Cryptography for Secure and Private Databases: Enabling Practical Data Access without Compromising Privacy

by

Matthew Daniel Green

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

January, 2009

Abstract

In 2006 America Online's research division leaked the web search histories of more than 600,000 of their customers. While this data had been stripped of customer names and identifying information, it nevertheless revealed deeply private information about these individuals' identities and interests.

Access to information is becoming fundamental to our society, whether it is a web search or a look at one's health records. While much research has considered the problem of securing data *within* the database, there exist applications where the content of the users' queries is more sensitive. For example, a doctor who queries a medical records database may inadvertently reveal information that can harm his patient's interests (*e.g.,* queries by a disease specialist might indicate a potential infection, and thus impact insurance coverage decisions).

In this work we propose *privacy-preserving* databases in which a central database serves a pool of users without learning their query pattern. These systems will have several competing requirements. First, we require that the database operator learn *nothing* about which items the user is asking for, or even the user's identity. This guarantee must hold according to a strong security definition that takes into account the possibility of a malicious operator who tampers with the protocol. Secondly, we require that the database operator retain the ability to control access to items within the database. This seems quite challenging, however, since access control appears to be fundamentally incompatible with our desired privacy requirements.

A promising technology for constructing oblivious databases is Oblivious Transfer (OT). In a k-out-of-N OT protocol, a Sender with a collection of N messages interacts with a Receiver such that the Receiver obtains any k of the messages, and no information about the rest of the database. For its part, the Sender learns nothing about *which* messages the Receiver requested. Unfortunately, while a k-items-out-of-N policy can be considered a basic form access control, it is not powerful enough for many practical applications. Furthermore, many existing OT constructions are vulnerable to *selective-failure* attacks that may effectively compromise user privacy if undertaken by a malicious database operator.

In this work we propose several methods that address these problems *efficiently* and under strong definitions of security. We will then show how these techniques may be combined in order to produce a complete solution. Specifically, we propose:

ABSTRACT

1. Two new protocols for k-out-of-N Oblivious Transfer (OT) based on techniques from the field of Identity Based Encryption (IBE). Proposed by Shamir [Sha84] and realized by Boneh-Franklin [BF01], IBE is a powerful technology that greatly simplifies key distribution. We formalize the notion of using this system to *blindly* extract keys, and show how the primitive can be used to construct efficient fully-simulatable OT protocols (previous OT constructions are either inefficient, are proven according to unrealistic security definitions, or require strong complexity assumptions).

2. A third OT protocol that is secure in the strong Universal Composability (UC) model of Canetti [Can01]. Not only does this protocol meet a strong definition of security, but it can be generically composed with any other UC-secure protocol (including itself). This is important in the case of databases where many users may concurrently access the same database. To our knowledge, this is the first efficient adaptive OT construction to meet this definition.

3. A technique for providing strong and *history-dependent* access control for an oblivious database. In this model, the user is prevented from requesting items that are not permitted by her policy, while the database operator learns nothing more about the content of her requests. Our constructions are based on a new form of stateful anonymous credentials. Finally, we show how these technologies can be combined to produce a practical oblivious database.

The contributions of this work are both theoretical and practical. In particular, we believe that the notion of constructing Oblivious Transfer from Identity-Based Encryption may ultimately help to expand our understanding of both primitives. Simultaneously, the constructions we propose achieve high efficiency under strong security definitions. Ultimately, we believe that this is the first work to thoroughly consider the practical tradeoffs of constructing privacy-preserving databases.

Thesis Readers:

ABSTRACT

Susan Hohenberger
Assistant Professor & Advisor
Department of Computer Science
Johns Hopkins University

Gerald Masson
Professor
Department of Computer Science
Johns Hopkins University

abhi shelat
Assistant Professor
Department of Computer Science
University of Virginia

Giuseppe Ateniese
Associate Professor
Department of Computer Science
Johns Hopkins University

Acknowledgements

A great many people have supported me in the writing of this thesis.

This work would not have been possible without the support of my advisor and friend Susan Hohenberger, who — despite being a busy new faculty member — made time to share her wealth of knowledge with me. Similarly, I would never have reached this point without the encouragement of Avi Rubin, who brought me to Johns Hopkins in the first place and has provided me with invaluable advice ever since.

I owe a great debt to the faculty, students and visitors with whom I had the pleasure of working during my time here. In particular I thank Giuseppe Ateniese for bringing me to the field of cryptography; Fabian Monrose for giving a solid grounding in computer security; Breno de Medeiros for helping me make sense out of it all; Scott Coull for his valuable collaboration on the final portion of this thesis; Anna Lisa Ferrara; Kevin Fu; Lucas Ballard; Seny Kamara; Reza Curtmola; Zachary Peterson; Josh Mason; Sujata Garera; Darren Davis; and in particular, my good friend Sam Small. I also extend thanks to my partners Stephen Bono and Adam Stubblefield at Independent Security Evaluators, for putting up with my unavailability while I was off writing this thesis.

I also extend my sincere thanks to Gerald Masson and abhi shelat for serving on my committee, and to Brent Waters for keeping me honest. On a practical note, I thank the National Science Foundation, who graciously funded this research under grant CNS-0716142, and Microsoft Research for their support under Susan Hohenberger's New Faculty Fellowship.

Without the encouragement of my family I would never have begun this project, and certainly would not have finished it. I thank my parents for giving me the gentle prodding I needed to pursue this degree, and — more importantly — for having the foresight to install a DECWriter II at our house when I was six years old, thus ensuring me a bright future in the field of Computer Science. I thank my sister for putting up with me, and for showing me all the creative things I could do with it.

Finally, and most importantly of all: I owe an undying debt of gratitude to my beloved wife Melissa, whose faith, love and understanding have sustained me through everything. She has given me everything I have, and made me everything I am. It is my greatest hope that I will be able to give to her even a small fraction of the happiness that she has given me.

ACKNOWLEDGEMENTS

December 26, 2008
Baltimore, Maryland

Contents

List of Figures

Chapter 1

Introduction

In 2006 America Online's research division leaked the web search histories of more than 600,000 of their customers [Gon06]. Although the data had been stripped of customer names and identifying information, it quickly became apparent that these protections were insufficient. Within several days, news organizations had discovered deeply private information about these individuals' identities and interests [BJ06, Zel06]— information derived solely from the customer's query patterns.

Much of the work on database security has focused on securing data *within* the database. However, in many applications it is the query information that is particularly sensitive. For example, a doctor who queries a medical records database may inadvertently reveal information about his patients (*e.g.,* queries by an HIV specialist could indicate a possible infection). In the hands of an insurer or employer, this information might severely harm a patient's interests. Similarly, many publicly-searchable patent databases receive a wealth of information from their users. When combined with the searcher's identity (or employer), these query patterns can reveal sensitive corporate intelligence.

Unfortunately, we as users are increasingly dependent on the goodwill and discretion of third parties to guard this information. Indeed, this dependence is likely to increase as the industry moves from a traditional model where companies operate their own databases, to a model where databases operations are outsourced to third parties. This approach — loosely referred to as "cloud computing" — is being heavily promoted by market leaders such as Google and Microsoft [Mar07, Bak07].

We stipulate that naïve solutions to this problem exist *when the database operator does not care how the how the data is accessed*. For example, a database operator can simply publish the entire database to its users (via an efficient distribution mechanism such as BitTorrent). In practice, however, it is quite common for databases to contain sensitive records that should only be accessed by specific users, often under particular conditions. These database implementations must include an access control mechanism to limit which records should be made available to specific users. Traditional databases enforce such con-

trols through knowledge of the user's identity and items requested. Enforcing an access control policy in a private database seems almost a contradiction, since the database operator must (by definition) be kept in the dark about the user's identity and request.

COMMON TECHNIQUES. Some researchers have proposed to hide users' query patterns by associating users with *pseudonyms* [CH05]. This approach adds a layer of indirection between user identities and actual database requests. Provided that the pseudonym-to-identity mapping is managed by a trusted third party, this approach can protect the user's privacy while still allowing the database operator to enforce a meaningful access control policy on pseudonymous users. Unfortunately, these solutions can leak some information: even if the mapping is secret, a malicious party can still link queries from the same user (pseudonym). The data gained can be substantial: for example, it should be possible to determine the specialization of a pseudonymous doctor by examining the patient files they access. Since it is difficult to determine *a priori* how much information might be leaked in a specific application, this approach should be viewed with caution.

An alternative solution is to enforce honest behavior through legislation. Prominent examples include the US Health Insurance Portability and Accountability Act (HIPAA) which mandates policies for handling medical data [Uni96] and the broader European Union Directive on Data Protection [Par95]. Unfortunately, legislative approaches suffer from jurisdictional challenges and are often unable to keep pace with changing technology. Database systems may be compromised by outsiders, who install malicious software to monitor private user transactions [Bos08]. Furthermore, database operators may disobey the law without detection— either maliciously or due to negligence by system administrators (see *e.g.,* [Hru08, CNN05]).

OUR APPROACH. We believe that the approaches above are insufficient to address privacy concerns for many sensitive database applications. Given the rapidly changing nature of the technology, we believe the problem requires a *technical* approach that provides a strong guarantees of user privacy, and does not depend on the vagaries of a specific application. Simultaneously we will show how such a database may still enforce sophisticated (and history-dependent) access control policies limiting which records each user may obtain.

While query privacy has been discussed in the literature (*e.g.,* [CKGS98, AIR01, NP99b, CNs07]), all of the previous works lack at least one of the requirements that we believe are necessary: (1) privacy for user identities, (2) privacy for queries, and (3) strong access control for the database operator. Worse, many previous techniques relied on strong cryptographic assumptions and/or were quite inefficient. In this work we will propose several cryptographic building blocks that solve these problems, and show how they can be combined into privacy-preserving databases.

Let us now describe these building blocks:

Adaptive Oblivious Transfer. In an Oblivious Transfer (OT) protocol, as proposed by Rabin [Rab81] and generalized by Even, Goldreich and Lempel [EGL82] and Brassard, Crépeau and Robert [BCR86], a Sender with a collection of messages interacts with a

Receiver such that the Receiver obtains only a subset of the messages, and no information about the rest of the database. For its part, the Sender learns nothing about *which* messages the Receiver requested. The generalized case of k-out-of-N OT [BCR86] — in which the Receiver obtains k messages from an N message collection — seems an obvious candidate for the construction of oblivious databases.

However, many existing k-out-of-N protocols have limitations. To be viable for use in database applications, an OT protocol must permit access by many users to the same database, and must be *adaptive — i.e.,* must permit the user to form its queries based on previous items received [NP99b]. Additionally, it is desirable that the protocol admit security proofs under reasonable definitions and complexity assumptions. Finally, the protocol must be efficient in terms of communication and computation.

Unfortunately, few existing adaptive OT protocols meet these requirements. Many require large number of rounds or costs that render them impractical. Among more efficient constructions, the vast majority have been analyzed under a weak security definition known as "half-simulation". In 1999, Naor and Pinkas [NP99a] showed that protocols secure under this definition admit practical attacks that may compromise Receiver security. The focus of our investigation, therefore, will be on developing protocols secure under strong definitions such as "full-simulation" security — in which the security of the protocol is evaluated with respect to an ideal world where a trusted party conducts all operations — and (even stronger) Canetti's *Universal Composability* framework [Can01] which provides similar guarantees with a strong property that protocols can be generically composed.

Oblivious Access Controls from Anonymous Credentials. Anonymous Credentials, first proposed by Chaum [Cha85], allow users to prove various properties about themselves (such as membership in an organization), without revealing their identity. These primitives make an excellent building block for an anonymous access control system, and are in fact being used to control access to computer systems [CH02]. We observe that this building block can be extended to encode both the user's identity, as well as a highly-complex and *history-dependent* access control policy describing which items a user may access. More importantly, we show that these credentials can be efficiently integrated into the fabric of our Oblivious Transfer protocols, allowing database operators to enforce flexible access control policies without learning the identity of the user or the items s/he requests.

OUR CONTRIBUTIONS. We will now detail the specific technical contributions addressed by this work.

1. **Blind Identity Based Encryption.** First, we introduce a building block, which is of independent interest. In identity-based encryption (IBE) [Sha84], there is an *extraction* protocol where a user submits an identity string to a master authority who then returns the corresponding decryption key for that identity. We formalize the notion of *blindly* executing this protocol, in a strong sense; where the authority does not learn the identity nor can she cause failures dependent on the identity, and the user learns nothing beyond the normal extraction protocol. In §4.3, we describe efficient *blind extraction* protocols satisfying this definition for several well-known IBE schemes.

2. **Fully Simulatable OT.** In 2007, Camenisch, Neven and shelat [CNs07] proposed an efficient *fully-simulatable* adaptive OT protocol secure in the standard model. However, their protocol depends on a new strong decisional assumptions in bilinear groups. We present new protocols that support efficient, fully-simulatable Oblivious Transfer that can be realized under any blind IBE scheme. Using the blind IBE schemes presented in §4.3, our protocols can be realized under relatively weaker computational assumptions than previous work.

3. **Universally-Composable Adaptive OT**. Building on our previous result, we construct two additional protocols that are, to our knowledge, the first practical, adaptive OT protocol secure in the Universal Composability model of Canetti [Can01]. Our protocol requires substantially fewer communication rounds that the previous protocols. Additionally, we show that this protocol permits many anonymous receivers to interact with a single database operator. Our construction is secure under bilinear assumptions in the standard model.

4. **Stateful Anonymous Credentials for Oblivious Access Control.** We then present a efficient construction for anonymously and privately enforcing an arbitrary access control policy on the contents of an oblivious database. Our construction permits the enforcement of complex and *history-dependent* access control policies across a large group of users, without compromising the identity of the requesting user or the content requested. While this access control mechanism has many applications, we focus on integrating it with the Oblivious Transfer schemes presented in this work.

   Previous efforts in this area have addressed only a portion of this problem. In 2001, Aiello, Ishai and Reingold [AIR01] proposed a protocol for "priced oblivious transfer", in which users are permitted to "purchase" database items using a descending balance. Unfortunately, their approach — which relied on a server-side counter — cannot provide user anonymity and is fundamentally vulnerable to selective-failure attacks. Furthermore, it provides only the most limited form of access control. In this work, we will develop a framework and compiler for enforcing arbitrary, programmable access control policies on an oblivious database. Our techniques make use of "signatures with efficient protocols" due to Camenisch and Lysyanskaya [CL04].

OUTLINE OF THIS WORK. Let us now describe the format of the remaining sections.

Chapter 2 provides a brief overview of Oblivious Transfer (OT) from its conception to the development of efficient *adaptive* k-out-of-N schemes. This chapter also includes an overview of the modern simulation-based security definitions for Oblivious Transfer, and some intuition for how OT can be adapted to a multi-party setting.

Chapter 3 details some of the notation and cryptographic preliminaries for the constructions we will present in later chapters. This chapter includes formal definitions of

the bilinear groups in which our schemes are set, as well as the complexity-theoretic assumptions that we will use to prove their security.

Chapter 4 details two new Oblivious Transfer protocols constructed from "Blind" Identity Based Encryption and provides several compatible instantiations of the latter primitive. Specifically:

1. In §4.2 we describe generic constructions for (1) *non-adaptive* Oblivious Transfer of k messages out of an N message collection OT_k^N, and argue that this approach is secure in the standard model provided that an appropriate IBE scheme is available. We then describe (2) a modification to this protocol that leads to *adaptive* $\mathsf{OT}_{k\times 1}^N$ in the random oracle model .
2. In §4.3 we present several concrete instantiations of Blind IBE schemes based on the Boneh-Boyen selective-ID IBE [BB04a], and the Waters IBE (with optimizations due to Naccache and Chaterjee-Sarkar [Nac05, CS05]). We also present a protocol for an *anonymous* IBE based on the Boyen-Waters scheme [BW06]. When the OT schemes of §4.2 are instantiated with any of these Blind IBE schemes, the resulting OT protocol will be secure under the Decisional Bilinear Diffie-Hellman assumption.

Chapter 5 details a construction for a k-out-of-N OT ($\mathsf{OT}_{k\times 1}^N$) that is secure in Canetti's Universal Composability (UC) framework [Can01]. This construction makes use of the efficient non-adaptive Zero-Knowledge and Witness-Indistinguishable proofs due to Groth and Sahai [GS08], which allow us to achieve a construction that is optimal in terms of communications rounds and may also be concurrently composed.

Chapter 6 describes an approach to achieving strong access controls for an oblivious database via a new concept which we refer to as "Stateful Anonymous Credentials". We describe this new primitive in isolation and then show how it can be attached to either the $\mathsf{OT}_{k\times 1}^N$ of §4.2 *or* an efficient $\mathsf{OT}_{k\times 1}^N$ due to Camenisch, Neven and shelat [CNs07].

Chapter 7 concludes by presenting several remaining open problems in this area.

The Appendices to this work contain several additional contributions which are referenced throughout this work, including an alternate UC-secure construction for adaptive Oblivious Transfer, as well as clarifying notes and security proofs for the contributions within the main body.

PREVIOUS PUBLICATIONS. Portions of this work have previously been published in other venues. Much of Chapter 4 appeared in the Proceedings of ASIACRYPT 2007 [GH08b]. Similarly, Chapter 5 contains material that was originally published in the Proceedings of ASIACRYPT 2008 [GH08c]. Chapter 6 is based on work that will appear in the Proceedings of PKC 2009 [CGH09]. We will provide a detailed citation at the start of each chapter.

Chapter 2

Oblivious Transfer

In the 1970s, a physicist named Steven Weisner proposed a technique for transmitting two messages such that at most one is received, but with a paradoxical feature: the sender does not learn *which* of the two messages arrived [Wei83]. Weisner's technique relied on the quantum properties of individual photons transmitted from the sender to a polarizing filter at the receiver's side. Though theoretically interesting, Weisner's approach was incompatible with existing communications networks. (However, Weisner's concepts became the foundation of the field of *quantum cryptography*, and several compatible photon transmission networks have since been built [sec08].)

Years later, a related concept was independently discovered by Michael Rabin [Rab81], who showed how it could be achieved using cryptographic techniques over standard communications networks. Rabin's protocol allowed a Sender to transmit a single message such that it would be received with probability exactly $1/2$. He named this protocol "Oblivious Transfer" (OT). It was later shown by Crépeau [Cré87] and Even, Goldreich and Lempel [EGL82] that $\mathsf{OT}_{\frac{1}{2}}$ implies more powerful variants, including a 1-out-of-2 protocol (OT_1^2) similar to Weisner's, in which a Receiver obtains one of two possible messages (where the message choice is either random, or explicitly made by the receiver). Brassard, Crépeau and Robert [BCR86] further generalized this concept to k-out-of-N OT (OT_k^N), a two-party protocol in which a Sender with messages $M_1, \ldots, M_N$ and a Receiver with indices $\sigma_1, \ldots, \sigma_k \in [1, N]$ interact in such a way that at the end the Receiver obtains $M_{\sigma_1}, \ldots, M_{\sigma_k}$ without learning anything about the other messages. Simultaneously, the Sender does not learn anything about $\sigma_1, \ldots, \sigma_k$.

Oblivious transfer has a particular significance as OT_1^4 is a key building block for secure multi-party computation [Yao86, GMW87, Kil88]. In fact, it has been shown that OT_1^4 is "complete" for that primitive, meaning that secure multi-party computation can be constructed in a black-box manner given *only* an appropriate Oblivious Transfer protocol [Kil88].

OT_k^N is a useful and interesting tool in its own right for constructing oblivious

Protocol	Rounds	Comm.	Assumption
Half Simulation:			
Rabin81 OT_2^1 [Rab81]	1 1/2	O(1)	Factoring
Kalai05 OT_1^2 [Kal05, HK07]	1 1/2	O(1)	Smooth Projective Hashing
BCR86 OT_k^N [BCR86]	O(1)	O(κN)	Quadratic Residuosity (QR)
NP99 $\mathsf{OT}_{k\times 1}^N$ [NP99b]	$\ell k \log N + 1/2$	–	Sum Consistent Synthesizers + ℓ-round OT_1^2
CT05 $\mathsf{OT}_{k\times 1}^N$ [CT05]	$O(k) + 1/2$	$O(N)$	Decisional DH (in ROM)
Full Simulation OT_k^N*:*			
This work OT_k^N §4.2.1	$O(k)$	$O(N)$	Decisional Bilinear DH
Full Simulation $\mathsf{OT}_{k\times 1}^N$*:*			
CNS07 [CNs07]	$4k + 1/2$	$O(N)$	y-Power Decisional DH + q-Strong DH
CNS07 [CNs07]	$O(k)$	$O(N)$	Unique blind signature (in ROM)
This work $\mathsf{OT}_{k\times 1}^N$ §4.2.2	$O(k)$	$O(N)$	Decisional Bilinear DH (in ROM)
UC OT_1^2*:*			
PVW08 OT_1^2 [PVW08]	1	O(N)	DDH/QR/Lattice assumptions ($\mathcal{F}_{CRS}$-hybrid)
DNO08 OT_1^2 [DNO08]	O(1)	O(N)	DLIN ($\mathcal{F}_{KR}$-hybrid)
UC $\mathsf{OT}_{k\times 1}^N$*:*			
This work $\mathsf{OT}_{k\times 1}^N$ §5.2	$k + 1/2$	$O(N)$	SXDH + DLIN + q-HLRSW ($\mathcal{F}_{CRS}$-hybrid)

Figure 2.1: A survey of adaptive and non-adaptive Oblivious Transfer protocols.

databases. Along these lines, Naor and Pinkas pointed out that existing OT_k^N protocols may be insufficient for database applications, since they do not permit an *adaptive* query pattern, where the sender may obtain $M_{\sigma_{i-1}}$ before deciding on σ_i [NP99b]. They proposed protocols for adaptive OT ($\mathsf{OT}_{k\times 1}^N$), using various components including an OT_1^2 scheme. While this and other works demonstrated the existence of transformations from OT_2^1 to the generalized forms OT_1^N and $\mathsf{OT}_{k\times 1}^N$, such *black-box* constructions are quite inefficient.

Developing efficient efficient adaptive protocols appears to be a more difficult and involved process than the non-adaptive protocols. Indeed, even finding the right security definition has proven challenging. Historically, many OT constructions were analyzed under a "half-simulation" definition, where the Sender and Receiver's security are described by a combination of simulation and game-based definitions. Naor and Pinkas [NP99b] showed that schemes analyzed under this definition may admit practical attacks on the Receiver's privacy.

2.1 Prior Work and Recent Developments

The definition of security for oblivious transfer has been evolving. Informally, security is defined with respect to an ideal-world experiment in which the Sender and Receiver exchange messages via a trusted party. An OT protocol is secure if, for every real-world cheating Sender (resp., Receiver) we can describe an ideal-world counterpart who gains as much information from the ideal-world interaction as from the real protocol. Bellare and Micali [BM89] presented the first practical OT_1^2 protocol to satisfy this intuition in the honest-but-curious model. This was followed by practical OT protocols due to Naor and Pinkas [NP99a, NP99b, NP01] in the half-simulation model where the simulation-based model (described above) is used only to show Sender security and Receiver security is

defined by a simpler game-based definition. Almost all efficient OT protocols are proven secure with respect to the half-simulation model, *e.g.,* [NP99b, NP99a, NP01, DHRS04, OK04, Kal05, CT05].

Unfortunately, Naor and Pinkas demonstrated that this model permits *selective-failure* attacks, in which a malicious Sender can induce transfer failures that are dependent on the message that the Receiver requests [NP99b]. In these attacks, the Sender structures its messages such that for certain Receiver inputs, the protocol will always fail. In practice, this can lead to a condition where an unsuspecting Receiver might attempt to re-initiate the protocol, thus leaking valuable information about its selection. These attacks are possible because the half-simulation definition does not enforce correctness of the Sender's inputs. (In a half-simulation security proof, the Sender is free to transmit any messages it wants, provided that it learns no information about the Receiver's selections at the conclusion of the protocol.) While this may seem a subtle distinction, many protocols with half-simulation security proofs seem quite difficult to adapt to the full-simulation definition. This can be problematic, as OT is a fundamental building block for many other protocols, which will often inherit the limitations of the underlying OT.

Efficient Adaptive OT Protocols. Recently, Camenisch, Neven, and shelat [CNs07] proposed practical $\mathsf{OT}^N_{k\times 1}$ protocols that are secure in a "full-simulation" model, where the security of both the Sender and Receiver are simulation-based. These simulatable OT protocols are particularly nice because they can be used to construct other cryptographic protocols in a simulatable fashion. More specifically, Camenisch *et al.* [CNs07] provide two distinct results. First, they show how to efficiently construct $\mathsf{OT}^N_{k\times 1}$ generically from any unique blind signature scheme in the random oracle model. The two known efficient unique blind signature schemes due to Chaum [Cha82] and Boldyreva [Bol03] both require *interactive* complexity assumptions: one-more-inversion RSA and chosen-target CDH, respectively. (Interestingly, when instantiated with Chaum signatures, this construction coincides with a prior one of Ogata and Kurosawa [OK04] that was analyzed in the half-simulation model.) Second, they provide a clever $\mathsf{OT}^N_{k\times 1}$ construction in the standard model based on dynamic complexity assumptions, namely the q-Power Decisional Diffie-Hellman (*i.e.*, in a bilinear setting $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, given $(g, g^x, g^{x^2}, \ldots, g^{x^q}, H)$ where $g \leftarrow \mathbb{G}$ and $H \leftarrow \mathbb{G}_T$, distinguish $(H^x, H^{x^2}, \ldots, H^{x^q})$ from random values) and q-Strong Diffie-Hellman (q-SDH) assumptions. (Unfortunately, Cheon showed that q-SDH requires larger than commonly used security parameters [Che06]). These dynamic (including interactive) assumptions seem significantly stronger than those, such as DDH and quadratic residuosity, used to construct efficient $\mathsf{OT}^N_{k\times 1}$ schemes in the half-simulation model [NP99b, CT05].

Thus, while quite elegant, the protocols of Camenisch *et al.* have two primary drawbacks that motivate further research in this area. Specifically:

1. The Camenisch *et al.* protocols depend for their security on the unforgeability of a *unique* blind signature in the Random Oracle model (the two known constructions of which require *interactive* complexity assumptions), or alternatively on a new strong

q-based decisional assumption (q-PDDH) in the Standard Model. It is desirable to consider protocols secure under weaker assumptions.

2. The Standard Model protocol makes use of adversarial rewinding in its security proof, and may not be secure under concurrent composition. We would like to consider protocols secure under stronger definitions such as Canetti's Universal Composability model [Can01].

We remark that our focus is on *adaptive* OT protocols, since these are required for the construction of Oblivious Databases. However, three recent works have also considered full-simulation and UC-secure OT protocols in the *non-adaptive* setting. Lindell [Lin08] recently proposed several efficient and fully-simulatable OT_1^2 protocols secure under weaker assumptions than those used in this work, *e.g.,* DDH and Quadratic Residuosity. Peikert, Vaikuntanathan and Waters [PVW08] recently proposed a framework for constructing (non-adaptive) OT_1^2 using "messy keys", and showed how to realize these in the Universal Composability (UC) model of Canetti [Can01] under DDH, Quadratic Residuosity, or lattice assumptions. Similarly, Damgrd, Nielsen and Orlandi proposed an alternative OT_1^2 using an alternative setup assumption and Groth-Sahai proofs.

2.2 Formal Definitions for Fully-Simulatable OT

We will now provide formal definitions for non-adaptive OT_k^N and adaptive $\mathsf{OT}_{k\times 1}^N$. To maintain consistency with earlier work, we generalize the definitions of Camenisch *et al.* [CNs07]. While that work focuses solely on *adaptive* OT, our definitions also consider the non-adaptive version of the primitive.

Definition 2.2.1 (k-out-of-N Oblivious Transfer (OT_k^N, $\mathsf{OT}_{k\times 1}^N$)) An oblivious transfer scheme is a tuple of algorithms $(\mathsf{S_I}, \mathsf{R_I}, \mathsf{S_T}, \mathsf{R_T})$. During the initialization phase, the Sender and the Receiver run an interactive protocol, where the Sender runs $\mathsf{S_I}(M_1, \ldots, M_N)$ to obtain state value S_0, and the Receiver runs $\mathsf{R_I}()$ to obtain state value R_0. Next, during the transfer phase, the Sender and Receiver interactively execute $\mathsf{S_T}, \mathsf{R_T}$, respectively, k times as described below.

Adaptive OT. In the adaptive $\mathsf{OT}_{k\times 1}^N$ case, for $1 \leq i \leq k$, the i^{th} transfer proceeds as follows: the Sender runs $\mathsf{S_T}(S_{i-1})$ to obtain state value S_i, and the Receiver runs $\mathsf{R_T}(R_{i-1}, \sigma_i)$ where $1 \leq \sigma_i \leq N$ is the index of the message to be received. The receiver obtains state information R_i and the message M'_{σ_i} or $\perp$ indicating failure.

Non-adaptive OT. In the non-adaptive OT_k^N case the parties execute the protocol as in the previous case; however, for each round $i < k$ the algorithm $\mathsf{R_T}(R_{i-1}, \sigma_i)$ **does not** output a message. At the end of the the k^{th} transfer $\mathsf{R_T}(R_{k-1}, \sigma_k)$ outputs the full collection $(M'_{\sigma_1}, \ldots, M'_{\sigma_k})$ where for $j = 1, \ldots, N$ each M'_{σ_j} is a valid message or the symbol $\perp$

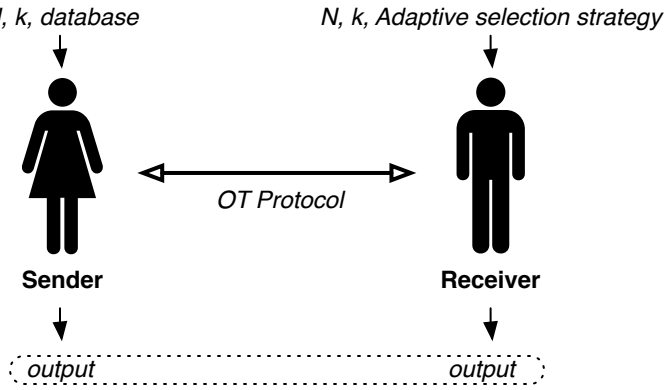

Figure 2.2: "Real world" experiment. The Sender is given N, k and $M_1, \ldots, M_N$. The Receiver is given a selection strategy Σ that dictates the next message it should request (based on previous messages received). The two interact using the OT protocol. The output of the experiment is the concatenation of both parties' outputs.

indicating protocol failure. (In a non-adaptive scheme, the k transfers do not necessarily require a corresponding number of communication rounds.)

Security. We now address the security definition for Oblivious Transfer. Informally, we will consider two experiments. In the "Real experiment" (figure 2.2), the Sender is given an N-item database and the Receiver a (possibly adaptive) strategy for obtaining items from this database. The pair will then interact using a cryptographic OT protocol such that the Receiver obtains up to k items. The output of this experiment is whatever output the Sender and Receiver produce at the termination of the protocol.

In the Ideal experiment (figure 2.2), the Sender and Receiver are given the same inputs as in the previous experiment. However in this hypothetical world, the two parties interact via a trusted party that honestly adheres to the following protocol: (1) it receives a set of messages $M_1^*, \ldots, M_N^*$ from the Sender (these may not be the same messages input to the experiment) along with a bitmap $b_1, \ldots, b_k$ indicating which transactions should succeed or fail, (2) it receives requests for indices $\sigma_1^*, \ldots, \sigma_k^*$ from the Receiver (either one at a time, or as a group), and (3) for $i \in [1, k]$ it responds to each request by returning $M_{\sigma_i^*}^*$ (if $b_i = 1$) or a failure notice $\perp$ (if $b_i = 0$).

Informally, we say that an OT is full-simulation secure if no (malicious) Sender or Receiver can succeed with significantly higher probability against the Real world experiment than an adversary playing the same position in the Ideal world experiment. This guarantee is powerful, since the Ideal world experiment clearly protects the interests of both parties in the protocol. However, while intuitive, this definition is tricky to formalize since we must define what we mean by an adversary succeeding in either world.

