*

For an $\mathsf{OT}^N_{k\times 1}$ scheme ($\mathsf{S_I}$, $\mathsf{S_T}$, $\mathsf{R_I}$, $\mathsf{R_T}$), define the honest sender Sn algorithm as the one which runs $\mathsf{S_I}(m_1, \ldots, m_N)$ in the initialisation phase, runs $\mathsf{S_T}$ in all following interactions and always outputs $S_k = \epsilon$ as its final output. Define the honest receiver Rn as the algorithm which runs $\mathsf{R_I}$ in the initialisation phase, runs $\mathsf{R_T}(R_{i-1}, \sigma_i)$, where in the i^{th} interaction, Σ is used to generate the index σ_i, and returns the list of received messages $R_k = (m_{\sigma_1}, \ldots, m_{\sigma_N})$ as its final output.

Ideal experiment. *In the experiment $\mathbf{Ideal}_{\hat{\mathsf{S}}',\hat{\mathsf{R}}'}(N, k, m_1, \ldots, m_N, \Sigma)$ the (possibly cheating) sender algorithm $\hat{\mathsf{S}}'(m_1, \ldots, m_N)$ generates messages $m_1^*, \ldots, m_N^*$ and hands these to the trusted party T. In each of the k transfer phases, T receives a bit b_i from the sender $\hat{\mathsf{S}}'$ and an index σ_i^* from the (possibly cheating) receiver $\hat{\mathsf{R}}'(\Sigma)$. If $b_i = 1$ and $\sigma_* \in \{1, \ldots, N\}$ then T hands $m^*_{\sigma_i^*}$ to the receiver; otherwise, it hands $\perp$ to the receiver. At the end of the*

k^{th} transfer, $\hat{\mathsf{S}}'$ and $\hat{\mathsf{R}}'$ output a string S_k and R_k; the output of the experiment is the pair (S_k, R_k).

Note that the sender's bit b models its ability to make the current transfer fail. However, the sender's decision to do so is independent of the index σ_i that is being queried by the receiver. This captures the strongest notion of coherence and excludes schemes that allow a sender to cause selective failure. As above, the ideal sender $\mathsf{Sn}'(m_1, \ldots, m_N)$ is defined as sending messages $m_1, \ldots, m_N$ to the trusted party in the initialisation phase, sending $b_i = 1$ in all transfer phases, and using $S_k = \epsilon$ as its final output. Define the honest ideal receiver Rn' as the algorithm which generates its selection indices σ_i through Σ and submits these to the trusted party. Its final output consists of all the messages it received $R_k = (m_{\sigma_i}, \ldots, m_{\sigma_N})$.

Sender Security. *$\mathsf{OT}^N_{k\times 1}$ is said to be $(t, t', t_{\mathsf{D}}, \epsilon)$-sender-secure if for any real-world cheating receiver $\hat{\mathsf{R}}$ running in time t, there exists an ideal-world receiver $\hat{\mathsf{R}}'$ running in time t' such that for any $N \in [1, t]$, any messages $m_1, \ldots, m_N$, and any selection algorithm Σ, a distinguisher D running in time t_{D} does not have probability of success greater than ϵ in distinguishing the distributions*

$$\mathbf{Real}_{\mathsf{Sn},\hat{\mathsf{R}}}(N, k, m_1, \ldots, m_N, \Sigma) \text{ and } \mathbf{Ideal}_{\mathsf{Sn}',\hat{\mathsf{R}}'}(N, k, m_1, \ldots, m_N, \Sigma).$$

Receiver Security. *$\mathsf{OT}^N_{k\times 1}$ is said to be $(t, t', t_{\mathsf{D}}, \epsilon)$-receiver-secure if for any real-world cheating sender $\hat{\mathsf{S}}$ running in time t, there exists an ideal-world sender $\hat{\mathsf{S}}'$ running in time t' such that for any $N \in \mathbb{N}$, any $k \in [0, N]$, any messages $m_1, \ldots, m_N$, and any selection strategy Σ, no distinguisher D running in time t_{D} has success probability greater than ϵ in distinguishing the distributions*

$$\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{Rn}}(N, k, m_1, \ldots, m_N, \Sigma) \text{ and } \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{Rn}'}(N, k, m_1, \ldots, m_N, \Sigma).$$

4.2.3 Simulatable Oblivious Transfer

Green and Hohenberger [GH07] build on the simulatable oblivious transfer presented by Camenish *et al.* [CNS07], focusing on adaptive and non-adaptive OT protocols. Both the adaptive and non adaptive schemes presented in [GH07] are given formal definitions, which are consistent with Definition 23. The schemes are realised under the IBE-to-OT transformation Green and Hohenberger provide, which we outline in Figure 4.1. We present their work here as it represents the the first application of blind IBE.

The non-adaptive construction. A non-adaptive OT_k^N construction without random oracles can be instantiated using the blind IBE schemes presented in Section 3.3 and the transform in Figure 4.1. It relies on the existence of a blind IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{BlindExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ where Setup is a system parameter generating algorithm, $\mathsf{BlindExtract}$ is an interactive blind key extraction protocol, $\mathsf{Encrypt}$ is an encryption algorithm and $\mathsf{Decrypt}$ is a decryption algorithm.

The sender executes Setup, and sends $params$ to the receiver. The sender constructs encryptions of N messages M_i under identity id_i and sends the resulting ciphertexts to the receiver. In order to retrieve k messages, the receiver runs the $\mathsf{BlindExtract}$ protocol for k identities of his choice. The receiver uses the resulting k decryption keys to decrypt and recover messages of his choosing. The blinding property ensures that a cheating receiver gains no information about the messages corresponding to secret keys he did not extract, while ensuring that a cheating sender does not learn the identities extracted.

It is fully-simulatable under the following modifications. The sender must prove, using zero knowledge, knowledge of the value msk. Instead of transmitting the ciphertext, the sender transmits only a commitment to a collision-resistant hash of the ciphertext vector. The actual ciphertexts are sent at the end of the k^{th} round, along with a proof that the commitment to the hash of the ciphertexts can be opened.

The adaptive construction. An adaptive $\mathsf{OT}_{k\times 1}^N$ protocol can be instantiated using the blind IBE schemes presented in Section 3.3 and the transform in Figure 4.2. It is fully-simulatable and is efficient in terms of communication cost and round-efficency. However, it is only secure in the random oracle model. The main advantage of this scheme is that the use of the blind IBE scheme means the complexity assumption on which the protocols are based is the DBDH assumption, whereas the scheme of Camenisch *et al.* [CNS07] requires interactive complexity assumptions. This is due to the requirement for unique blind signatures in their construction, of which there are currently only two constructions [Cha82, Bol03].The scheme presented in Figure 4.2 can be combined with the scheme of Camenisch *et al.* to achieve a standard model variant. However, this protocol relies on strong assumptions, which require larger than normal security parameters.

The constructions presented in Figure 4.1 and Figure 4.2 require the $\mathsf{IsValid}(params, id, ct)$ ciphertext correctness test. Firstly, this check verifies the group parameters are valid and for $ct = (X, Y, Z)$, all the values are in the correct groups and that the following relation holds:

$$e(Y, F(id)) = e(Z, g).$$

Initialisation Phase The sender and receiver begin by agreeing on parameters for a commitment scheme, such as Pedersen's commitment scheme, and a collision resistant hash function H. During the initialisation phase, the sender runs $\mathsf{S_I}(M_1, \ldots, M_N)$ and the receiver runs $\mathsf{R_I}()$.

1. The sender runs $\mathsf{Setup}(1^\kappa)$, which outputs systems parameters $params$ and the master secret msk.
2. For $j = 1, \ldots, n$, the sender computes $C_j \leftarrow \mathsf{Encrypt}(params, j, M_j)$, and $(\mathcal{C}, \mathcal{D}) \leftarrow \mathsf{Commit}(H(C_1, \ldots, C_N))$.
3. The sender sends $(params, \mathcal{C})$ to receiver.
4. The sender executes the proof of knowledge $PoK\{(msk) : (params, msk) \in \mathsf{Setup}(1^\kappa)\}$. If the proof does not verify, the receiver aborts.

Transfer phase During the transfer phase the sender runs $\mathsf{S_T}()$ and the receiver runs $\mathsf{R_T}(\sigma_i)$.

1. Sender sends $(C_1, \ldots, C_N)$ to the receiver.
2. The sender executes the proof of knowledge $PoK\{(\mathcal{D}) : \mathsf{Decommit}(H(C_1, \ldots, C_N), \mathcal{C}, \mathcal{D}) = 1\}$
3. If the proof does not verify, or if for any i, $\mathsf{IsValid}(params, i, C_i) \neq 1$, the receiver aborts.
4. During the i^{th} transfer, the sender and receiver execute $\mathsf{BlindExtract}$ for identity σ_i. Set $M'_{\sigma_i} \leftarrow \perp$ if $\mathsf{BlindExtract}$ fails, else set $M'_{\sigma_i} \leftarrow \mathsf{Decrypt}(params, \sigma_i, sk_{\sigma_i}, C_{\sigma_i})$.
5. The sender output is $(msk, \mathcal{D})$ and the receiver output is $(params, \mathcal{C}, M'_{\sigma_1}, \ldots, M'_{\sigma_N})$.

Figure 4.1: Transformation IBE-to-OT_k^N

Initialisation Phase During the initialisation phase, the sender runs $\mathsf{S_I}(M_1, \ldots, M_N)$ and the receiver runs $\mathsf{R_I}()$.