To solve this problem we will make use of the *simulation* paradigm. Specifically, for every (adversarial) Sender $\hat{\mathsf{S}}$ (resp. Receiver $\hat{\mathsf{R}}$) in the Real world, we will show that there

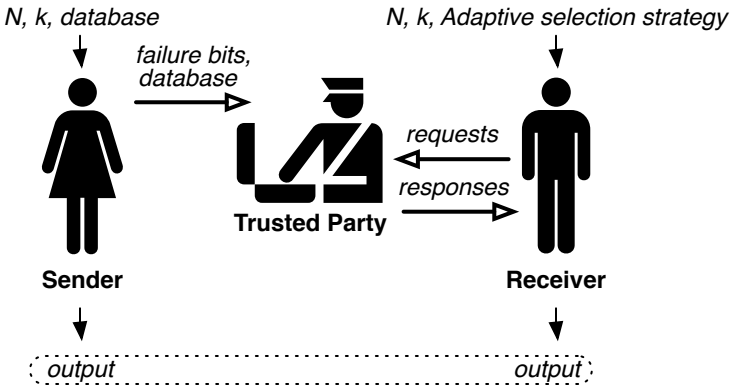

Figure 2.3: "Ideal world" experiment. The Sender and Receiver are given the same inputs as in the Real world experiment. However, the two interact with a trusted party to exchange messages. The output of the experiment is the sum of both parties' outputs.

must exist an corresponding adversarial $\hat{\mathsf{S}}'$ (resp. $\hat{\mathsf{R}}'$) in the Ideal world, such that the output of the Ideal experiment conducted between this ideal adversary and an "honest" counterparty is *computationally indistinguishable* the output of the Real experiment conducted between the real adversary and its honest counterparty.

We now formalize these definitions.

Definition 2.2.2 (Full Simulation Security.) Full-simulation security for OT_k^N, $\mathsf{OT}_{k\times 1}^N$ is defined according to the following experiments. Note that, as in [CNs07] we do not explicitly specify auxiliary input to the parties, but note that this information can be provided in order to achieve sequential composition.

Real experiment. In experiment $\mathbf{Real}_{\hat{\mathsf{S}},\hat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma)$ the possibly cheating sender $\hat{\mathsf{S}}$ is given messages $(M_1, \ldots, M_N)$ as input and interacts with possibly cheating receiver $\hat{\mathsf{R}}(\Sigma)$, where Σ is a selection algorithm that on input the full collection of messages thus far received, outputs the index σ_i of the next message to be queried. At the beginning of the experiment, both $\hat{\mathsf{S}}$ and $\hat{\mathsf{R}}$ output initial states (S_0, R_0). In the adaptive case, for $1 \leq i \leq k$ the sender computes $S_i \leftarrow \hat{\mathsf{S}}(S_{i-1})$, and the receiver computes $(R_i, M_i') \leftarrow \hat{\mathsf{R}}(R_{i-1})$. In the non-adaptive case, the Receiver obtains no messages until the k^{th} round, and therefore the selection strategy Σ must be non-adaptive. At the end of the k^{th} transfer the output of the experiment is (S_k, R_k).

We will define the honest Sender algorithm $\mathbf{S}$ as one that runs $\mathsf{S}_\mathsf{I}(M_1, \ldots, M_N)$ in the first phase, during each transfer runs $\mathsf{S}_\mathsf{T}()$ and outputs $S_k = \varepsilon$ as its final output. The honest Receiver $\mathbf{R}$ runs R_I in the first phase, and $\mathsf{R}_\mathsf{T}(R_{i-1}, \sigma_i)$ at the i^{th} transfer, and outputs $R_k = (M'_{\sigma_1}, \ldots, M'_{\sigma_k})$ as its final output.

Ideal experiment. In experiment $\mathbf{Ideal}_{\hat{\mathsf{S}}',\hat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma)$ the possibly cheating sender algorithm $\hat{\mathsf{S}}'$ generates messages $(M_1^*, \ldots, M_N^*)$ and transmits them to a trusted

party T. In the i^{th} round $\hat{\mathsf{S}}'$ sends a bit b_i to T; the possibly cheating receiver $\hat{\mathsf{R}}'(\Sigma)$ transmits σ_i^* to T. In the adaptive case, if $b_i = 1$ and $\sigma_i^* \in \{1, \ldots, N\}$ then T hands $M^*_{\sigma_i{}^*}$ to $\hat{\mathsf{R}}'$. If $b_i = 0$ then T hands $\perp$ to $\hat{\mathsf{R}}'$. Note that in the non-adaptive case, T caches its responses to $\hat{\mathsf{R}}'$ and delivers the full collection at the conclusion of the k^{th} round. After the k^{th} transfer the output of the experiment is (S_k, R_k).

We will define the honest Sender algorithm S' as one that transmits $\mathsf{S}_{\mathsf{I}}(M_1, \ldots, M_N)$ to T in the first phase, and outputs $S_k = \varepsilon$ as its final output. The honest Receiver $\mathbf{R}$ sends σ_i to T at the i^{th} transfer, and outputs $R_k = (M^*_{\sigma_1}, \ldots, M^*_{\sigma_k})$ as its final output.

Let $\ell(\cdot)$ be a polynomially-bounded function. We now define Sender and Receiver security in terms of the experiments above.

Sender Security. An $\mathsf{OT}^N_{k\times 1}$ provides Sender security if for every real-world p.p.t. receiver $\hat{\mathsf{R}}$ there exists a p.p.t. ideal-world receiver $\hat{\mathsf{R}}'$ such that $\forall N = \ell(\kappa)$, $k \in [1, N]$, $(M_1, \ldots, M_N)$, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma) \overset{c}{\approx} \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma)$$

Receiver Security. $\mathsf{OT}^N_{k\times 1}$ provides Receiver security if for every real-world p.p.t. sender $\hat{\mathsf{S}}$ there exists a p.p.t. ideal-world sender $\hat{\mathsf{S}}'$ such that $\forall N = \ell(\kappa)$, $k \in [1, N]$, $(M_1, \ldots, M_N)$, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\hat{\mathsf{S}},\mathbf{R}}(N, k, M_1, \ldots, M_N, \Sigma) \overset{c}{\approx} \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \Sigma)$$

2.3 Universally Composable Security

A stronger notion of security is the Universal Composability framework [Can01] allows for the design of concurrent and composable cryptographic protocols, which are important properties in any practical deployment of an oblivious database. Canetti and Fischlin showed that OT cannot be UC-realized without trusted setup assumptions such as the existence of a Common Reference String (CRS) [CF01]. This is formally referred to as the $\mathcal{F}_{CRS}$-hybrid model, and is assumed by the constructions of Peikert *et al.* [PVW08] as well as those in this work.

As in [PVW08], we will work in the standard UC framework with static corruptions, where all parties are modeled as p.p.t. interactive Turing machines. Security of protocols is defined by comparing the protocol execution to an *ideal process* for carrying out the desired task. More formally, there is an *environment* $\mathcal{Z}$ whose task is to distinguish between two worlds: ideal and real. In the ideal world, "dummy parties" (some of whom may be corrupted by the *ideal adversary* $\mathcal{S}$) interact with an *ideal functionality* $\mathcal{F}$. In the real world, parties (some of whom may be corrupted by the *real world adversary* $\mathcal{A}$) interact with each other according to some protocol π. We refer to Canetti [Can01, Can08] for a fuller description, as well as a definition of the ideal world ensemble $\mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}}$ and the

Functionality $\mathcal{F}_{CRS}^{\mathcal{D},P}$

Upon receiving input $(\mathsf{sid}, \mathsf{crs})$ from party P, first verify that $p \in \mathcal{P}$; else ignore the input. If there is no value r recorded, then choose and record $r \leftarrow D$. Finally send output $(\mathsf{sid}, \mathsf{crs}, r)$ to P.

Figure 2.4: Ideal functionality for the common reference string [Can08].

real world ensemble $\mathsf{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}$. We use the established notion of a protocol π *securely realizing* an ideal functionality $\mathcal{F}$ as:

Definition 2.3.1 *Let $\mathcal{F}$ be a functionality. A protocol π* UC-realizes *$\mathcal{F}$ if for any adversary $\mathcal{A}$, there exists a simulator $\mathcal{S}$ such that for all environments $\mathcal{Z}$,*

$$\mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

Canetti and Fischlin showed that OT cannot be UC-realized without a trusted setup assumption [CF01]. Thus, as in [CLOS02, PVW08], we assume the existence of an honestly-generated Common Reference String (crs), and work in the so-called $\mathcal{F}_{CRS}$-hybrid model. The functionality is parameterized by a distribution D and a set $\mathcal{P}$ of recipients. For our purposes, $\mathcal{P}$ will include the OT Sender and Receiver only. Here the environment learns about the reference string from the adversary, and thus the simulator can set up a string with "trapdoor information", etc.

Figure 2.4 describes the $\mathcal{F}_{CRS}$ functionality and Figure 2.5 describes the $\mathcal{F}_{OT}^{N\times 1}$ functionality.

We briefly mention that there are techniques for designing and analyzing multiple OT protocols which use a single reference string; i.e., a multi-session extension. One might worry that if multiple protocols now share some joint state, then they can no longer be analyzed separately and then composed later. Fortunately, this is addressed by *universal composition with joint state* (JUC) [CR03] and could be done in our case. A second issue with sharing the reference string is that we make no guarantee about the security of protocols which use the same reference string in ways other than those specified by the OT protocol, and here we explicitly assume that the crs is only available to certain parties. This is at odds with the notion that the crs is a "global" entity, however, there are strong impossibility results for UC-realizing OT in a setting where the crs is available to everyone (including the environment) and can no longer be crafted by the simulator. There are models, such as the *augmented CRS* functionality $\mathcal{F}_{\mathrm{ACRS}}$ [CDPW07], which overcome these impossibility results, but we do not explore these advanced UC issues with respect to our OT construction in this work.

Functionality $\mathcal{F}_{OT}^{N\times 1}$

$\mathcal{F}_{OT}^{N\times 1}$ proceeds as follows, parameterized with integers N, ℓ and running with an oblivious transfer Sender $\mathbf{S}$, a receiver $\mathbf{R}$ and an adversary $\mathcal{S}$.

- Upon receiving a message $(\mathsf{sid}, \mathsf{sender}, m_1, \ldots, m_N)$ from $\mathbf{S}$, where each $m_i \in \{0,1\}^\ell$, store $(m_1, \ldots, m_N)$.
- Upon receiving a message $(\mathsf{sid}, \mathsf{receiver}, \sigma)$ from $\mathbf{R}$, check if a $(\mathsf{sid}, \mathsf{sender}, \ldots)$ message was previously received. If no such message was received, send nothing to $\mathbf{R}$. Otherwise, send $(\mathsf{sid}, \mathsf{request})$ to $\mathbf{S}$ and receive the tuple $(\mathsf{sid}, b \in \{0,1\})$ in response. Pass (sid, b) to the adversary, and: If $b = 0$, send $(\mathsf{sid}, \perp)$ to $\mathbf{R}$. If $b = 1$, send (sid, m_σ) to $\mathbf{R}$.

Figure 2.5: Ideal functionality for adaptive Oblivious Transfer, based on the OT_1^2 definition from [CLOS02].

2.4 On Multiple Receivers

OT is traditionally described as a two-party protocol between one Sender and one Receiver. We will our main constructions in this setting. However, since we are motivated by the application of OT to database systems, we would also like to support applications where multiple users share a single database, *i.e.,* one Sender and multiple Receivers. Naively this can be accomplished by requiring the database to run separate OT protocol instances with each user. However, this approach can be quite inefficient, and moreover does not ensure *consistency* in the database viewed by individual Receivers. In Chapter 5 we address this by strengthening our security definition to include the additional requirement that all Receivers "view" the same database, i.e., the database owner cannot selectively alter the messages in the database when interacting with different receivers – on query σ from *any* receiver, he must return a value in $\{m_\sigma, \perp\}$.

Chapter 3

Cryptographic Preliminaries

THE next several chapters describe techniques for constructing privacy-preserving databases. Before we present these cryptographic protocols, we must first describe certain concepts and notation used in our presentation. Within this chapter, we include a description of the cryptographic setting in which we will base our protocols, as well as the complexity-theoretic assumptions that we will use in our security proofs.

3.1 Model and Notation

We will begin by describing the notation that will be used throughout this work. By *p.p.t* we will denote a probabilistic polynomial-time Turing machine.

SECURITY PARAMETER. Our cryptographic protocols make use of an adjustable security parameter κ. We will generally provide this in unary representation 1^κ.[1] This parameter may be passed explicitly, or can be implicitly incorporated into other parameters (*e.g.,* group parameters, public keys) that are provided as input.

POLYNOMIAL AND NEGLIGIBLE FUNCTIONS. Let $poly(\cdot)$ as a polynomial function. We define a *negligible* function $\nu(\cdot)$ such that for all polynomial functions $poly(\cdot)$ and sufficiently large n the value $\nu(n) < 1/poly(n)$.

COMPUTATIONAL INDISTINGUISHABILITY. Let $\{A_\kappa\}_{\kappa\in N}$ and $\{B_\kappa\}_{\kappa\in N}$ be ensembles of probability distributions where A_κ, B_κ are probability distributions over $\{0,1\}^{poly(\kappa)}$ for some polynomial function $poly(\cdot)$. We will express the computational indistinguishability of these distributions by $\{A_\kappa\}_{\kappa\in N} \stackrel{c}{\approx} \{B_\kappa\}_{\kappa\in N}$. Quoting the definition of Pass and shelat [Pas08], the ensembles $\{A_\kappa\}_{\kappa\in N}$ and $\{B_\kappa\}_{\kappa\in N}$ are *computationally indistinguishable* if for all polynomial-time adversaries D then for some negligible $\nu(\cdot)$ and $\forall\kappa \in N$:

[1] This string should be read as a κ-bit string consisting solely of 1 bits. It is included so that the running time of the cryptographic algorithm can be specified as a function of the input size (κ).

$$|\Pr[t \leftarrow A_\kappa, D(t) = 1] - \Pr[t \leftarrow B_\kappa, D(t)]| \leq \nu(\kappa)$$

ALGEBRAIC NOTATION. By $\mathbb{G} = \langle g \rangle$ we indicate that g is a generator of the cyclic group $\mathbb{G}$. For consistency of notation we will use multiplicative notation throughout this work, though we note that some candidate implementations require the use of additive groups.

Models of Computation. In our security proofs we will model all parties as non-uniform probabilistic polynomial-time turning machines. We will prove several of our constructions secure in the *standard model* of computation, in which we will assume only the hardness of certain complexity theoretic assumptions. However, some of our proofs will be set in the *random oracle model* which assumes the existence of idealized random functions [BR93]. Some recent works have demonstrated a strong separation between the two models: specifically, there exist certain cryptosystems which are secure in the random oracle model, but become insecure when the random oracle is instantiated with any deterministic function or function family, *e.g.,* [CGH04]. Thus, where possible we will emphasize proofs without random oracles.

3.2 Bilinear Groups

Many of the protocols in this work require prime-order groups supporting an efficient bilinear map. Candidate groups were brought to cryptographers' attention with the famous attack of Menezes, Vanstone and Okamoto [MVO91], and were first used in protocols by Joux and Nguyen [Jou00, JN01] for applications such as one-round tripartite key agreement. These groups have been used to construct a wide variety of cryptographic protocols, notably including Identity-Based Encryption [BF01, BB04a, Wat05].

We will now provide definitions for bilinear groups.

Definition. Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be multiplicative cyclic groups of prime order q, and let e be a function of the form $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. We say that e is a bilinear map if it satisfies the following requirements:

1. **Non-degeneracy.** If $\langle g \rangle = \mathbb{G}_1$ and $\langle \tilde{g} \rangle = \mathbb{G}_2$, then $\langle e(g, \tilde{g}) \rangle = \mathbb{G}_T$.
2. **Bilinearity.** If $g, \tilde{g}$ generate $\mathbb{G}_1, \mathbb{G}_2$ respectively, then for $a, b \in \mathbb{Z}_q$ it holds that $e(g^a, \tilde{g}^b) = e(g, \tilde{g})^{ab}$.
3. **Efficiency.** The mapping e is efficiently computable.

The description above is known as the *asymmetric* setting, and it closely describes the properties of all known instantiations. However, some of our protocols will require a *symmetric* bilinear map operating on a single group $\mathbb{G}$ and taking the form $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. In practice, the symmetric setting may be constructed from the asymmetric when

there is an efficiently-computable isomorphism $\psi : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and $\hat{e}$ is implemented as $e : \mathbb{G}_1 \times \psi(\mathbb{G}_1) \rightarrow \mathbb{G}_T$. With the appropriate notational modifications, all of the conditions listed above must apply in this setting as well. We will use both settings in this work.

Parameter Generation. Our protocols will assume the existence of a p.p.t. algorithm $\mathsf{BMsetup}$ that, on input a security parameter 1^κ, outputs the parameters γ for a bilinear mapping. In the asymmetric setting, we will specify that $\gamma = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \tilde{g}, e)$ where g generates $\mathbb{G}_1$, $\tilde{g}$ generates $\mathbb{G}_2$, the groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ have prime order q, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. In the symmetric setting, we will have $\gamma = (q, \mathbb{G}, \mathbb{G}_T, g, e)$. Our schemes require that the correctness of these parameters be publicly verifiable (Chen *et al.* [CCS07] describe efficient techniques for verifying these parameters in a typical instantiation).

3.2.1 Concrete Settings

We now briefly outline several relevant facts about known instantiations of bilinear groups. We will keep this discussion at a high level, and point the reader to [Men05, GPS06] for a detailed tutorial.

All known bilinear groups are constructed such that $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of points on some elliptic curve E over a prime-order finite field $\mathbb{F}_p$. The group $\mathbb{G}_T$ is usually a multiplicative subgroup over a related extension field $\mathbb{F}_p^k$ where k is the embedding degree of the curve. In curves with low embedding degree, the bilinear map can be implemented using the Weil or Tate pairings, which can be computed efficiently using Miller's algorithm [Mil04].

A notable feature of the elliptic curve setting is the absence of sub-exponential-time algorithms for directly solving the discrete logarithm problem (DLP) within a curve subgroup (given $g, h \in \mathbb{G}$, the discrete logarithm problem is that of finding the a value x such that $g^x = h$). However, Menezes *et al.* [MVO91] showed that in curves of low embedding degree, the Weil or Tate pairing can be used to transfer a problem instance into the extension field $\mathbb{F}_p^k$ where sub-exponential DLP solving are known (*e.g.,*[AH99]). Thus, in bilinear groups, the hardness of the DLP determined both by the order of the elliptic curve subgroup (q) *and* the size of the extension field (p^k).

Size of group elements. The selection of the curve has implications for the security and efficiency of protocols set in bilinear groups. For example, to achieve an "80-bit" security level in $\mathbb{G}_1$ (*i.e.,* solving the DLP requires approximately 2^{80} operations) we must select $q \approx 2^{160}$ to compensate for Pollard's rho algorithm [Pol78], and p, k such that $p^k \approx 2^{1024}$ to deal with field-based solvers. In many cases, it is possible to construct the group $\mathbb{G}_1$ such that $|q| \approx |p|$, and thus elements of $\mathbb{G}_1$ can be represented with approximately $|p| = 160$ bits.[2] While the representation of $\mathbb{G}_1$ will be quite compact, elements in $\mathbb{G}_T$ must be at least six times as large (1024 bits) to retain security. In general, the representation of $\mathbb{G}_2$

[2]Note that a point consists of two elements (x, y) in $\mathbb{F}_k$. However, it is possible to compute y from x and the least-significant bit of y.

will be between three and six times that of $\mathbb{G}_1$.[3] The reader should keep these figures in mind when evaluating our protocols of Chapters 4 and 5.

The hardness of Decisional Diffie Hellman. In symmetric bilinear groups, the availability of a bilinear map permits an efficient solution to the Decisional Diffie-Hellman (DDH) problem in $\mathbb{G}$ (specifically: given $(\langle g \rangle = \mathbb{G}, g^a, g^b, Q)$, decide whether $Q = g^{ab}$). One can solve such an instance by testing $e(g, Q) \stackrel{?}{=} e(g^a, g^b)$. Our constructions of Chapter 5 will use asymmetric ("SXDH") groups [Sco02, BBS04, BGdMM05] in which the Decisional Diffie-Hellman problem is assumed to be hard within both $\mathbb{G}_1$ and $\mathbb{G}_2$.

Given the ease of solving DDH in symmetric bilinear groups, the SXDH assumption may seem a strong additional assumption. However, we note that the *only* known method for solving DDH in known bilinear groups is to apply the bilinear map (pairing) on two elements within the same group. In asymmetric bilinear groups, the pairing must be combined with an efficiently-computable *distortion map* $\psi : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ that permits the map to "operate" on elements within the same group. Verheul proved that in certain curves, various choices of $\mathbb{G}_1$ do not admit efficiently computable distortion maps to $\mathbb{G}_2$ (and vice-versa) [Ver04]. This rules out the only known technique for solving DDH within these subgroups. Thus, although the result does not rule out the possibility of an alternative DDH-solving approach, such an outcome seems unlikely.

3.3 Complexity Assumptions

Our protocols will be use a variety of complexity assumptions. Many of these assumptions are made in symmetric and asymmetric bilinear groups. However, our constructions in Chapter 6 will use components that are secure in the RSA setting.

3.3.1 Comparing cryptographic assumptions

The past several years have seen a rapid increase in the number of new complexity assumptions used by cryptographers. This phenomenon is partly due to the introduction of bilinear-map based cryptography, whose new settings require new cryptographic assumptions. It can be further explained by a renewed push to develop constructions whose security does not depend on random oracles.

Unfortunately, introducing new assumptions is a risky process, and many new constructions are proven secure only by assuming the the hardness of some complex, unstudied problem. Indeed, Cheon recently illustrated the limitations of this approach with his attack on the widely-used p-Strong Diffie-Hellman problem [Che06].

Some efforts have been made to address this problem. Shoup's Generic Group model can be used to evaluate the complexity of a problem within an ideal cyclic group that has

[3] Typically elements of $\mathbb{G}_2$ are on the curve over the extension field $\mathbb{F}_p^k$ or $\mathbb{F}_p^{k/2}$. See [Men05].

no special structure [Sho97]. While this is a useful first step, a proof in such an artificial model should be considered at most an argument in favor of the assumption.

Thus, when evaluating constructions we need to be mindful of the assumptions used (and introduced) in their security proofs. In particular, we wish to use well-known mathematical problems with particular properties that make us more confident in their validity. These properties include (1) ease of description (preferably, the problem instance should be constant size regardless of the Adversary's behavior, (2) non-interactivity (since interactive assumptions are relatively harder to falsify), and (3) a constant-size solution space (there should be one, or a relatively small number of valid solutions). We will informally characterize the known assumptions into three categories of increasing "risk":

1. **Static assumptions.** These assumptions have only a single solution, and have a constant-size description that does not depend on the Adversary's behavior.
2. **Dynamic assumptions**. These assumptions are non-interactive, but have a description that must vary in size depending on the Adversary's behavior, *e.g.,* the number of signing queries that will be requested in a signature scheme. They may also have one or more valid solution.
3. **Interactive assumptions.** These assumptions provide the Adversary with an oracle to which it may send chosen inputs. These assumptions are relatively difficult to falsify, since a problem instance cannot be efficiently described.

The p-Strong Diffie-Hellman and p-Power Decision Diffie-Hellman assumptions used by the $\mathsf{OT}^N_{k\times 1}$ of Camenisch *et al.* are examples of dynamic assumptions, since each has a description that is variable, and ultimately linear in the number of messages in the OT database. The p-SDH problem also has an exponential number of valid solutions. A major goal of our constructions of Chapter 4 will be to replace these assumptions with static assumptions such as the Decisional Bilinear Diffie-Hellman assumption (see below).

3.3.2 Bilinear Settings

We first define two well-known complexity assumptions that are believed to hold in *symmetric* bilinear groups.

Definition 3.3.1 (Computational Diffie-Hellman (CDH) [DH76]) *Let* $\mathsf{BMsetup}(1^\kappa) \rightarrow (q, g, \mathbb{G}, \mathbb{G}_T, e)$. *The* Computational Diffie-Hellman *assumption holds in* $\mathbb{G}_1$ *(resp.* $\mathbb{G}_2, \mathbb{G}_T$*) if for all p.p.t. adversaries* Adv*, the following probability is strictly less than* $1/\mathrm{poly}(\kappa)$*:*

$$\Pr[a, b \xleftarrow{\$} \mathbb{Z}_q : \mathsf{Adv}(\gamma, g^a, g^b) = g^{ab}]$$

While CDH is generally believed to hold in bilinear groups, the *Decisional Diffie-Hellman* assumption (DDH) is known *not* to hold in the image group $\mathbb{G}_1$ of a symmetric bilinear map, and may or may not hold in asymmetric bilinear groups.

Definition 3.3.2 (Decisional Bilinear Diffie-Hellman (DBDH) [BF01])
Let $\mathsf{BMsetup}(1^\kappa) \rightarrow (q, g, \mathbb{G}, \mathbb{G}_T, e)$. *The* Decisional Bilinear Diffie-Hellman *assumption holds if for all p.p.t. adversaries* Adv*, the following probability is strictly less than 1/2 +* $1/\mathrm{poly}(\kappa)$*:*

$$\Pr[a, b, c, d \xleftarrow{\$} \mathbb{Z}_q;\ x_0 \leftarrow e(g,g)^{abc};\ x_1 \leftarrow e(g,g)^d; \\ z \leftarrow \{0,1\} :\ \mathsf{Adv}(\gamma, g^a, g^b, g^c, x_z) = z]$$

We now present several assumptions set in *asymmetric* bilinear groups.

Definition 3.3.3 (Computational Co-Diffie-Hellman (Co-CDH) [BLS01]) *Let* $\mathsf{BMsetup}(1^\kappa) \rightarrow (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$. *The* Computational Co-Diffie-Hellman *assumption holds in* $\mathbb{G}_1, \mathbb{G}_2$*) if for all p.p.t. adversaries* Adv*, the following probability is strictly less than* $1/\mathrm{poly}(\kappa)$*:*

$$\Pr[a, b \xleftarrow{\$} \mathbb{Z}_q :\ \mathsf{Adv}(\gamma, g^a, \tilde{g}^b) = \tilde{g}^{ab}]$$

We remark that co-CDH is simply a variant of Computational Diffie Hellman, where the Adversary's input is split across two distinct groups. It can alternatively be described by swapping the order and placement of the groups $\mathbb{G}_1, \mathbb{G}_2$ above.

Definition 3.3.4 (Decision Linear Assumption (DLIN) [BBS04]) *Let* $\mathsf{BMsetup}(1^\kappa) \rightarrow (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$. *The* Decision Linear Assumption *holds if for all p.p.t. adversaries* Adv*, the following probability is strictly less than 1/2 +* $1/\mathrm{poly}(\kappa)$*:*

$$\Pr[a, b, c, d \xleftarrow{\$} \mathbb{Z}_q; f \leftarrow g^c; \tilde{f} \leftarrow \tilde{g}^c; h \leftarrow g^d; \tilde{h} \leftarrow \tilde{g}^d; \\ x_0 \leftarrow h^{a+b}; x_1 \xleftarrow{\$} \mathbb{G}_1; z \leftarrow \{0,1\} : \mathsf{Adv}(\gamma, g, \tilde{g}, f, \tilde{f}, h, \tilde{h}, g^a, f^b, x_z) = z].$$

Note that this is a weaker asymmetric version of the original DLIN assumption of Boneh, Boyen and Shacham [BBS04], which was set in symmetric groups.

Definition 3.3.5 (Symmetric External Diffie-Hellman Assumption (SXDH) [Sco02, BBS04, BGdMM
Let $\mathsf{BMsetup}(1^\kappa) \rightarrow \gamma = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$. *The* Symmetric External Diffie-Hellman *assumption holds if the Decisional Diffie-Hellman problem is hard within both* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.More formally, for all p.p.t. adversaries* Adv*, the following two probabilities are each strictly less than 1/2 +* $1/\mathrm{poly}(\kappa)$*:*

1. $\Pr[g \xleftarrow{\$} \mathbb{G}_1; a, b \xleftarrow{\$} \mathbb{Z}_q; x_0 \leftarrow g^{ab}; x_1 \xleftarrow{\$} \mathbb{G}_1; z \leftarrow \{0,1\} : \mathsf{Adv}(\gamma, g^a, g^b, x_z) = z]$
2. $\Pr[\tilde{g} \xleftarrow{\$} \mathbb{G}_2; a, b \xleftarrow{\$} \mathbb{Z}_q; \tilde{x}_0 \leftarrow \tilde{g}^{ab}; \tilde{x}_1 \xleftarrow{\$} \mathbb{G}_2; z \leftarrow \{0,1\} : \mathsf{Adv}(\gamma, \tilde{g}^a, \tilde{g}^b, \tilde{x}_z) = z]$.

We remark that the SXDH assumption is implied by the following assumption:

Definition 3.3.6 (p-Hidden LRSW Assumption) *Let* $\mathsf{BMsetup}(1^\kappa) \rightarrow \gamma = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$. *The* p-Hidden LRSW Assumption *holds if for all p.p.t. adversaries* Adv*, the following probability is strictly less than* $1/\mathrm{poly}(\kappa)$*:*

$$\begin{aligned}\Pr[s, t \xleftarrow{\$} \mathbb{Z}_q; \tilde{S} \leftarrow \tilde{g}^s, \tilde{T} \leftarrow \tilde{g}^t; \forall i \in [1 \ldots p], x_i, y_i \xleftarrow{\$} \mathbb{Z}_q, b_i \leftarrow g^{y_i}, \tilde{b}_i \leftarrow \tilde{g}^{y_i};\\ A \leftarrow \mathsf{Adv}(\gamma, \tilde{S}, \tilde{T}, \{b_1, b_1^{s+x_1 st}, b_1^{x_1}, b_1^{x_1 t}, g^{x_1}, \tilde{b}_1\}, \ldots, \{b_p, b_p^{s+x_p st}, b_p^{x_p}, b_p^{x_p t}, g^{x_p}, \tilde{b}_p\}) :\\ A = (a_1, a_2, a_3, a_4, a_5, a_6) \wedge x \notin \{x_1, \ldots, x_p\} \wedge x \in \mathbb{Z}_q^* \wedge a_1 \in \mathbb{G}_1 \wedge\\ a_2 = a_1^{s+xst} \wedge a_3 = a_1^x \wedge a_4 = a_1^{xt} \wedge a_5 = g^x \wedge e(a_1, \tilde{g}) = e(g, a_6)].\end{aligned}$$

This is a new assumption introduced by this work. However, related formulations of the above assumption in an oracle-setting, where the x_i values are chosen dynamically by Adv, are the LRSW assumption which was introduced by Lysyanskaya *et al.* [LRSW99] and the Strong LRSW assumption of Ateniese *et al.* [ACdM05]. We eliminate the oracle and instead give q random tuples, which are also slightly changed. To provide evidence in support of the above assumption, we show in Appendix C.2 that it admits a proof in Shoup's generic group model [Sho97].

3.3.3 RSA Setting

The anonymous credential protocols used in Chapter 6 may be set in the RSA cryptographic setting. Let p, q be large safe primes (*i.e.,* for some primes p', q' we can express $p = 2p' + 1$ and $q = 2q' + 1$), and let $n = pq$ be an RSA modulus. By $\mathbb{Z}_n^*$ we denote a group consisting of the set of all elements in $[1, n-1]$ which are relatively prime to n. We define the following cryptographic assumption in this setting:

Definition 3.3.7 (Strong RSA Assumption [BP97, FO97]) *Given an RSA modulus* n *and a random element* $g \in \mathbb{Z}_n^*$*, it is hard to compute* $h \in \mathbb{Z}_n^*$ *and integer* $e > 1$ *such that* $h^e \equiv g \bmod n$*. The modulus* n *is of a special form* pq*, where* $p = 2p' + 1$ *and* $q = 2q' + 1$ *are safe primes.*

3.4 Zero-Knowledge and Witness Indistinguishable Proofs

Zero knowledge (ZK) and Witness-Indistinguishable proofs allow one party (Prover) to convince another (Verifier) of the validity of a statement, *without* leaking additional information [GMR89]. Such proofs exist for all languages in NP [GMR89]. In this work will a related tool: the zero-knowledge *proof of knowledge*, which allows the Prover to demonstrate knowledge of a witness that satisfies a particular statement, without revealing any information to the verifier. A witness indistinguishable proof of knowledge has a similar

property, but satisfies only the weaker property that the Verifier not learn which witness was used to form the proof. We note that every ZK proof is implicitly a WI proof (but not necessarily the reverse).

In particular, protocols of Chapter 4 will use interactive proofs of knowledge (which can be made non-interactive through the use of random oracles). In Chapter 5 we will construct non-interactive proofs using the Groth-Sahai proof system [GS08]. When referring to a zero-knowledge or witness-indistinguishable proof, we will use the notation of Camenisch and Stadler [CS97]. For instance, to describe a zero-knowledge proof of knowledge of values x and r such that the statement $y = g^x h^r$ holds and $1 \leq x \leq n$, we will write:

$$ZKPoK\{(x, r) : y = g^x h^r \wedge (1 \leq x \leq n)\}$$

All values not in enclosed in $()$'s are assumed to be known to the verifier. We will denote witness-indistinguishable proofs by $WIPoK$. Wherever possible we will specify proofs using the weaker WI requirement, though a zero-knowledge proof will naturally suffice.

We now *informally* sketch some general requirements for these proof systems, with formal definitions provided in later chapters.

Correctness. Given an honestly-generated proof (and any necessary global parameters) an honest verifier will accept the proof with probability 1.

Extractability (Soundness). We require that all *p.p.t.* adversaries have at most negligible probability of convincing an honest Verifier to accept a proof of an invalid statement. In the case of a proof-of-knowledge, we formalize this by mandating the existence of a knowledge extractor that can be used (under appropriate circumstances, see below) to obtain the values being proved, with at most a negligible probability of failure.

Witness Indistinguishability. We require that all *p.p.t.* adversaries have at most negligible advantage in the following game. Allow the adversary to choose a statement and two distinct satisfying witnesses W_0, W_1. Select a random $b \xleftarrow{\$} \{0, 1\}$ and conduct a proof with the adversary using witness W_b, to obtain the adversary's guess b'. The adversary's advantage is defined by $|\Pr[b' = b] - 1/2|$.

Zero-Knowledge. We require that all *p.p.t.* adversaries have at most negligible advantage in the following game. Allow the adversary to choose a statement S and a satisfying witness W. Select a bit $b \xleftarrow{\$} \{0, 1\}$, and if $b = 0$ conduct the proof with the adversary based on W. If $b = 1$ conduct a *simulated* proof that is *not* based on W. Finally, obtain the adversary's output b'. The adversary's advantage is defined by $|\Pr[b' = b] - 1/2|$.

Thus, for every proof of knowledge we require a technique for extracting the knowledge being proved. Zero-knowledge proofs also require a technique for simulating proofs without knowledge of a witness (provided one exists). These processes will require capabilities that are not available to parties in a real environment: *e.g.,*the ability to rewind other participants and/or control "trusted" global parameters such as a common reference string.

3.4.1 Interactive Known Discrete-Logarithm Proofs

We use known zero-knowledge techniques for proving statements about discrete logarithms, such as (1) proof of knowledge of a discrete logarithm modulo a prime [Sch91], (2) proof that a committed value lies in a given integer interval [CFT98, CM99, Bou00], and also (3) proof of the disjunction or conjunction of any two of the previous [CDS94]. These protocols are secure under the discrete logarithm assumption, although some implementations of (2) also require the Strong RSA assumption. While the basic protocols are *honest-verifier* zero-knowledge, they can be efficiently converted to standard zero-knowledge [CDM00].

Note that we can apply the Fiat-Shamir heuristic [FS86] to make such proofs non-interactive in the random oracle model.

3.4.2 Non-interactive Groth-Sahai Proofs

The Groth-Sahai proof system [GS08] permits a variety of efficient non-interactive proofs of the satisfiability of one or more pairing product equations. For variables $\{\mathcal{X}\}_{1...m} \in \mathbb{G}_1, \{\mathcal{Y}\}_{1...n} \in \mathbb{G}_2$ and constants $\{\mathcal{A}\}_{1...n} \in \mathbb{G}_1, \{\mathcal{B}\}_{1...m} \in \mathbb{G}_2, a_{i,j} \in \mathbb{Z}_q$, and $t_T \in \mathbb{G}_T$, these equations have the form:

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{Y}_i) \prod_{i=1}^{m} e(\mathcal{X}_i, \mathcal{B}_i) \prod_{i=1}^{m} \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{Y}_j)^{a_{i,j}} = t_T$$

Groth and Sahai show how to construct Witness Indistinguishable proof-of-knowledge of a satisfying witness to such an equation, in prime-order groups where the SXDH or Decision Linear assumptions hold. The proof system they describe can be composed over multiple equations involving the same variables. Additionally, they point out that in some special cases, their techniques can be strengthened to provide Zero Knowledge. Unlike the interactive proofs used in [CNs07, GH08b], the Groth-Sahai proofs do not use adversarial rewinding in their security analysis.

Groth-Sahai Commitments [GS08]. At the core of the Groth-Sahai system is a homomorphic commitment scheme to elements of $\mathbb{G}_1$ or $\mathbb{G}_2$.[4] The public parameters for the commitment scheme can be generated in one of two ways. Method (1) leads to a perfectly-binding commitment scheme, while method (2) leads to a perfectly-*hiding* scheme. Note that the two parameter distributions are computationally indistinguishable under the SXDH assumption. When the GS commitment parameters are configured according to method (1), they are equivalent to an Elgamal encryption of a group element, and can be decrypted by a party that knows a trapdoor to the commitment parameters. When commitments are configured according to method (2), a "simulation" trapdoor can be used on random commitments to open them to any value g^x (or $\tilde{g}^x$) for known x.

[4] As noted in [GS08, BCKL08] the commitment scheme can also be used to commit to elements of $\mathbb{Z}_q$, though we use this only in the context of simulating proofs.

The Proof System. We now describe the proof system at a high level, adopting some notation and exposition from [BCKL08]. For this description we will conceal many of the underlying details, though the reader can refer to [GS08, BCKL08] for a more detailed explanation. The proof system contains the following (possibly probabilistic) polynomial time algorithms:

- $\mathsf{GSSetup}(\gamma)$. On input $\gamma \in \mathsf{BMsetup}(1^\kappa)$, outputs a string GS containing parameters for the proof system. This string embeds binding parameters for the G-S commitment scheme.
- $\mathsf{GSProve}\,(GS, S, W)$. On input a statement S describing the equation, and a satisfying witness $W \in \langle \{\mathcal{X}\}_{1\ldots m}, \{\mathcal{Y}\}_{1\ldots n} \rangle$, outputs a proof π. To formulate this proof, a commitment $\hat{C}_i$ is generated for each element in W. The proof embeds openings to the commitments in such a way that a prover can ascertain that S is verifiably satisfied, and yet the elements of W remain hidden.
- $\mathsf{GSVerify}(GS, \pi)$. Verifies the proof π (using the commitments and opening values) and outputs ACCEPT if π is valid, REJECT otherwise. (For compactness of notation, we will specify that π embeds the statement S).

Above we describe the proof system in normal operation. Our security proofs additionally use:

- $\mathsf{GSExtractSetup}(\gamma)$. Outputs GS (distributed identically to the output of $\mathsf{GSSetup}(\gamma)$) and an extraction trapdoor td_{ext} containing a trapdoor for the commitment scheme. This trapdoor permits an extraction of a valid witness from the commitments embedded within a proof.
- $\mathsf{GSExtract}(GS, td_{ext}, \pi)$. Given a proof π and the extraction trapdoor, extracts $\mathcal{X}_i$ or $\mathcal{Y}_i$ from each commitment $\hat{C}_i$, and outputs the witness $W = \langle \{\mathcal{X}\}_{1\ldots M}, \mathcal{Y}\}_{1\ldots N} \rangle$ that satisfies the equations.
- $\mathsf{GSSimulateSetup}(\gamma)$. Outputs parameters GS' that are computationally indistinguishable from the output of $\mathsf{GSSetup}(\gamma)$, as well as a simulation trapdoor td_{sim} which consists of a simulation trapdoor for the commitment scheme.
- $\mathsf{GSSimProve}(GS', td_{sim}, S)$. Given simulation parameters GS' and trapdoor td_{sim}, outputs a proof π of statement S that such that $\mathsf{GSVerify}(GS', \pi) = \text{ACCEPT}$. Note that this algorithm operates on certain restricted classes of statements (see below).

In the general case, Groth-Sahai proofs provide strong Witness Indistinguishability in groups where the SXDH assumption holds. However, in the special case where in all equations being simultaneously satisfied, the value $t_T = 1$ (or t_T can be decomposed in a specific way), then it is also possible to form proofs that meet a strong definition of composable Zero-Knowledge. We will further discuss the set of statements for which Zero-Knowledge proofs are possible below, and momentarily refer to this class as $\vec{S}_{ZK}$. We now discuss the security properties of the proof system:

Correctness. For honestly-generated GS and π, $\mathsf{GSVerify}(GS, \pi)$ will always output ACCEPT.

Extractability (Soundness). For $(GS, td_{ext}) \in \mathsf{GSExtractSetup}(\gamma)$ and some π (embedding a statement S): if $\mathsf{GSVerify}(GS, \pi)$ outputs ACCEPT then with probability 1 the algorithm $\mathsf{GSExtract}(GS, td, \pi)$ extracts a witness W that satisfies S.

Composable Witness Indistinguishability. We first require that the parameters generated by $\mathsf{GSSimulateSetup}(\gamma)$ be computationally indistinguishable from the parameters generated by $\mathsf{GSSetup}(\gamma)$. We additionally require that all *p.p.t.* adversaries $\mathcal{A}$ have advantage 0 in the following game. Hand $\mathcal{A}$ the parameters $GS' \leftarrow \mathsf{GSSimulateSetup}(\gamma)$, and allow $\mathcal{A}$ to output (S, W_0, W_1) where S is a statement and W_0, W_1 are distinct satisfying witnesses. Select $b \xleftarrow{\$} \{0, 1\}$, give $\mathcal{A}$ the proof $\pi \leftarrow \mathsf{GSProve}(GS', S, W_b)$, and collect its guess b'. $\mathcal{A}$'s advantage is defined as $|\Pr[b = b'] - 1/2|$.

Composable Zero-Knowledge. We again require that the parameters generated by $\mathsf{GSSimulateSetup}(\gamma)$ be computationally indistinguishable from the parameters generated by $\mathsf{GSSetup}(\gamma)$. We additionally require that all *p.p.t.* adversaries $\mathcal{A}$ have advantage 0 in the following game. Generate $(GS', td_{sim}) \leftarrow \mathsf{GSSimulateSetup}(\gamma)$, and give GS' to $\mathcal{A}$. Allow $\mathcal{A}$ to output (S, w) where $S \in \vec{S}_{ZK}$ and w is a satisfying witness. Let $\pi_0 \leftarrow \mathsf{GSProve}(GS', S, w), \pi_1 \leftarrow \mathsf{GSSimProve}(GS', td_{sim}, S)$. Select $b \xleftarrow{\$} \{0, 1\}$, give $\mathcal{A}$ the proof π_b, and collect its guess b'. $\mathcal{A}$'s advantage is $|\Pr[b = b'] - 1/2|$.

Note that GS proofs can be defined over multiple pairing product equations. In this case, satisfiability implies knowledge of a witness for each statement. In our constructions, we will denote a GS proof statement using the notation of Camenisch and Stadler [CS97]. For instance, $NIWI_{GS}\{(a_1, a_2) : e(a_1, a_2)e(g, h^{-1}) = 1 \wedge e(a_2, g_2)e(d_2^{-1}, a_3) = 1\}$ represents a non-interactive Witness Indistinguishable proof of knowledge, formed under parameters GS, of a witness $W = \langle a_1, a_2 \rangle$ that satisfies both statements. All values not in enclosed within the initial $()$'s are assumed to be known to the verifier. We will alternatively use the notation $NIZK$ to denote a Zero-Knowledge proof.

Statements with Zero-Knowledge Proofs. While Groth and Sahai [GS08] generally accomplish Witness-Indistinguishable (WI) proofs, they note that certain classes of pairing-product statements admit Zero-Knowledge proofs as well. In order to prove a statement in Zero-Knowledge (as per the definition above), a simulator must be able to produce a simulated proof π without being given specific knowledge of a witness to the statement. Note that if the simulator can compute a valid witness by itself, then it is sufficient to simply use a WI proof. For instance, in the special case where $t_T = 1$ for a pairing product equation, the simulator can always compute a satisfying witness by selecting each $\mathcal{X}_i$ or $\mathcal{Y}_i$ to be g^0 or $\tilde{g}^0$ respectively.

Groth and Sahai further observe that more complex statements can be made Zero Knowledge by applying the simulation trapdoor for the Groth-Sahai commitment scheme.

This trapdoor allows the simulator to open a random commitment to any g^x or $\tilde{g}^x$ (for known x), and can be applied such that the same commitment is opened *differently* for each equation within the statement. In some cases, we may need to re-write a statement in order to construct a ZK proof. For example, consider the proof $NIWI_{GS}\{(a) : e(a,d) = e(g,h)\}$ made on variable a and constants d, g, h. By adding a second variable b we obtain the equivalent $NIZK$ statement:

$$NIZK\{(a,b) : e(a,d)e(b,h^{-1}) = 1 \ \wedge \ e(b,g)e(g^{-1},g) = 1\}$$

Note that the equivalence holds by the property that $b = g$ is the only valid solution to the revised equation. However, we can simulate the statement by opening the appropriate commitments such that $a = b = g^0$ in the first equation, while in the second equation $b = g$. We will use similar techniques to simulate the Zero-Knowledge proofs in our constructions.

3.5 Commitment Schemes

Commitment schemes can be thought of as the digital equivalent of a physical envelope. Specifically, they allow a party to bind itself to a particular value without revealing it. At a later point, the party may reveal, or "decommit" this value. For a commitment scheme to be secure, it must be both *binding* and *hiding*. The binding property ensures that the committing party cannot change the value it has committed to, while hiding ensures that the commitment does not reveal the value (until the committing party reveals it).

We will describe a commitment scheme as a (possibly probabilistic) algorithms $(\mathsf{CSetup}, \mathsf{Commit}, \mathsf{Decommit})$ that operate as follows.

$\mathsf{CSetup}(1^\kappa)$. On input security parameter 1^κ, outputs public parameters ρ.
$\mathsf{Commit}(\rho, M)$. On input a message $M \in \{0,1\}^*$, outputs a commitment/decommitment pair $(\mathcal{C}, \mathcal{D})$.
$\mathsf{Decommit}(\rho, M, \mathcal{C}, \mathcal{D})$. On input $M, \mathcal{C}, \mathcal{D}$, outputs 1 if $\mathcal{D}$ decommits $\mathcal{C}$ to M, or 0 otherwise.

Informally a commitment scheme is computationally (resp. perfectly) *binding* if no polynomial-time (resp. unbounded) adversary can produce decommitments $\mathcal{D}, \mathcal{D}'$ and distinct messages M, M' such that $\mathsf{Decommit}(\rho, M, \mathcal{C}, \mathcal{D}) = 1$ and $\mathsf{Decommit}(\rho, M', \mathcal{C}, \mathcal{D}') = 1$. The commitment scheme is computationally (resp. perfectly) hiding if no polynomial-time (resp. unbounded) adversary gains any information about the underlying message M from $\rho, \mathcal{C}$.

Proving Knowledge of a Decommitment. Our constructions in Chapter 4 will require an *efficient* (possibly interactive) zero-knowledge protocol for proving knowledge of a decommitment $\mathcal{D}$ with respect to $(\rho, M, \mathcal{C})$. We will denote this proof as $ZKPoK\{(\mathcal{D}) : \mathsf{Decommit}(\rho, M, \mathcal{C}, \mathcal{D}) = 1\}$.

Instantiations. In our protocols of Chapter 4 we suggest using the Pedersen commitment scheme [Ped92] based on the discrete logarithm assumption, in which the public parameters are a group of prime order q, and random generators $(g_0, \ldots, g_m)$. In order to commit to the values $(v_1, \ldots, v_m) \in \mathbb{Z}_q^m$, pick a random $r \in \mathbb{Z}_q$ and set $\mathcal{C} = g_0^r \prod_{i=1}^m g_i^{v_i}$ and $\mathcal{D} = r$. Schnorr's technique [Sch91] can be used to efficiently prove knowledge of the value $\mathcal{D} = r$. The Pedersen scheme is computationally binding and perfectly hiding.

3.6 Signatures with Efficient Protocols

Our protocols in Chapter 6 make us of *signatures with efficient protocols*, or "p-signatures" [CL02, CL04, BCKL08]. In particular, we will use a signature scheme due to Camenisch and Lysyanskaya (CL) [CL02], which possesses two efficient protocols: (1) a protocol for a user to obtain a signature on the value(s) in a Pedersen (or Fujisaki-Okamoto) commitment [Ped92, FO97] without the signer learning anything about the message(s), and (2) a proof of knowledge of a signature on a committed value. CL signatures are based on the Strong RSA [BP97, FO97] assumption. We can easily substitute these for other bilinear signatures with efficient protocols [BB04a, CL04], though we will not provide explicit details on this usage.

We now briefly outline the Camenisch-Lysyanskaya signature scheme [CL02]. To generate a key, compute a special RSA modulus $n = pq$ (where p, q are safe primes) and three values $a, b, c \xleftarrow{\$} QR_n$. Set $pk = (n, a, b, c)$ and $sk = (p, q)$. To sign a message $m \in [0, 2^\ell]$ (for some parameter ℓ), select a random prime e and a random number s (of specific lengths described in [CL02]), and compute the signature σ such that $\sigma^e \equiv a^m b^s c \ mod \ n$. Signature verification consists of checking the previous equation and verifying that e has the appropriate length.

Note that the constructions of Chapter 5 will use a variant of a bilinear signature scheme that is also due to Camenisch and Lysyanskaya [CL04]. This scheme should not be confused with the p-signature described above.

3.7 Identity-Based Encryption

Identity-Based Encryption (IBE), proposed by Shamir [Sha84] and realized by Boneh-Franklin and Cocks [BF01, Coc01] is an alternative to public-key encryption where users' identities serve as their public key. An IBE scheme supports two types of players: a single master authority ($\mathcal{P}$), and multiple users ($\mathcal{U}$) who obtain their secret keys from the PKG. These players make use of the algorithms Setup, Encrypt, Decrypt and the protocol Extract. Let us provide some input/output specification for these protocols with intuition for what they do.

Notation: Let $\mathcal{I}$ be the identity space and $\mathcal{M}$ be the message space. We write

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

- In the $\mathsf{Setup}(1^\kappa, c(\kappa))$ algorithm, on input a security parameter 1^κ and a description of an the identity space $|\mathcal{I}| \leq 2^{c(\kappa)}$ where $c(\cdot)$ is a computable, polynomially-bounded function, the master authority $\mathcal{P}$ outputs master parameters $params$ and a master secret key msk.
- In the $\mathsf{Extract}(\mathcal{P}(params, msk), \mathcal{U}(params, id)) \rightarrow (id, sk_{id})$ protocol, an honest user $\mathcal{U}$ with identity $id \in \mathcal{I}$ obtains the corresponding secret key sk_{id} from the master authority $\mathcal{P}$ or outputs an error message. The master authority's output is **the identity** id or an error message.[5] (Note that $\mathcal{P}$ is permitted to abort the protocol selectively based on id.)
- In the $\mathsf{Encrypt}(params, id, m)$ algorithm, on input identity $id \in \mathcal{I}$ and message $m \in \mathcal{M}$, any party can output ciphertext C.
- In the $\mathsf{Decrypt}(params, id, sk_{id}, C)$ algorithm, on input a ciphertext C, the user with sk_{id} outputs a message $m \in \mathcal{M}$ or the distinguished symbol ϕ.

Throughout the remainder of the text we will assume that $params$ defines $\mathcal{I}$ and $\mathcal{M}$.

Security of IBE. Traditionally, there are various levels of ciphertext security that an IBE scheme might meet: security against chosen-plaintext attack (CPA) vs. security against the stronger chosen-ciphertext attack (CCA), security against selective-identity attacks [CHK04] vs. security against the stronger adaptive-identity attacks [BF01]. Fortunately, our OT protocols in §4.2 require only the weakest ciphertext security notion: selective-identity security against chosen-plaintext attack ($\mathsf{IND\text{-}sID\text{-}CPA}$). We now define this notion.

Definition 3.7.1 (Selective-Identity Secure IBE ($\mathsf{IND\text{-}sID\text{-}CPA}$) [CHK04]) Let κ be a security parameter, $c(\cdot)$ be a polynomially-bounded function, $|\mathcal{I}| \leq 2^{c(\kappa)}$ and $\mathcal{M}$ be the message space. An IBE is $\mathsf{IND\text{-}sID\text{-}CPA}$-secure if every p.p.t. adversary $\mathcal{A}$ has an advantage negligible in κ for the following game:

1. $\mathcal{A}$ outputs a target identity $id^* \in \mathcal{I}$.
2. Run $\mathsf{Setup}(1^\kappa, c(\kappa))$ to obtain $(params, msk)$, and give $params$ to $\mathcal{A}$.
3. $\mathcal{A}$ may run the $\mathsf{Extract}$ protocol with an oracle $O_{params,msk}(\cdot)$ polynomially many times, where on any input $id \neq id^*$ in $\mathcal{I}$, the oracle returns sk_{id}, and on any other input, the oracle returns an error message.
4. $\mathcal{A}$ outputs two messages $m_0, m_1 \in \mathcal{M}$ where $|m_0| = |m_1|$. Select a random bit b and give $\mathcal{A}$ the challenge ciphertext $c^* \leftarrow \mathsf{Encrypt}(params, id^*, m_b)$.

[5] The canonical definition of IBE [BF01] specifies an extraction *algorithm*. Note however that given such an algorithm, one can define a simple $\mathsf{Extract}$ protocol as: (1) $\mathcal{U}$ transmits id, (2) if $id \in \mathcal{I}$, $\mathcal{P}$ runs the extraction algorithm on $(params, msk, id)$ to obtain sk_{id} and returns this value (or an error), (3) user checks the validity of sk_{id} by encrypting a polynomially-bounded number of random messages and verifying their correct decryption.

5. $\mathcal{A}$ may continue to query oracle $O_{params,msk}(\cdot)$ under the same conditions as before.
6. $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

We define $\mathcal{A}$'s advantage in the above game as $|\Pr[b' = b] - 1/2|$.

On stronger notions of ciphertext security for IBE. A stronger notion of ciphertext security for IBE schemes is adaptive-identity security (IND-ID-CPA) [BF01], which strengthens the IND-sID-CPA definition by allowing $\mathcal{A}$ to delay selecting the target identity id^* until the start of step (4) in the above game. In §4.3, we show blind IBE schemes satisfying both IND-sID-CPA and IND-ID-CPA security. Fortunately, our oblivious transfer applications in §4.2 require only IND-sID-CPA-security (because the "identities" will be fixed integers from 1 to $\mathrm{poly}(\kappa)$), some additional applications in §6.2 require the stronger IND-ID-CPA-security.

Chapter 4

Fully Simulatable Oblivious Transfer from Blind IBE

This chapter is based on joint work with Susan Hohenberger. An extended abstract was originally published in Kaoru Kurosawa (Ed.): Advances in Cryptology - ASIACRYPT 2007, volume 4833 of Lecture Notes in Computer Science, pages 265–282, Springer-Verlag, 2007 [GH08b].

In this chapter we will investigate an approach to constructing OT_k^N and $\mathsf{OT}_{k\times 1}^N$ protocols using techniques from the field of Identity-Based Encryption (IBE). Our techniques will be both efficient and secure under a strong *fully-simulatable* definition. While Camenisch *et al.* also proposed efficient and fully-simulatable protocols for $\mathsf{OT}_{k\times 1}^N$, realizing those protocols securely requires either *interactive* or strong *dynamic* p-based assumptions such as p-Power Decisional Diffie-Hellman (*i.e.*, in a bilinear setting $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, given $(g, g^x, g^{x^2}, \ldots, g^{x^q}, H)$ where $g \leftarrow \mathbb{G}$ and $H \leftarrow \mathbb{G}_T$, distinguish $(H^x, H^{x^2}, \ldots, H^{x^q})$ from random values) and p-Strong Diffie-Hellman (p-SDH) [BB04b].

Given the complexity of these assumptions, it is interesting to develop new protocols that achieve the same properties using static assumptions. In the following chapter we propose, to our knowledge, the first efficient and fully-simulatable OT schemes secure under *static* complexity assumptions (*e.g.*, DBDH, where given (g, g^a, g^b, g^c), it is hard to distinguish $e(g, g)^{abc}$ from random). We summarize our results as follows.

Intuition behind the Constructions. Oblivious Transfer protocols can be roughly divided into two categories. Let's restrict our attention to non-adaptive OT_1^N for the moment. In approach (1), which is used by [Rab81, EGL82, Lin08, PVW08], the Receiver transmits a collection of specially-formed encryption keys to the Sender, who encrypts each message and returns the N ciphertexts to the Receiver. The protocol is secure provided that the encryption keys are formed such that a Receiver is able to decrypt at most *one* of the resulting ciphertexts. In approach (2), which is used by [CT05, FIPR05, CNs07, GH08b]

and this work, the Sender encrypts the message collection under keys of her own choosing, and— in some interactive protocol with the Receiver— helps to decrypt *one* ciphertext.

While both approaches can be used to implement adaptive OT in theory, the first approach requires that the Sender generate a new set of ciphertexts at *each* transfer stage (for *each* receiver), requiring at least $O(N \cdot k)$ cost. Even worse, the Sender might be able to maliciously change the database between transfers and present different versions of the database to different receivers.

The latter approach is much better suited for the adaptive case. A single database can be committed to and then each decryption can be performed in constant computational and communication cost, for a total $O(N + k)$ cost. This approach is taken by the fully-simulatable protocols of [CNs07], which both use rewinding in their simulations to (1) simulate proofs and (2) extract knowledge.[1]

Our Approach. First, we introduce a building block, which is of independent interest. In identity-based encryption (IBE) [Sha84], there is an *extraction* protocol where a user submits an identity string to a master authority who then returns the corresponding decryption key for that identity. We formalize the notion of *blindly* executing this protocol, in a strong sense; where the authority does not learn the identity nor can she cause failures dependent on the identity, and the user learns nothing beyond the normal extraction protocol. This concept has similarities to recent work by Goyal [Goy07], in which a user wishes to hide certain characteristics of an extracted IBE key from the authority. We call IBE schemes supporting efficient blind extraction protocols: *blind IBE*, for short.

Second, we will present an efficient and fully-simulatable OT_k^N protocol that can be constructed from any blind IBE scheme meeting our definitions (with the additional assumption of a secure commitment scheme). When implemented with the concrete Blind IBE schemes of §4.3, our constructions will be secure under only DBDH. Intuitively, consider the following OT_k^N construction. The Sender runs the IBE setup algorithm and sends the corresponding public parameters to the Receiver. Next, for $i = 1$ to N, the Sender encrypts M_i under identity "i" and sends this ciphertext to the Receiver. To obtain k messages, the Receiver blindly extracts k decryption keys for identities of his choice and uses these keys to decrypt and recover the corresponding messages. While this simple protocol does not appear to be simulatable, we are able to appropriately modify it. (Indeed, one must also be cautious of possibly malformed ciphertexts, as we discuss later.) Our constructions from blind IBE are inspired by the Camenisch *et al.* [CNs07] generic construction from unique blind signatures. Indeed, recall that the secret keys sk_{id} of any fully-secure IBE can be viewed as signatures by the authority on the message "id" [BF01]. Camenisch *et al.* [CNs07] require *unique* blind signatures, whereas we do not; however, where they require unforgeability, we require that our "blind key extraction" protocol does not jeopardize

[1] Along the same lines, the half-simulation protocols of [NP99b, FIPR05] use a form of oblivious pseudo-random function evaluation (OPRF) to encrypt and obliviously decrypt the message database. Unfortunately, the evaluation protocols described in those works appear vulnerable to selective-failure attacks, and the modifications necessary to achieve UC security (or full simulation) seem substantial.

the semantic security of the IBE.

Third, we present an efficient and fully-simulatable $\mathsf{OT}^N_{k\times 1}$ protocol constructed from blind IBE in the random oracle model. We discuss how to remove these oracles at an additional cost. This improves on the complexity assumptions required by the comparable random-oracle scheme in Camenisch *et al.* [CNs07], although we leave the same improvement for their adaptive construction without random oracles as an open problem.

Finally, in §4.3, we will describe efficient *blind extraction* protocols satisfying this definition for several concrete IBE schemes, including those due to Boneh and Boyen [BB04a] and Waters [Wat05] (using a generalization proposed independently by Naccache [Nac05] and Chatterjee and Sarkar [CS05]). The latter protocol is similar to a blind signature scheme proposed by Okamoto [Oka06]. In section §6.2 we will also discuss the independent usefulness of blind IBE to other applications, such as blind signatures, anonymous email, and encrypted keyword search.

4.1 Blind Identity-Based Encryption

In an identity-based encryption (IBE) scheme, senders encrypt messages using the recipient's *identity* as the public key. The concept was first proposed by Shamir [Sha84]; however, the first IBE schemes were realized several years later by Boneh and Franklin [BF01] and by Cocks [Coc01]. Beyond encryption applications, IBE has also led to the development of a variety of novel cryptographic protocols, such as secret handshakes [BDS+03], public-key searchable encryption [BCOP04, WBDS04], CCA-secure public-key encryption [CHK04], and digital signatures [BLS01].

In §3.7 we described a traditional IBE scheme. A *blind IBE* scheme consists of the same players, together with the same algorithms Setup, Encrypt, Decrypt and yet we replace the protocol Extract with a new protocol BlindExtract which differs only in the authority's output:

- In the $\mathsf{BlindExtract}(\mathcal{P}(params, msk), \mathcal{U}(params, id)) \rightarrow (\text{nothing}, sk_{id})$ protocol, an honest user $\mathcal{U}$ with identity $id \in \mathcal{I}$ obtains the corresponding secret key sk_{id} from the master authority $\mathcal{P}$ or outputs an error message. The master authority's output is **nothing** or an error message.

We now define security for blind IBE, which informally is any IND-sID-CPA-secure (or IND-ID-CPA-secure) IBE scheme with a BlindExtract protocol that satisfies two properties:

1. Leak-free Extract: a potentially malicious user cannot learn anything by executing the BlindExtract protocol with an honest authority which she could not have learned by executing the Extract protocol with an honest authority; moreover, as in Extract, the user must know the identity for which she is extracting a key.

2. Selective-failure Blindness: a potentially malicious authority cannot learn anything about the user's choice of identity during the BlindExtract protocol; moreover, the

authority cannot cause the BlindExtract protocol to fail in a manner dependent on the user's choice.

Of course, a protocol realizing the functionality BlindExtract (in a fashion that satisfies the properties above) is a special case of secure two-party computation [Yao86, GMW87, Kil88]. However, using generic tools may be inefficient, so as in the case of blind signature protocols, we seek to optimize this specific computation. Indeed, recall that sk_{id} in an adaptive-identity secure IBE can be viewed as a signature by the authority on message id (see §6.2). Thus, our BlindExtract protocol (for an adaptive-identity secure IBE) *is* a blind signature scheme, but the converse implication is not necessarily true. Our leak-free extraction property is much stronger than the common *one-more unforgeability* requirement of blind signatures. Moreover, we will not require adaptive-identity security for the IBE in our OT applications. Let us now formally state these properties.

Definition 4.1.1 (Leak-Free Extract) A protocol $\mathsf{BlindExtract} = (\mathcal{P},\mathcal{U})$ associated with an IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ is *leak free* if for all efficient adversaries $\mathcal{A}$, there exists an efficient simulator $\mathcal{S}$ such that for every value κ and polynomial $c(\cdot)$, no efficient distinguisher D can distinguish the output of Game Real from Game Ideal with non-negligible advantage:

Game Real: Run $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa, c(\kappa))$ and publish $params$. Each time D requests it, $\mathcal{A}$ chooses an identity id and atomically executes the BlindExtract protocol with $\mathcal{P}$: $\mathsf{BlindExtract}(\mathcal{P}(params, msk), \mathcal{A}(params, id))$. $\mathcal{A}$'s output (which is the output of the game) includes the list of identities and extracted keys.

Game Ideal: A trusted party runs $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa, c(\kappa))$ and publishes $params$. Each time D's requests it, $\mathcal{S}$ chooses an identity id and queries the trusted party to obtain the output of $\mathsf{Extract}(params, msk, id)$, if $id \in \mathcal{I}$ and $\perp$ otherwise. $\mathcal{S}$'s output (which is the output of the game) includes the list of identities and extracted keys.

In the games above, BlindExtract and Extract are treated as atomic operations. Hence D and $\mathcal{A}$ (or $\mathcal{S}$) may communicate at any time except during the execution of those protocols. Additionally, while we do not explicitly specify that auxiliary information is given to the parties, this information must be provided in order to achieve the sequential composition property required by our OT protocols in §4.2.

This definition implies that the identity id (for the key being extracted) is *extractable* from the BlindExtract protocol— with all but negligible probability— since for every adversary there exists a $\mathcal{S}$ that must be able to interact with a black-box $\mathcal{A}$ to learn which identities to submit to the trusted party. We will make use of this observation later. Another nice property of this definition is that any key extraction protocol with leak-freeness (regardless of whether blindness holds or not) composes into the existing security definitions for IBE. (This would not necessarily be true of a blind signature protocol for the same type of signatures.) We state this formally below.

Lemma 4.1.2 *If* $\Pi = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is an* IND-sID-CPA*-secure (resp.,* IND-ID-CPA*) IBE scheme and* BlindExtract *associated with* Π *is leak-free, then* $\Pi' = (\mathsf{Setup}, \mathsf{BlindExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is an* IND-sID-CPA*-secure (resp.,* IND-ID-CPA*) IBE scheme.*

Next, we define the second property of *blindness*. We use a strong notion of blindness called *selective-failure blindness* proposed recently by Camenisch et al. [CNs07], ensuring that even a malicious authority is unable to induce BlindExtract protocol failures that are dependent on the identity being extracted.