1. The sender runs $\mathsf{Setup}(1^\kappa)$, which outputs systems parameters $params$, msk, and chooses a collision-resistant hash function $H : \mathcal{M} \rightarrow \{0,1\}^n$.
2. The sender selects random $W_1, \ldots, W_N \in \mathcal{M}$ and for $j = 1, \ldots, n$, computes $A_j \rightarrow \mathsf{Encrypt}(params, j, W_j)$, $B_j \rightarrow H(W_j) \oplus M_j$ and $C_j = (A_j, B_j)$.
3. The sender executes the proof of knowledge $PoK\{msk : (params, msk) \in \mathsf{Setup}(1^\kappa)\}$.
4. Sender sends $(params, C_1, \ldots, C_N)$ to the receiver.
5. If the proof of knowledge fails, or if $\mathsf{IsValid}(params, i, C_i) \neq 1$, the receiver aborts the transfer. Otherwise the sender outputs $S_0 = (params, msk)$ and the receiver outputs $R_0 = (params, C_1, \ldots, C_N)$.

Transfer phase During the transfer phase the sender runs $\mathsf{S_T}(S_{i-1})$ and the receiver runs $\mathsf{R_T}(R_{i-1}, \sigma_i)$.

1. During the i^{th} transfer, the sender and receiver execute $\mathsf{BlindExtract}$ for identity σ_i.
2. The receiver computes $M'_{\sigma_i} \rightarrow B_{\sigma_i} \oplus H(\mathsf{Decrypt}(params, \sigma_i, sk_{\sigma_i}, A_{\sigma_i}))$ or $\perp$ if $\mathsf{BlindExtract}$ fails.
3. The sender output is $S_i = S_{i-1}$ and the receiver output is $R_i = (R_{i-1}, M'_{\sigma_i})$.

Figure 4.2: Transformation IBE-to-$\mathsf{OT}^N_{k\times 1}$

4.3 Public-Key Encryption with Oblivious Keyword Search using Blind Identity-Based Encryption

Searchable encryption schemes provide an important mechanism to protect data while keeping it available to be searched and accessed. In a common approach for their construction, the encrypting entity chooses one or several keywords that describe the content of each encrypted record of data. To perform a search, a user obtains a trapdoor for a keyword of her choosing and uses this trapdoor to find all the data described by this keyword.

In joint work with Camenisch *et al.* [CKRS09], we present a searchable encryption scheme that allows users to perform adaptive, oblivious searches by keywords on encrypted data in a public key setting and decrypt the search results. The novel contribution of our scheme is that it does not require a user to reveal their search term in order to obtain the corresponding trapdoor. The resulting scheme is called public key encryption with oblivious keyword search (PEOKS). PEOKS is an extension of public key encryption with keyword search (PEKS) in which users obtain trapdoors from the secret key holder without revealing the keywords. Our PEOKS scheme is constructed by using the *committed blind anonymous IBE* we constructed, as presented in Section 3.4.1.

4.3.1 Oblivious Keyword Search

Oblivious keyword search (OKS) [OK04] generalises oblivious transfer by associating a keyword with messages, rather than a σ index as used in oblivious transfer. As such, oblivious transfer can be seen as a particular case of oblivious keyword search, where the unique keyword associated with a record is the index value σ_i.

Oblivious keyword search involves a sender and a receiver. The sender generates a set of message-keyword pairs, $M = \{(M_{W_1}, W_1), \ldots, (M_{W_N}, W_N)\}$, where the keywords belong to a keyword space W_{sp}. The receiver chooses a keyword $\hat{W}_i$ to obtain the message $M_{\hat{W}_i}$. Assume, without loss of generality, that each keyword is linked to at most one message. To retrieve all data associated with keyword $\hat{W}_i$, it is possible to construct a message $M_{\hat{W}_i}$ to include all the relevant information . Using oblivious keyword search, the privacy properties of oblivious transfer are maintained. The receiver obtains a message $M_{\hat{W}_i}$ in such a way that the sender learns nothing about $\hat{W}_i$, and the receiver does not learn anything about the other messages. Additionally, oblivious keyword search may be adaptive.

Definition 24 ($\mathsf{OKS}^N_{k\times 1}$ [OK04].) *An $\mathsf{OKS}^N_{k\times 1}$ scheme is a tuple of algorithms (S_I, R_I, S_T, R_T) which are run in matched pairs. In the initialization phase, the sender runs $S_I((M_{W_1}, W_1), \ldots, (M_{W_N}, W_N))$, where each $W_j \in W_{sp}$, $1 \leq j \leq N$, is a keyword, and obtains state value S_0. The receiver runs $R_I()$ and obtains state value R_0. During the transfer phase sender and receiver execute (S_T, R_T) k times. During the i^{th} transfer, $1 \leq i \leq k$, the sender runs $S_T(S_{i-1})$ to obtain S_i, and the receiver runs $R_T(R_{i-1}, \hat{W}_i)$, for $\hat{W}_i \in W_{sp}$, to obtain state value R_i and the message $M'_{\hat{W}_i}$ (or $\perp$ indicating failure), where $M'_{\hat{W}_i}$ is used to indicate message $M_{\hat{W}_i}$ has been received.*

4.3.2 Public-key Encryption with Keyword Search

Public key encryption with keyword search (PEKS) addresses the problem of searching data that has been encrypted using a public key. The capacity to search encrypted text for particular keywords is delegated by a central authority. This allows a third party, an entity such as an email gateway, to test if the encrypted text contains a keyword. The email recipient, Alice, does not want to allow the gateway decrypt all of her messages, merely to determine if the keyword is present in the encrypted text. Using PEKS, as described by Boneh *et al.* [BDCOP04], Alice can generate a key which will allow the third party to identify the relevant encrypted messages containing a given keyword. PEKS is also possible using IBE systems, known as identity-based PEKS.

Definition 25 (Non-interactive PEKS.) *A non-interactive PEKS scheme consists of the following polynomial time randomised algorithms:*

$\mathsf{KeyGen}(k)$**:** *given as input a security parameter k, generates a public/private key pair A_{pub}, A_{priv}.*

$\mathsf{PEKS}(A_{pub}, W)$**:** *given as input a public key A_{pub}, and a keyword W, produces a searchable encryption of W.*

$\mathsf{Trapdoor}(A_{priv}, W)$**:** *given as input Alice's private key and a keyword W, produces a trapdoor T_W.*

$\mathsf{Test}(A_{pub}, S, T_W)$**:** *given as input Alice's public key, a searchable encryption $S = \mathsf{PEKS}(A_{pub}, W)$, outputs yes if $W = W'$ and no otherwise.*

PEKS is reliant on both the sender, Bob, and the recipient, Alice, adhering to the scheme. Bob, when sending a message, generates the ciphertext using a standard public key system. When the

ciphertext is generated, he appends the PEKS of each keyword to the message. That is, to send message m with keywords $W_1, \ldots, W_m$, Bob constructs

$$E_{A_{pub}}(m)|\mathsf{PEKS}(A_{pub}, W_1)|\cdots|\mathsf{PEKS}(A_{pub}, W_m)$$

where A_{pub} is Alice's public key, $E_{A_{pub}}(m)$ is the encryption of the message m under the public key and PEKS is an algorithm. Alice generates a trapdoor T_{W_i} corresponding to each W_i and gives the keywords W_i to the third party. These trapdoors allow the holder to test if $W = W'$, given access to the protocol $\mathsf{Test}(A_{pub}, S, T_{W'})$ and T_W. If $W \neq W'$, the third party learns nothing about W'.

A PEKS scheme implies IBE [BDCOP04]. To capture the security of PEKS schemes, indistinguishability under chosen-plaintext attack (IND-CPA) is usually used. Under IND-CPA for PEKS, an adversary cannot distinguish between two searchable encryptions for keywords of his choice, even with access to an oracle providing trapdoors for any non-challenge keywords. Additionally, we require that the searchable encryptions do not leak any information about the information contained within.

Definition 26 (PEKS-IND-CPA) *A PEKS scheme is said to be PEKS-IND-CPA secure if there exists a negligible function $\nu(k)$ such that:*

$$\begin{aligned}
&\Pr[WSet \leftarrow \emptyset; (s_k, params) \leftarrow SetupPEKS(1^k);\\
&\quad (W_0, W_1, M_0, M_1, state) \leftarrow A^{\leftrightarrow Oracle_{Trapdoor}(\cdot)}(params)\\
&\quad \wedge W_0, W_1 \notin WSet; b \leftarrow \{0,1\}; c \leftarrow \{0,1\};\\
&\quad S_{W_b, M_c} \leftarrow PEKS(params, W_b, M_c);\\
&\quad (b', c') \leftarrow A^{\leftrightarrow Oracle_{Trapdoor}(\cdot)}(params, S_{W_b, M_c}, state) :\\
&\quad b = b' \wedge c = c'] < 1/4 + \nu(k).
\end{aligned}$$

Using $Oracle_{Trapdoor}(W)$, if $W \in WSet$ then it returns $\perp$; otherwise it adds W to the set $WSet \leftarrow WSet \cup \{W\}$ and it returns $T_W \leftarrow Trapdoor(params, s_k, W)$.