Definition 4.1.3 (Selective-Failure Blindness (SFB) [CNs07]) *A protocol* $P(\mathcal{A}(\cdot), \mathcal{U}(\cdot,\cdot))$ *is said to be selective-failure blind if every p.p.t. adversary* $\mathcal{A}$ *has a negligible advantage in the following game: First,* $\mathcal{A}$ *outputs* $params$ *and a pair of identities* $id_0, id_1 \in \mathcal{I}$*. A random* $b \in \{0,1\}$ *is chosen.* $\mathcal{A}$ *is given black-box access to two oracles* $\mathcal{U}(params, id_b)$ *and* $\mathcal{U}(params, id_{b-1})$*. 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* $sk_{b-1} \neq \perp$ *then* $\mathcal{A}$ *receives* $(\perp, \varepsilon)$*. If* $sk_b \neq \perp$ *and* $sk_{b-1} = \perp$ *then* $\mathcal{A}$ *receives* $(\varepsilon, \perp)$*. If* $sk_b = \perp$ *and* $sk_{b-1} = \perp$ *then* $\mathcal{A}$ *receives* $(\perp, \perp)$*. Finally,* $\mathcal{A}$ *outputs its guess* b'*. We define* $\mathcal{A}$*'s advantage in the above game as* $|\Pr[b' = b] - 1/2|$.

We thus arrive at the following definition.

Definition 4.1.4 (Secure Blind IBE) *A blind IBE* $\Pi = (\mathsf{Setup}, \mathsf{BlindExtract}, \mathsf{Encrypt}, \mathsf{Decrypt})$ *is called* IND-sID-CPA-secure *(resp.* IND-ID-CPA*) if and only if: (1)* Π *is* IND-sID-CPA*-secure (resp.* IND-ID-CPA*), and (2)* BlindExtract *is leak free and selective-failure blind.*

4.1.1 Additional Properties for a Blind IBE Scheme

Our constructions for OT_k^N and $\mathsf{OT}_{k\times 1}^N$ in §4.2 will require a blind IBE scheme with two additional properties, which we describe below.

Efficient PoK of master secret key. Our OT contructions will make use of an efficient zero-knowledge proof of knowledge protocol for the statement $ZKPoK\{(msk) : (params, msk) \in \mathsf{Setup}(1^\kappa, c(\kappa))\}$. If we were not concerned about efficiency, we could accomplish this proof using general techniques [Yao86, GMW87, Kil88]. Fortunately, in §4.3 we show that this proof can be conducted efficiently for a number of Blind IBE constructions.

Committing IBE. To construct our OT protocols, we will require that our blind IBE schemes be *committing*. This property is related to committing encryption [CFGN96], but deals with the fact that IBE decryption keys may be extracted from malicious parties. Intuitively, we want to ensure that a given ciphertext C_{id} always decrypts to the "correct"

plaintext, even when we are using decryption keys that have been extracted from a malicious PKG.

Somewhat more formally, we require that an adversary playing the role of the PKG is unable to generate an identity/ciphertext pair (id, C) and— by conducting the extraction protocol with an honest party— any two keys sk_{id}, sk'_{id} such that $\mathsf{Decrypt}(params, id, sk_{id}, C) \neq \mathsf{Decrypt}(params, id, sk'_{id}, C)$. We observe that this property holds trivially for any IBE scheme where identity keys are "unique" (there is at most one decryption key per identity). However, in certain schemes (*e.g.,* the Boneh-Boyen scheme [BB04a]), there are many valid decryption keys for a given identity. This may lead to a condition where some incorrectly-formed ciphertexts will decrypt to different values depending on which secret key is used.

To address schemes with this property, we will define a publicly-computable ciphertext correctness checking algorithm, which we denote by $\mathsf{IsValid}(params, id, C)$. The *correctness* property for the IsValid algorithm is that it outputs 1 for all honestly-generated parameters and ciphertexts. The algorithm's behavior in the case of maliciously-generated input is implicitly contained within the following definition:

Definition 4.1.5 (Committing IBE) *An IBE scheme (resp., blind IBE) is* committing *if and only if: (1) it is* IND-sID-CPA*-secure (resp., secure in the sense of definition 4.1.4) and (2) every p.p.t. adversary $\mathcal{A}$ has an advantage negligible in κ for the following game: First, $\mathcal{A}$ outputs $params, id \in \mathcal{I}$ and a ciphertext C. If $\mathsf{IsValid}(params, id, C) \neq 1$ then abort. Otherwise, the challenger, on input $(params, id)$, runs the* Extract *(resp.,* BlindExtract*) protocol with $\mathcal{A}$ twice to obtain purported keys sk_{id}, sk'_{id}. $\mathcal{A}$'s advantage is defined as:*

$$|\Pr\left[\mathsf{Decrypt}(params, id, sk_{id}, C) \neq \mathsf{Decrypt}(params, id, sk'_{id}, C)\right]|$$

4.2 OT Constructions

We now turn our attention to constructing efficient and fully-simulatable oblivious transfer protocols. Our constructions may be instantiated with any efficient blind IBE that satisfies Definition 4.1.5 (provided that there is an efficient proof of knowledge for the IBE master secret). In particular, we focus on building (non-adaptive) OT_k^N and (adaptive) $\mathsf{OT}_{k\times 1}^N$ protocols, in which a Sender and Receiver transfer up to k messages out of an N-message set. In the non-adaptive model [BCR86, NP99a], the Receiver requests all k messages simultaneously. In the adaptive model [NP99b], the Receiver may request the messages one at a time, using the result of previous transfers to inform successive requests. Intuitively, the Receiver should learn only the messages it requests (and nothing about the remaining messages), while the Sender should gain no information about *which* messages the Receiver selected.

Full-simulation vs. half-simulation security. Security for oblivious transfer is defined using the real-world/ideal-world paradigm. In the real world, a Sender and Receiver interact

directly according to the protocol, while in the ideal world, the parties interact via a trusted party. Informally, a protocol is secure if, for every real-world cheating Sender (resp., Receiver) we can describe an ideal-world counterpart who gains as much information from the ideal-world interaction as from the real protocol. Much of the oblivious transfer literature uses the simulation-based definition only to show *Sender* security, choosing to define Receiver security by a simpler game-based definition. Naor and Pinkas demonstrated that this weaker "half-simulation" approach permits *selective-failure* attacks, in which a malicious Sender induces transfer failures that are dependent on the message that the Receiver requests [NP99b]. Recently, Camenisch *et al.* [CNs07] proposed several practical $\mathsf{OT}^N_{k\times 1}$ protocols that are secure under a "full-simulation" definition, using adaptive (*e.g.*, q-PDDH) or interactive (*e.g.*, one-more-inversion RSA) assumptions. We now enhance their results by demonstrating efficient full-simulation OT^N_k and $\mathsf{OT}^N_{k\times 1}$ protocols secure under static complexity assumptions (*e.g.*, DBDH).

4.2.1 Non-adaptive OT^N_k in the Standard Model

Given a committing blind IBE scheme Π, it is tempting to consider the following "intuitive" protocol: First, the Sender runs the IBE Setup algorithm and sends $params$ to the Receiver. Next, for $i = 1, \ldots, N$ the Sender transmits an encryption of message M_i under identity "i". To obtain k messages, the Receiver extracts decryption keys for identities $(\sigma_1, \ldots, \sigma_k)$ via k distinct executions of BlindExtract, and uses these keys to decrypt the corresponding ciphertexts. If Π is a blind IBE secure in the sense of definition 4.1.5, then a cheating Receiver gains no information about the messages corresponding to secret keys he did not extract. Similarly, with additional precautions, a cheating Sender does not learn the identities extracted. However, it seems difficult to show this protocol is fully-simulatable, because the ideal Sender would have to form the N ciphertexts *before* learning the messages that k of them must decrypt to!

Fortunately, we are able to convert this simple idea into the fully-simulatable OT^N_k protocol shown in Figure 4.1. We require only the following modifications: first, we have the Sender prove knowledge of the value msk using appropriate zero-knowledge techniques.[2] Then, rather than transmitting the ciphertext vector during the first phase of the protocol, the Sender transmits only a *commitment* to the ciphertext vector,[3] and sends the actual ciphertexts at the end of the k^{th} round together with a proof that she can open the commitment to the ciphertext vector. (She does *not* open the commitment; she only proves that she knows how to do so.)

Note that when using a commitment scheme it is important to specify how the commitment parameters will be generated. In this case, the commitment scheme must be at

[2] In §4.3.1.2, we describe how to conduct these proofs efficiently for the practical blind IBE constructions we consider.

[3] In practice, it is sufficient to commit to a collision-resistant hash of the ciphertext vector, which will improve efficiency.

$\underline{\mathsf{S}_\mathsf{I}(M_1,\ldots,M_N),\mathsf{S}_\mathsf{T}()}$	$\underline{\mathsf{R}_\mathsf{I}(),\mathsf{R}_\mathsf{T}(\sigma_1,\ldots,\sigma_k)}$
Sender and Receiver agree on parameters for a commitment scheme.	
1. Generate $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa, c(\kappa))$.	
2. For $j = 1,\ldots,N$, set $C_j \leftarrow \mathsf{Encrypt}(params, j, M_j)$.	
3. Compute $(\mathcal{C},\mathcal{D}) \leftarrow \mathsf{Commit}\,(H(C_1,\ldots,C_N))$.	
4. Send $(params,\mathcal{C})$ to Receiver.	
5. Conduct $ZKPoK\{(msk) : (params, msk) \in \mathsf{Setup}(1^\kappa, c(\kappa))\}$.	
	6. If the proof does not verify, abort.
For $i = 1,...,k$, run $\mathsf{BlindExtract}(\mathcal{P}(params, msk),\mathcal{U}(params,\sigma_i)) \rightarrow (\cdot, sk_{\sigma_i})$.	
Following the k^{th} extraction:	
1. Send the ciphertexts $(C_1,\ldots,C_N)$ to the Receiver.	
2. Conduct $ZKPoK\{(\mathcal{D}) : \mathsf{Decommit}\,(\langle C_1,\ldots,C_N\rangle,\mathcal{C},\mathcal{D}) = 1\}$.	
	3. If the proof does not verify, or for any i the output of $\mathsf{IsValid}(params, i, C_i) \neq 1$, then abort and output $\perp$.
	4. For $i = 1$ to k: If $\mathsf{BlindExtract}$ on σ_i failed, set $M'_{\sigma_i} \leftarrow \perp$; else, set M'_{σ_i} to the value $\mathsf{Decrypt}(params, \sigma_i, sk_{\sigma_i}, C_{\sigma_i})$.
Output $(msk,\mathcal{D})$	Output $(params,\mathcal{C}, C_1,\ldots,C_N, M'_{\sigma_1},\ldots,M'_{\sigma_k})$.

Figure 4.1: OT_k^N from any committing blind IBE, with input messages $M_1,\ldots,M_N \in \mathcal{M}$. We present the $\mathsf{S}_\mathsf{I},\mathsf{R}_\mathsf{I},\mathsf{S}_\mathsf{T},\mathsf{R}_\mathsf{T}$ algorithms in a single protocol flow.

least computationally binding (against an adversarial Sender), and also hiding (against an adversarial Receiver). Thus, these parameters may be generated by a trusted party, or by one of the parties in the protocol. For instance, when using the Pedersen commitment scheme [Ped92], it is sufficient to have the Receiver generate the commitment parameters at the start of the protocol.

4.2.1.1 Security Analysis

Theorem 4.2.1 (Full-simulation Security of the OT_k^N Scheme) *If Π is a* committing *blind IBE scheme secure in the sense of definition 4.1.5, and* $(\mathsf{CSetup},\mathsf{Commit},\mathsf{Decommit})$ *is a secure commitment scheme, then the OT_k^N protocol of figure 4.1 is sender-secure and receiver-secure in the full-simulation model.*

We now prove Theorem 4.2.1. Note that when Π is instantiated using the blind IBE schemes

from section 4.3 and the Pedersen commitment scheme [Ped92], we obtain a OT_k^N scheme secure under the Decisional Bilinear Diffie Hellman (DBDH) assumption.[4] The proof is divided into two parts, one to show that the OT scheme meets the *sender security* property, and a second to show *receiver security*.

Proof of Sender Security (Theorem 4.2.1). For any real-world cheating receiver $\hat{\mathsf{R}}$ we can construct an ideal-world receiver $\hat{\mathsf{R}}'$ such that the "real" and "ideal" experiments are computationally indistinguishable. More formally, define some set of negligible functions where $\nu_n(\cdot)$ indicates the n^{th} function. Then $\forall$ p.p.t. D:

$$\begin{aligned}\mathbf{Pr}\left[\, D(\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N,k,M_1,\ldots,M_N,\Sigma)) = 1 \,\right] - \\ \mathbf{Pr}\left[\, D(\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N,k,M_1,\ldots,M_N,\Sigma)) = 1 \,\right] \leq \nu_1(\kappa)\end{aligned}$$

To describe the construction of $\hat{\mathsf{R}}'$ we will begin with the real-world experiment, and modify elements via a series of games until we arrive at the ideal-world experiment. For notational convenience, let $\mathbf{Adv}\left[\,\mathbf{Game\ i}\,\right]$ be D's advantage in distinguishing the output of $\mathbf{Game\ i}$ from the $\mathbf{Real}$ distribution.

$\mathbf{Game\ 0}$. In this game the honest real-world sender $\mathbf{S}(\mathbf{M_1},\ldots,\mathbf{M_N})$ interacts with the real-world cheating receiver $\hat{\mathsf{R}}$. Clearly $\mathbf{Adv}\left[\,\mathbf{Game\ 0}\,\right] = 0$.

$\mathbf{Game\ 1}$. In this game, we employ the knowledge extractor for $\mathsf{BlindExtract}$ to extract from $\hat{\mathsf{R}}$ each of the identities $(\sigma_1,\ldots,\sigma_k)$ from the k sequential executions of the $\mathsf{BlindExtract}$ protocol.[5] If the knowledge extractor fails for any execution, set $\hat{\mathsf{R}}'$'s output to $\perp$. Let $\mathbf{Pr}\left[\,error\,\right]$ be the probability that the knowledge extractor fails during any given execution, then $\mathbf{Adv}\left[\,\mathbf{Game\ 1}\,\right] - \mathbf{Adv}\left[\,\mathbf{Game\ 0}\,\right] \leq (k \cdot \mathbf{Pr}\left[\,error\,\right])$. Since Π is *leak-free*, it must hold that $k \cdot \mathbf{Pr}\left[\,error\,\right] \leq \nu_2(\kappa)$, and thus, $\mathbf{Adv}\left[\,\mathbf{Game\ 1}\,\right] \leq \nu_2(\kappa)$.

$\mathbf{Game\ 2}$. In this game, we replace the proof-of-knowledge:

$$PoK\{(\mathcal{D}) : \mathsf{Decommit}\,(H(C_1,\ldots,C_N),\mathcal{C},\mathcal{D}) = 1\}$$

with a simulated proof of the same statement. By the zero-knowledge property of this proof, D's advantage in distinguishing the simulated proof from a correctly-generated proof must be at most negligible in κ. Therefore, $\mathbf{Adv}\left[\,\mathbf{Game\ 2}\,\right] - \mathbf{Adv}\left[\,\mathbf{Game\ 1}\,\right] \leq \nu_3(\kappa)$.

[4] The Pedersen scheme is secure under the Discrete Logarithm Assumption, which is implied by DBDH.

[5] Note that the leak-freeness definition implies that for every adversary adversary $\mathcal{A}$, there exists a simulator that queries the trusted party and produces indistinguishable output (including the extracted identities). We can use this simulator as a black box to construct our extractor, which must fail with at most negligible probability.

Game 3. In this game, the commitment $\mathcal{C}$ is replaced with a commitment to a random value. We define the probability that D distinguishes this condition as $\mathbf{Adv}[dec]$, and note that $\mathbf{Adv}[dec] \leq \nu_4(\kappa)$ by the hiding property of a secure commitment scheme. Therefore, $\mathbf{Adv}[\mathbf{Game\ 3}] - \mathbf{Adv}[\mathbf{Game\ 2}] \leq \nu_4(\kappa)$.

Game 4. In the final game, we alter the ciphertext vector $(C_1, \ldots, C_N)$ to produce a new vector $(C'_1, \ldots, C'_N)$ as follows: for $j = 1, \ldots, N$ if $j \notin (\sigma_1, \ldots, \sigma_k)$, set $C'_j \leftarrow \mathsf{Encrypt}(params, j, M' \xleftarrow{\$} \mathcal{M})$, and otherwise set $C'_j \leftarrow C_j$. By Lemma 5.2.9 below, the security properties of Π imply that $\mathbf{Adv}[\mathbf{Game\ 3}] - \mathbf{Adv}[\mathbf{Game\ 2}] \leq \nu_5(\kappa)$.

Summing the differences between the above games, it is clear that $\mathbf{Adv}[\mathbf{Game\ 4}]$ is negligible, and therefore no p.p.t. algorithm can distinguish the distribution of **Game 4** from **Game 0**. The ideal-world receiver $\hat{\mathsf{R}}'$ is an algorithm that runs $\hat{\mathsf{R}}$, and (1) issues it a random commitment, (2) extracts the values $(\sigma_1, \ldots, \sigma_k)$ from $\hat{\mathsf{R}}$'s executions of the BlindExtract protocol, (3) transmits these values to the trusted party T to receive $(M_{\sigma_1}, \ldots, M_{\sigma_k})$. Next, (4) for $i = 1, \ldots, k$, $\hat{\mathsf{R}}'$ sets $C'_{\sigma_i} = \mathsf{Encrypt}(params, \sigma_i, M_{\sigma_i})$ and each of the remaining ciphertexts to encryptions of a random message, and (5) sends $(C'_1, \ldots, C'_N)$ to $\hat{\mathsf{R}}$ along with a simulated proof of knowledge of the opening of the commitment.

Lemma 4.2.2 (Indistinguishability of Ciphertexts)
$\mathbf{Adv}[\mathbf{Game\ 4}] - \mathbf{Adv}[\mathbf{Game\ 3}] \leq \nu_5(\kappa)$ if Π is a blind IBE scheme secure in the sense of definition 4.1.4 (or definition 4.1.5).

Proof sketch. We show, via a series of hybrids, that no p.p.t. D distinguishes **Game 4** from **Game 3** except with negligible probability, as long as (1) the PoK of msk is zero-knowledge, and (2) Π is both leak-free and IND-sID-CPA-secure.

Zero-Knowledge and Leak-freeness. Consider a pair of hybrid games. Hybrid 0 is identical to **Game 3**, except that $\mathbf{S}$ *simulates* the PoK of msk. Clearly the zero-knowledge property of Π ensures that this hybrid is indistinguishable from **Game 3**. Hybrid 1 extends the previous hybrid as follows: $\mathbf{S}$ does not run Setup, but is instead given $params$ and an oracle $O_{params,msk}(\cdot)$ with which it may run the Extract protocol. Each time $\hat{\mathsf{R}}$'s initiates the BlindExtract protocol with $\mathbf{S}$, use the knowledge extractor for BlindExtract to obtain the identity id that $\hat{\mathsf{R}}$ is attempting to extract, then use $O_{params,msk}$ to extract sk_{id} and simulate a correct response to $\hat{\mathsf{R}}$. Note that by the definition of Leak-freeness, this hybrid must be indistinguishable from **Game 3**.

IND-sID-CPA security. Now assume by contradiction that some D distinguishes hybrid 1 from **Game 4**. If this is the case, then we show how to construct an adversary $\mathcal{A}$ that wins the IND-sID-CPA game against Π with non-negligible advantage. This proof is just a standard hybrid argument, but we provide it for completeness. Beginning with hybrid 1 from above, we describe an additional $(N - k)$ hybrids, where the final hybrid is **Game 4**.

Each hybrid j is identical to hybrid $(j-1)$ except that the distribution of the ciphertext vector is different at some position which we denote by ℓ. Specifically, in hybrid $(j-1)$, the ciphertext C_ℓ encrypts M_ℓ, while in hybrid j the ciphertext C_ℓ encrypts a random message. If D distinguishes the first and last hybrids with non-negligible probability, then clearly there must exist a D' that distinguishes some pair of consecutive hybrids $(j, j-1)$ with non-negligible probability.

Consider these two hybrids, and let ℓ be the position at which the ciphertext vectors differ. The IND-sID-CPA adversary $\mathcal{A}$ outputs $id^* = \ell$ and receives $params$. It then runs D' (which controls $\hat{\mathsf{R}}$) and conducts the initial stage of the OT protocol as in hybrid 1 (this involves queries to a key extraction oracle as in the IND-sID-CPA game). Select $M^* \xleftarrow{\$} \mathcal{M}$ and output (M_ℓ, M^*) to obtain the challenge ciphertext C^*. Construct a ciphertext vector $\vec{C}$ with the correct distribution for hybrid $(j-1)$ (by encrypting either a real message or a random message at each position as appropriate)— however, at the ℓ^{th} position, set $C_\ell \leftarrow C^*$. Send $\vec{C}$ to $\hat{\mathsf{R}}$ and complete the protocol. Let b' be D''s output. Output b'.

Note that when C^* encrypts M_ℓ, D's view is that of hybrid $j-1$, and when C^* encrypts M^*, D's view is that of hybrid j. Thus, if D outputs 1 with probability α in the first, case, and probability β in the second, then $\mathcal{A}$ wins the IND-sID-CPA game with non-negligible advantage $\frac{|\beta-\alpha|}{2}$.

By the hybrid argument, therefore, D's advantage must be negligible for each of the hybrids, and thus by summation of all hybrids we obtain $\mathbf{Adv}\,[\,\mathbf{Game\ 4}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 3}\,] \leq \nu_5(\kappa)$.

□

□

Proof of Receiver Security (Theorem 4.2.1). For any real-world cheating sender $\hat{\mathsf{S}}$ we can construct an ideal-world sender $\hat{\mathsf{S}}'$ such that no p.p.t. algorithm D can distinguish the distributions $\mathbf{Real}_{\hat{\mathsf{S}},\mathbf{R}}(N, k, M_1, \ldots, M_N, \Sigma)$ and $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \Sigma)$. We arrive at the ideal-world sender via a series of games. Again let $\mathbf{Adv}\,[\,\mathbf{Game\ i}\,]$ be D's advantage in distinguishing the output of $\mathbf{Game\ i}$ from the $\mathbf{Real}$ distribution.

> $\mathbf{Game\ 0}$. In this game the honest real-world receiver $\mathbf{R}$ interacts with the real-world cheating sender $\hat{\mathsf{S}}$. Clearly $\mathbf{Adv}\,[\,\mathbf{Game\ 0}\,] = 0$.
>
> $\mathbf{Game\ 1}$. In this game, the simulator uses the knowledge extractor for $PoK\{msk : (params, msk) \in \mathsf{Setup}(1^\kappa, c(\kappa))\}$ to extract msk. If the extractor fails or outputs an invalid msk, set $\mathbf{R}$'s output to $\perp$. Since this extractor fails with probability negligible in κ, then $\mathbf{Adv}\,[\,\mathbf{Game\ 1}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 0}\,] \leq \nu_1(\kappa)$.
>
> $\mathbf{Game\ 2}$. In this game, the simulator replaces the k executions of BlindExtract with executions on random identities $(\sigma'_1, \ldots, \sigma'_k)$. If the i^{th} execution fails, record $b_i \leftarrow 0$, otherwise set $b_i \leftarrow 1$. By Lemma 4.2.3, $\mathbf{Adv}\,[\,\mathbf{Game\ 2}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 1}\,] \leq (k \cdot \nu_2(\kappa))$ if Π is selective-failure blind.

Game 3. In this game, this simulator verifies that for all $j \in (\sigma'_1, \ldots, \sigma'_k)$, the condition $\mathsf{Decrypt}(sk_{\sigma_j}, C_{\sigma_j}) = \mathsf{Decrypt}(\mathsf{Extract}(msk, \sigma_j), C_{\sigma_j})$ holds. If this does not hold, then set $\mathbf{R}$'s output to $\perp$. By Lemma 4.2.4, $\mathbf{Adv}[\mathbf{Game\ 2}] - \mathbf{Adv}[\mathbf{Game\ 1}] \leq \nu_3(\kappa)$ if Π is a *committing* blind IBE.

Summing the differences beteen the above games, it is clear that $\mathbf{Adv}[\mathbf{Game\ 3}] - \mathbf{Adv}[\mathbf{Game\ 0}]$ is negligible, and therefore no p.p.t. algorithm can distinguish the distribution of $\mathbf{Game\ 3}$ from $\mathbf{Game\ 0}$. The ideal-world sender $\hat{\mathsf{S}}'$ is an algorithm that performs all of the changes between the games above, and on learning $(M_1, \ldots, M_N, b_1, \ldots, b_k)$ transmits these values to the trusted party T.

Lemma 4.2.3 (Blindness of Extractions)
$\mathbf{Adv}[\mathbf{Game\ 2}] - \mathbf{Adv}[\mathbf{Game\ 1}] \leq (k \cdot \nu_2(\kappa))$ if Π is selective-failure blind in the sense of definition 4.1.4.

Proof sketch. By contradiction, let D be a p.p.t. distinguisher that controls $\hat{\mathsf{S}}$ and distinguishes the distributions of $\mathbf{Game\ 2}$ and $\mathbf{Game\ 1}$ with advantage $> k \cdot \nu_2(\kappa)$. This implies that D can distinguish two experiments that differ only in the distribution of extracted identities. We conduct our proof using a standard hybrid argument: beginning with $\mathbf{Game\ 1}$ define a series of k intermediate hybrids during each of which a single execution of $\mathsf{BlindExtract}$ is altered from using a "real" identity σ_j to some random $\sigma'_j \xleftarrow{\$} [1, N]$. The last hybrid is equivalent to $\mathbf{Game\ 2}$. If D successfully distinguishes the first and last hybrids, then $\exists D', j$ such that D' distinguishes hybrid $(j-1)$ from hybrid j with maximal probability $> \nu_2(\kappa)$. We use D' to construct an adversary $\mathcal{A}$ with non-negligible advantage in winning the selective-failure blindness game against Π.

$\mathcal{A}$ runs D' and conducts the protocol with $\hat{\mathsf{S}}$ as in $\mathbf{Game\ 1}$ up to the point where $\mathbf{R}$ initiates the $\mathsf{BlindExtract}$ protocol. At all but the ℓ^{th} execution of $\mathsf{BlindExtract}$, $\mathcal{A}$ selects the appropriate identity distribution (σ_k or σ'_k) for hybrid $(j-1)$. At the ℓ^{th} execution, $\mathcal{A}$ selects $\sigma'_\ell \xleftarrow{\$} [1, N]$ and outputs $(params, \sigma_\ell, \sigma'_\ell)$ as the first move of the selective-failure blindness game. Now $\mathcal{A}$ forwards the messages from the first oracle, $\mathcal{U}_b$ directly to $\hat{\mathsf{S}}$, returning $\hat{\mathsf{S}}$'s responses until the $\mathsf{BlindExtract}$ protocol run is complete. When D' ultimately outputs a bit b', $\mathcal{A}$ outputs b' as its guess.

Note that when $b = 0$, the ℓ^{th} extraction is conducted on σ_ℓ, and thus the game has the correct distribution for hybrid $(j-1)$. When $b = 1$, the extraction is conducted on random σ'_ℓ and thus the game has the correct distribution for hybrid j. If D' outputs 1 with probability α when presented with hybrid $(j-1)$ and probability β when presented with hybrid j, then $\mathcal{A}$ guesses correctly and wins the selective-failure blindness game with probability $\frac{|\beta - \alpha|}{2}$. If we assume that $|\beta - \alpha| > \nu_2(\kappa)$ then $\mathcal{A}$ wins with non-negligible advantage. Since contradicts our assumption about Π, then D' succeeds with probability $\leq \nu_2(\kappa)$ and thus D succeeds with probability $\leq k \cdot \nu_2(\kappa)$.

We conclude our sketch by observing that $\hat{\mathsf{S}}$ commits to the ciphertext vector in the first stage of the protocol. Assuming that $H(\cdot)$ is collision resistant, and the commitment scheme is binding, then $\hat{\mathsf{S}}$'s choice of $(C_1, \ldots, C_N)$ is independent of all subsequent actions including executions of BlindExtract. □

Lemma 4.2.4 (Committing IBE) $\mathbf{Adv}\,[\,\mathbf{Game\ 3}\,] - \mathbf{Adv}\,[\,\mathbf{Game\ 2}\,] \leq \nu_3(\kappa)$ if Π is committing in the sense of definition 4.1.5.

Proof sketch. Let D be a p.p.t. distinguisher that distinguishes the distributions of $\mathbf{Game\ 3}$ and $\mathbf{Game\ 2}$ with non-negligible advantage. This implies that for some j it is the case that with non-negligible probability $\hat{\mathsf{S}}$ (in cooperation with D) outputs at least one ciphertext C_{σ_j} such that $\mathsf{Decrypt}(sk_{\sigma_j}, C_{\sigma_j}) \neq \mathsf{Decrypt}(\mathsf{Extract}(msk, \sigma_j), C_{\sigma_j})$, while simultaneously the statement $\mathsf{IsValid}(params, \sigma_j, C_{\sigma_j}) = 1$ (since this condition is ensured by the protocol). Thus, by definition the algorithm $\hat{\mathsf{S}}$ must succeed in the game of definition 4.1.5 with non-negligible probability. Since Π is a committing IBE scheme, then we can bound D's advantage as $\leq \nu_3(\kappa)$. □

□

4.2.2 Adaptive $\mathsf{OT}^N_{k\times 1}$ in the Random Oracle Model

While our first protocol is efficient and full-simulation secure, it permits only *non-adaptive* queries. For many practical applications (*e.g.*, oblivious retrieval from a large database), we desire a protocol that supports an adaptive query pattern. We approach this goal by first proposing an efficient $\mathsf{OT}^N_{k\times 1}$ protocol secure in the random oracle model. The protocol, which we present in Figure 4.2, requires an IBE scheme with a super-polynomial message space (as in the constructions of §4.3), and has approximately the same efficiency as the construction with random oracles of Camenisch *et al.* [CNs07]. However, their construction requires unique blind signatures and the two known options due to Chaum [Cha82] and Boldyreva [Bol03] both require interactive complexity assumptions. When instantiated using the blind IBE schemes in §4.3, our protocols can be based on the DBDH assumption.

4.2.2.1 Security Analysis

Theorem 4.2.5 (Full-simulation Security of the $\mathsf{OT}^N_{k\times 1}$ Scheme) *If Π is a committing blind IBE scheme secure in the sense of 4.1.5, and $H(\cdot)$ is modeled as a random oracle, then the $\mathsf{OT}^N_{k\times 1}$ protocol of figure 4.2 is sender-secure and receiver-secure in the full-simulation model.*

We now sketch a proof of theorem 4.2.5. A nice feature of this proof is our ability to use the random oracle $H(\cdot)$ in place of the extractor for BlindExtract. We also note that when

$\underline{\mathsf{S}_\mathsf{I}(M_1, \ldots, M_N)}$	$\underline{\mathsf{R}_\mathsf{I}()}$
1. Select $(params, msk) \leftarrow \mathsf{Setup}(1^\kappa, c(\kappa))$.	
2. Select random $W_1, \ldots, W_N \in \mathcal{M}$, and for $j = 1, \ldots, N$ set:	
— $A_j \leftarrow \mathsf{Encrypt}(params, j, W_j)$	
— $B_j \leftarrow H(j \Vert W_j) \oplus M_j$	
— $C_j = (A_j, B_j)$	
3. Conduct $ZKPoK\{(msk) : (params, msk) \in \mathsf{Setup}(1^\kappa, c(\kappa))\}$.	
4. Send $(params, C_1, \ldots, C_N)$ to Receiver.	
	5. If the proof fails to verify or for any i $\mathsf{IsValid}(params, i, C_i) \neq 1$, abort and set $M'_{\sigma_1}, \ldots, M'_{\sigma_k} \leftarrow \perp$.
Output $S_0 = (params, msk)$	Output $R_0 = (params, C_1, \ldots, C_N)$
$\underline{\mathsf{S}_\mathsf{T}(S_{i-1})}$	$\underline{\mathsf{R}_\mathsf{T}(R_{i-1}, \sigma_i)}$
In the i^{th} transfer, $\mathbf{R}$ runs BlindExtract on identity σ_i, to obtain sk_{σ_i}.	
	1. If BlindExtract fails, then set M'_{σ_i} to $\perp$.
	2. Else set $t \leftarrow \mathsf{Decrypt}(params, \sigma_i, sk_{\sigma_i}, A_{\sigma_i})$ and set $M'_{\sigma_i} \leftarrow B_{\sigma_i} \oplus H(i \Vert t)$.
Output $S_i = S_{i-1}$	Output $R_i = (R_{i-1}, M'_{\sigma_i})$.

Figure 4.2: Adaptive $\mathsf{OT}^N_{k\times 1}$ from any committing blind IBE, with $M_1, \ldots, M_N \in \{0,1\}^n$. We model $H : \mathcal{M} \rightarrow \{0,1\}^n$ as a random oracle.

implemented with one of the IBE schemes in §4.3, the OT protocol is secure under the DBDH assumption.

Proof sketch. **Sender Security.** For any real-world cheating receiver $\hat{\mathsf{R}}$ we can construct an ideal-world receiver $\hat{\mathsf{R}}'$ such that no p.p.t. algorithm D can distinguish the distributions $\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \Sigma)$ and $\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \Sigma)$. $\hat{\mathsf{R}}'$ interacts with $\hat{\mathsf{R}}$ and the trusted party as follows. $\hat{\mathsf{R}}'$ first runs $\mathsf{Setup}(1^\kappa, c(\kappa))$ to generate the scheme parameters, proves knowledge of msk, and sends $(C_1, \ldots, C_N)$ formed by setting $(B_1, \ldots, B_N)$ to be random bitstrings and computing $(A_1, \ldots, A_N)$ as usual. $\hat{\mathsf{R}}'$ now simulates the random oracle $H : \mathcal{M} \rightarrow \{0,1\}^{|M_1|}$, observing $\hat{\mathsf{R}}$'s queries. Whenever $\hat{\mathsf{R}}$ calls $H(\cdot)$ on a value W_{σ_i} (for some $i \in [1, N]$), $\hat{\mathsf{R}}'$ queries the trusted party to obtain M_{σ_i}. If the trusted party outputs $\perp$, then $\hat{\mathsf{R}}'$ causes the BlindExtract protocol to fail. Otherwise, $\hat{\mathsf{R}}'$ now programs the random oracle so that $H(i \Vert W_{\sigma_i}) = B_{\sigma_i} \oplus M_{\sigma_i}$. If a p.p.t. D can distinguish the real and

ideal-world distributions then it must be the case that either (a) D breaks the IND-sID-CPA or Leak-Free security of the IBE scheme Π, or (b) the proof-of-knowledge on msk is not zero knowledge.

Receiver Security. Our proof of receiver security is almost identical to that of the non-adaptive OT protocol above. For any real-world cheating sender $\hat{\mathsf{S}}$ we can construct an ideal-world sender $\hat{\mathsf{S}}'$ such that no p.p.t. D can distinguish the distributions $\mathbf{Real}_{\hat{\mathsf{S}},\mathbf{R}}(N, k, M_1, \ldots, M_N, \Sigma)$ and $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathbf{R}'}(N, k, M_1, \ldots, M_N, \Sigma)$. $\hat{\mathsf{S}}'$ interacts with $\hat{\mathsf{S}}$ and the trusted party as follows. When $\hat{\mathsf{S}}$ proves knowledge of the value msk, use the appropriate knowledge extractor to obtain msk. Use msk to decrypt the ciphertext vector $(C_1, \ldots, C_N)$ as per the protocol, and transmit the resulting messages $(M_1, \ldots, M_N)$ to the trusted party T. At the i^{th} protocol round, run BlindExtract on a random identity σ_i'. If BlindExtract fails, send $b_i = 0$ to T, otherwise send $b_i = 1$. Based on the selective-failure blindness property of the IBE scheme Π, any failures in the BlindExtract protocol are independent of the values $(\sigma_1, \ldots, \sigma_k)$ actually extracted by an ideal-world honest receiver. If a p.p.t. D can distinguish the real and ideal-world distributions then it must be the case that either (a) $\hat{\mathsf{R}}$ breaks the selective-failure blindness property of Π, (b) Π is not committing, or (c) the extractor for msk failed.

□

4.2.3 A Note on Adaptive $\mathsf{OT}^N_{k\times 1}$ in the Standard Model

The random-oracle $\mathsf{OT}^N_{k\times 1}$ presented above is reasonably efficient both in terms of communication cost and round-efficiency. Ideally, we would like to construct a protocol of comparable efficiency in the standard model. We could construct an $\mathsf{OT}^N_{k\times 1}$ protocol by compiling k instances of the non-adaptive OT^N_k from §4.2.1. Each protocol round would consist of a 1-out-of-N instance of the protocol, with new IBE parameters and new a vector of ciphertexts $(C_1, \ldots, C_N)$. To ensure that each round is consistent with the previous rounds, the Sender would need to prove that the underlying plaintexts remain the same from round to round. This can be achieved using standard proof techniques, but is impractical for large values of k or N.

A better approach would be to modify the OT^N_k above to perform *blind decryption* of ciphertexts, rather than blindly extracting keys. Given such a protocol, we might be able to simulate the correct decryption of a ciphertext, opening it to the value of our choice. Unfortunately, the existing schemes are either CPA-secure (which is insufficient for or purposes) or secure under unreasonable assumptions about the plaintext distribution. However, we might achieve blind decryption by adapting some of our IBE-based techniques. In fact, several efficient transformations exist that allow for the conversion of IND-sID-CPA-secure IBE schemes into CCA-secure Public Key Encryption [CHK04, BMW05]. However, it seems quite difficult to produce blind decryption protocols from these schemes. Thus, we leave the development of an appropriate blind decryption protocol as an open problem.

Fortunately, there are alternative approaches to achieving $\mathsf{OT}^N_{k\times 1}$ in the standard model. In the next chapter, we will propose an very different approach to this problem that achieves *universally-composable* security without random oracles, under relatively stronger assumptions than those used in this chapter.

4.3 Efficient Instantiations of Blind IBE

In this section, we describe efficient BlindExtract protocols for: (1) the IND-sID-CPA-secure IBE due to Boneh and Boyen [BB04a], (2) the IND-ID-CPA-secure IBE proposed independently by Naccache [Nac05] and Chatterjee-Sarkar [CS05] which is a generalized version of Waters' IBE [Wat05], and (3) the *anonymous* IND-ID-CPAscheme of Boyen and Waters. Note that in §4.3.1.2 we will be adding some additional features to these IBE schemes; these are needed by the oblivious transfer protocols in §4.2.

4.3.1 BlindExtract protocols for the Boneh-Boyen and Waters schemes

Since these two of these schemes share a similar structure, we'll begin by describing their common elements.

Setup$(1^\kappa, c(k))$**:** Let $\gamma = (q, g, \mathbb{G}, \mathbb{G}_T, e)$ be the output of BMsetup(1^κ). Choose random elements $h, g_2 \in \mathbb{G}$ and a random value $\alpha \in \mathbb{Z}_q$. Set $g_1 = g^\alpha$. Finally, select a function $F : \mathcal{I} \to \mathbb{G}$ that maps identities to group elements. (The descriptions of F and $\mathcal{I}$ will be defined specific to the schemes below.) Output $params = (\gamma, g, g_1, g_2, h, F)$ and $msk = g_2^\alpha$.

Extract**:** Identity secret keys are of the form: $sk_{id} = (d_0, d_1) = (g_2^\alpha \cdot F(id)^r, g^r)$, where $r \in \mathbb{Z}_q$ is randomly chosen by the master authority. Note that the correctness of these keys can be publicly verified using a test described below.

Encrypt$(params, id, M)$**:** Given an identity $id \in \mathcal{I}$, and a message $M \in \mathbb{G}_T$, select a random $s \in \mathbb{Z}_q$ and output the ciphertext $C = (e(g_1, g_2)^s \cdot M, g^s, F(id)^s)$.

Decrypt$(params, id, sk_{id}, c_{id})$**:** On input a decryption key $sk_{id} = (d_0, d_1) \in \mathbb{G}^2$ and a ciphertext $C = (X, Y, Z) \in \mathbb{G}_T \times \mathbb{G}^2$, output $M = X \cdot e(Z, d_1)/e(Y, d_0)$.

Next, we'll describe the precise format of the secret keys sk_{id} and corresponding BlindExtract protocols for particular IBEs.

4.3.1.1 A BlindExtract Protocol for the Boneh-Boyen scheme

In the Boneh-Boyen IBE [BB04a], $\mathcal{I} \subseteq \mathbb{Z}_q$ and the function $F : \mathcal{I} \to \mathbb{G}$ is defined as $F(id) = h \cdot g_1^{id}$. A secret key for identity id, where $r \in \mathbb{Z}_q$ is random, is:

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

The protocol $\mathsf{BlindExtract}(\mathcal{P}(params, msk), \mathcal{U}(params, id))$ is described in Figure 4.3. Recall that $\mathcal{U}$ wants to obtain sk_{id} without revealing id, and $\mathcal{P}$ wants to reveal no more than sk_{id}. Let Π_{BB} be the blind IBE that combines algorithms Setup, Encrypt, Decrypt with the protocol BlindExtract in Figure 4.3.

$\underline{\mathcal{P}(params, msk)}$	$\underline{\mathcal{U}(params, id)}$
	1. Choose $y \xleftarrow{\$} \mathbb{Z}_q$.
	2. Compute $h' \leftarrow g^y g_1^{id}$ and send h' to $\mathcal{P}$.
	3. Execute $WIPoK\{(y, id) : h' = g^y g_1^{id}\}$.
4. If the proof fails to verify, abort.	
5. Choose $r \xleftarrow{\$} \mathbb{Z}_q$.	
6. Compute $d_0' \leftarrow g_2^{\alpha} \cdot (h'h)^r$.	
7. Compute $d_1' \leftarrow g^r$.	
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 $z \xleftarrow{\$} \mathbb{Z}_q$; otherwise, output $\perp$ and abort.
	11. Compute $d_0 \leftarrow (d_0'/(d_1')^y) \cdot F(id)^z$ and $d_1 \leftarrow d_1' \cdot g^z$.
	12. Output $sk_{id} = (d_0, d_1)$.

Figure 4.3: A BlindExtract protocol for the Boneh-Boyen IBE.

Theorem 4.3.1 *Under the DBDH assumption, blind IBE Π_{BB} is secure (according to Definition 4.1.4); i.e.,* BlindExtract *is leak-free, selective-failure blind, and committing.*

*Proof.*We will first address the properties of IND-sID-CPA and selective-failure blindness. Further below, we will show that the proposed scheme meets the definition of Committing IBE.

We begin by observing that the Setup, Encrypt, Decrypt algorithms of Π_{BB} are identical to the original Boneh-Boyen (H)IBE [BB04a] instantiated with only one level. Thus, when Π_{BB} is considered with the key extraction algorithm of [BB04a], it is IND-sID-CPA-secure by the original proof of security. To prove the remaining properties, we must show that the BlindExtract protocol in Figure 4.3 is both leak free and selective-failure blind. We

begin with leak freeness, which requires the existence of an efficient simulator $\mathcal{S}$ such that no efficient distinguisher D can distinguish Game Real (where $\mathcal{A}$ is interacting with an honest $\mathcal{P}$ running the BlindExtract protocol) from Game Ideal (where the ideal adversary $\mathcal{S}$ is given access to a trusted party executing the ideal Extract protocol).

We describe the ideal adversary $\mathcal{S}$ as follows:

1. On input $params$ from the trusted party, $\mathcal{S}$ hands $params$ to a copy of $\mathcal{A}$ it runs internally.
2. Each time $\mathcal{A}$ engages $\mathcal{S}$ in a BlindExtract protocol, $\mathcal{S}$ behaves as follows. In the first message of the protocol, $\mathcal{A}$ must send to $\mathcal{S}$ a value h' and prove knowledge of values (y, id) such that $h' = g^y \cdot g_1^{id}$. If the proof fails to verify, $\mathcal{S}$ aborts. Since this proof of knowledge is implemented using the *extractable* techniques mentioned in §3.4, $\mathcal{S}$ can efficiently extract the values (y, id).
3. Next, $\mathcal{S}$ submits id to the trusted party, who returns the valid secret key for this identity $sk_{id} = (d_0, d_1) = (g_2^{\alpha} \cdot F(id)^r, g^r)$ for some random $r \in \mathbb{Z}_q$.
4. Finally, $\mathcal{S}$ computes the pair $(d'_0, d'_1) = (d_0 \cdot d_1^y, d_1)$ and returns these values to $\mathcal{A}$.

Observe that the responses of $\mathcal{S}$ are always correctly formed (as $\mathcal{A}$ can verify) and drawn from the same distribution as those of $\mathcal{P}$. Thus, Game Real and Game Ideal are indistinguishable to both $\mathcal{A}$ and D. We also note (as above) that the identity id being requested by $\mathcal{A}$ is efficiently *extractable* (by an extractor with special rewind capabilities not available to $\mathcal{P}$).

Next, we turn out attention to selective-failure blindness for protocol BlindExtract $= (\mathcal{P}, \mathcal{U})$. Here $\mathcal{A}$ outputs $params$ and two identities $id_0, id_1 \in \mathcal{I}$. Then a random bit b is chosen. Next, $\mathcal{A}$ is given black-box access to two oracles $\mathcal{U}(params, id_b)$ and $\mathcal{U}(params, id_{b-1})$. The $\mathcal{U}$ algorithms conduct the BlindExtract protocol (with $\mathcal{A}$ playing the role of $\mathcal{P}$), and 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 $sk_{b-1} \neq \perp$ then $\mathcal{A}$ receives $(\perp, \varepsilon)$. If $sk_b \neq \perp$ and $sk_{b-1} = \perp$ then $\mathcal{A}$ receives $(\varepsilon, \perp)$. If $sk_b = \perp$ and $sk_{b-1} = \perp$ then $\mathcal{A}$ receives $(\perp, \perp)$. $\mathcal{A}$ then tries to predict b, which we want to argue he cannot do with non-negligible advantage over guessing.

First, we observe that in this protocol, $\mathcal{U}$ speaks first and sends to $\mathcal{A}$ a value h' uniformly distributed in $\mathbb{G}$ and then performs a zero-knowledge proof of knowledge $PoK\{(y, id) : h' = g^y \cdot g_1^{id}\}$. Suppose that $\mathcal{A}$ runs one or both of his oracles up to this point. Now, it is $\mathcal{A}$'s turn to speak, and at this point, his views so far are computationally indistinguishable. Let's assume that $\mathcal{A}$ must now return two values $(d'_0, d'_1) \in \mathbb{G}^2$ to the first oracle. Suppose $\mathcal{A}$ chooses this pair using any strategy he wishes. At the point $\mathcal{A}$ fixes on two values, he is able to *predict* the output sk_i of these oracles $\mathcal{U}(params, id_b)$ with non-negligible advantage as follows:

1. $\mathcal{A}$ checks if $e(g_1, g_2) \cdot e(d'_1, h' \cdot h) = e(d'_0, g)$ holds. If the test fails, record $sk_0 \leftarrow \perp$.
2. Next, $\mathcal{A}$ choses any two values $(d'_0, d'_1) \in \mathbb{G}^2$ for the second oracle, performs the same check and, in the event of failure, records $sk_1 \leftarrow \perp$.

3. If either test failed, then: if $sk_0 = \perp$ and $sk_1 \neq \perp$, output $(\perp, \varepsilon)$. If $sk_0 \neq \perp$ and $sk_1 = \perp$, output $(\varepsilon, \perp)$. If both tests failed, output $(\perp, \perp)$.
4. If both test succeeded, then: $\mathcal{A}$ initiates BlindExtract with itself on (id_0, id_1) (playing the roles of $\mathcal{U}$ and $\mathcal{P}$). If either protocol run fails, abort.[6] Otherwise output the returned keys (sk_0, sk_1).

This prediction is correct, because $\mathcal{A}$ is performing the same check as the honest $\mathcal{U}$, and when both tests succeed, outputting a pair of valid secret keys obtained via $\mathsf{BlindExtract}(params, id)$, as does $\mathcal{U}$. But at a higher-level, note that if $\mathcal{A}$ is able to predict the final output of its oracles accurately, then $\mathcal{A}$'s advantage in distinguishing $\mathcal{U}(params, id_b)$ and $\mathcal{U}(params, id_{b-1})$ is the same without this final output. Thus, all of $\mathcal{A}$'s advantage must come from distinguishing the earlier messages of the oracles. Since these oracles only send one uniformly random value $h' \in \mathbb{G}$ and then perform a zero-knowledge proof of knowledge about the representation of h' with respect to public values, we know from the security of the underlying proof that $\mathcal{A}$ cannot distinguish between them with non-negligible probability.

□

4.3.1.2 A BlindExtract Protocol for the Waters scheme

In the generalized version of Waters' IBE [Wat05], proposed independently by Naccache [Nac05] and Chatterjee and Sarkar [CS05], the identity space $\mathcal{I}$ is the set of bit strings of length N, where N is polynomial in κ, represented by n blocks of ℓ bits each. The function $F : \{0,1\}^N \to \mathbb{G}$ is defined as $F(id) = h \cdot \prod_{j=1}^{n} u_j^{a_j}$, where each $u_j \in \mathbb{G}$ is randomly selected by the master authority and each a_j is an ℓ-bit segment of id. Naccache discusses practical IBE deployment with $N = 160$ and $\ell = 32$ [Nac05]. A secret key for identity id, where $r \in \mathbb{Z}_q$ is random, is:

$$sk_{id} = (d_0, d_1) = (g_2^{\alpha} \cdot F(id)^r, g^r) = (g_2^{\alpha} \cdot (h \cdot \prod_{j=1}^{n} u_j^{a_j})^r, g^r).$$

The protocol $\mathsf{BlindExtract}(\mathcal{P}(params, msk), \mathcal{U}(params, id))$ is described in Figure 4.4. Line 4 of the protocol uses a range proof (*e.g.*, $0 \leq a_i < 2^{\ell}$) that can be performed exactly or, by shortening each a_i by a few bits, can be done at almost no additional cost [CFT98, CM99, Bou00]. Let Π_{Waters} be the blind IBE that combines Setup, Encrypt, Decrypt with the BlindExtract protocol described above.

Theorem 4.3.2 *Under the DBDH assumption, blind IBE* Π_{Waters} *is secure (according to Definition 4.1.4); i.e.,* BlindExtract *is both leak-free and selective-failure blind.*

[6] Note that $\mathcal{A}$ only reaches this step if $\mathcal{U}$'s two previous executions of the protocol have succeeded. If that event occurs with non-negligible probability, then $\mathcal{A}$ successfully obtains (sk_0, sk_1) with non-negligible probability.

$\mathcal{P}(params, msk)$	$\mathcal{U}(params, id)$
	1. Parse id into ℓ-bit chunks $(a_1, \ldots, a_n)$.
	2. Choose $y \xleftarrow{\$} \mathbb{Z}_q$.
	3. Compute $h' \leftarrow g^y \cdot \prod_{j=1}^{n} u_j^{a_j}$. Send h' to $\mathcal{P}$.
	4. Execute $WIPoK\{(y, a_1, \ldots, a_n) :$ $h' = g^y \cdot \prod_{j=1}^{n} u_j^{a_j} \wedge 0 \leq a_i < 2^\ell,$ for $i = 1$ to $n\}$
5. If the proof fails to verify, abort.	
6. Choose $r \xleftarrow{\$} \mathbb{Z}_q$.	
7. Compute $d_0' \leftarrow g_2^\alpha \cdot (h'h)^r$.	
8. Compute $d_1' \leftarrow g^r$.	
9. Send (d_0', d_1') to $\mathcal{U}$.	
	10. Check that $e(g_1, g_2) \cdot e(d_1', h'h) = e(d_0', g)$.
	11. If the check passes, choose $z \xleftarrow{\$} \mathbb{Z}_q$; otherwise, output $\perp$ and abort.
	12. Compute $d_0 \leftarrow (d_0'/(d_1')^y) \cdot F(id)^z$ and $d_1 \leftarrow d_1' \cdot g^z$.
	13. Output $sk_{id} = (d_0, d_1)$.

Figure 4.4: A BlindExtract protocol for the generalized Waters IBE.

Proof sketch. This proof follows the outline of the proof of Theorem 4.3.1 almost identically. Again we observe that IND-ID-CPA security can be shown via the original proof by Naccache [Nac05]. To satisfy leak freeness, the simulator $\mathcal{S}$ operates exactly as before: starting up an internal copy of $\mathcal{A}$ in step (1), extracting the values (y, id) from $\mathcal{A}$ in step (2), querying the trusted party for $sk_{id} = (d_0, d_1) \leftarrow \mathsf{Extract}(msk, id)$ in step (3), and returning the pair $(d_0', d_1') = (d_0 \cdot d_1^y, d_1)$ to $\mathcal{A}$ in step (4). Although the internal structure of the secret keys in the Naccache-Waters IBE differ from those of the Boneh-Boyen IBE, the key observation here is that $\mathcal{S}$ doesn't need to know anything about this structure to compute the correct response in step (4).

To satisfy selective-failure blindness, we first observe that the prediction of $\mathcal{U}$'s final output is done exactly as before. Thus, $\mathcal{A}$ must be able to distinguish the oracles after seeing only a value h' again uniformly distributed in $\mathbb{G}$ and a zero-knowledge proof of knowledge about the representation of h' with respect to public values. We conclude that this advantage must be negligible.

We conclude by noting that the argument from above can be used (unchanged) to show that this scheme is a committing IBE. □

The Committing property. In the case of blind IBE schemes Π_{BB} and Π_{Waters}, we can implement the check $\mathsf{IsValid}(params, id, C)$ by first verifying the group parameters γ are valid (see [CCS07]), then verifying that for any $params$ and $C = (X, Y, Z)$, all the values are in the correct groups and the following relation holds:

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

Recall that the *correctness* property for the IsValid algorithm is that it outputs 1 for all honestly-generated parameters and ciphertexts. From the description of Π_{BB} and Π_{Waters}, it is easy to see that IsValid is correct.

Theorem 4.3.3 *Combined with the* IsValid *algorithm defined above, both* Π_{BB} *and* Π_{Waters} *are committing blind IBE schemes (in the sense of definition 4.1.5).*

Proof sketch. For the purposes of this sketch, we will assume that all key extraction is performed via the BlindExtract protocol. Recall that in both schemes $params = (\gamma, g, g_1, g_2, h, F)$, $msk = g_2^{\alpha}$. Then for any message $M \in \mathbb{G}_T$, identity $id \in \mathcal{I}$ it holds that $\exists s, r \in \mathbb{Z}_q$ such that well-formed ciphertexts and keys can be expressed as follows:

$$C_{id} = (X, Y, Z) = (e(g_1, g_2)^s \cdot M, g^s, F(id)^s) \tag{4.1}$$

$$sk_{id} = (d_0, d_1) = (g_2^{\alpha} \cdot F(id)^r, g^r) \tag{4.2}$$

In both Π_{BB} and Π_{Waters}, the BlindExtract protocol includes a correctness check on the returned secret key. When the group parameters γ are valid, this check ensures that the user's output will either be $\perp$, or a key of the form shown in equation 4.2 above.[7] Similarly, the Decrypt algorithm includes a validity check to ensure that (a) the group parameters γ are correct (this check may be probabilistic, but is inaccurate with at most negligible probability), and (b) ciphertexts are of the form shown in equation 4.1. A failure in the BlindExtract check causes that protocol to output $\perp$, and a failed ciphertext check will cause Decrypt to output ϕ regardless of which secret key is used.

Now consider a malicious master authority $\mathcal{A}$ with non-negligible advantage in the game of 4.1.5. For $\mathcal{A}$ to succeed, it must hold that neither execution of BlindExtract with $\mathcal{A}$ outputs $\perp$, and $\mathsf{Decrypt}(params, id, sk_{id}, C) \neq \mathsf{Decrypt}(params, id, sk'_{id}, C)$. This implies that the (possibly probabilistic) group parameter check was conducted twice on γ, and succeeded at least once (else both calls to Decrypt would output ϕ). We denote by β the probability that $\mathcal{A}$ succeeds when the parameters γ are *not* valid.

In the event that the group parameters are valid and $\mathcal{A}$ succeeds, then by the ciphertext/key validity checks in BlindExtract and Decrypt, it must be the case that C, sk_{id}, sk_{id} all have the correct form for (respectively) some values $s, r_1, r_2 \in \mathbb{Z}_q$ and yet $\mathsf{Decrypt}(params, id, sk_{id}, C) \neq \mathsf{Decrypt}(params, id, sk'_{id}, C)$. Yet, by examining the math

[7] For some known $y \in \mathbb{Z}_q$ selected by the user, this test can be written as the comparison $e(g_1, g_2) \cdot e(d'_1, F(id)g^y) = e(d_0 g^{yr}, g)$.

of the decryption algorithm we see that this cannot be the case. The following equation *must* hold for every tuple $s, r_1, r_2 \in \mathbb{Z}_q$:

$$\mathsf{Decrypt}(params, id, sk_{id}, C) = \mathsf{Decrypt}(params, id, sk'_{id}, C)$$

$$X \cdot \frac{e(F(id)^s, g^{r_1})}{e(g^s, g_2^\alpha \cdot F(id)^{r_1})} = X \cdot \frac{e(F(id)^s, g^{r_2})}{e(g^s, g_2^\alpha \cdot F(id)^{r_2})} = \frac{X}{e(g^s, g_2^\alpha)}$$

$\mathcal{A}$'s advantage in the game as therefore bounded by β, the probability that at least one execution of the group parameter check incorrectly accepts γ as valid. Since the definition of the group parameter check ensures that β is negligible in κ, we conclude our proof. □

4.3.2 Boyen-Waters Anonymous IBE

Some of the applications we propose — *e.g.,* oblivious keyword search [OK04] — require a blind IBE scheme with the additional property of *anonymity*. This property is the identity-based equivalent of the more traditional "key privacy" [BBDP01]. In an anonymous IBE scheme, an adversary with access to a ciphertext cannot determine which identity the ciphertext was encrypted under.[8] This property is quite useful, especially as Boneh, DiCrescenzo, Ostrovsky and Persiano [BCOP04] show that it is sufficient for constructing public-key searchable encryption.

In 2006, Boyen and Waters [BW06] proposed an anonymous IBE secure under the DBDH and Decision Linear assumptions in symmetric bilinear groups. While this scheme is related to the the Boneh-Boyen construction, the key extraction protocol is quite different. As a result, we must develop the BlindExtract from scratch. (Independently of this work, Camenisch, Kohlweiss, Durán and Sheedy proposed a second protocol for blindly extracting keys in the Boyen-Waters scheme [CKDS09]. Their protocol differs from ours in that it makes use of an additively-homomorphic encryption scheme and uses a greater number of rounds.)

Let us now recall the basic elements of the Boyen-Waters IBE:

$\mathsf{Setup}(1^\kappa, c(k))$**:** Let $\gamma = (q, g, \mathbb{G}, \mathbb{G}_T, e)$ be the output of $\mathsf{BMsetup}(1^\kappa)$. Choose random generators $g, g_0, g_1 \in \mathbb{G}$ and random $\omega, t_1, t_2, t_3, t_4 \in \mathbb{Z}_q$. Output $params = [\Omega = e(g,g)^{t_1 t_2 \omega}, g, g_0, g_1, v_1 = g^{t_1}, v_2 = g^{t_2}, v_3 = g^{t_3}, v_4 = g^{t_4}]$ and $msk = [\omega, t_1, t_2, t_3, t_4]$.

$\mathsf{Extract}$**:** The master authority generates $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_q$ and for a given id outputs a secret keys are of the form:

$$\left[g^{r_1 t_1 t_2 + r_2 t_3 t_4}, g^{-\omega t_2}(g_0 g_1^{id})^{-r_1 t_2}, g^{-\omega t_1}(g_0 g_1^{id})^{-r_1 t_1}, (g_0 g_1^{id})^{-r_2 t_4}, (g_0 g_1^{id})^{-r_2 t_3}\right]$$

[8] Naturally the adversary *will* be able to test the ciphertext against any identity secret keys it possesses.

As before, the correctness of the key can be easily tested.

Encrypt$(params, id, M)$**:** Given an identity $id \in \mathcal{I}$, and a message $M \in \mathbb{G}_T$, select random $s, s_1, s_2 \in \mathbb{Z}_q$ and output the ciphertext $C = [C', C_0, C_1, C_2, C_3, C_4] = \left[\Omega^s M, (g_0 g_1^{id})^s, v_1^{s-s_1}, v_2^{s_1}, v_3^{s-s_2}, v_4^{s_2}\right]$.

Decrypt$(params, id, sk_{id}, c_{id})$**:** On input a decryption key $sk_{id} = (d_0, d_1, d_2, d_3, d_4)$ and a ciphertext $C = (C', C_0, C_1, C_2, C_3, C_4)$, output $M = C' e(C_0, d_0) e(C_1, d_1) e(C_2, d_2) e(C_3, d_3) e(C_4, d_4)$.

The BlindExtract **Protocol.** The blind extraction protocol for the Boyen-Waters scheme differs from that of the previous schemes in two important ways:

1. It does not produce a "correctly"-formed IBE decryption key. Nonetheless, it is possible to perform decryption using the returned key. We specify the protocol as well as the modified decryption algorithm.
2. The key returned from the protocol does not satisfy the strong definition of selective-failure blindness proposed in the previous sections. Instead, we assume that keys returned from this protocol will *not* be revealed to an adversary.

$\underline{\mathcal{P}(params, msk)}$	$\underline{\mathcal{U}(params, id)}$
	1. Choose $v \xleftarrow{\$} \mathbb{Z}_q$.
	2. Compute $h' \leftarrow (g_0 g_1^{id})^v$ and send h' to $\mathcal{P}$.
	3. Conduct $WIPoK\{(v, id) : h' = g_0^v g_1^{v \cdot id}\}$.
4. If the proof fails to verify, abort.	
5. Choose $r_1, r_2 \xleftarrow{\$} \mathbb{Z}_q$.	
6. Set $[d'_0, d'_1, d'_2, d'_3, d'_4] \leftarrow [g^{r_1 t_1 t_2 + r_2 t_3 t_4}, g^{-\omega t_2} h'^{-r_1 t_2}, g^{-\omega t_1} h'^{-r_1 t_1}, h'^{-r_2 t_4}, h'^{-r_2 t_3}]$	
7. Send $[d'_0, d'_1, d'_2, d'_3, d'_4]$ to $\mathcal{U}$.	
	7. Unblind the returned value by computing: $\left[d_0, \bar{d}_1, \bar{d}_2, d_3, d_4\right] \leftarrow \left[d'_0, d_1'^{1/v}, d_2'^{1/v}, d_3'^{1/v}, d_4'^{1/v}\right]$
	8. Test the key by encrypting a random message and using the modified Decrypt algorithm.
	9. Output $sk_{id} = \left[d_0, \bar{d}_1, \bar{d}_2, d_3, d_4\right]$.

Figure 4.5: A BlindExtract protocol for the Boyen-Waters anonymous IBE.

Modified Decryption. The extracted key $\left[d_0, \bar{d}_1, \bar{d}_2, d_3, d_4\right]$ is not correctly formed as in standard extraction. The difference is isolated to the components $\bar{d}_1$ and $\bar{d}_2$ which can be written as $\bar{d}_1 = g^{\frac{-\omega}{v} t_2} (g_0 g_1^{id})^{-r_1 t_2}$ and $\bar{d}_2 = g^{\frac{-\omega}{v} t_1} (g_0 g_1^{id})^{-r_1 t_1}$.

Note that this key can still be used to decrypt. However, we must slightly alter the decryption process. Given a ciphertext $[C', C_0, C_1, C_2, C_3, C_4]$, a key, and the value v, the revised Decrypt algorithm is simply:

$$M' = C' \left(e(C_0, d_0)e(C_1, \bar{d}_1)e(C_2, \bar{d}_2)e(C_3, d_3)e(C_4, d_4)\right)^v$$

We can show that correctness still holds by expanding the terms. Define the value $K \in \mathbb{G}_T$ as:

$$\begin{aligned} K = e\left((g_0 g_1^{id})^s, g^{r_1 t_1 t_2 + r_2 t_3 t_4}\right) \\ \cdot e\left(v_1^{s-s_1}, g^{\frac{-\omega}{v} t_2} (g_0 g_1^{id})^{-r_1 t_2}\right) \\ \cdot e\left(v_2^{s_1}, g^{\frac{-\omega}{v} t_1} (g_0 g_1^{id})^{-r_1 t_1}\right) \\ \cdot e\left(v_3^{s-s_2}, (g_0 g_1^{id})^{r_2 t_4}\right) \\ \cdot e\left(v_4^{s_2}, (g_0 g_1^{id})^{r_2 t_3}\right) \\ = e(g, g)^{-\frac{\omega t_1 t_2 s}{v}} \end{aligned}$$

Then by definition $M' = C' K^v$.

Theorem 4.3.4 *Under the DBDH assumption, the above blind IBE is secure in the* IND-ID-CPA *sense, and* BlindExtract *is leak-free.*

We sketch a *partial* proof of Theorem 4.3.4 in Appendix C.1 which uses standard techniques.