Consistency in PEKS can be divided into two parts [ABC+08]. The first requires that, given a valid searchable encryption, trapdoor pair computed using the same keyword, $Test$ never outputs

$\perp$. More formally, given $W \in \{0,1\}^*$ and $m \in \{0,1\}^*$:

$$\Pr[(s_k, params) \leftarrow \mathsf{Setup}(1^k); S_{W,M} \leftarrow \mathsf{PEKS}(params, W, M);$$
$$T_W \leftarrow \mathsf{Trapdoor}(params, msk, W);$$
$$M' \leftarrow Test(params, S_{W,M}, T_W) : M' = M] = 1$$

The second condition focuses on consistency and states that when a searchable encryption / trapdoor pair were computed using different keywords, then algorithm $Test$ should output $\perp$. No known PEKS scheme fulfills this *perfect consistency* property. Abdalla *et al.* [ABC$^+$08] considered two relaxations for consistency, statistical and computational:

Definition 27 (Consistency for PEKS) *Consider a PEKS scheme $(SetupPEKS, PEKS, Trapdoor, Test)$ and the following probability Pr:*

$$\Pr[(s_k, params) \leftarrow \mathsf{Setup}(1^k); (W, W', M) \leftarrow A(params);$$
$$S_{W,M} \leftarrow \mathsf{PEKS}(params, W, M); T_{W'} \leftarrow \mathsf{Trapdoor}(params, msk, W');$$
$$M' \leftarrow \mathsf{Test}(params, S_{W,M}, T_{W'}) : M' \neq \perp]$$

The PEKS scheme is said to be:

perfectly consistent *if, for all computationally unbounded adversaries, $P = 0$.*

statistically consistent *if, for all computationally unbounded adversaries, there exists a negligible function $\nu(k)$ such that $P \leq \nu(k)$.*

computationally consistent *if, for all probabilistic polynomial time adversaries, there exists a negligible function $\nu(k)$ such that $P \leq \nu(k)$.*

4.3.3 Construction of the PEOKS Scheme

In joint work with Camenisch *et al.* [CKRS09], we construct a PEOKS scheme, using the suitable anonymous IBE scheme presented in Section 3.4.1 and the generic transformation by Abdalla *et al.* [ABC$^+$08] from IBE to PEKS. The transformation is presented in Figure 4.3.

This generic transformation takes as input the algorithms Π of a secure IBE scheme and returns a PEKS scheme $\Upsilon = (\mathsf{KeyGen}, \mathsf{PEKS}, \mathsf{Trapdoor}, \mathsf{Test})$.

Given an IBE scheme with algorithms $(\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ the PEKS scheme is as follows:

$\mathsf{Setup}(1^k)$**:** On input a security parameter k, run $\mathsf{Setup}(1^k)$ to obtain the secret key msk and $params$, the parameters of the scheme.

$\mathsf{PEKS}(params, W, M)$**:** On input a keyword W and a message M, it picks a random value $C_2 \in \{0,1\}^k$ and computes $C_1 = \mathsf{Encrypt}(params, W, C_2||M)$. It outputs the tuple $S_{W,M} = (C_1, C_2)$.

$\mathsf{Trapdoor}(params, msk, W)$**:** The trapdoor T_W associated with the keyword W is the secret key sk_W associated with this keyword (acting as an identity), so it can be obtained by running $T_W = \mathsf{Extract}(params, msk, W)$.

$\mathsf{Test}(params, S_{W,M}, T_{W'})$**:** On input the searchable encryption $S_{W,M}$ and the trapdoor $T_{W'}$, it outputs M if $C_2||M = Decrypt(T_{W'}, C_1)$ and $\perp$ otherwise.

Figure 4.3: Transformation IBE-to-PEKS

Transformation to PEOKS We begin by extending the definition of PEKS. Definition 28 of PEKS extends that of Definition 25 by encoding a secret m into the PEKS element S_W generated by the PEKS algorithm. This secret m is returned by Test when a match occurs.

Definition 28 (PEKS**.)** *A* PEKS *scheme consists of the following algorithms:*

$\mathsf{KeyGen}(1^k)$**:** *given as input a security parameter k, generates a public/private key pair A_{pub}, A_{priv}.*

$\mathsf{PEKS}(A_{pub}, W, m)$**:** *given as input a public key A_{pub}, a keyword W and a message m, produces a searchable encryption S_W of m under W.*

$\mathsf{Trapdoor}(A_{pub}, A_{priv}, W)$**:** *given as input Alice's public and private keys, and a keyword W, produces a trapdoor T_W that allows searches for the keyword W.*

$\mathsf{Test}(A_{pub}, S_W, T'_W)$**:** *given as input Alice's public key, a searchable encryption S_W, and a trapdoor T'_W, outputs the message m encoded in S_W if $W = W'$ and $\perp$ otherwise.*

We construct a PEOKS scheme consisting of the algorithms Υ of such a PEKS scheme, a secure commitment scheme Commit used to commit to keywords and a BlindTrapdoor protocol, which we use in place of the standard Trapdoor algorithm. The transformation is presented in Figure 4.4.

Using the BlindTrapdoor protocol achieves the oblivious property of our PEOKS scheme. With this protocol, the user obtains trapdoors from the $\mathcal{TGC}$ without revealing the keywords. By

using commitments in conjunction with the blind anonymous IBE scheme presented in Section 3.4, the user can assure the $\mathcal{TGC}$ that she is authorised to perform searches using this keyword.

The BlindTrapdoor protocol is as follows:

BlindTrapdoor$(\mathcal{U}(A_{pub}, W, open_W), \mathcal{TGC}(A_{pub}, A_{priv}, C))$ generates a trapdoor T_W for a keyword W by running an interactive blind key extraction protocol between $\mathcal{U}$ and $\mathcal{TGC}$, BlindExtract$(\mathcal{U}(A_{pub}, W, open_W), \mathcal{KGC}(A_{pub}, A_{priv}, C))$. If $C = \mathsf{Commit}(W, open_W)$, $\mathcal{U}$'s output is the trapdoor T_W and the output of $\mathcal{TGC}$ is empty. Otherwise both parties output $\perp$.

The properties of *leak freeness* and *selective-failure blindness* follow from the BlindExtract protocol in the underlying blind IBE scheme.

Definition 29 (PEOKS) *A PEOKS scheme* $(\Upsilon, \mathsf{BlindTrapdoor}, \mathsf{Commit})$ *is secure if and only if: (1) the underlying* Υ *is a secure PEKS scheme, (2)* Commit *is a secure commitment scheme and (3)* BlindTrapdoor *is instantiated using a* BlindExtract *protocol that is* leak-free *and* selective-failure blind.

Theorem 3 *Given a PEKS scheme (*Setup,PEKS,Trapdoor,Test*) and a protocol* BlindTrapdoor*, if the PEKS scheme is PEKS-IND-CPA secure and protocol* BlindTrapdoor *is instantiated using a* BlindExtract *protocol that is leak free (with commitment), then the PEOKS scheme (*Setup*,* PEOKS*,* BlindTrapdoor*,* Test*) is PEOKS-IND-CPA secure.*

Proof. We begin by showing that if an adversary has non-negligible advantage in winning the PEOKS-IND-CPA game when the BlindExtract protocol underlying the BlindTrapdoor protocol is leak free, then we can build an algorithm A that has non-negligible advantage in winning the PEKS-IND-CPA game.

Let E be a probabilistic polynomial time adversary that has non-negligible advantage in winning the PEOKS-IND-CPA game with the protocol BlindExtract being leak free. We can build a polynomial time algorithm A that has non-negligible advantage in winning the PEKS-IND-CPA game as follows. First, A hands to E the parameters of the scheme. At every stage, A answers the oracle queries of E by acting as the $\mathcal{TGC}$ in the BlindExtract protocol. This is possible as the leak freeness property ensures that the protocol can be simulated. A uses rewinding capabilities to extract the keyword that is being queried and passes it to $Oracle_{\mathsf{Extract}}$ and receives the corresponding trapdoor, which allows A to simulate the protocol.

Given an anonymous committed blind IBE scheme with algorithms $(\mathsf{Setup}, \mathsf{BlindExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ the PEOKS scheme is as follows:

$\mathsf{KeyGen}(1^k)$**:** On input a security parameter k, runs IBE algorithm $\mathsf{Setup}(1^k)$ and returns the key pair (A_{pub}, A_{priv}) and the secret key and parameters $(msk, params)$ of the IBE scheme.

$\mathsf{PEOKS}(A_{pub}, W, M)$**:** On input public key A_{pub}, message M and a keyword W, it computes a searchable encryption $S_{W,M}$ for keyword W as follows:

1. Generate a random value $C_2 \in \{0,1\}^k$.
2. Compute $C_1 = \mathsf{Encrypt}(A_{pub}, W, m|C_2)$.
3. Output the tuple $S_W = (C_1, C_2)$.

$\mathsf{BlindTrapdoor}(\mathcal{TGC}(A_{pub}, A_{priv}, C), \mathcal{U}(A_{pub}, W, open_W))$**:** The input of $\mathcal{TGC}$ is the key pair $A_{pub}, A_{priv})$ and a commitment $C = Commit(W', open_{W'})$ to a keyword, and the input of $\mathcal{U}$ is the public key of the key pair, keyword W and a value $open_W$. Generate the trapdoor T_W for W by running the protocol $\mathsf{BlindExtract}(\mathcal{TGC}(A_{pub}, A_{priv}, C), \mathcal{U}(A_{pub}, W, open_W))$. The output of the user is the trapdoor T_W or $\perp$ if the protocol fails and the output of the TGC is nothing or $\perp$.

$\mathsf{Test}(A_{pub}, S_W, T_{W'})$**:** On input the public key A_{pub}, a searchable encryption S_W parsed as (C_1, C_2) and a trapdoor $T_{W'}$, compute $M = \mathsf{Decrypt}(A_{pub}, T_{W'}, C_1)$. If $M = m|C_2$, outputs the message m encoded in S_W; if there is no match, outputs $\perp$.