4.3.3 On Other IBEs and HIBEs

We have not considered hierarchical IBE schemes [GS02, BB04a, Wat05, CS06, GH08a] in this work, since we do not need this capability for the OT protocols described here. Nonetheless, it is worth pointing out that Boneh and Boyen [BB04a], Waters [Wat05] and Chatterjee and Sarkar [CS06] do admit hierarchical delegation. In these constructions, the number of elements comprising an identity secret key grow with the depth of the hierarchy, but each piece is similar in format to the original keys and our same techniques would apply.

Let us briefly summarize what we know about efficient BlindExtract protocols for other IBE schemes. First, random oracle based IBEs [BF01, Coc01] appear to be less suited to developing efficient BlindExtract protocols than their standard model successors. This is in part due to the fact that the identity string is hashed into an element in $\mathbb{G}$ in these schemes, instead of represented as an integer exponent, which makes our proof of knowledge techniques unwieldy. We were not able to find BlindExtract protocols for the Boneh

and Franklin [BF01], Cocks [Coc01], or the recent Boneh-Gentry-Hamburg [BGH07] IBEs with running time better than $O(|\mathcal{I}|)$, where $\mathcal{I}$ is the identity space. Additionally, we did not consider the efficient IBE of Gentry [Gen06], as our focus was on developing schemes secure under *static* complexity assumptions.

On other committing blind IBE schemes.. We note that several existing IBE schemes, *e.g.,* that of Gentry [Gen06] seem incompatible with the notion of a committing IBE, since keys come in multiple forms which may not be easily distinguished. However, this might be rectified by adding zero-knowledge proofs of correctness to the key extraction protocol. We conclude with a general observation: that any "unique" secure blind IBE is implicitly committing. Borrowing from the language of signatures, we define a *unique* IBE as having one valid identity secret key for each identity in the system. Since the schemes presented herein are not unique, we might simplify our constructions by looking for such schemes.

4.4 Other Applications of Blind IBE

Privacy-preserving delegated keyword search. Several works use IBE as a building-block for *public-key searchable encryption* [BCOP04, WBDS04]. These schemes permit a keyholder to delegate search capability to other parties. For example, Waters, Balfanz, Durfee and Smetters [WBDS04] describe a searchable encrypted audit log in which a third party auditor is granted the ability to independently search the encrypted log for specific keywords. To enable this function, a central authority generates "trapdoors" for the keywords that the auditor wishes to search on. In this scenario, the trapdoor generation authority necessarily learns each of the search terms. This may be problematic in circumstances where the pattern of trapdoor requests reveals sensitive information (*e.g.*, the name of a user under suspicion). By using blind and partially-blind IBE, we permit the authority to generate trapdoors, yet learn no information (or only partial information) about the search terms.[9]

Blind and partially-blind signature schemes. Moni Naor observed that each adaptive-identity secure IBE implies an existentially unforgeable signature scheme [BF01]. By the same token, an adaptive-identity secure blind IBE scheme implies an unforgeable, selective-failure blind signature scheme. This result applies to the adaptive-identity secure Π_{Waters} protocol of §4.3.1.2, and to the selective-identity secure protocol Π_{BB} when that scheme is instantiated with appropriately-sized parameters and a hash function (see §7 of [BB04a]). The efficient $\mathsf{BlindExtract}$ protocol for the adaptive-identity secure Π_{Waters} scheme can also be used to construct a *partially-blind* signature, by allowing the signer (the master authority) to supply a portion of the input string. Partially-blind signatures have many applications, such as document timestamping and electronic cash [MS98].

[9] Boneh *et al.* [BCOP04] note that keyword search schemes can be constructed from any *key anonymous* IBE scheme. Thus, a practical implementation might use the Boyen-Waters scheme described above [BW06].

Temporary anonymous identities. In a typical IBE, the master authority can link users to identities. For some applications, users may wish to remain anonymous or pseudonymous. By employing (partially-)blind IBE, an authority can grant temporary credentials without linking identities to users or even learning which identities are in use.

Chapter 5

Universally Composable Adaptive Oblivious Transfer

This chapter is based on joint work with Susan Hohenberger that appears in Josef Pieprzyk (Ed.): Advances in Cryptology - ASIACRYPT 2008, Volume 5350 of Lecture Notes in Computer Science, pages 179–197, Springer-Verlag, 2008 [GH08c].

In the previous chapter, we proposed an efficient adaptive $\mathsf{OT}^N_{k\times 1}$ protocol based on IBE techniques. To our knowledge, this protocol and those of Camenisch *et al.* [CNs07] represent the only efficient *adaptive* OT protocols secure in a strong, full-simulation definition. Unfortunately, the adaptive protocols of the previous chapter and the "generic" protocol of Camenisch *et al.* are proven secure in the Random Oracle model, which has been shown to admit proofs of security for demonstrably insecure protocols [CGH04]. At the same time, the standard-model protocols of Camenisch *et al.* use interactive zero-knowledge protocols which rely on rewinding for their security proofs. Thus, these protocols are secure only under sequential composition, and cannot be proven secure under current composition.

In this chapter, we take a different approach to constructing OT protocols, which allows them to be simultaneously efficient, adaptive, universally composable and globally consistent. This is, to our knowledge, the first such *practical* adaptive OT secure in the UC security model.

Intuition behind the Construction. An appealing naive approach to realizing UC-secure adaptive OT would be to modify the protocols of Chapter 4, or the standard of Camenisch, Neven and shelat [CNs07]— *e.g.,* by simply replacing rewinding-based proofs with the non-interactive proof techniques of Groth and Sahai [GS08]. Unfortunately, this is non-trivial for two reasons. First, the Groth-Sahai techniques provide broad support for non-interactive, *witness indistinguishable* proofs of algebraic assertions in bilinear groups, but only provide non-interactive, *zero-knowledge* proofs for a restricted class of algebraic as-

sertions. Unfortunately, the proof statements required by [CNs07] fall outside of this class, and it does not seem easy to rectify this problem. Secondly, the protocols mentioned above require some form of extraction (e.g., extracting the chosen index from the adversarial Receiver or extracting the secret encryption keys from the adversarial Sender) for proofs containing elements of $\mathbb{Z}_q$; unfortunately, Groth-Sahai proofs of knowledge are f-extractable (but not fully extractable), where only some one-way function of the witness, $f(w)$, can be extracted (e.g., g^w) and not the witness w itself. Dealing with this limitation would necessitate substantial changes to the protocols.

Instead, our construction starts from scratch. While we follow the "assisted decryption" framework used throughout this work, we are able to do so without the need for strong p-based decisional assumptions. We instead base the security of the ciphertexts in our scheme on the Decision Linear assumption [BBS04]. Finally, since the Groth-Sahai proofs have not yet been shown to be either simulation-sound or UC in general, we develop techniques that permit UC simulation (even in the advanced case where multiple receivers interact with a single sender).

5.1 Building Blocks

We now describe several of the building blocks that will be used in our construction.

Groth-Sahai Proofs. Our constructions will make use of the Groth-Sahai proof system, which is described in detail in §3.4.2.

Modified CL Signatures. Our constructions use a weak variant of the Camenisch-Lysanskyaya signature scheme [CL04], altered to operate on messages in $\mathbb{G}_1$. Whereas CL signatures rely on the interactive oracle LRSW assumption to achieve security against adaptive chosen-message attacks, in the context of our construction we will require only a non-interactive p-Hidden LRSW assumption to achieve a weaker property (unforgeability given a set of signatures on *random* messages).

$\mathsf{CLNKeyGen}(\gamma, g, \tilde{g})$**.** On input $\gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \dots)$ and generators $(g, \tilde{g})$, select $s, t \xleftarrow{\$} \mathbb{Z}_q$ and set $\tilde{S} \leftarrow \tilde{g}^s, \tilde{T} \leftarrow \tilde{g}^t$. Output $vk = (\gamma, g, \tilde{g}, \tilde{S}, \tilde{T})$, and $sk = (vk, s, t)$.

$\mathsf{CLNSign}_{sk}(m)$**.** On input a message $m \in \mathbb{G}_1$, select $w \xleftarrow{\$} \mathbb{Z}_q$ and output the signature $\mathsf{sig} = (g^w, m^w, g^{ws} m^{wst}, m^{wt}, \tilde{g}^w) \in \mathbb{G}_1^4 \times \mathbb{G}_2$.

$\mathsf{CLNVerify}_{vk}(\mathsf{sig}, m)$**.** On input the value $m \in \mathbb{G}_1$ and $\mathsf{sig} = (a_1, a_2, a_3, a_4, \tilde{a}_5)$, verify that $e(g, \tilde{a}_5) = e(a_1, \tilde{g}) \ \wedge \ e(m, \tilde{a}_5) = e(a_2, \tilde{g}) \ \wedge \ e(a_2, \tilde{T}) = e(a_4, \tilde{g}) \ \wedge \ e(a_3, \tilde{g}) = e(a_1 a_4, \tilde{S})$.

Note that the verification algorithm can be represented as a set of pairing product equations, and thus it is possible to prove knowledge of a pair (m, sig) using the GS proof system. To prove knowledge of m, sig, first select $y \xleftarrow{\$} \mathbb{Z}_q$, compute $\mathsf{sig}' = \langle a'_1, a'_2, a'_3, a'_4, \tilde{a}'_5 \rangle =$

$\langle a_1^y, a_2^y, a_3^y, a_4^y, \tilde{a}_5^y \rangle$ and release the pair $a'_1, \tilde{a}'_5$ along with the following witness indistinguishable proof:

$$\pi = NIWI_{GS}\{(m, a'_2, a'_3, a'_4) :$$
$$e(m, \tilde{a}'_5)e(a'_2, \tilde{g}^{-1}) = 1 \ \wedge \ e(a'_2, \tilde{T})e(a'_4, \tilde{g}^{-1}) = 1 \ \wedge \ e(a'_3, \tilde{g})e(a'^{-1}_4, \tilde{S}) = e(a'_1, \tilde{S})\}$$

The verifier checks both the proof and the fact that $e(a'_1, \tilde{g}) = e(g, \tilde{a}'_5)$.

Selective-message Secure Boneh-Boyen Signatures. Our constructions also make use of a weak signature scheme built from the Boneh-Boyen selective-ID IBE scheme [BB04a] (§4).

$\mathsf{BBKeyGen}(\gamma, g_1, \tilde{g}_1)$**.** On input $\gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \dots)$ and bases $(g_1, \tilde{g}_1)$, select $\alpha, z \xleftarrow{\$} \mathbb{Z}_q$, $g \leftarrow g_1^{1/\alpha}$, $\tilde{g} \leftarrow \tilde{g}_1^{1/\alpha}$, $g_2 \leftarrow g^z$, $\tilde{g}_2 \leftarrow \tilde{g}^z$, $h \xleftarrow{\$} \mathbb{G}_1$. Output $vk = (\gamma, g, \tilde{g}, g_1, g_2, h, \tilde{g}_2)$, and $sk = (vk, g_2^\alpha)$.

$\mathsf{BBSign}_{sk}(m)$**.** On input a message $m \in \mathbb{G}_1$, select $r \xleftarrow{\$} \mathbb{Z}_q$ and output the signature $\mathsf{sig} = ((mh)^r g_2^\alpha, \tilde{g}^r, g^r) \in \mathbb{G}_1^2 \times \mathbb{G}_2$.

$\mathsf{BBVerify}_{vk}(\mathsf{sig}, m)$**.** On input $m \in \mathbb{G}_1$ and $\mathsf{sig} = (s_1, \tilde{s}_2, s_3)$, verify that $e(s_1, \tilde{g})/e(mh, \tilde{s}_2) = e(g_1, \tilde{g}_2)$ and $e(g, \tilde{s}_2) = e(s_3, \tilde{g})$.

We can prove knowledge of a pair (m, sig) as follows. Select $y \xleftarrow{\$} \mathbb{Z}_q$ and set $\mathsf{sig}' = (s'_1, \tilde{s}'_2, s'_3) = (s_1(mh)^y, \tilde{s}_2\tilde{g}^y, s_3 g^y)$. Output $\tilde{s}'_2, s'_3$ and the witness indistinguishable proof:

$$\pi = NIWI_{GS}\{(m, s'_1) : e(s'_1, \tilde{g})e(m, \tilde{s}'^{-1}_2) = e(h, \tilde{s}'_2)e(g_1, \tilde{g}_2)\}$$

The verifier checks the proof and the fact that $e(g, \tilde{s}'_2) = e(s'_3, \tilde{g})$.

Double-Trapdoor BBS Encryption. Boneh, Boyen and Shacham [BBS04] describe a semantically-secure encryption scheme based on the Decision Linear (DLIN) assumption. We extend their scheme into a *two-key* (double-trapdoor) encryption scheme with a public consistency check. In this system, we can encrypt a message under two distinct public keys pk_1, pk_2, such that *either* of the corresponding secret keys sk_1, sk_2 will decrypt the ciphertext. For every well-formed ciphertext, it must be the case that decryption will produce the same message regardless of which secret key is used. To satisfy this requirement, we also define a *publicly-computable* check for ciphertext well-formedness (*i.e.,* the check does not require knowledge of either secret key).

Let $\mathsf{BMsetup}(1^\kappa) \rightarrow \gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g})$. Define global parameters $h, \tilde{h}$ such that $e(g, \tilde{h}) = e(\tilde{g}, h)$, and for $i \in [1, 2]$ select $sk_i \leftarrow (x_i, y_i \in_R \mathbb{Z}_q)$ and $pk_i \leftarrow (h^{1/x_i}, h^{1/y_i}, \tilde{h}^{1/x_i}, \tilde{h}^{1/y_i} \in \mathbb{G}_1^2 \times \mathbb{G}_2^2)$. To encrypt a message $m \in \mathbb{G}_1$ under $pk_1 = (u_1, v_1), pk_2 = (u_2, v_2)$, first select random values $r, s \in \mathbb{Z}_q$ and output the ciphertext $(u_1^r, v_1^s, u_2^r, v_2^s, h^{r+s}m)$. To decrypt a message $(c_1, \dots, c_5)$ under $sk_1 = (x_1, y_1)$, output $c_5/(c_1^{x_1} \cdot c_2^{y_1})$. To decrypt under $sk_2 = (x_2, y_2)$, output $c_5/(c_3^{x_2} \cdot c_4^{y_2})$. Note that the structure of a ciphertext can be verified using the bilinear map, by checking that $e(c_1, \tilde{u}_2) = e(u_1, \tilde{c}_3) \ \wedge \ e(c_2, \tilde{v}_2) = e(v_1, \tilde{c}_4)$ It is straightforward to show that the scheme above is semantically-secure under the DLIN assumption.

Protocol OTA

OTA is parameterized by the algorithms
(OTGenCRS, OTInitialize, OTRequest, OTRespond, OTComplete).

When $\mathbf{S}$ is activated with $(\mathsf{sid}, \mathsf{sender}, \langle M_1, \ldots, M_N \rangle)$:

1. $\mathbf{S}$ queries $\mathcal{F}_{CRS}$ with $(\mathsf{sid}, \mathbf{S}, \mathbf{R})$ and receives $(\mathsf{sid}, \mathsf{crs})$. $\mathbf{R}$ then queries $\mathcal{F}_{CRS}$ with $(\mathsf{sid}, \mathbf{S}, \mathbf{R})$ and receives $(\mathsf{sid}, \mathsf{crs})$. $\mathcal{F}_{CRS}$ responds to these queries with $\mathsf{crs} \leftarrow \mathsf{OTGenCRS}(1^\kappa)$.
2. $\mathbf{S}$ computes $(T, sk) \leftarrow \mathsf{OTInitialize}(\mathsf{crs}, M_1, \ldots, M_N)$, sends (sid, T) to $\mathbf{R}$ and stores (sid, T, sk).

When $\mathbf{R}$ is activated with $(\mathsf{sid}, \mathsf{receiver}, \sigma)$, and $\mathbf{R}$ has previously received (sid, T) and $(\mathsf{sid}, \mathsf{crs})$:

1. If $\mathbf{S}$ was not previously activated with $(\mathsf{sid}, \mathsf{sender}, M_1, \ldots, M_N)$, do nothing.
2. $\mathbf{R}$ runs $(Q, Q_{priv}) \leftarrow \mathsf{OTRequest}(\mathsf{crs}, T, \sigma)$, sends (sid, Q) to $\mathbf{S}$ and stores (sid, Q_{priv}).
3. $\mathbf{S}$ gets (sid, Q) from $\mathbf{R}$, runs $R \leftarrow \mathsf{OTRespond}(\mathsf{crs}, T, sk, Q)$, and sends (sid, R) to $\mathbf{R}$.
4. $\mathbf{R}$ receives (sid, R) from $\mathbf{S}$, and outputs $(\mathsf{sid}, \mathsf{OTComplete}(\mathsf{crs}, T, R, Q_{priv}))$.

Figure 5.1: A high-level outline of the $\mathsf{OT}^N_{k\times 1}$ protocol, with details of each algorithm described in Section 5.2. We make no explicit mention of the value k, the total transfers permitted by the Sender, because our protocol does not depend on it. The Sender may choose to stop answering the Receiver's queries at any point, in which case OTRespond outputs "reject" and OTComplete accepts this as the message $\perp$.

5.2 Construction

Our adaptive oblivious transfer protocol, $\mathsf{OT}^N_{k\times 1}$ follows the framework described in Figure 5.1. This work provides two possible instantiations of the algorithms (OTGenCRS, OTInitialize, OTRequest, OTRespond, OTComplete). We present our main construction below (and also present the alternative realization in Appendix A.1).

$\mathsf{OTGenCRS}(1^\kappa)$. Given security parameter κ, generate parameters for a bilinear mapping $\gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \tilde{g}) \leftarrow \mathsf{BMsetup}(1^\kappa)$. Compute $GS_S \leftarrow \mathsf{GSSetup}(\gamma)$ and $GS_R \leftarrow \mathsf{GSSetup}(\gamma)$. Choose $a, b, c \xleftarrow{\$} \mathbb{Z}_q$, and set $(g_1,\ g_2,\ h,\ \tilde{g}_1,\ \tilde{g}_2,\ \tilde{h}) \leftarrow (g^a, g^b, g^c, \tilde{g}^a, \tilde{g}^b, \tilde{g}^c)$. Output $\mathsf{crs} = (\gamma,\ GS_S,\ GS_R,\ g_1,\ g_2,\ h,\ \tilde{g}_1,\ \tilde{g}_2,\ \tilde{h})$. (At the end of this chapter, we describe how this common reference string can be replaced

by a common random string.)

$\mathsf{OTInitialize}(\mathsf{crs}, m_1, \ldots, m_N)$. This algorithm is executed by the Sender. On input a collection of N messages and the crs, it outputs a commitment to the database, T, for publication to the Receiver, as well as a Sender secret key, sk. We treat messages as elements of $\mathbb{G}_1$, since there exist efficient mappings between strings in $\{0,1\}^\ell$ and elements in $\mathbb{G}_1$ (e.g., [BF01, ACdM05]).

1. Parse crs to obtain $GS_S, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h}$ and γ.
2. Choose random values $x_1, x_2 \in \mathbb{Z}_q$.
3. Set $(u_1, u_2) \leftarrow (h^{1/x_1}, h^{1/x_2})$, $(\tilde{u}_1, \tilde{u}_2) \leftarrow (\tilde{h}^{1/x_1}, \tilde{h}^{1/x_2})$.
4. Set $(vk_1, sk_1) \leftarrow \mathsf{CLNKeyGen}(\gamma, u_1, \tilde{u}_1)$, $(vk_2, sk_2) \leftarrow \mathsf{CLNKeyGen}(\gamma, u_2, \tilde{u}_2)$ and $(vk_3, sk_3) \leftarrow \mathsf{BBKeyGen}(\gamma, u_1, \tilde{u}_1)$.
5. Set $pk \leftarrow (u_1, u_2, \tilde{u}_1, \tilde{u}_2, vk_1, vk_2, vk_3)$.
6. For $j = 1, \ldots, N$ encrypt each message m_j as:
   (a) Select random $r, s, t \in \mathbb{Z}_q$.
   (b) Compute $\mathsf{sig}_1 \leftarrow \mathsf{CLNSign}_{sk_1}(u_1^r)$, $\mathsf{sig}_2 \leftarrow \mathsf{CLNSign}_{sk_2}(u_2^s)$, and $\mathsf{sig}_3 \leftarrow \mathsf{BBSign}_{sk_3}(u_1^r u_2^s)$.
   (c) Set $C_j \leftarrow (u_1^r,\ u_2^s,\ g_1^r,\ g_2^s,\ m_j \cdot h^{r+s},\ \mathsf{sig}_1,\ \mathsf{sig}_2,\ \mathsf{sig}_3)$.
7. Set $T \leftarrow (pk, C_1, \ldots, C_N)$ and $sk \leftarrow (x_1, x_2)$. Output (T, sk).

Each ciphertext C_j above can be thought of as a signcryption where it is the *randomness* for each ciphertext that is signed, rather than the plaintext itself. Each plaintext m_j is encrypted under S's public key u_1, u_2, as well as a "key" g_1, g_2 drawn from crs. This "double-trapdoor" encryption is necessary for the security proof of the OT scheme.

To verify the format of each ciphertext $C_j = (c_1, \ldots, c_5, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$ in T, anyone can check that $\mathsf{CLNVerify}_{vk_1}(c_1, \mathsf{sig}_1)$, $\mathsf{CLNVerify}_{vk_2}(c_2, \mathsf{sig}_2)$, and $\mathsf{BBVerify}_{vk_3}(c_1 c_2, \mathsf{sig}_3)$ each succeed, and that $e(c_1, \tilde{g}_1) = e(c_3, \tilde{u}_1) \wedge e(c_2, \tilde{g}_2) = e(c_4, \tilde{u}_2)$.

$\mathsf{OTRequest}(\mathsf{crs}, T, \sigma)$. This algorithm is executed by a Receiver. On input T generated by the Sender, along with an item index σ, generates a query Q for transmission to the Sender.

1. Parse T as $(pk, C_1, \ldots, C_N)$, and ensure that it is correctly formed (see above). If T is not correctly formed, abort the protocol. (This is only necessary on the first transfer.)
2. Parse crs to obtain $(GS_R, \tilde{h})$, and parse pk as $(u_1, u_2, \tilde{u}_1, \tilde{u}_2, vk_1, vk_2, vk_3)$. Parse the σ^{th} ciphertext C_σ as $(c_1, \ldots, c_5, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$.
3. Select random $v_1, v_2 \in \mathbb{Z}_q$.
4. Set $d_1 \leftarrow (c_1 \cdot u_1^{v_1}), d_2 \leftarrow (c_2 \cdot u_2^{v_2})$, $t_1 \leftarrow h^{v_1}$, $t_2 \leftarrow h^{v_2}$.

5. Use the Groth-Sahai techniques and reference string GS_R to compute a Witness Indistinguishable proof π that the values d_1, d_2 pertaining to the ciphertext C_σ (which the Receiver wishes to have the Sender help him open) have the correct structure:

$$\pi = NIWI_{GS_R}\{(c_1, c_2, t_1, t_2, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3) : \\ e(c_1, \tilde{h})e(t_1, \tilde{u}_1) = e(d_1, \tilde{h}) \ \wedge \ e(c_2, \tilde{h})e(t_2, \tilde{u}_2) = e(d_2, \tilde{h}) \wedge \\ \mathsf{CLNVerify}_{vk_1}(c_1, \mathsf{sig}_1) = 1 \ \wedge \ \mathsf{CLNVerify}_{vk_2}(c_2, \mathsf{sig}_2) = 1 \wedge \\ \mathsf{BBVerify}_{vk_3}(c_1c_2, \mathsf{sig}_3) = 1\}$$

6. Set request $Q \leftarrow (d_1, d_2, \pi)$, and private state $Q_{priv} \leftarrow (Q, \sigma, v_1, v_2)$. Output (Q, Q_{priv}).

To explain what is happening in the statement of step (5), first observe that the signature proofs of knowledge ensure that the values c_1, c_2 and the product (c_1c_2) each correspond to a valid signature held by the Receiver. The remaining equations ensure that the values d_1, d_2 correspond to "blinded" versions of the elements c_1, c_2. These checks guarantee that the witness used by the Receiver, and thus the decryption request being made, corresponds to one of the N ciphertexts published by the Sender.

$\mathsf{OTRespond}(\mathsf{crs}, T, sk, Q)$. This algorithm is executed by the Sender. If the Sender does not wish to answer any more requests for the Receiver, then the Sender outputs the message "reject". Otherwise, the Sender processes the Receiver's request Q as:

1. Parse crs to obtain $(GS_R, \tilde{g}, \tilde{h})$, and parse T as $(pk, C_1, \ldots, C_N)$, and sk as (x_1, x_2).
2. Parse pk (from T) as $(u_1, u_2, \tilde{u}_1, \tilde{u}_2, vk_1, vk_2, vk_3)$.
3. Parse Q as (d_1, d_2, π) and verify proof π using GS_R. Abort if check fails.
4. Set $a_1 \leftarrow d_1^{x_1}$, $a_2 \leftarrow d_2^{x_2}$, and $s \leftarrow a_1 \cdot a_2$.
5. Use the Groth-Sahai techniques and reference string GS_S to formulate a zero-knowledge proof[1] that the decryption value s is properly computed:

$$\delta = NIZK_{GS_S}\{(a_1, a_2) : e(a_1, \tilde{u}_1)e(d_1^{-1}, \tilde{h}) = 1 \\ \wedge \ e(a_2, \tilde{u}_2)e(d_2^{-1}, \tilde{h}) = 1 \ \wedge \ e(a_1a_2, \tilde{h})e(s^{-1}, \tilde{h}) = 1\}$$

The third equation ensures that $s = a_1 \cdot a_2$, while the first two, since the values $(u_1, d_1, u_2, d_2, \tilde{h})$ are known to both parties, ensure that $a_1 = d_1^{x_1}$ and $a_2 = d_2^{x_2}$.

[1] We present a simplified version of this proof above. However, to permit simulation, we must add a third variable $\tilde{a}_3 = \tilde{h}$ and re-write the proof as $NIZK_{GS_S}\{(a_1, a_2, \tilde{a}_3) : e(a_1, \tilde{u}_1)e(d_1^{-1}, \tilde{a}_3) = 1 \ \wedge \ e(a_2, \tilde{u}_2)e(d_2^{-1}, \tilde{a}_3) = 1 \ \wedge \ e(a_1a_2, \tilde{a}_3)e(s^{-1}, \tilde{a}_3) = 1 \ \wedge \ e(u_1, \tilde{a}_3) = e(u_1, \tilde{h})\}$. See the full version for details.

6. Output $R \leftarrow (s, \delta)$.

$\mathsf{OTComplete}(\mathsf{crs}, T, R, Q_{priv})$. This algorithm is executed by the Receiver. On input R generated by the Sender in response to a request Q, along with state Q_{priv}, outputs a message m or $\perp$. If R is the message "reject", then the Receiver outputs $\perp$. Otherwise, the Receiver does:

1. Parse crs to obtain (GS_S, h). Parse T as $(pk, C_1, \ldots, C_N)$, R as (s, δ), and Q_{priv} as (Q, σ, v_1, v_2).
2. Verify proof δ using GS_S. If verification fails, output $\perp$.
3. Parse C_σ to obtain the first five elements $(c_1, \ldots, c_5)$ and output $m = c_5/(s \cdot h^{-v_1} \cdot h^{-v_2})$. Map this element to a value in $\{0,1\}^\ell$ [ACdM05].

5.2.1 Efficiency Analysis

When the protocol in Figure 5.1 is implemented using the algorithms described above, we obtain a $(k + 1/2)$-round protocol with communications cost $O(N + k)$, where $k \leq N$. More concretely, the crs is comprised of 7 elements in $\mathbb{G}_1$ and 7 elements of $\mathbb{G}_2$, the Sender's public key contains 5 elements in $\mathbb{G}_1$ and 6 elements in $\mathbb{G}_2$. Each of the N ciphertexts in T requires 15 elements in $\mathbb{G}_1$ and 3 elements in $\mathbb{G}_2$. Moreover, each item transfer involves transmission of 68 elements of $\mathbb{G}_1$ and 38 elements of $\mathbb{G}_2$ from Receiver to Sender, and then 20 elements of $\mathbb{G}_1$ and 18 elements of $\mathbb{G}_2$ from Sender to Receiver. The message space of our OT protocol is elements in $\mathbb{G}_1$, which will be sufficient for transferring a symmetric encryption key to unlock a file of arbitrary size.

5.2.2 Security Analysis

Theorem 5.2.1 *Instantiated with the above algorithms,* OTA *securely realizes the functionality* $\mathcal{F}_{OT}^{N\times 1}$ *in the* $\mathcal{F}_{CRS}$-hybrid *model under the DLIN, and* p*-Hidden LRSW assumptions.*

5.2.2.1 Intuition

Let us now provide some intuition behind this proof, with the full proof directly below. When either the Sender or the Receiver is corrupted, we wish to describe a simulator $\mathcal{S}$ such that it can interact with the ideal functionality $\mathcal{F}_{OT}^{N\times 1}$ (which we'll denote simply as $\mathcal{F}$) and the environment $\mathcal{Z}$ appropriately; i.e., $\mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$.

Simulating the case where only S is corrupted. We first consider the case where the real-world adversary $\mathcal{A}$ corrupts the Sender, and thus $\mathcal{S}$ must interact with $\mathcal{F}$ as the ideal Sender and with (an internal copy of) $\mathcal{A}$ as a real-world Receiver. Here $\mathcal{S}$ does the following:

1. Ask $\mathcal{A}$ to begin an OT protocol, and set the crs *for these two parties* by running $\gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g \in \mathbb{G}_1, \tilde{g} \in \mathbb{G}_2) \leftarrow \mathsf{BMsetup}(1^\kappa)$, $GS_S \leftarrow \mathsf{GSSetup}(\gamma)$, $GS_R \leftarrow \mathsf{GSSetup}(\gamma)$, selecting random elements $a_1, a_2 \in \mathbb{Z}_q$, and setting $g_1^{a_1} = g_2^{a_2} = h$ (and a corresponding relationship for $\tilde{g}_1, \tilde{g}_2, \tilde{h}$). Set $\mathsf{crs} = (\gamma, GS_S, GS_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$. When the parties query $\mathcal{F}_{CRS}$, return $(\mathsf{sid}, \mathsf{crs})$.
2. Obtain the database commitment T from $\mathcal{A}$. Verify that T is well-formed, abort if not. Otherwise, $\forall i \in [1, N]$ use a_1, a_2 to decrypt each ciphertext $C_i = (c_1, \ldots, c_5, \ldots)$ as $m_i = c_5/(c_3^{a_1} c_4^{a_2})$. Map each element $m_i \in \mathbb{G}_1$ to a string in $\{0,1\}^\ell$ [ACdM05]. Send $(\mathsf{sid}, \mathbf{S}, m_1, \ldots, m_N)$ to $\mathcal{F}$.
3. Upon receiving $(\mathsf{sid}, \mathsf{request})$ from $\mathcal{F}$, return $\mathsf{OTRequest}(\mathsf{crs}, T, 1)$ to $\mathcal{A}$. This response includes two random values d_1, d_2 and a non-interactive witness indistinguishable proof π with respect to $GS_R \in \mathsf{crs}$ that d_1, d_2 are "blinded" values corresponding to ciphertext C_1. This proof can be performed honestly and without rewinding.
4. If $\mathcal{A}$ issues a "reject" message or responds with anything other than a value in $\mathbb{G}_1$ and a valid NIZK proof, then $\mathcal{S}$ tells $\mathcal{F}$ to fail the request by sending message $(\mathsf{sid}, 0)$. Otherwise, $\mathcal{S}$ sends the message $(\mathsf{sid}, 1)$ to $\mathcal{F}$.

The indistinguishability argument here follows from the indistinguishability of the crs (which is identically distributed to a real crs), the perfect extraction of the messages in step (2),[2] and the Witness Indistinguishability of the GS proof π issued during each request phase, which guarantees that $\mathcal{A}$ (the corrupt Sender) cannot distinguish a request to decrypt C_1 from a request to decrypt any other valid ciphertext. Thus, $\mathcal{S}$ can adequately mimic its response pattern.