Figure 4.4: Transformation IBE-to-PEOKS

When E outputs the challenge keywords W_0, W_1 and the challenge messages M_0, M_1, A uses them as its own challenges. Given the ciphertext S_{W_b,M_c}, A passes it to E. Finally, E outputs bits b' and c', and A outputs these bits. Since E has non-negligible advantage in winning the PEOKS-IND-CPA game even when the protocol $BlindTrapdoor$ is leak free and it is clear that in this case E does not get any knowledge from the protocol, then A wins the PEKS-IND-CPA game with non-negligible advantage by outputting b' and c'.

We assume the PEKS-IND-CPA security of the underlying PEKS scheme and the commitment scheme Commit is binding. From this, we can say it is not possible to construct a distinguisher D that has non-neglibilble advantage in distinguishing between Game Real and Game Ideal of the PEOKS scheme. It is also not possible to construct an adversary $\mathcal{A}$ that has non-negligible advantage in breaking the binding value of the commitment scheme Commit. □

Definition 30 (Consistency and security for PEOKS) *A PEOKS scheme (*Setup, PEOKS, BlindTrapdoor, Test*) is secure if an only if it is PEOKS-IND-CPA secure and protocol* BlindExtract *underlying the* BlindTrapdoor *protocol is selective-failure blind. It is consistent if the underlying PEKS scheme (*Setup, PEKS, Trapdoor, Test*) is consistent.*

PEOKS allows a user to perform a search on an encrypted database without revealing the search keyword. This property of hiding the search terms from the trusted third party affords the user a greater level of privacy than in previous schemes. It also prevents the trusted third party from learning what sort of information is encrypted. Keywords such as 'USA', 'bomb' can be used to identify messages of interest. Such keywords must be chosen carefully to avoid false positives. Consider the message Alice sends Bob after her holidays: 'Just back from USA - the flights cost a bomb!'. This message will be returned as a false positive, which represents an invasion of Alice and Bob's privacy.

4.3.4 Application of Public-Key Encryption with Oblivious Keyword Search

Authorised Private Searches on Public-key Encrypted Data

We apply PEOKS to the setting of public key encrypted databases to enable oblivious searches. The construction is similar to the audit log presented above. Each data record is encrypted using a fresh random symmetric key and associated with several searchable encryptions. Each searchable encryption is generated using input of a keyword that describes the content of the record, and

a secret message that contains the symmetric key. Once an investigator obtains a trapdoor that matches a searchable encryption (i.e., both were computed on input the same keyword), she is returned the symmetric key that allows her to decrypt the record.

In constructing authorised oblivious private searches, we aim to ensure that neither the keywords of interest for the investigator nor the search results are revealed. To achieve the first property, the PEOKS scheme is employed. The investigator runs protocol BlindTrapdoor with the trapdoor generation entity ($\mathcal{TGC}$) in order to retrieve a trapdoor for a committed keyword in a blind manner. The committed blind extraction allows the $\mathcal{TGC}$ to construct policies detailing the restrictions on the data that a particular investigator can obtain. To enforce these restrictions, the $\mathcal{TGC}$ requires the investigator to prove in zero-knowledge that the keyword used to compute the commitment belongs to a certain language. Also consider a party (such as a judge) charged with deciding which keywords can be utilized by the investigator, and describe how the investigator obtains a search warrant from the judge and shows it to the $\mathcal{TGC}$. The judge and the $\mathcal{TGC}$ are only required to be involved in providing search warrants and trapdoors respectively, and can remain off-line when not required to perform these tasks.

To obscure the search results, a data structure is described that allows the use of a private information retrieval (PIR scheme) and that integrates concepts from [CGKO06] to improve the efficiency of the searches[1]. Since the PIR queries are made on encrypted data, a further requirement is that the investigator does not learn anything about data for which she is not authorised to retrieve a trapdoor. Due to the public key setting, the database only stores the public key of the PEOKS scheme.

Details on data storage. We consider a data structure in which only one searchable encryption per keyword is computed, which allows each data record to be described by several keywords. Once the investigator finds the searchable encryption that matches her trapdoor, she receives the information needed to decrypt all the data records described by the corresponding keyword. This mechanism of data storage allows for an efficient search (not all the searchable encryptions need to be tested) and is privacy enhancing in so far as it hides the number of keywords that describe a record from the investigator.

Using encrypted linked lists, store the encrypted nodes at random positions in the PIR database.

[1]The amount of PIR queries may give some indication about the number of records retrieved. This information can be hidden through dummy transactions up to an upper limit on the number of matching records.

We do this in order to hide the node corresponding to a given linked list [CGKO06], and construct one linked list per keyword. Each node in the linked list contains the information required to retrieve and decrypt one record associated with the keyword. A node contains a PIR query index P_R for the data record and the key K_R used to encrypt the record. It also stores a PIR query index to the next node on the list, and the key used to encrypt it. To encrypt the nodes and the records of data, we employ a symmetric encryption algorithm $\mathsf{Encrypt}$.

The data holder adds a keyword W for which no searchable encryption has previously been computed. To generate a searchable encryption for the keyword, she chooses a symmetric key K_{N_1}, and runs algorithm $\mathsf{PEKS}(A_{pub}, W, K_{N_1}||P_{N_1})$ to compute the searchable encryption. P_{N_1} is the PIR query index to the first node of the list and K_{N_1} is the symmetric key used to encrypt this node. She then builds the node $N_1 = (P_R, K_R, P_{N_2}, K_{N_2})$, computes $\mathsf{Encrypt}(K_{N_1}, N_1)$, and stores the node in the position given by P_{N_1}. Finally, she deletes P_{N_1} and K_{N_1} from memory but keeps values P_{N_2} and K_{N_2}. P_{N_2} and K_{N_2} are the PIR query index and the key for the next node in the list. In position P_{N_2} is stored a flag to indicate the end of the list.

When the data holde chooses this keyword to describe another record R', she builds the second node $N_2 = (P_{R'}, K_{R'}, P_{N_3}, K_{N_3})$, runs $\mathsf{Encrypt}(K_{N_2}, N_2)$, and stores the encrypted node in the position given by P_{N_2}. She deletes P_{N_2} and K_{N_2} from memory but keeps P_{N_3} and K_{N_3} to facilitate adding another node to the list. She also stores the flag in P_{N_3}. This iterative procedure is applied as many times as required.

Authorizing and performing private searches. An investigator that wants to search the encrypted database uses the following procedure:

1. The investigator requests authorisation from the judge to search a given database for a particular keyword W. Assuming the investigator holds the relevant credentials, the judge grants a warrant. In practice, the investigator runs an interactive protocol with the judge, which returns to the investigator a credential $cred$ with attribute W from the judge.

2. The investigator requests a trapdoor from the $\mathcal{TGC}$. This is a three step process:
   (a) The investigator creates a commitment $C = Commit(W, open_W)$ to the keyword W for which she wants to receive a trapdoor, and sends C to the $\mathcal{TGC}$.
   (b) The investigator and the $\mathcal{TGC}$ run the interactive protocol to verify the validity of the credential presented by the investigator and the claim that the keyword used to compute the

commitment is the same as the keyword contained in the credential's attributes.

(c) The investigator and the $\mathcal{TGC}$ execute the BlindTrapdoor protocol, with investigator input $A_{pub}, W, open_W$ and $\mathcal{TGC}$ input A_{pub}, A_{priv}, C. The protocol returns no output to the $\mathcal{TGC}$, and a trapdoor T_W to the investigator.

3. The investigator downloads the list of PEKS elements for all the keywords.

4. If an investigator performs a successful Test for a PEKS element (using the correct trapdoor), the algorithm returns the key and PIR query index pair that correspond to the first node of the list. The investigator uses the PIR scheme to retrieve the node and the first record. As above, each node returns sufficient information to link to the next node, until all data related to the keyword have been returned.

4.4 Anonymous Key Issuing

In anonymous key issuing (AKI), there are two seemingly conflicting requirements: a user's identity must not be leaked to the $\mathcal{KGC}$, yet once authenticated, the user must be able to retrieve her private key. This objective is to enhance a user's privacy by preventing the $\mathcal{KGC}$ from learning the identity associated with a key request.

We present three schemes. The first, by Sui *et al.* [SCH+05], uses the blinding property discussed in Section 3.2 with signatures. The second, by Chow [Cho09], has the user present a signed certificate of her identity to prevent the $\mathcal{KGC}$ from viewing the identity requested. An identity certifying authority (ICA) signs this certificate, which is similar to the $\mathcal{KGC}$ issuing a blind signature on the identity. Lastly, in joint work with Gray [SG09], we present a framework to achieve the same goals using blind IBE schemes. Our scheme allows a $\mathcal{KGC}$ to issue a private key to an authenticated user without learning either the identity requested or the requesting user.

4.4.1 Separable and Anonymous Key Issuing

A standard approach to authentication in IBE schemes is to have a separate registration authority RA that is responsible for authenticating users and their credentials. This authentication is similar in practice to the PKI registration authority. Sui *et al.* [SCH+05] propose a separable and anonymous key issuing scheme. The scheme involves a separation of duties between the RA and $\mathcal{KGC}$. This facilitates the $\mathcal{KGC}$ generating the private key for a user in such a way that only the legitimate

requesting user, as authenticated by the RA, can retrieve it. It aims to prevent both the $\mathcal{KGC}$ and an eavesdropper from learning the identity of the user.

The scheme uses blinding techniques, as described in Section 3.2. Their scheme is motivated by privacy concerns, as well as maintaining the property of IBE schemes that a user is not required to pre-register. The following short blind signature scheme is required.

KeyGen**:** Given as input $P \in \mathbb{G}_p$ is a point of prime order p and hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_p$ where $\mathbb{G}_p$ is an abelian group, choose $s \in \mathbb{Z}_q^*$ as the secret key sk_{signer}. Return public parameters of $\mathbb{G}_p, p, P, H$ and the public key $pk_{sign} = P^s$ where P is a point on an elliptic curve and $\mathbb{G}_p = \langle P \rangle$.

Sign**:** The sender chooses a random $r \in \mathbb{Z}_q^*$, computes $\overline{m} = H(m)^r$ and sends $\overline{m}$ to the signer. The signer computes $\overline{\sigma} = X(\overline{m}^s)$, where $X(\cdot)$ is the x co-ordinate of the element, and sends it to the sender. To retrieve the signature, the sender computes $\sigma = \overline{\sigma}^{r^{-1}}$.

Verify**:** Given as input pk_{signer}, m, $\sigma(m)$, find $y, \in \mathbb{F}_q$ such that $S = (\sigma, y)$ is a point of order p in $E(\mathbb{F}_q)$. Test if either $e(S, P) = e(H(m), pk_{signer})$ or $e(S, P)^{-1} = e(H(m), pk_{signer})$.

The protocol describe in Figure 4.5 achieves separable key issuing. The user is required to have registered their identity id and a corresponding access password $password$ in advance with an RA. This requirement of the system loses one of the advantageous features of IBE schemes. Two $\mathcal{KGC}$ entities are required in this protocol, both with access to databases containing the tuple $(id, password)$. The hashes of these values may be pre-computed and stored in the database held by each $\mathcal{KGC}$.

The key requests are conducted in a manner that prevents the $\mathcal{KGC}$ from learning the request directly. However, the presence of the tuple id and $password$ in the databases held by each $\mathcal{KGC}$, along with their hash, means that a $\mathcal{KGC}$ can link the key request to the identity using the hash of the tuple. As anonymity is usually described as having the property of unlinkability [PH05], the scheme described in Figure 4.5 can not be deemed anonymous with respect to the $\mathcal{KGC}$.

Additionally, the need for the user to pre-register their (id, $password$) tuple prior to the key extraction phase means that the the IBE scheme no longer has the property of spontaneity. In a spontaneous scheme, there is no requirement on the user to have performed an action before the key generation.

Given $\mathcal{KGC}_1$ and $\mathcal{KGC}_2$, their (jointly generated) public key is $pk_{\mathcal{KGC}} = P^{s_1 s_2}$ where $pk_{\mathcal{KGC}1} = P^{s_1}$ and $pk_{\mathcal{KGC}_2} = P^{s_2}$ and $s_1, s_2 \in \mathbb{Z}_q^*$. Suppose $\mathcal{KGC}_1$, $\mathcal{KGC}_2$ have access to databases as described above. The user $\mathcal{U}$ executes the following interactive key extraction protocol with each $\mathcal{KGC}$.

1. $\mathcal{U}$ chooses a random r_1, computes $Q_1 = H(id)^{r_1}, T_1 = H(password)^{r_1}$ and sends T_1, Q_1 to $\mathcal{KGC}_1$.
2. $\mathcal{KGC}_1$ tests the validity of the id, $password$ tuple by checking that $e(Q_1, T_1) = e(H(id), H(password))$ holds for a tuple in the database. If it does, $\mathcal{KGC}_1$ computes $S_1 = Q_1^{s_1}, \sigma_1' = T_1^{s_1}$ and sends (S_1, σ_1') to $\mathcal{U}$.
3. $\mathcal{U}$ verifies the blinded partial private key by checking $e(S_1, P) = e(Q_1, pk_{\mathcal{KGC}_1})$. $\mathcal{U}$ verifies the signature by checking $e(\sigma_1', P) = e(T_1, pk_{\mathcal{KGC}_1})$. If both hold, $\mathcal{U}$ unblinds to obtain the partial private key $H(id)^{s_1}$ and signature $\sigma_1 = H(password)^{s_1}$.
4. $\mathcal{U}$ selects a random r_2, computes $Q_2 = H(id)^{s_1 r_2}$, $T_2 = H(password)^{\frac{1}{r_2}}$ and sends Q_2, T_2 to $\mathcal{KGC}_2$.
5. $\mathcal{KGC}_2$ checks the validity of the request by testing $e(Q_2, T_2) = e(H(id), \sigma_1)$ holds, and the validity of the signature by checking $e(\sigma_1, P) = e(H(password), pk_{\mathcal{KGC}_1})$. If they hold, $\mathcal{KGC}_2$ computes $S_2 = Q_2^{s_2}$ and sends S_2 to $\mathcal{U}$.
6. $\mathcal{U}$ verifies the blinded private key by checking $e(S_2, P) = e(Q_2, pk_{\mathcal{KGC}_2})$. If this holds, she obtains the private key by unblinding $sk_{id} = S_2^{\frac{1}{r_2}} = H(id)^{s_1 s_2}$.

Figure 4.5: Separable and Anonymous Key Issuing without Key Escrow

4.4.2 Anonymous Private Key Issuing

Chow [Cho09] presents an anonymous private key issuing protocol that consists of the Setup and KeyGen protocols of an IBE scheme, as well as the following four polynomial algorithms.

IKeyGen: The ICA generates a public/private key pair for certification, pk_{cert}, sk_{cert}.

SigCert: Given as input identity id, sk_{cert}, the ICA outputs a certificate $cert$ for id and some auxiliary information aux to $\mathcal{U}$.

IssueKey / ObtainKey: Given as input the master public key mpk, id and $cert$, $\mathcal{U}$ receives the secret key sk_{id} as output. Given as input msk and $cert$, the $\mathcal{KGC}$ receives nothing as output.

Chow presents the new security notion of *$\mathcal{KGC}$ anonymous ciphertext indistinguishability* (ACI-KGC). It protects against adversaries who hold msk but not the identity list.

Definition 31 *An IBE scheme is (t, q_E, ϵ) ACI-KGC secure if all t-time adversaries making at most q_E embedded-identity encryption oracle queries have advantage at most ϵ in winning the following game:*

Experiment Exp $^{aci-kgc}_{IBE,\mathcal{A}}(\lambda)$

$params \leftarrow \mathsf{Setup}(1^\lambda)$; $id^* \leftarrow \{0,1\}^n$;

$(mpk, st) \leftarrow \mathcal{A}(\mathsf{KeyGen}, params)$; *if* $mpk \notin params$ *then return 0;*

$(m_0^*, m_1^*, st) \leftarrow \mathcal{A}^{\mathsf{EncO}(mpk, id^*)}(\cdot)('find', mpk, st)$

If $\{m_0^*, m_1^*\} \not\subseteq \mathsf{MsgSp}(\lambda) or |m_0^*| \neq |m_1^*|$ *then return 0*;

$b \leftarrow \{0,1\}$; $ct \leftarrow \mathsf{Encrypt}(mpk, id^*, m_b^*)$; $b' \leftarrow \mathcal{A}^{\mathsf{EncO}\,mpk, id^*}(\cdot)('guess', ct, st)$;

If $b \neq b'$, *then return 0 else return 1.*

Chow's protocol uses a modified variant of Gentry's anonymous IBE scheme [Gen06], as presented in Section 2.6.7. This modification is used to achieve the property of ACI-KGC. The architecture presented requires a non-colluding ICA. The modification consists of separating the master key generation from the Setup phase, as shown in Figure 4.6.

4.4.3 Anonymous Key Issuing using Blind Identity-Based Encryption

The anonymous key issuing protocol we contribute [SG09] focuses on the Setup and Extract phases of IBE, the Encrypt and Decrypt are as outlined in Naccache's scheme presented in Section 2.6.5. As in Chow's scheme, master key generation is separated from the Setup stage, reducing further the level of trust required in the $\mathcal{KGC}$. The $\mathcal{KGC}$ remains the only entity holding the master secret key, but it is prevented by this design from maliciously choosing system parameters. This scheme requires only one authenticated connection with the user, and the $\mathcal{KGC}$ does not interact directly with the key-requesting user at any point.

We present a framework for the partial-blind IBE, along with the adaptations required to achieve partial-blind and double-blind IBE. The data flow outlined below centres on the interactive PartialBlindExtract protocol, presented in Section 3.5. The $\mathcal{KGC}$ holds the msk, and thus is the only entity involved that can generate keys. The objective is to prevent it from holding an identity list.

An Authentication Authority ($\mathcal{AA}$) in introduced, and is responsible for authenticating the

1. Setup: The trusted initialiser chooses group $\mathbb{G}$ according to the security parameter, and selects g, h_1, h_2, h_3 randomly from $\mathbb{G}$. It also chooses a hash function $H : \{0,1\}^n \rightarrow \mathbb{Z}_q$ from a family of universal one-way hash functions. The public parameter $params$ is g, h_1, h_2, h_3, H).

2. MKeyGen: The $\mathcal{KGC}$ chooses a random exponent $\alpha \in \mathbb{Z}_q$ and sets $g_1 = g^\alpha \in \mathbb{G}$. The master public/private key pair is given by ($mpk = g_1, msk = g$).