Simulating the case where only R is corrupted. Next, we consider the case where the real world adversary $\mathcal{A}$ corrupts the Receiver, and thus $\mathcal{S}$ must interact with $\mathcal{F}$ as the ideal Receiver and with (and internal copy of) $\mathcal{A}$ as real-world Receiver. This case requires that the $p = N$ for the *p-Hidden LRSW* assumption. Here $\mathcal{S}$ does the following:

1. Ask $\mathcal{A}$ to begin an OT protocol, and set the crs *for these two parties* by running $\gamma = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g \in \mathbb{G}_1, \tilde{g} \in \mathbb{G}_2) \leftarrow \mathsf{BMsetup}(1^\kappa)$, $(GS_S, td_{sim}) \leftarrow \mathsf{GSSimulateSetup}(\gamma)$ and $(GS_R, td_{ext}) \leftarrow \mathsf{GSExtractSetup}(\gamma)$. Select random elements for $g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h}$. Set $\mathsf{crs} \leftarrow (\gamma, GS_S, GS_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$. When the parties query $\mathcal{F}_{CRS}$, return $(\mathsf{sid}, \mathsf{crs})$.
2. $\mathcal{S}$ must commit to a database of messages for $\mathcal{A}$ *without* knowing the messages $m_1, \ldots, m_N$. Thus, $\mathcal{S}$ simply commits to random junk messages, and sends the corresponding T to $\mathcal{A}$.
3. When $\mathcal{A}$ makes a transfer request, $\mathcal{S}$ uses td_{ext} to extract the witness W corresponding to $\mathcal{A}$'s decryption request from the NIWI proof. (This extraction is done via opening perfectly-binding commitments which are included in the WI proof and does not

[2]Note that a ciphertext that passes the validity check can be represented as $C = (u_1^r, u_2^s, g_1^r, g_2^s, h^{r+s} m, \ldots)$ for some $r, s \in \mathbb{Z}_q$, and when (g_1, g_2, h) have the relationship described above, decryption using a_1, a_2 always produces m.

require any rewinding.) This witness includes the first two elements (c_1, c_2) of the ciphertext that $\mathcal{A}$ is requesting to decrypt, and from these it is possible to determine the index σ' of the ciphertext that $\mathcal{A}$ has requested to open.

4. $\mathcal{S}$ now sends $(\mathsf{sid}, \mathbf{R}, \sigma')$ to $\mathcal{F}$ to obtain the real $m_{\sigma'}$ message.
5. Finally, $\mathcal{S}$ returns a response to $\mathcal{A}$ which opens $C_{\sigma'}$ to $m_{\sigma'}$ and then uses td_{sim} to simulate an NIZK proof that this opening is correct. The NIZK proof here is designed in such a way that simulation is always possible and no rewinding is necessary.

The indistinguishability argument here follows from the indistinguishability of the crs (from a real crs), the indistinguishability of the "fake" database T, the ability to extract witnesses from the NIWI proofs, and the zero-knowledge property of "fake" NIZK proofs. In particular, note that the N-*Hidden LRSW* assumption ensures that any decryption request made by the receiver corresponds to a valid ciphertext from the database T (if $\mathcal{A}$ produces a proof π embedding invalid ciphertext values, we can use $\mathcal{A}$ to solve N-*Hidden LRSW* or the co-CDH problem [BLS01], which is implied by N-*Hidden LRSW*).[3] Unlike the protocol of [CNs07] we are able to base the semantic security of the ciphertexts on a standard decisional assumption (the Decision Linear assumption). This is possible because the full ciphertext can be constructed using only the DLIN input (see the note on Ciphertext security below). Notice that $\mathcal{S}$ is never *both* simulating and extracting via the same (subsection of the) common reference string; indeed, we do not require that the proofs be simulation-sound.

Simulating the remaining cases. When both the Receiver and Sender are corrupted, $\mathcal{S}$ knows the inputs to $\mathbf{S}$ and $\mathbf{R}$ and can simulate a protocol execution by generating the real messages exchanged between the two parties. In the case where neither party is corrupted, then: when $\mathcal{S}$ receives messages of the form (sid, b_i) indicating that transfers have occurred, $\mathcal{S}$ generates a simulated transcript between the honest $\mathbf{S}$ and $\mathbf{R}$. In this case, $\mathcal{S}$ runs the protocol as specified, using as $\mathbf{S}$'s input a random database $(\hat{m}_1, \ldots, \hat{m}_N)$, and (for each transfer), $\mathbf{R}$'s input $\sigma' = 1$. If in the i^{th} transfer $b_i = 0$ then $\mathbf{S}$'s responds with an invalid R (the empty string). Else, $\mathbf{S}$ returns a valid response as in the protocol.

Ciphertext security. We briefly elaborate on the security of the ciphertexts in our scheme. To prove security when Receiver is corrupted, we must show that a ciphertext vector encrypting random messages is indistinguishable from a vector encrypting the real message database. We argue that this is the case under the Decision Linear assumption. Let $D = (g, \tilde{g}, f, \tilde{f}, h, \tilde{h}, g^a, f^b, z_d)$ be a candidate Decision Linear tuple. We consider a simulation that behaves as follows:

1. Set $u_1 = g, u_2 = f, \tilde{u}_1 = \tilde{g}, \tilde{u}_2 = \tilde{f}$. Select random $y_1, y_2 \in \mathbb{Z}_q$, and set $g_1 =$

[3] Note that we are using both an existentially unforgeable signature scheme, as well as a selective-ID IBE scheme that has been "retasked" as signature scheme. The latter leads to a signature that is only secure for a polynomial-sized, fixed message space. In the full version, we show that this limitation is acceptable given that we are signing the product of other messages which have been signed using the stronger signature scheme. Since there are at most a polynomial number of such products, the construction is secure.

$u_1^{y_1}, g_2 = u_2^{y_2}$ (and similarly for $\tilde{g}_1, \tilde{g}_2$). Fix $\mathsf{crs} \leftarrow (\gamma, GS'_S, GS'_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$.

2. Generate $(vk_1, sk_1), (vk_2, sk_2), (vk_3, sk_3)$ as in normal operation. Set $pk = (u_1, u_2, \tilde{u}_1, \tilde{u}_2, vk_1, vk_2, vk_3)$.
3. For $i = 1$ to N, choose fresh random $s, t_1, t_2 \in \mathbb{Z}_q$ and set $c_1 = g^{as} g^{st_1}, c_2 = f^{bs} f^{st_2}$. Set C_i:
$$C_i = (c_1, c_2, c_1^{y_1}, c_2^{y_2}, z_d^s h^{s(t_1+t_2)} m_j, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$$
where $\mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3$ are generated normally using the proper secret keys.
4. Set $T \leftarrow (pk, C_1, \ldots, C_N)$.
5. The simulation answers requests from the malicious Receiver by extracting from its proof and simulating correct responses (as described above.)

Note that in the above, if $z_d = h^{a+b}$, then the above simulation perfectly encrypts $(m_1, \ldots, m_N)$. However, when z_d is a random element of $\mathbb{G}_1$, then the ciphertexts correspond to encryptions of random elements in $\mathbb{G}_1$. Now, suppose for the sake of contradiction, that there exists an environment $\mathcal{Z}$ who can distinguish case one from case two with non-negligible probability ϵ. Then, it is easy to see that we can use $\mathcal{Z}$ to decide Decision Linear.

We will now present the full security proof.

5.2.2.2 Security Proof

Proof of Theorem 5.2.1. Let $\mathcal{A}$ be a static adversary that interacts with parties $\mathbf{S}, \mathbf{R}$ running protocol OTA parameterized with the algorithms of section 5.2. We construct an adversary $\mathcal{S}$ for the ideal functionality $\mathcal{F}_{OT}^{N\times 1}$. $\mathcal{S}$ begins by invoking a copy of $\mathcal{A}$ and running a simulated interaction with the environment $\mathcal{Z}$ and the parties running the protocol. $\mathcal{S}$ proceeds as follows.

Simulating the communication with $\mathcal{Z}$. Every input value that $\mathcal{S}$ receives from $\mathcal{Z}$ is written into the adversary $\mathcal{A}$'s input tape. Similarly, every output value written by $\mathcal{A}$ on its output tape is copied to $\mathcal{S}$'s own output tape (to be read by $\mathcal{S}$'s environment $\mathcal{Z}$).

Simulating the case where only $\mathbf{R}$ is corrupted. Let $\gamma \leftarrow \mathsf{BMsetup}(1^\kappa)$, then compute $(GS'_S, td_{sim}) \leftarrow \mathsf{GSSimulateSetup}(\gamma)$ and $(GS'_R, td_{ext}) \leftarrow \mathsf{GSExtractSetup}(\gamma)$. Generate the remaining elements of crs normally, and set $\mathsf{crs} \leftarrow (\gamma, GS'_S, GS'_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$. When the parties query $\mathcal{F}_{CRS}$, return $(\mathsf{sid}, \mathsf{crs})$.

$\mathcal{S}$ initiates the communication with $\mathcal{A}$ by generating a random message database $\hat{m}_1, \ldots, \hat{m}_N \xleftarrow{\$} \mathbb{G}_1$, computing $T \leftarrow \mathsf{OTInitialize}(\mathsf{crs}, \hat{m}_1, \ldots, \hat{m}_N)$ and sending (sid, T) to $\mathcal{A}$ as if from $\mathbf{S}$. Next, whenever $\mathcal{A}$ outputs (sid, Q), $\mathcal{S}$ performs as follows. First, it parses Q as (d_1, d_2, π) and (if π is valid) computes $\mathsf{GSExtract}(\mathsf{crs}, td_{ext}, \pi)$ to extract a satisfying witness $W = (\omega_1, \omega_2, \omega_3, \omega_4, \ldots)$. Parse T as $(pk, C_1, \ldots, C_N)$, and for each ciphertext $C_i = (c_1, c_2, \ldots)$ determine whether $(\omega_1, \omega_2) = (c_1, c_2)$. If no matching ciphertext is found

(or multiple ciphertexts match), then $\mathcal{S}$ aborts the simulation and gives no further messages to $\mathcal{A}$.

Otherwise, let σ' be the index of the matching ciphertext: $\mathcal{S}$ sends $(\mathsf{sid}, \mathsf{receiver}, \sigma')$ to $\mathcal{F}_{OT}^{N\times 1}$. When $\mathcal{F}_{OT}^{N\times 1}$ outputs $(\mathsf{sid}, m_{\sigma'})$ for $m_{\sigma'} \neq \perp$, $\mathcal{S}$ formulates the response $s = (c_5\omega_3\omega_4)/m_{\sigma'}$ and— using the simulation trapdoor td_{sim}— simulates the zero-knowledge proof δ' indicating that s is correctly formed according to the statement defined in the $\mathsf{OTRespond}$ algorithm (see Lemma 5.2.8 for details on simulating this proof). $\mathcal{S}$ then sends $R \leftarrow (s, \delta')$ to $\mathcal{A}$ as if from $\mathbf{S}$. $\mathcal{S}$ repeats this process for each request received from $\mathcal{A}$.

Simulating the case where only $\mathbf{S}$ is corrupted. Our simulation proceeds as follows. Let $\gamma \leftarrow \mathsf{BMsetup}(1^\kappa)$, then compute $GS_S \leftarrow \mathsf{GSSetup}(\gamma)$ and $GS_R \leftarrow \mathsf{GSSetup}(\gamma)$. Select $g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h}$ such that $g_1^{y_1} = g_2^{y_2} = h$ (and $\tilde{g}_1^{y_1} = \tilde{g}_2^{y_2} = \tilde{h}$) for (y_1, y_2) known to the simulator. Set $\mathsf{crs} \leftarrow (\gamma, GS_S, GS_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$. When the parties query $\mathcal{F}_{CRS}$, return $(\mathsf{sid}, \mathsf{crs})$.

$\mathcal{S}$ activates $\mathcal{A}$ and receives the message (sid, T) that would be $\mathcal{A}$'s first move in a real execution with $\mathbf{R}$. $\mathcal{S}$ verifies that T is correctly-structured, using the public check described in §5.2. (If T does not pass this check, $\mathcal{S}$ will instruct $\mathcal{F}_{OT}^{N\times 1}$ to fail on all message requests from $\mathbf{R}$.) Otherwise, $\mathcal{S}$ parses T as $(pk, C_1, \ldots, C_N)$ and for $i = 1$ to N first parses ciphertext C_i into $(c_1, c_2, c_3, c_4, c_5, \ldots)$, then computes $m'_i \leftarrow c_5/(c_3^{y_1} c_4^{y_2})$. $\mathcal{S}$ decodes each $m'_1, \ldots, m'_N$ to a value in $\{0,1\}^\ell$ and sends $(\mathsf{sid}, \mathsf{sender}, m'_1, \ldots, m'_N)$ to $\mathcal{F}_{OT}^{N\times 1}$.

Whenever $\mathcal{F}_{OT}^{N\times 1}$ outputs (sid) to the dummy $\mathbf{S}$ (indicating that $\mathbf{R}$ has initiated a transfer request), $\mathcal{S}$ computes $(Q, Q_{priv}) \leftarrow \mathsf{OTRequest}(\mathsf{crs}, T, 1)$ and hands (sid, Q) to $\mathcal{A}$ as if from $\mathbf{R}$. When $\mathbf{S}$ returns (sid, R), $\mathcal{S}$ checks whether $\mathsf{OTComplete}(\mathsf{crs}, T, R, Q_{priv}) = \perp$. If so, then $\mathcal{S}$ sets $b \leftarrow 0$, and $b \leftarrow 1$ otherwise. $\mathcal{S}$ returns (sid, b) to $\mathcal{F}_{OT}^{N\times 1}$.

Simulating the case where neither party is corrupted. When $\mathcal{S}$ receives k messages of the form (sid, b_i) indicating that transfers have occurred, $\mathcal{S}$ generates a simulated transcript between the honest $\mathbf{S}$ and $\mathbf{R}$. In this case, $\mathcal{S}$ runs the protocol as specified, using as $\mathbf{S}$'s input the random database $(\hat{m}_1, \ldots, \hat{m}_N)$, and (for each transfer), $\mathbf{R}$'s input $\sigma = 1$. If in the i^{th} transfer $b_i = 0$ then $\mathbf{S}$'s responds with an invalid R (the empty string). Else, $\mathbf{S}$ returns a valid response as in the protocol.

Simulating the case where both parties are corrupted. In this case $\mathcal{S}$ knows the inputs to $\mathbf{S}$ and $\mathbf{R}$ and can simulate a protocol execution by generating the real messages exchanged between the two parties.

We now address the environment's ability to distinguish the ideal execution from the real protocol execution. This is shown via the following claims.

Claim 5.2.2 *When $\mathcal{A}$ corrupts only $\mathbf{R}$, then $\mathsf{IDEAL}_{\mathcal{F}_{OT}^{N\times 1},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$ under the Decision Linear and N-*Hidden LRSW *assumptions.*

Proof. Consider the simulation described above. We will begin with the real-world protocol execution, where $\mathbf{R}$ interacts with an honest $\mathbf{S}$ that knows the message database. We

will then show via a series of hybrids that the real execution transcript is computationally indistinguishable from the simulated transcript. For notational convenience, we define $\mathbf{Pr}[\,\mathbf{Game\ i}\,]$ as the probability that environment $\mathcal{Z}$ distinguishes the transcript of **Game** i from that of the real execution. We now describe the cases:

Game 0. This is the real-world protocol execution, where $\mathbf{R}$ interacts with an honest $\mathbf{S}$ running protocol OTA on message database $(m_1, \ldots, m_N)$. Clearly $\mathbf{Pr}[\,\mathbf{Game\ 0}\,] = 0$.

Game 1 (Parameter switching). This execution proceeds as above, except that we compute $(GS'_S, td_{sim}) \in \mathsf{GSSimulateSetup}(\gamma)$, $(GS'_R, td_{ext}) \in \mathsf{GSExtractSetup}(\gamma)$, and substitute GS'_S, GS'_R in place of the honestly-generated parameters GS_S, GS_R (td_{sim}, td_{ext} are not revealed). When the parties query $\mathcal{F}_{CRS}$, return $\mathsf{crs} = (\gamma, GS'_S, GS'_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$. Note that if the SXDH assumption holds in $\mathbb{G}_1, \mathbb{G}_2$, then $(GS'_S, GS'_R) \stackrel{c}{\approx} (GS_S, GS_R)$ (Lemma 5.2.6) and thus $|\mathbf{Pr}[\,\mathbf{Game\ 1}\,] - \mathbf{Pr}[\,\mathbf{Game\ 0}\,]| \leq \nu_1(\kappa)$.

Game 2 (Extracting R's selections). This execution proceeds as above, except that for transfer phase $i = 1$ to k, we compute a candidate for $\mathbf{R}$'s selection σ'_i by extracting from its PoK π. Parse $\mathbf{R}$'s i^{th} request (sid, Q_i) to obtain (d_1, d_2, π) and (provided that π is valid) run $\mathsf{GSExtract}(\mathsf{crs}, td_{ext}, \pi)$ to extract a satisfying witness $W = (\omega_1, \omega_2, \omega_3, \omega_4, \ldots)$. Parse T as $(pk, C_1, \ldots, C_N)$, and for each ciphertext $C_i = (c_1, \ldots, c_9)$ determine whether $(\omega_1, \omega_2) = (c_1, c_2)$. Let σ'_i be the index of the matching ciphertext. If no matching ciphertext is found (or multiple ciphertexts match), then output EXTRACT-FAIL to $\mathcal{Z}$ and send no further messages to $\mathbf{R}$. By Lemma 5.2.7, this event will occur with negligible probability under the N-*Hidden LRSW* assumption; thus $|\mathbf{Pr}[\,\mathbf{Game\ 2}\,] - \mathbf{Pr}[\,\mathbf{Game\ 1}\,]| \leq \nu_2(\kappa)$.

Game 3 (Simulating S's responses). This execution proceeds as above, except that we will formulate each of $\mathbf{S}$'s transfer responses independently of sk. We parse $C_{\sigma'_i}$ to obtain $(c_1, \ldots, c_5)$ and compute $s' = (c_5\omega_3\omega_4)/m_{\sigma'_i}$. Let S be the statement proved by the Sender during the $\mathsf{OTRespond}$ algorithm: we compute a simulated PoK $\delta' \leftarrow \mathsf{GSSimProve}(GS_S, td_{sim}, S)$ and set $R' \leftarrow (s', \delta')$. Note that in order to simulate a response during transfer i, it is only necessary to know the subset of messages, $(m_{\sigma'_1}, \ldots, m_{\sigma'_i})$. By Lemma 5.2.8, the transcript including these responses is computationally indistinguishable from the distribution with valid PoKs. Thus $|\mathbf{Pr}[\,\mathbf{Game\ 3}\,] - \mathbf{Pr}[\,\mathbf{Game\ 2}\,]| \leq \nu_3(\kappa)$.

Game 4 (Substituting the ciphertexts). This execution proceeds as above, except that we replace $\mathbf{S}$'s first message with (sid, T') where $T' \in \mathsf{OTInitialize}(\mathsf{crs}, \hat{m}_1, \ldots, \hat{m}_N)$ for $\hat{m}_1, \ldots, \hat{m}_N \stackrel{\$}{\leftarrow} \mathbb{G}_1$. For $i = 1$ to k, we also modify the i^{th} transfer phase such that $\mathbf{S}$'s response is (sid, R'_i) for $R'_i = (s', \delta')$

computed as in **Game 3**, except that we must now compute the PoK δ on a possibly invalid statement S. By Lemma 5.2.9, the hardness of the Decision Linear problem implies that the distribution of messages is indistinguishable from the real execution, even though s' may be *incorrectly* formed with respect to S. Thus $|\mathbf{Pr}[\mathbf{Game\ 4}] - \mathbf{Pr}[\mathbf{Game\ 3}]| \leq \nu_4(\kappa)$.

Notice that the distribution produced in $\mathbf{Game\ 4}$ is identical to that of our simulation. By summation, we have that $\mathbf{Pr}[\mathbf{Game\ 4}] \leq \nu_5(\kappa)$ and thus $\mathsf{IDEAL}_{\mathcal{F}_{OT}^{N\times 1},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$ under the N-Hidden LRSW and Decision Linear assumptions.

□

Claim 5.2.3 *When $\mathcal{A}$ corrupts only* $\mathbf{S}$*, then* $\mathsf{IDEAL}_{\mathcal{F}_{OT}^{N\times 1},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$ *under the N-Hidden LRSW assumptions.*

Proof. Consider the simulation described above. Again we begin with the real-world protocol execution, where $\mathbf{S}$ interacts with an honest $\mathbf{R}$ that chooses messages according to an arbitrary selection strategy Σ. We then show via a series of hybrids that the real execution transcript is computationally indistinguishable from the simulated transcript.

Game 0. This is the real-world protocol execution, where $\mathbf{S}$ interacts with an honest $\mathbf{R}$ running protocol OTA using selection strategy Σ. Clearly $\mathbf{Pr}[\mathbf{Game\ 0}] = 0$.

Game 1 (Parameter generation). This execution proceeds as above, except that we select elements of crs such that $g_1^x = g_2^y = h$ (and $\tilde{g}_1^x = \tilde{g}_2^y = \tilde{h}$) for known (x, y). When the parties query $\mathcal{F}_{CRS}$, return $\mathsf{crs} = (\gamma, GS_S, GS_R, g_1, g_2, h, \tilde{g}_1, g_2, h)$. Note that the distribution of crs is identical to the normal distribution. Thus $|\mathbf{Pr}[\mathbf{Game\ 1}] - \mathbf{Pr}[\mathbf{Game\ 0}]| = 0$.

Game 2 (Substituting R's queries). Next, during transfer $i = 1$ to k, we modify the transcript by generating $Q_i' \leftarrow \mathsf{OTRequest}(T, 1)$ and replacing $\mathbf{R}$'s request with (sid, Q_i'). Let $Q' = (d_1', d_2', \pi')$. Observe that for any $i \in [1, N]$, where $C_i = (c_1, \ldots, c_5, \ldots)$, we can express d_1', d_2' as $c_1 u_1^{v_1'}, c_2 u_2^{v_2'}$ for some v_1', v_2'. Thus for every C_i there exists a witness $(c_1, c_2, h^{v_1'}, h^{v_2'}, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$ that satisfies the pairing product equation S_π. By the Witness-Indistinguishability property of the Groth-Sahai proof system, the value Q_1' is indistinguishable from a request formed on a different $\sigma_j \in [1, N]$. Thus $|\mathbf{Pr}[\mathbf{Game\ 2}] - \mathbf{Pr}[\mathbf{Game\ 1}]| \leq \nu_1(\kappa)$.

Game 2 has an identical distribution to our simulation, and $\mathbf{Pr}[\mathbf{Game\ 2}] \leq \nu_1(\kappa)$. It remains to show that the in our simulation the distribution of messages obtained by an ideal $\mathbf{R}$ interacting with $\mathcal{F}_{OT}^{N\times 1}$ are identical to the messages recovered by an honest $\mathbf{R}$ running the protocol directly with $\mathbf{S}$. This implies that for every set of indices $(\sigma_1, \ldots, \sigma_k)$ the plaintexts $(m_{\sigma_1}', \ldots, m_{\sigma_k}')$ obtained by $\mathcal{S}$— which decrypts the ciphertexts in T with

the trapdoor (x, y)— are identical to the messages recovered by an honest $\mathbf{R}$ running the protocol with $\mathbf{S}$.

$\mathbf{S}$'s initial output T embeds $pk = (u_1, u_2, \dots)$. Let (a, b) be $\mathbf{S}$'s secret key, which is implicitly defined by $u_1^a = u_2^b = h$. We observe that if T passes the validity check run by honest $\mathbf{R}$, then each ciphertext C_i can be expressed as $(u_1^r, u_2^s, g_1^r, g_2^s, m_i h^{r+s}, \dots)$ for some $r, s \in \mathbb{Z}_q$ and $m_i \in \mathbb{G}_1$. Since the simulator constructed g_1, g_2 such that $g_1^x = g_2^y = h$ then it necessarily holds that $(p_1^{ra} p_2^{sb}) = (g_1^{rx} g_2^{sy}) = h^{r+s}$. Let us consider an honest $\mathbf{R}$ that requests index i from $\mathbf{S}$. It selects $v_1, v_2 \in \mathbb{Z}_q$ and sets $d_1 = u_1^r u_1^{v_1}, d_2 = u_2^s u_2^{v_2}$, sending request $Q = (d_1, d_2, \pi)$. Let $R = (s, \delta)$ be the response from $\mathbf{S}$. If PoK δ verifies, then (by the soundness property of the proof system) with all but negligible probability $s = d_1^a d_2^b$ and the honest $\mathbf{R}$ computes the message as $(m_i h^{s+r})/(d_1^a d_2^b h^{-v_1} h^{-v_2}) = m_i$. This is identical to the decryption obtained by $\mathcal{S}$ using the trapdoor (x, y), which produces $h^{s+r}/(g_1^{rx} g_2^{sy}) = m_i$. Thus, the distribution of messages given to $\mathcal{F}_{OT}^{N\times 1}$ by $\mathcal{S}$ is indistinguishable from the distribution of messages obtained by running the protocol directly with $\mathbf{S}$.

□

Claim 5.2.4 *When $\mathcal{A}$ corrupts neither $\mathbf{S}$ nor $\mathbf{R}$, then* $\mathsf{IDEAL}_{\mathcal{F}_{OT}^{N\times 1},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$ *under the Decision Linear and N-*Hidden LRSW *assumptions.*

We omit a formal proof of this claim, but note that it re-uses techniques identical to those of the previous claims. Specifically, we replace the Sender's initial message T with a commitment to a random database, and show that this random database is indistinguishable from a real database under the Decision Linear assumption (as in Claim 5.2.2). We then argue that by the Witness-Indistinguishability property of the Groth-Sahai proof system, the extractions on message index 1 are indistinguishable from extractions on other message indices (as in Claim 5.2.3).

Claim 5.2.5 *When $\mathcal{A}$ corrupts both $\mathbf{S}$ and $\mathbf{R}$, then* $\mathsf{IDEAL}_{\mathcal{F}_{OT}^{N\times 1},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$.

We omit a formal proof of this claim.

Lemma 5.2.6 *Under the SXDH assumption (implied by N-*Hidden LRSW*, the parameters generated by* GSSetup *(and* GSExtractSetup*) are computationally indistinguishable from those produced by* GSSimulateSetup.

We refer the reader to the work of Groth and Sahai [GS08] for a proof of this theorem.

Lemma 5.2.7 *Under the N-*Hidden LRSW *and co-CDH*[4] *assumptions, the probability that* $\mathbf{S}$ *outputs* EXTRACT-FAIL *in* **Game 3** *is negligible.*

[4]Computational co-Diffie-Hellman (CDH) is implied by *N-Hidden LRSW*; thus, no new assumptions are being introduced here.

Proof sketch. Let $T = (pk, C_1, \ldots, C_N)$ be honestly-generated as in **Game 3**. Consider $\mathcal{A}$'s request (sid, Q_i) at transfer $i \in [1, k]$, and parse Q_i as (d_1, d_2, π) where π is a PoK (of the statement described in the definition of $\mathsf{OTRequest}$) using parameters GS'_R. Note that the simulator knows the trapdoor td_{ext} corresponding to GS'_R, and can therefore extract a satisfying witness $W = (\omega_1, \omega_2, \ldots) \leftarrow \mathsf{GSExtract}(GS_R, td_{ext}, \pi)$ (in the general case, extraction succeeds with probability $\geq 1 - \nu(\kappa)$ by the Soundess property of the Groth-Sahai proof system).[5] Since extraction fails with at most negligible probability, then if $\mathbf{S}$ outputs EXTRACT-FAIL with non-negligible probability, then it must be that that for $j \in [1, N]$ there is either (a) *no* single ciphertext $C_j = (c_{j,1}, \ldots, c_{j,5}, \ldots)$ such that $(\omega_1, \omega_2) = (c_{j,1}, c_{j,2})$, *or* (b) there are multiple ciphertexts for which the relation holds.

We can easily dispose of case (b): since T is honestly generated, then for each ciphertext $(c_{j,1}, \ldots, c_{j,5}, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$, the values $c_{j,1}, c_{j,2}$ are uniformly distributed in $\mathbb{G}_1$. Therefore, the probability is negligible that any two distinct ciphertexts are identical in the first two elements. This it remains only to address case (a) where there is no ciphertext C_j such that $(c_{j,1}, c_{j,2}) = (\omega_1, \omega_2)$. This condition can be further divided into two sub-cases:

1. Where for $i \neq j$ there exists some pair of ciphertexts C_i, C_j such that $\omega_1 = c_{i,1}$ and $\omega_2 = c_{j,2}$.
2. Where there is no pair of ciphertexts such that the above condition holds, *i.e.,* either ω_1 or ω_2 is not contained within any ciphertext in T.

We now show that if $\mathcal{A}$ outputs a PoK satisfying condition (1) then we can use its response to solve the co-CDH problem, and if $\mathcal{A}$ satisfies condition (2) we can solve N-*Hidden LRSW*. We now describe each of the two simulations:

Case 1: co-CDH. We consider the case where $\mathcal{A}$ produces $(\omega_1, \omega_2) = (c_{i,1}, c_{j,2})$ for $i \neq j$, and show that an $\mathcal{A}$ that produces such a query can be used to solve the Computational co-Diffie-Hellman problem in $\mathbb{G}_1, \mathbb{G}_2$, *i.e.,* given $(g, g^a, g^b, \tilde{g}, \tilde{g}^a, \tilde{g}^b)$ for $a, b \in_R \mathbb{Z}_q$, solve for g^{ab}. The intuition behind this argument is that the final component sig_3 is a signature on the product $(c_1 c_2)$. This signature is built from the Boneh-Boyen selective-ID IBE scheme from [BB04a] (§4), and a forger of this scheme can be used to solve the co-CDH problem in asymmetric bilinear groups.[6] Our reduction is based on the one given by Boneh and Boyen, although we reduce to co-CDH. Since the N-*Hidden LRSW* assumption implies the hardness of co-CDH, we are not introducing a new assumption.

Given an input $(g, g^a, g^b, \tilde{g}, \tilde{g}^a, \tilde{g}^b)$ to the co-CDH problem: select random values $u, v, w, y \xleftarrow{\$} \mathbb{Z}_q$. Set $(u_1, u_2) \leftarrow (g^a, g^{au})$, $(\tilde{u}_1, \tilde{u}_2) \leftarrow (\tilde{g}^a, \tilde{g}^{au})$ and $h' \leftarrow (g^a)^{-v} g^w$. Generate $(vk_1, sk_1), (vk_2, sk_2)$ as in the normal scheme, but set $vk_3 = (\gamma, g, \tilde{g}, g^a, g^b, h', \tilde{g}^b)$.

[5] In fact, for parameters $GS'_R \in \mathsf{GSExtractSetup}()$, Groth-Sahai proofs are *perfectly* extractable [GS08].

[6] Note that a selective-ID IBE scheme implies a secure signature scheme *only* if the message space is polynomial in κ. Since in this case $\mathcal{A}$ succeeds by proving knowledge of a signature on message $(c_{i,1} c_{j,2})$ for some $i \neq j$, we have naturally restricted the total number of valid messages to $N^2 - N$.

Set $pk \leftarrow (u_1, u_2, \tilde{u}_1, \tilde{u}_2, vk_1, vk_2, vk_3)$. Randomly select two ciphertext indices i^*, j^* such that $i^* \neq j^*$.

Now for $i = 1$ to N, choose r_i, s_i, y_i uniformly from $\mathbb{Z}_q$ with the restriction that $(r_{i^*} + us_{j^*}) = v \bmod p$. Set $z_i = (r_i + us_i) \bmod p$. Generate $\mathsf{sig}_1 \leftarrow \mathsf{CLNSign}_{sk_1}(u_1^r), \mathsf{sig}_2 \leftarrow \mathsf{CLNSign}_{sk_2}(u_2^s)$, and set $\mathsf{sig}_3 \leftarrow \left((g^b)^{\frac{-w}{z_i - v}}((g^a)^{z_i - v} g^w)^{y_i}, (\tilde{g}^b)^{\frac{-1}{z_i - v}} \tilde{g}^{y_i}, (g^b)^{\frac{-1}{z_i - v}} g^{y_i}\right)$. Construct the i^{th} ciphertext as:

$$C_i = \left(u_1^{r_i}, u_2^{s_i}, g_1^{r_i}, g_2^{s_i}, m_j h^{r_i + s_i}, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3\right)$$

(Note that the sig_3 has the correct distribution. Let $\hat{y}_i = y_i - b/(z_i - v)$, and re-write $(\tilde{g}^b)^{\frac{-1}{z_i - v}} \tilde{g}^{y_i} = \tilde{g}^{y_i - b/(z_i - v)} = \tilde{g}^{\hat{y}_i}$ (and similarly for the third element). We can then express the first element $(g^b)^{\frac{-w}{z_i - v}}((g^a)^{z_i - v} g^w)^{y_i}$ as $(g^b)^a \left((g^a)^{z_i - v} g^w\right)^{y_i - \frac{-b}{z_i - v}} = (g^{ab}) \left(g^{az_i} h'\right)^{\hat{y}_i} = g_1^a((g^a)^{r_i + us_i} h')^{\hat{y}_i}$.)