3. IKeyGen: The ICA generates a key pair (pk_{cert}, sk_{cert}) for the signature scheme.

4. SigCert: For $id \in \{0,1\}^n$, the ICA creates the certificate $cert$ = (σ,com,aux) by randomly picking aux from the decommitment-string space and generating a signature σ = Commit(id, aux) by running the signing algorithm.

5. ObtainKey(mpk, id, $cert$, aux) $\leftrightarrow$ IssueKey(msk, $cert$):

   (a) $\mathcal{U}$ and $\mathcal{KGC}$ engage in a secure two-party computational protocol with $\mathcal{U}$ input random $r \in \mathbb{Z}_q$, id, aux and $\mathcal{KGC}$ input of α. $\mathcal{KGC}$ receives private output of $x = (\alpha - id)r$ if com = Commit(id, aux) or $x = \perp$ otherwise.

   (b) If $x \neq \perp$, $\mathcal{KGC}$ randomly picks $\tau_{id,1} \in \mathbb{Z}_q$ and computes $usk'_{cert} = (usk'_1 = (h_1 g^{-\tau_{id,1}})^{\frac{1}{x}}, usk'_2 = \tau_{id,1})$.

   (c) $\mathcal{U}$ outputs $(usk_1, usk_2)) = (usk'^r_1, usk'_2) = ((h_1 g^{-\tau_{id,1}})^{\frac{1}{(\alpha - id)}}, usk'_2)$.

Figure 4.6: Chow's Anonymous Private Key Issuing

user's credentials. The user passes her identity, along with blinding elements and any necessary credentials to the $\mathcal{AA}$. If the user's credentials are valid, the $\mathcal{AA}$ constructs the required blinding (full or partial) of the identity, and passes it to the $\mathcal{KGC}$. The $\mathcal{AA}$ must be trusted not to impersonate users, and not to collude with the $\mathcal{KGC}$ by revealing individual identities or the identity list. Should the $\mathcal{AA}$ construct a blinding of an identity dishonestly, it will be detected by the user when the private key is returned.

It is possible to achieve anonymous key issuing in the absence of the $\mathcal{AA}$ using blind IBE. However, the advantages of using this third party are twofold. Firstly, the user $\mathcal{U}$ never authenticates to the $\mathcal{KGC}$. This prevents the $\mathcal{KGC}$ from learning who has requested keys. Secondly, the credentials that $\mathcal{U}$ presents can be trivially checked by the $\mathcal{AA}$. An architectural view of the scheme is shown in Figure 4.7.

1. $\mathcal{AA}$ runs Setup(1^λ) and outputs the system parameters $params$ for security parameter $\lambda \in \mathbb{N}$, with message space MsgSp.

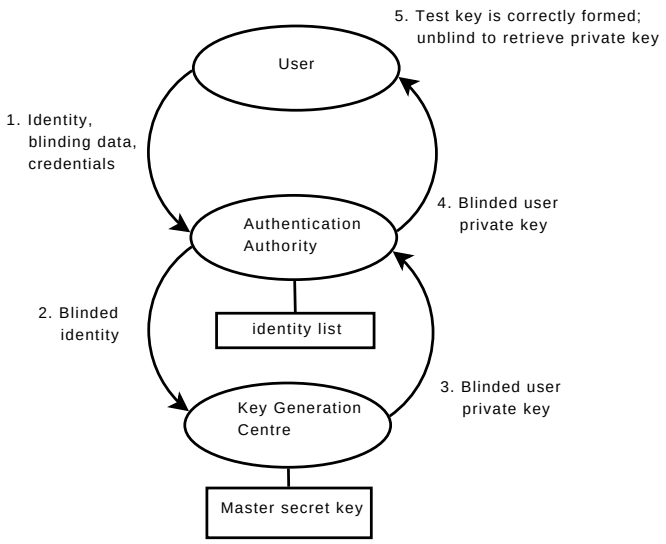

Figure 4.7: Anonymous Key Issuing using Blind Identity-Based Encryption

2. $\mathcal{KGC}$ receives $params$ and runs MKeyGen($params$) to output the master secret msk and public key $params$ conforming to $params$.

Note that this change does not affect the original security guarantees of Nacchache's IBE scheme.

The first step of the protocol is for the user to generate her randomly generated blinding values β, y as per step 1 of the PartialBlindExtract protocol (Section 3.5), choose her identity and gather the necessary credentials for the identity. The blinding values are kept private from $\mathcal{AA}$. $\mathcal{U}$ passes to $\mathcal{AA}$ her identity $v = (v_i)$, relevant credentials $cred$ and sufficient blinding information $u_i^{\beta}, g^{\beta y}, u'^{\beta}$ to allow $\mathcal{AA}$ to construct the blinding of the identity.

If $\mathcal{U}$ convinces $\mathcal{AA}$ that she is entitled to the identity requested, $\mathcal{AA}$ constructs the blind identity hash X. This prevents the $\mathcal{KGC}$ from learning any elements of the identity. $\mathcal{KGC}$ constructs the blinded private key d'_v, which is returned to $\mathcal{AA}$ in step three. $\mathcal{AA}$ is unable to unblind d'_v as it does not have the required blinding values, and returns d'_v to $\mathcal{U}$.

$\mathcal{U}$ tests that d'_v is correctly constructed. At this point, if $\mathcal{AA}$ or $\mathcal{KGC}$ have behaved maliciously, $\mathcal{U}$ will not receive a correctly formed key for her identity and this test will fail, causing her to reject d'_v. If the test is passed, she uses her blinded values to retrieve her private key.

AKI data flow

1. $\mathcal{U} \rightarrow \mathcal{AA}$: $v_i, u_i^{\beta}, g^{\beta y}, u'^{\beta}, cred$
2. $\mathcal{AA} \rightarrow \mathcal{KGC}$: $X = (g^{\beta y} u'^{\beta} \prod_{i=1}^{n} u_i^{\beta v_i})$
3. $\mathcal{KGC} \rightarrow \mathcal{AA}$: $d'_v = (d'_1, d'_2)$
4. $\mathcal{AA} \rightarrow \mathcal{U}$: $d'_v = (d'_1, d'_2)$
5. $\mathcal{U}$ tests key : $e(g_1, g_2) \cdot e(d'_2, u' \prod_{i=1}^{n} u_i^{v_i}) = e(g, d'_1)$

   $\mathcal{U}$ unblinds : choose $z \in \mathbb{Z}_q$ and compute

   $d_v = \left(d'_1/(d'_2)^y \cdot (u' \prod_{i=1}^{n} u_i^{v_i})^z, d'_2 \cdot g^z\right)$.

Note that neither the $\mathcal{KGC}$ nor the $\mathcal{AA}$ know d_1 or d_2, where $d_v = (d_1, d_2)$.

Partial-blind Adaptation In the partial-blind variant of the scheme, $\mathcal{KGC}$ can insist on certain elements being present in the identity string and is assured of their presence as they are unblinded. $\mathcal{KGC}$ could insist on a certain expiration date or some similar generic element of the identity string. Such elements can be visible to $\mathcal{KGC}$. This requires changing step 2 in the data flow to contain a partial-blinding of the identity.

Partial-blind IBE AKI data flow

2. $\mathcal{AA} \rightarrow \mathcal{KGC}$: $X = (g^{\beta y} u'^{\beta} \prod_{i=1,\gamma_i=1}^{n} u_i^{\beta v_i})$,
   (u_i^{β}, v_i) where $\gamma_i = 0$ and g^{β}, u'^{β}.

Double-blind Adaptation In the double-blind variant of the scheme, $\mathcal{KGC}$ can insist on certain elements being present in the identity string as above, which are unblinded. $\mathcal{KGC}$ can also insert elements into the identity string which remain unknown to $\mathcal{U}$, which therefore are double blinded. The identity string in this case consists of the partially-blinded elements presented to $\mathcal{AA}$ by the user and the double-blinded elements inserted by $\mathcal{KGC}$.

This requires changing step 2 in the data flow to contain a double-blinding of the identity.

Double-blind AKI IBE data flow

2. $\mathcal{AA} \rightarrow \mathcal{KGC}$: $X = (g^{\beta y} u'^{\beta} \prod_{i=1,\gamma_i=1}^{m} u_i^{\beta v_i})$,
   (u_i^{β}, v_i) where $\gamma_i = 0$ and g^{β}, u'^{β} for $i \in \{1, m\}$
   u_i^{β} for $i \in \{m+1, n\}$.

4.5 Unique Receipt Issuing using Double-blind IBE

A natural requirement of IBE schemes is for the user to know the identity corresponding to her private key, so double-blind IBE may seem unintuitive. We motivate its usefulness with a unique receipt issuing protocol [SG09]. Blind signatures are used in a host of schemes from e-cash [Cha82] to online lotteries [LC09].

Lotteries are a common way for charities to raise money, and are characterised by a large number of off-line participants. In traditional lotteries, users can buy tickets over a relatively long period in advance. Lotteries must be fair and publicly verifiable. We propose using the double-blind IBE scheme to construct a receipt issuing scheme, and apply it to online lotteries.

The Scheme

The following is a simple protocol that allows a user to anonymously purchase a lottery ticket electronically. The scheme is anonymous, unless the participant makes a claim on the lottery, in which case she reveals her name.

Purchase The participant constructs her purchase request as the identity string $numbers|\overline{name}$, where $numbers$ contains the k numbers of her choosing for the forthcoming draw from n numbers and $\overline{name}$ contains her name, blinded. The partially-blinded identity string is sent to the Lottery Agent (LA). It is trivial to include other details in the identity string, such as the date of the draw. The LA executes the interactive DoubleBlindExtract protocol with the participant, generating the private key corresponding to the participant's identity string.

The LA adds some private validation information, $lotto_{id}$, to the identity string. This string is used to hold unique draws and ensure tickets are one-time use only. The resulting private key corresponds to the participant's lottery ticket. The LA places the blinded identity, corresponding to X in the DoubleBlindExtract protocol, and the chosen numbers $numbers$ onto a public bulletin board. This board is not made public until ticket sales are closed, but is available prior to numbers being drawn.

The lottery occurs, and numbers are drawn and announced. The fairness and random generation properties can be achieved using traditional methods such as weighing the lottery balls.

Claim A participant who wants to make a claim on the lottery draw sends the identity string $numbers|name$ to the LA. The LA encrypts some nonce $\mathcal{N}$ using $numbers|name|lotto_{id}$

as the identity string. The ciphertext $ct = \{\mathcal{N}\}_{numbers|name|lotto_{id}}$ is passed as a challenge to the participant . If the participant can decrypt the challenge and retrieve the nonce, she has proved ownership of the ticket (private key) corresponding to the numbers. A valid claim by the participant should allow him to send the decrypted nonce to the LA and for her claim to be upheld. Failure to decrypt the nonce correctly is identified as a false claim.

The protocol is shown in Figure 4.8.

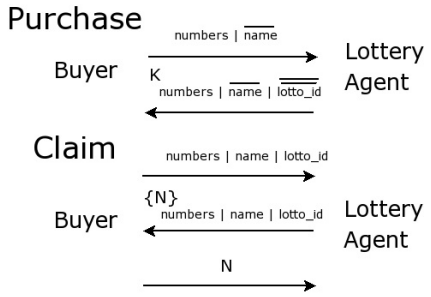

Figure 4.8: Online Lottery Protocol

Discussion

Online lotteries have a variety of requirements with are necessary to achieve a fair and unbiased scheme.

Security A participant attempting to forge a wining ticket should not succeed

Proof. The lottery ticket is the private key corresponding to the string $numbers|name|lotto_{id}$. In order for a cheating participant to forge such a key, she is required to learn both the master secret msk of the IBE scheme as well as the private lottery validity information $lotto_{id}$ for the specific draw to generate a forgery.

□

Correctness Participants must receive the ticket corresponding to numbers of their choosing. Her choices cannot be falsified.

Proof. The LA is prevented from falsifying the participant's choice of numbers by the key correctness test. If they are excluded from the identity string, the hash of the string used by the participant to verify the key will not correspond to the elements of the identity string she

has chosen. Similarly, an eavesdropper cannot manipulate the identity string to falsify the participant's choice of numbers without being caught in the same manner.

□

Anonymity / Privacy Given a ticket, it should not be possible to link the identity of a participant with her choices. Only owner of the winning ticket should be identified.

Proof. This is property is achieved in two ways. Firstly, the LA is unable to link a key request with the resulting unblinded key. Secondly, the participant identity is blinded in the identity string presented to the lottery agent. It is only necessary to reveal this identity in the case of a claim on the lottery. □

Publicly verifiable Participants and observers can observe the lottery and verify the winning result.

Proof. The presence of each lottery choice on the bulletin board along with the participant's blinded element of the identity string affords individual and universal verifiability. Individuals can check that their ticket has been correctly recorded and observers can check that a winning ticket was recorded on the bulletin board before the lottery draw occurred. □

Pre-registration not required Participants are not required to pre-register in order to buy tickets.

Proof. One of the features of identity-based encryption schemes is there is no pre-registration step required. Our scheme retains this property. □

k-out-of-n choice Participants choose k values out of a possible n in the lottery.

Proof. Participants are restricted to k elements in the non-blinded part of the identity string. Any attempt to include additional values in the blinded part is easily detected if the participant claims a prize. □

4.6 Conclusion

In this chapter, we have revisited the concept of blind identity-based encryption. We have provided some insight into the uses it has in practical applications, and sketched the resulting applications. We began with an application of blind identity based encryption to oblivious transfer, the scenario which motivated its original design. We then presented an adapted public key encryption with

keyword search, which uses blind identity based encryption to obscure the keywords from the key extraction entity. We showed that blind identity based encryption provides a natural solution to anonymous keyword search. Finally, we presented a unique receipt issuing protocol, and its application to online lotteries.

Chapter 5

Conclusion

5.1 Review

In this thesis we have demonstrated that blind identity-based encryption schemes can be used to address the level of trust required in the key generation centres. Commonly known as the key escrow problem, the ability of a key generation centre to produce multiple keys, or multiple copies of a single key, for a given identity is an issue that has received considerable attention in the literature.

We addressed this problem by focusing on the identity string. If a key generation centre does not learn the identity string belonging to a user during the extraction phase, its ability to interfere with the user's communications is reduced. It no longer holds an identity list consisting of all the identities for which a key has been requested. While a key generation centre can still generate a private key for any given identity, it no longer has the advantage of this identity list. It is no longer privy to them during the extraction phase, and is reduced to trial and error guessing of relevant identity strings.

We investigated beneficial features of the blinding property and of existing identity-based encryption schemes. We presented the first construction of anonymous blind identity-based encryption. Using the underlying anonymous identity-based encryption scheme we constructed, we generated the corresponding blind extraction protocol. Anonymity is a desirable feature for use in conjunction with the blindness property, as it prevents a ciphertext from leaking the identity string.

We have extended the scope of blind extraction protocols to incorporate the existing property of partial-blindness. Partial-blindness allows the key generation centre to place restrictions on the content of the identity string without the need for proof systems. We proposed the novel property

of double-blindness. Double-blindness represents the first such construction that allows a key generation centre to embed elements into an identity string without the user learning them. As with all our schemes, we have contributed security definitions and arguments to accompany each construction.

We have demonstrated that constructions that blind the identity string have useful applications. Our public-key encryption with oblivious keyword search advances existing search schemes by obscuring the search term. The data holder does not learn anything about the encrypted data it holds. Furthermore, the trapdoor generation entity does not learn anything about the requested search terms. This is a valuable advancement as such terms can reveal information on the encrypted data.

In our anonymous key issuing application, we proposed a protocol that not only prevents a key generation centre from learning the identity string but also prevents it from learning the identities of the requesting users. By removing the need for users to authenticate to the key generation centre directly, the identity of the users as well as their chosen identity strings remain unknown. This represented the first such scheme in which the key generation centre does not learn the identity string or the identity of the requesting user and cannot link the key issuing protocol to either.

We proposed an online lotto system as a motivating application for double-blind identity-based encryption. We provided simple security arguments in support of the scheme.

5.2 Open Questions

Anonymity of identity-based encryption schemes does not account for the situation where a key generation centre, holding the master secret, tests the ciphertext to see if it has been encrypted using a particular identity string. Recent work [IP08] defines the concept of Key Anonymity with respect to the Authority. This work focuses on key-encapsulation methods. The construction of a blind identity-based encryption scheme that provides this form of security to the user would be a positive development.

The concept of double-blinding requires further study. It represents a paradoxical problem in that a user does not learn her full public-key. In our schemes, we use it as a signature scheme. An open problem is to distil its properties into a strong motivating application using double-blind identity based encryption as an encryption scheme. Such paradoxical problems are often the most interesting in cryptography.

The key generation centre is always required to be the entity holding the master secret. The

holy grail of identity-based systems is to somehow construct a key that prevents a key generation centre from reproducing it and from generating it independently of the authentic requesting user. Our work has confirmed that the level of trust in the key generation centre can be reduced. This provides further support that the continued questioning of the level of trust required in the key generation centre is a valid pursuit.

Bibliography

[ABC+08] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. *Journal of Cryptology*, 21(3):350–391, 2008.

[AF96] M. Abe and E. Fujisaki. How to date blind signatures. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 1996)*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1996.

[AIR01] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2001)*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2001.

[AO00] M. Abe and T. Okamoto. Provably secure partially blind signatures. In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 2000)*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286. Springer, 2000.

[ARP03] S.S. Al-Riyami and K.G. Paterson. Tripartite authenticated key agreement protocols from pairings. In *Proceedings of the IMA Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 332–359. Springer, 2003.

[BB04a] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2004)*, volume 3027, pages 223–238. Springer, 2004.

[BB04b] D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 2004)*, volume 3152, pages 443–459. Springer, 2004.

[BBDP01] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 2001)*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer, 2001.

[BD08] J. Birkett and A.W. Dent. Relations among notions of plaintext awareness. In *Proceedings of the International Workshop on Practice and Theory in Public-key Cryptography: (PKC 2008)*, Lecture Notes in Computer Science, pages 47–64. Springer, 2008.

[BDCOP04] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2004)*, volume 3027, pages 506–522. Springer, 2004.

[BDMN06] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: Framework and applications. In *2006 IEEE Symposium on Security and Privacy*, pages 184–198, 2006.

[BDPR98] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 1998)*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.

[BF01] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 2001)*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, 2001.

[Bla05] I.F Blake. *Advances in Elliptic Curve Cryptography*, volume 317 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2005.

[BM05] W. Bagga and R. Molva. Policy-based cryptography and applications. In *Financial Cryptography and Data Security: (FC 2005)*, volume 3570 of *Lecture Notes in Computer Science*, pages 72–87. Springer, 2005.

[Bol03] A. Boldyreva. Efficient threshold signature, multisignature and blind signature schemes based on the gap-diffie-hellman-group signature scheme. In *Proceedings of the International Workshop on Practice and Theory in Public-key Cryptography Proceedings: (PKC 2003)*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 62–73. ACM New York, NY, USA, 1993.

[BR95] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in cryptology (EUROCRYPT 1994)*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1995.

[Bra00] S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000.

[BW06] X. Boyen and B. Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 2006)*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.

[Can04] J. C. Cannon. *Privacy: what developers and IT professionals should know*. Addison-Wesley Professional, 2004.

[CGH04] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004.

[CGKO06] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proceedings of the ACM Conference on Computer and Communications Security: (CCS 2006)*, pages 79–88. ACM New York, NY, USA, 2006.

[Cha82] D. Chaum. Blind signatures for untraceable payments. In *Proceedings of Advances in Cryptology (CRYPTO 1983)*, volume 82 of *Lecture Notes in Computer Science*, pages 23–25, 1982.

[Cha83] D. Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1983.

[CHK07] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, 2007.

[Cho09] S. S. M. Chow. Removing escrow from identity-based encryption. In *Proceedings of the International Conference on Practice and Theory in Public Key Cryptography: Public Key Cryptography (PKC 2009)*, volume 5443 of *Lecture Notes in Computer Science*, pages 256–276. Springer, 2009.

[CKRS09] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In *Proceedings of the International Conference on Practice and Theory in Public Key Cryptography: Public Key Cryptography (PKC 2009)*, volume 5443 of *Lecture Notes in Computer Science*, pages 196–214. Springer, 2009.

[CNS07] J. Camenisch, G. Neven, and A. Shelat. Simulatable adaptive oblivious transfer. In *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2007)*, volume 4515 of *Lecture Notes in Computer Science*, pages 573–590. Springer, 2007.

[Coc01] C. Cocks. An identity-based encryption scheme based on quadratic residues. In *Proceedings of the IMA International Conference on Cryptography and Coding*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer, 2001.

[CS06] S. Chatterjee and P. Sarkar. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In *Proceedings of the International Conference on Information Security and Cryptology (ICISC 2005)*, volume 3935 of *Lecture Notes in Computer Science*, pages 424–440. Springer, 2006.

[Den06] A. 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.

[DH76] W. Diffie and M. Hellman. New directions in cryptology. *IEEE Transaction on Information Theory*, 22:644–654, 1976.

[DL99] W. Diffie and S. E. Landau. *Privacy on the Line*. MIT Press Cambridge, Mass, 1999.

[Elg85] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Gea87] C. Gearty. The Courts and Recent Exercises of the Prerogative. *The Cambridge Law Journal*, pages 372–374, 1987.

[Gen06] C. Gentry. Practical identity-based encryption without random oracles. In *Proceedings of the International Cryptology Conference: (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.

[GH07] M. Green and S. Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 2007)*, volume 4833 of *Lecture Notes in Computer Science*, pages 265–282. Springer, 2007.

[GHK06] D. Galindo, J. Herranz, and E. Kiltz. On the generic construction of identity-based signatures with additional properties. In *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 2006)*, volume 4284 of *Lecture Notes in Computer Science*, pages 178–193. Springer, 2006.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 291–304. ACM Press New York, NY, USA, 1985.

[GÓhS07] S.D. Galbraith, C. Ó hÉigeartaigh, and C. Sheedy. Simplified pairing computation and security implications. *Journal of Mathematical Cryptology*, 1(3):267–281, 2007.

[Gol01] O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.

[Goy07] V. Goyal. Reducing trust in the PKG in identity-based cryptosystems. In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 2007)*, volume 4622 of *Lecture Notes in Computer Science*, pages 430–447. Springer, 2007.

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GSW08] V. Goyal, A. Sahai, and B. Waters. Black-box accountable authority identity-based encryption. In *Proceedings of the ACM conference on Computer and Communications security: (CCS 2008)*, pages 427–436. ACM New York, NY, USA, 2008.

[HL02] J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2002)*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer, 2002.

[HPFS02] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile, RFC 3280, 2002.

[IP08] M. Izabachene and D. Pointcheval. New anonymity notions for identity-based encryption. In *Proceedings of the International Conference on Security and Cryptography for Networks*, volume 5229 of *Lecture Notes in Computer Science*, pages 375–391. Springer, 2008.

[JLO97] A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures (extended abstract). In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 1997)*, volume 1294 of *Lecture Notes in Computer Science*, pages 150–164. Springer, 1997.

[Jou00] A. Joux. A one-round protocol for tripartite Diffie-Hellman. In *Proceedings of the Algorithmic Number Theory Symposium Conference: (ANTS-IV)*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[JWO75] F. G. Jacobs, R. C. A. White, and C. Ovey. *The European convention on human rights*. Clarendon press, 1975.

[Kil88] J. Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the Annual ACM Symposium on Theory of Computing: (STOC 1988)*, pages 20–31. ACM, 1988.

[KM07] N. Koblitz and A.J. Menezes. Another look at "Provable Security". *Journal of Cryptology*, 20(1):3–37, 2007.

[LC09] J. S. Lee and C. C. Chang. Design of electronic t-out-of-n lotteries on the Internet. *Computer Standards & Interfaces*, 31(2):395–400, 2009.

[LV09] B. Libert and D. Vergnaud. Towards black-box accountable authority IBE with short ciphertexts and private keys. In *Proceedings of the International Workshop on Practice and Theory in Public Key Cryptography: (PKC 2009)*, pages 235–255. Springer, 2009.

[Mao03] W. Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall Professional Technical Reference, 2003.

[MH78] R. Merkle and M. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5):525–530, 1978.

[Mil04] V. S. Miller. The Weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

[MVOV97] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[Nac07] D. Naccache. Secure and practical identity-based encryption. *IET Image Processing*, 1(2):59–64, 2007.

[Nat08] United Nations. *Universal declaration of human rights*. United Nations Publications, 2008.

[NP99] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the annual ACM symposium on Theory of Computing*, pages 245–254. ACM Press New York, NY, USA, 1999.

[NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Annual ACM Symposium on Discrete Algorithms*, pages 448–457. Society for Industrial and Applied Mathematics, 2001.

[NP05] M. Naor and B. Pinkas. Computationally secure oblivious transfer. *Journal of Cryptology*, 18(1):1–35, 2005.

[OK04] W. Ogata and K. Kurosawa. Oblivious keyword search. *Journal of Complexity*, 20(2-3):356–371, 2004.

[Oka06a] T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *Proceedings of the Theory of Cryptography Conference: (TTC 2006)*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99. Springer, 2006.

[Oka06b] T. Okamoto. On pairing-based cryptosystems. In *Proceedings of the International Conference on Cryptology: Progress in Cryptology (VIETCRYPT 2006)*, volume 4341, pages 50–66. Springer, 2006.

[Pai99] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 1999)*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[Ped91] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 1991)*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[PH05] A. Pfitzmann and M. Hansen. M.: Anonymity, unlinkability, unobservability, pseudonymity, and identity management a consolidated proposal for terminology. version 0.26. Technical report, 2005. `http://www.freehaven.net/anonbib/cache/terminology`.

[PS96] D. Pointcheval and J. Stern. Provably secure blind signature schemes. In *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 1996)*, volume 1163 of *Lecture Notes in Computer Science*, pages 252–265. Springer, 1996.

[Rab81] M. Rabin. How to exchange secrets by oblivious transfer. Technical report, 1981. TR-81, Harvard Aiken Computation Laboratory, 1981.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sat07] T. Satoh. On pairing inversion problems. In *Proceedings of the International Conference on Pairing-Based Cryptography: (Pairing 2007)*, volume 4575 of *Lecture Notes in Computer Science*, pages 317–328. Springer, 2007.

[SCH+05] A. Sui, S. S. M. Chow, L. C. K. Hui, SM Yiu, KP Chow, WW Tsang, CF Chong, KH Pun, and HW Chan. Seperable and anonymous identity-based key issuing without secure channel. *Proceedings of the International Conference on Parallel and Distributed Systems: (ICPADS 2005)*, 2:275–279, 2005.

[SG09] C. Sheedy and D. Gray. Extensions to blind identity-based encryption and applications. Unpublished, 2009.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[Sha84] A. Shamir. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, 30(5):699–704, 1984.

[Sha85] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of Advances in Cryptology (CRYPTO 1984)*, volume 1440 of *Lecture Notes in Computer Science*, pages 47–53. Springer Verlag, 1985.

[SK08] C. Sheedy and P. Kumaraguru. A Contextual Method for Evaluating Privacy Preferences. In *Proceedings of Policies and Research in Identity Management: IFIP Wg 11.6 Working conference on Policies and Research in Identity Management (IDMAN 2007)*, pages 139–146. Springer, 2008.

[SOK00] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairings. In *Proceedings of the Symposium on Cryptography and Information Security: (SCIS2000)*, 2000.

[SW05] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2005)*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[TI89] S. Tsujii and T. Itoh. An ID-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communications*, 7(4):467–473, 1989.

[TO08] I. Teranishi and W. Ogata. Relationship between two approaches for defining the standard model PA-ness. In *Information Security and Privacy (ACISP 2008)*, volume 5107 of *Lecture Notes in Computer Science*, pages 113–127. Springer, 2008.

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 2005)*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–27. Springer, 2005.

[WBDS04] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters. Building an encrypted and searchable audit log. *Proceedings of the Annual Network and Distributed System Security Symposium: (NDSS 2004)*, 6, 2004.

[Yao82] A. C. Yao. Protocols for secure computations. In *Proceedings of the Annual IEEE Symposium on Foundations of Computer Science: (SFCS 82)*, pages 160–164, 1982.