Now set $T \leftarrow (pk, C_1, \ldots, C_N)$ and send T to $\mathcal{A}$. Whenever $\mathcal{A}$ submits a request $Q = (d_1, d_2, \pi)$ where π verifies correctly, use the extraction trapdoor to obtain the values $(\omega_1, \omega_2, \omega_3, \omega_4)$ and the values $s_1', \tilde{s}_2', s_3'$ corresponding to sig_3. Now:

1. If, for some $j \in [1, N]$, the pair $(\omega_1, \omega_2) = (u_1^{r_j}, u_2^{s_j})$: then output a valid response to $\mathcal{A}$ by selecting $s' = (h^{r_j + s_j} \omega_3 \omega_4)$, constructing the proof δ', and sending $R = (s', \delta')$ to $\mathcal{A}$.[7] Continue the simulation.
2. If $(\omega_1, \omega_2) = (u_1^{r_{i^*}}, u_2^{s_{j^*}})$, then compute $s_1'/s_3'^w$ as the solution to the co-CDH problem.
3. In all other cases, abort the simulation.

Observe that in case (2) the soundness of the G-S proof system ensures that for some y' we can represent $(s_1', \tilde{s}_2', s_3') = ((g^a)^v h)^{y'} g^{ab}, \tilde{g}^{y'}, g^{y'})$. By substitution we obtain $((g^a)^v (g^a)^{-v} g^w)^{y'} g^{ab}, \tilde{g}^{y'}, g^{y'}) = (g^{wy'} g^{ab}, \tilde{g}^{y'}, g^{y'})$, and thus $s_1'/s_3'^w = g^{ab}$. In this case, we can obtain the value g^{ab} and output a correct solution to the co-CDH problem.

Note that the distribution of the messages sent to $\mathcal{A}$ is identical to that of the real attack, and are independent of i^*, j^* in $\mathcal{A}$'s view. Therefore, if $\mathcal{A}$ produces $(\omega_1, \omega_2) = (c_{i',1}, c_{j',2})$ for $i' \neq j'$ with some non-negligible probability ϵ, then the approach above solves co-CDH with probability approximately $\frac{\epsilon}{N^2 - N}$, *i.e.,* the probability that that $(i', j') = (i^*, j^*)$.

Case 2: *N-Hidden LRSW.* In the case where either ω_1 or ω_2 is not contained within any ciphertext in T, then we will construct a solver for the *N-Hidden LRSW* problem. Our simulation proceeds as follows: given the *N-Hidden LRSW* instance $(g, \tilde{g}, S, T, \{b_1, b_1^{s + a_1 st}, b_1^{a_1}, b_1^{a_1 t}, g^{a_1}, \tilde{b}_1\}, \ldots, \{b_q, b_q^{s + a_q st}, b_q^{a_q}, b_q^{a_q t}, g^{a_q}, \tilde{b}_q\})$, randomly select $s \in \{1, 2\}$ representing one of the following two strategies.

[7] Note that we can simulate the proof δ', but this is not even necessary, since we can construct a valid witness to the statement.

Strategy 1. Select a secret key $sk = (x_1, x_2) \xleftarrow{\$} \mathbb{Z}_q^2$ for the OT scheme and select $u_1, u_2, u_1, u_2, h, \tilde{h}$ such that $u_1 = g$, $\tilde{u}_1 = \tilde{g}$, $u_1^{x_1} = u_2^{x_2} = h$ and $\tilde{u}_1^{x_1} = \tilde{u}_2^{x_2} = \tilde{h}$. Select values $(g_1, g_2, \tilde{g}_1, \tilde{g}_2)$ for crs such that $g_1 = u_1^t$ for random $t \in \mathbb{Z}_q$, and g_2 is a random element. Generate $(vk_2, sk_2), (vk_3, sk_3)$ as in the normal scheme, and set $vk_1 = (\gamma, g, \tilde{g}, S, T)$. To compute each ciphertext, set $\mathsf{sig}_1 \leftarrow (b_j, b_j^{a_j}, b_j^{s+a_j st}, b_j^{a_j t}, \tilde{b}_j)$ and compute $\mathsf{sig}_2, \mathsf{sig}_3$ normally. Select a random $y_j \in \mathbb{Z}_q$ and set $C_j \leftarrow (g^{a_j}, u_2^{y_j}, g^{a_j t}, g_2^{y_j}, g^{a_j x_1} h^{y_j} m_j, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$.
Strategy 2. Similar to the previous strategy, but formulate vk_2 and embed g^{a_j} in the second position of C_j.

Observe that since the values $a_1, \ldots, a_N$ from the N-*Hidden LRSW* instance are uniformly distributed, then T has the correct distribution. Next, answer $\mathcal{A}$'s queries using the key sk, extracting a witness $W = (\omega_1, \omega_2, \ldots)$ from the proof π. Note that from the witness W it is possible to obtain the full value of sig_1. If ever $\mathcal{A}$ outputs a PoK π such that for Strategy $s \in \{1, 2\}$ the extracted witness ω_s does not match any $c_{j,s} \in C_j$, then extract the witness values for the proof of signature s— and output these as $\langle a'_1, a'_2, a'_3, a'_4, \tilde{a}'_5$. Otherwise abort. This tuple represents a valid solution to the N-*Hidden LRSW* problem. Since all values are correctly distributed and s is outside of $\mathcal{A}$'s view, then we select the correct strategy with probability $1/2$.

To conclude our sketch, note that we have covered all cases where event EXTRACT-FAIL can occur. Thus if the event occurs with probability non-negligible in κ then we have an algorithm that solves N-*Hidden LRSW* or CDH with non-negligible probability.

□

Lemma 5.2.8 *Replacing* **S***'s honestly-generated responses (as in* **Game 2***) with simulated responses (as in* **Game 3***) results in a simulation that is computationally indistinguishable from that of* **Game 2***.*

Proof sketch. Consider a transcript where each response (sid, R) is replaced with a simulated response (sid, R'). Let $R = (s, \delta)$ be the honestly-generated response, and let $R' = (s', \delta')$ be the simulated response. To complete our argument, we must show that for any given reponse: (1) with probability at most $\nu(\kappa)$, the value $s \neq s'$, and (2) the PoK $\delta \overset{c}{\approx} \delta'$. This must hold for all $\mathcal{A}, \mathcal{Z}$.

Recall that pk embeds u_1, u_2 such that $u_1^{x_1} = u_2^{x_2} = h$ for some $x_1, x_2 \in \mathbb{Z}_q$. $\mathcal{A}$ initiates the transfer by sending a message (sid, Q) containing the values (d_1, d_2, π). Using the extraction algorithm, we obtain a witness $W = (\omega_1, \omega_2, \omega_3, \omega_4, \ldots)$ to the statement S_π. Note that a correctly-formed response will have the form $s = (d_1^{x_1} d_2^{x_2})$, and for $C_{\sigma'} = (c_1, \ldots, c_5, \ldots)$ a simulated response has the form $s' = (c_5 \omega_3 \omega_4)/m_{\sigma'}$, which we expand to $s' = (c_1^{x_1} c_2^{x_2} m_{\sigma'} h^{v_1} h^{v_2})/m_{\sigma'}$ for some (v_1, v_2). We omit a detailed expansion, but observe that by the statement S_π it holds that $d_1 = c_1 h^{v_1/x_1}$ and $d_2 = c_2 h^{v_2/x_2}$ and thus our simulated s' is identical to the correct response s.

Paraphrasing the composable zero-knowledge property of the Groth-Sahai proof system, when $(GS'_S, td_{sim}) \in \mathsf{GSSimulateSetup}()$ we can simulate a PoK $\delta' \leftarrow \mathsf{GSSimProve}(GS'_S, S_\delta)$ such that no adversary can distinguish δ' from a valid PoK. It is easy to show that we can simulate the statement S_δ. Recall that the δ is defined as:

$$\delta = NIZK_{GS_S}\{(a_1, a_2, \tilde{a}_3) : e(a_1, \tilde{u}_1)e(d_1^{-1}, \tilde{a}_3) = 1 \ \wedge \\ e(a_2, \tilde{u}_2)e(d_2^{-1}, \tilde{a}_3) = 1 \ \wedge \ e(a_1 a_2, \tilde{a_3})e(s^{-1}, \tilde{a}_3) = 1 \ \wedge \ e(u_1, \tilde{a}_3) = e(u_1, \tilde{h})\}$$

To simulate the proof, we must select commitments to represent $a_1, a_2, \tilde{a}_3$, and we then compute opening values such that each statement is satisfied. Note that using the simulation trapdoor td_{sim} we may open the commitment differently in each statement. To simulate a proof δ', set $a_1 = a_2 = a_3 = h^0$ and generate commitments to each value. In the first three statements, we open the third commitment to h^0. In the final statement, we use the simulation trapdoor to open the third commitment to h^1. Thus, all statements are satisfied. □

Lemma 5.2.9 *Let $m_1, \ldots, m_N \in \mathbb{G}_1$ be any message database, and $\hat{m}_1, \ldots, \hat{m}_N \in_R \mathbb{G}_1$ be a set of random messages. Also let all of* **S***'s responses be computed as in* **Game 3***. Under the Decision Linear assumption, no environment $\mathcal{Z}$ will distinguish the transcript where $T = \mathsf{OTInitialize}(\mathsf{crs}, m_1, \ldots, m_N)$ from the transcript where $T = \mathsf{OTInitialize}(\mathsf{crs}, \hat{m}_1, \ldots, \hat{m}_N)$ (except with negligible probability).*

Proof sketch. Let $D = (g, \tilde{g}, f, \tilde{f}, h, \tilde{h}, g^a, f^b, z_d)$ be a candidate Decision Linear tuple. Next, consider a simulation that behaves as follows:

1. Set $u_1 = g, u_2 = f, \tilde{u}_1 = \tilde{g}, \tilde{u}_2 = \tilde{f}$. Select random $y_1, y_2 \in \mathbb{Z}_q$, and set $g_1 = u_1^{y_1}, g_2 = u_2^{y_2}$ (and similarly for $\tilde{g}_1, \tilde{g}_2$). Fix $\mathsf{crs} \leftarrow (\gamma, GS'_S, GS'_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$.
2. Generate $(vk_1, sk_1), (vk_2, sk_2), (vk_3, sk_3)$ as in normal operation. Set $pk = (u_1, u_2, \tilde{u}_1, \tilde{u}_2, vk_1, vk_2, vk_3)$.
3. For $i = 1$ to N, choose fresh random $s, t_1, t_2 \in \mathbb{Z}_q$ and set $c_1 = g^{as} g^{st_1}, c_2 = f^{bs} f^{st_2}$. Set C_i:
$$C_i = (c_1, c_2, c_1^{y_1}, c_2^{y_2}, z_d^s h^{s(t_1+t_2)} m_j, \mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3)$$
where $\mathsf{sig}_1, \mathsf{sig}_2, \mathsf{sig}_3$ are generated normally using the appropriate secret keys.
4. Set $T \leftarrow (pk, C_1, \ldots, C_N)$.
5. The simulation proceeds as in **Game 3** at answer transfer requests.

Note that in the above, if $z_d = h^{a+b}$, then the above simulation perfectly encrypts $(m_1, \ldots, m_N)$. However, when z_d is a random element of $\mathbb{G}_1$, then the ciphertexts correspond to encryptions of random elements in $\mathbb{G}_1$. Now, suppose for the sake of contradiction, that there exists a $\mathcal{Z}$ who can distinguish case one from case two with non-negligible probability ϵ. Then, it is easy to see that we can use $\mathcal{Z}$ to decide Decision Linear. □

□

5.2.3 Sampling from a Common Random String

We briefly note that by the same arguments used above, the Reference String used in our construction can be replaced with a Common Random String. Note that crs embeds $(\gamma, GS_S, GS_R, g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h})$, for $GS_R, GS_S \in \mathsf{GSSetup}(\gamma)$ and $g_1, g_2, h, \tilde{g}_1, \tilde{g}_2, \tilde{h} \in_R \mathbb{G}_1^3 \times \mathbb{G}_2^3$. Each set of Groth-Sahai commitment parameters embeds a tuple in $\mathbb{G}_1$ (resp. $\mathbb{G}_2$). When GSSetup is used, the parameters are generated such that the parameters are a DDH tuple in the respective group, and when GSSimulateSetup is used, they are uniformly random. Under SXDH, the latter distribution is indistinguishable from the correct one, and thus we may sample the components of GS_S, GS_R uniformly. Since the parameters γ can be sampled from a random string [GOS06], then all elements of crs can therefore be derived from a uniformly *random* string when a source of common randomness is available.

5.3 On Multiple Receivers

Since we are motivated by the application of OT to database systems, we would also like to support applications where multiple users share a single database. Naively this can be accomplished by requiring the database to run separate OT protocol instances with each user. However, this approach can be quite inefficient, and moreover does not ensure *consistency* in the database viewed by individual Receivers. Consider a strengthening of the security definition of $\mathcal{F}_{OT}^{N\times 1}$ (in Figure 2.5) to include the additional requirement that all Receivers "view" the same database, i.e., the database owner cannot selectively alter the messages in the database when interacting with different receivers – on query σ from *any* receiver, he must return a value in $\{m_\sigma, \perp\}$. Fortunately, *consistency* is easy and inexpensive to achieve in our construction – simply alter $\mathcal{F}_{CRS}^{\mathcal{D},P}$ to return the *same* values (g_1, g_2, h) as part of the crs to *all* receivers and have the Sender publish one database commitment T to everyone, handling joint state via [CR03]. Intuitively, this captures consistency because the simulator can set the values (g_1, g_2, h) and then trapdoor decrypt all messages in T (see the description of BBS encryption above). Given the soundness of the GS proofs, all of the Sender's responses to any Receiver must be consistent with T, even if the other parts of their common reference strings are distinct. Note that it is not at all clear how *consistency* can be achieved efficiently *even in the non-adaptive setting* using prior UC results [PVW08], since there each Receiver provides her own encryption key for the Sender to bundle the messages in.

Chapter 6

Access Controls

This chapter is based on joint work with Scott Coull and Susan Hohenberger that will appear in Stanislaw Jarecki and Gene Tsudik (Ed.): The International Conference on Theory and Practice of Public-Key Cryptography - PKC 2009, Lecture Notes in Computer Science, Springer-Verlag, 2009 [CGH09].

It is a universal truth that where there is valuable information there will be a need for access controls. Content providers have long been in the habit of restricting how they hand out their data. Unfortunately, this requirement may seem to conflict with the privacy goals that we require from an Oblivious Database.

In previous chapters we have proposed several protocols for adaptive Oblivious Transfer, and promoted this primitive as a natural candidate for constructing Oblivious Databases. However, while $\mathsf{OT}^N_{k\times 1}$ provides a limited form of access control (a Receiver can obtain at most k out of N database records), such policies seem insufficient for practical applications. This raises further questions when we consider the problem of hiding a user's *identity* in a multi-user database (see §5.3).

Thus, to realize an *anonymous* and *oblivious* database for our users, we must couple it with some manner of enforceable access controls for the provider. We make two design choices that act as guiding principles for the design of our system. Our first is to maintain the strongest possible anonymity or privacy guarantees. We reject any solutions that use pseudonyms or allow for some form of transaction linking, since it is too difficult to infer what compromise to privacy might result.

Contributions. Our approach is to combine Oblivious Transfer with another important privacy-preserving primitive. Anonymous Credentials [Lys02, CL02, CL04], first proposed by Chaum, allow a user to prove certain attributes about themselves in zero-knowledge. In our protocols we will show how to embed the user's identity and a history-dependent access policy into her anonymous credential so that for each Database she can prove (in zero-knowledge) that she has the right to access the record that she is obliviously requesting.

Beyond integrating these systems, we present an extension to traditional anonymous credential systems which embeds the user's current state into the credential, and *dynamically* updates that state according to well-defined policies governing the user's actions. These *stateful anonymous credentials* are built on top of well-known signatures with efficient protocols [Lys02, CL02, CL04, BB04b]. Our constructions are secure in the standard model under basic assumptions, such as Strong RSA. Additionally, we introduce a technique for efficiently proving that a committed value lies in a *hidden* range that is unknown to the verifier, which may be of independent interest.

More importantly, we show how these components can be used to *efficiently* provide non-trivial, real-world access controls for oblivious databases. These access controls include the Brewer-Nash (Chinese Wall) [BN89] and Bell-LaPadula (Multilevel Security) [BL88] access control model, which are used in a number of settings, including financial institutions and classified government systems. In addition, we also show how to combine our anonymous credential system with several other anonymous and oblivious protocols, like blind signing protocols [CL02, CL04, GH08b] and searches over encrypted data [WBDS04]. We provide simulation-based security definitions for our stateful anonymous credentials, as well as an anonymous and oblivious database system with access controls.

Related Work. Several previous works sought to limit *anonymous* user actions, either directly within an existing protocol or through the use of anonymous credentials. Aiello, Ishai, and Reingold [AIR01] proposed *priced* oblivious transfer, in which each user is given a declining balance that is "spent" on each transfer. However, here user anonymity is not protected, and the protocol is also vulnerable to *selective-failure* attacks in which a malicious server induces faults to deduce the user's selections [NP99b, CNs07]. The more general concept of *conditional* oblivious transfer was proposed by Di Crescenzo, Ostrovsky, and Rajagopolan [COR99] and subsequently strengthened by Blake and Kolesnikov [BK04]. In conditional oblivious transfer, the sender and receiver maintain private inputs (x and y, respectively) to some publicly known predicate $q(\cdot, \cdot)$ (*e.g.,* the greater than equal to relation on integers). The items in the oblivious transfer scheme are encrypted such that the receiver can complete the oblivious transfer and recover her data if and only if $q(x, y) = 1$. In addition, techniques from e-cash and anonymous credentials have been used to place simple limitations on an anonymous user's actions, such as preventing a user from logging in more than once in a given time period [CHK$^+$06], authenticating anonymously at most k times [TFS04], or preventing a user from exchanging too much money with a single merchant [CHL06]. Rather than providing a specific type of limitation or restricting the limitation to a particular protocol, our proposed system instead provides a general method by which arbitrary access control policies can be implemented to a wide variety of anonymous and oblivious protocols.

6.1 Stateful Anonymous Credentials

The goal of typical anonymous credential systems is to provide users with a way of proving certain attributes about themselves (*e.g.,*age, or height) without revealing their identity. Users conduct this proof by obtaining a credential from some organization, and subsequently "showing" the credential without revealing their identity. A stateful anonymous credential system adds the additional notion of credential state, which the user may update over the lifetime of the credential. State updates are restricted to some well-defined *policy* dictated by the credential provider. In practice, this may limit the user to a finite number of states, or a particular ordering of states that must be arrived at in succession. The update protocol for a stateful credential must oblivious. In other words, it does not leak information about the credential's current state beyond what the user chooses to reveal. As with typical anonymous credential systems, the user's state and other attributes can be proved without revealing her identity.

At a high level, the stateful anonymous credential system, which is defined by the tuple of algorithms $(\mathsf{Setup}, \mathsf{ObtainCred}, \mathsf{UpdateCred}, \mathsf{ProveCred})$, operates as follows. First, the user and credential provider negotiate the use of a specified policy using the ObtainCred protocol. The negotiated policy determines the way in which the user will be allowed to update her credential. After the protocol completes, the user receives an anonymous credential that embeds her initial state in the policy, in addition to any other user attributes. Next, the user can prove (in zero-knowledge) that the credential she holds embeds a given state, or attribute, just as she would in other anonymous credential systems by using the ProveCred protocol. This allows the anonymous access to some service, while the entity checking the credential is assured of the user's attributes, as well as her state in the specified policy – in some cases, as we will show later, these proof can be done in such a way that the verifying entity learns nothing about the user's state or attributes. Finally, when the user wishes to update her credential to reflect a change in her state, she interacts with the credential provider using the UpdateCred protocol, during which she proves (again, in zero-knowledge) her current state and the existence of a transition in the policy from her current state to her intended next state. As with the ProveCred protocol, the provider learns nothing about the user other than the fact that her state change is allowed by the policy that was previously negotiated within the ObtainCred protocol.

Policy Model. To represent the policies for our stateful anonymous credential system, we use directed graphs, which can be thought of as a state machine that describes the user's behavior over time. We describe the *policy graph* Π_{pid} as the set of *tags* of the form $(id, S \rightarrow T)$, where id is the identity of the policy and $S \rightarrow T$ represents a directed edge from state S to state T. Thus, the user's credential embeds the identity of the policy id and the user's current state in the policy graph. When the user updates her credential, she chooses a tag, then proves that the policy id she is following is the same as what is provided in the tag and that the tag encodes an edge from her current state to her desired next state.

These policy graphs can be created in such a way that the users may reach a terminal

state, and therefore would be unable to continue updating (and consequently using) their credential. In this case, it may be possible for an adversary to perform traffic analysis to infer the policy that the user is following. To prevent this, we consider the use of *null transitions* in the graph. The null transitions occur as self-loops on the terminal states of the policy graph, and allow the user to update her credential as often as she wishes to prevent such traffic analysis attacks. However, the updates performed on these credentials only allow the user access to a predefined null resource. The specifics of this null resource are dependent on the anonymous protocol that the credential system is coupled with, and we describe an implementation for them in oblivious databases in Section 6.2.

While these policy graphs are rather simplistic, they can represent complicated policies. For instance, a policy graph can encode the user's history with respect to accessing certain resources up to the largest cycle in the graph. Moreover, we can extend the policy graph tags to include auxiliary information about the actions that the user is allowed to perform at each state. By doing so, we allow the graph to dynamically control the user's access to various resources according to her behavior and history, as well as her other attributes. In Section 6.2, we examine how to extend these policy graphs to provide non-trivial, real-world access control policies for oblivious databases, as well as a variety of other anonymous and oblivious application.

6.1.1 Protocol Descriptions and Definitions for Stateful Anonymous Credentials

A stateful anonymous credential scheme consists of four protocols: Setup, ObtainCred, UpdateCred, and ProveCred. We will now describe their input/output behavior and intended functionality.

Setup($\mathcal{U}(1^k), \mathcal{P}(1^k, \Pi_1, \ldots, \Pi_n)$)**:** The provider $\mathcal{P}$ generates parameters $params$ and a keypair $(pk_{\mathcal{P}}, sk_{\mathcal{P}})$ for the credential scheme. For each graph Π to be enforced, $\mathcal{P}$ also generates a cryptographic representation $\Pi_{\mathcal{C}}$ and publishes this value via an authenticated channel. Each user $\mathcal{U}$ generates a keypair and requests that it be certified by a trusted CA.

ObtainCred($\mathcal{U}(pk_{\mathcal{P}}, sk_{\mathcal{U}}, \Pi_{\mathcal{C}}), \mathcal{P}(pk_{\mathcal{U}}, sk_{\mathcal{P}}, \Pi_{\mathcal{C}}, S)$)**:** $\mathcal{U}$ identifies herself to $\mathcal{P}$ and then receives her credential Cred which binds her to a policy graph Π and starting state S.

UpdateCred($\mathcal{U}(pk_{\mathcal{P}}, sk_{\mathcal{U}}, \mathsf{Cred}, \Pi_{\mathcal{C}}, T), \mathcal{P}(sk_{\mathcal{P}}, \Pi_{\mathcal{C}}, D)$)**:** $\mathcal{U}$ and $\mathcal{P}$ interact such that Cred is updated from its current state to state T, but only if this transition is permitted by the policy Π. Simultaneously, $\mathcal{P}$ should not learn $\mathcal{U}$'s identity, attributes, or current state. To prevent replay attacks, $\mathcal{P}$ maintains a database D, which it updates as a result of the protocol.

ProveCred($\mathcal{U}(pk_{\mathcal{P}}, sk_{\mathcal{U}}, \mathsf{Cred}), \mathcal{P}(pk_{\mathcal{P}}, E)$)**:** $\mathcal{U}$ proves possession of a credential Cred in a particular state. To prevent re-use of credentials, $\mathcal{P}$ maintains a database E, which it updates as a result of the protocol.

A note on our model: In a traditional anonymous credential scheme (*e.g.,* [CL01]), the user may "show" its credential to many different organizations. We have simplified our protocol descriptions to reflect the assumption that a user need only show its credential to the original credential issuer. This model is sufficient for the applications we consider. We note that our credentials also function in the multi-organization model.

Security Definitions. Security definitions for anonymous credentials have traditionally been game-based. Unfortunately, the existing definitions may be insufficient for the applications considered in this work, as these definitions do not necessarily capture *correctness*. This can lead to problems when we integrate our credential system with oblivious transfer protocols (see *e.g.,* [NP99b, CNs07]). To capture the security requirements needed for our applications, we instead use a simulation-based definition, in which security of our protocols is analyzed with respect to an "ideal world" instantiation. We do not require security under concurrent executions, but rather restrict our analysis to atomic, sequential execution of each protocol. We do so because our constructions, which employ standard zero-knowledge techniques, require rewinding in their proof of security and thus are not concurrently secure. An advantage of the simulation paradigm is that our definitions will inherently capture correctness (*i.e.,* if parties honestly follow the protocols then they will each receive their expected outputs). Informally, the security of our system is encompassed by the following two definitions:

Provider Security: A malicious user (or set of colluding users) must not be able to falsely prove possession of a credential without first obtaining that credential, or arriving at it via an admissable sequence of credential updates. For our purposes, we require that the malicious user(s) cannot provide a proof of being in a state if that state is not present in her credential.

User Security: A malicious provider controlling some collection of corrupted users cannot learn any information about a user's identity or her state in the policy graph beyond what is available through auxiliary information from the environment.

Formalizing Definitions. Security for our protocols will be defined using the real-world/ideal-world paradigm, following the approach of [CNs07]. In the real world, a collection of (possibly cheating) users interact directly with a provider according to the protocol, while in the ideal world the parties interact via a trusted party. Informally, a protocol is secure if, for every real-world cheating combination of parties we can describe an ideal-world counterpart ("simulator") who gains as much information from the ideal-world interaction as from the real protocol. We note that our definitions will naturally enforce both *privacy* and *correctness*, but not necessarily *fairness*. It is possible that $\mathcal{P}$ will abort the protocol *before* the user has completed updating her credential or accessing a resource. This is unfortunately unavoidable in a two-party protocol.

Definition 6.1.1 (Security for a Stateful Anonymous Credential Scheme)
Full-simulation security for stateful anonymous credentials is defined according to the following experiments. Note that we do not explicitly specify auxiliary input to the parties, but this information can be provided in order to achieve sequential composition.

Real experiment. The real-world experiment $\mathbf{Real}_{\hat{\mathcal{P}},\hat{\mathcal{U}}_1,\ldots,\hat{\mathcal{U}}_\eta}(\eta, k, \Pi_1, \ldots, \Pi_\eta, \Sigma)$ is modeled as k rounds of communication between a possibly cheating provider $\hat{\mathcal{P}}$ and a collection of η possibly cheating users $\{\hat{\mathcal{U}}_1, \ldots, \hat{\mathcal{U}}_\eta\}$. In this experiment, $\hat{\mathcal{P}}$ is given the policy graph for each user $\Pi_1, \ldots, \Pi_\eta$, and the users are given an adaptive strategy Σ that, on input of the user's identity and current graph state, outputs the next action to be taken by the user.

At the beginning of the experiment, all users and the provider conduct the Setup procedure. At the end of this step, $\hat{\mathcal{P}}$ outputs an initial state P_1, and each user $\hat{\mathcal{U}}_i$ output state $U_{1,i}$. For each subsequent round $j \in [2, k]$, each user may interact with $\hat{\mathcal{P}}$ to update their credential as required by the strategy Σ. Following each round, $\hat{\mathcal{P}}$ outputs P_j, and the users output $(U_{1,j}, \ldots, U_{\eta,j})$. At the end of the k^{th} round the output of the experiment is $(P_k, U_{1,k}, \ldots, U_{\eta,k})$.

We will define the *honest* provider $\mathcal{P}$ as one that honestly runs its portion of Setup in the first round, honestly runs its side of the ObtainCred and ProveCred protocols when requested by a user at round $j > 1$, and outputs $P_k = params$. Similarly, an honest user $\mathcal{U}_i$ runs the Setup protocol honestly in the first round, and executes the user's side of the Setup, ObtainCred and ProveCred protocols, and eventually outputs the received value Cred along with all messages received.

Ideal experiment. In experiment $\mathbf{Ideal}_{\hat{\mathcal{P}}',\hat{\mathcal{U}}'_1,\ldots,\hat{\mathcal{U}}'_\eta}(\eta, k, \Pi_1, \ldots, \Pi\eta, \Sigma)$ the possibly cheating provider $\hat{\mathcal{P}}'$ sends the policy graphs to the trusted party $\mathcal{T}$. In each round $j \in [1, k]$, every user $\hat{\mathcal{U}}'_i$ (following strategy Σ) may send a message to $\mathcal{T}$ of the form $(\mathsf{update}, i, S_i, T_i)$ to update her credential using the UpdateCred protocol, or (prove, i, S_i) to prove her state using the ProveCred protocol.

- When $\mathcal{T}$ receives an update message, it checks $\hat{\mathcal{U}}'_i$'s current state and policy Π_i to determine whether the requested transition is allowed, setting a bit $b_{\mathcal{T}} = 1$ to so indicate. $\mathcal{T}$ sends $(\mathsf{update}, b_{\mathcal{T}})$ to $\hat{\mathcal{P}}'$, who responds with a bit $b_{\hat{\mathcal{P}}'} \in \{0, 1\}$ to $\mathcal{T}$ that indicates whether the update should succeed or fail. $\mathcal{T}$ returns $(b_{\hat{\mathcal{P}}'} \wedge b_{\mathcal{T}})$ to $\hat{\mathcal{U}}'_i$.
- For a prove message, $\mathcal{T}$ checks that $\hat{\mathcal{U}}'_i$ (setting $b_{\mathcal{T}}$ to so indicate), and relays $(\mathsf{prove}, S, b_{\mathcal{T}})$ to $\hat{\mathcal{P}}'$ who responds with a bit $b_{\hat{\mathcal{P}}'}$, and returns $(b_{\hat{\mathcal{P}}'} \wedge b_{\mathcal{T}})$ to $\hat{\mathcal{U}}'$.[1] Following each round, $\hat{\mathcal{P}}'$ outputs P_j, and the users output $(U_{1,j}, \ldots, U_{\eta,j})$. At the end of the k^{th} round the output of the experiment is $(P_k, V_j, U_{1,k}, \ldots, U_{\eta,k})$.

Let $\ell(\cdot), c(\cdot)$ be polynomially-bounded functions. We now define provider and user security in terms of the experiments above.

[1] Note that this reveals the current state S to $\hat{\mathcal{P}}'$. In section 6.2 we discuss techniques that also hide this information.

Provider Security. A stateful anonymous credential scheme is provider secure if for every collection of possibly cheating real-world p.p.t. receivers $\hat{\mathcal{U}}_1, \ldots, \hat{\mathcal{U}}_\eta$ there exists a collection of p.p.t. ideal-world receivers $\hat{\mathcal{U}}'_1, \ldots, \hat{\mathcal{U}}'_\eta$ such that $\forall \eta = \ell(\kappa)$, $k \in c(\kappa)$, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\mathcal{P},\hat{\mathcal{U}}_1,\ldots,\hat{\mathcal{U}}_\eta}(\eta, k, \Pi_1, \ldots, \Pi_\eta, \Sigma) \stackrel{c}{\approx} \mathbf{Ideal}_{\mathcal{P}l,\hat{\mathcal{U}}'_1,\ldots,\hat{\mathcal{U}}'_\eta}(\eta, k, \Pi_1, \ldots, \Pi_\eta, \Sigma)$$

User Security. A stateful anonymous credential scheme provides Receiver security if for every real-world p.p.t. provider $\hat{\mathcal{P}}$ who colludes with some collection of corrupted users, there exists a p.p.t. ideal-world provider $\hat{\mathcal{P}}'$ and users $\hat{\mathcal{U}}'$ such that $\forall \eta = \ell(\kappa)$, $k \in c(\kappa)$, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\hat{\mathcal{P}},\mathcal{U}_1,\ldots,\mathcal{U}_\eta}(\eta, k, \Pi_1, \ldots, \Pi_\eta, \Sigma) \stackrel{c}{\approx} \mathbf{Ideal}_{\hat{\mathcal{P}}',\mathcal{U}'_1,\ldots,\mathcal{U}'_\eta}(\eta, k, \Pi_1, \ldots, \Pi_\eta, \Sigma)$$

6.1.2 Hidden Range Proofs

Standard techniques [CFT98, CM99, CM99, Bou00] allow us to efficiently prove that a committed value lies in a *public* integer interval (*i.e.,*where the interval is known to both the prover and verifier). In our protocols, we sometimes need to *hide* this interval from the verifier, and instead have the prover show that a committed value lies between the openings of two other commitments.

Fortunately, this can be done efficiently as follows. Suppose we wish to show that $a \leq j \leq b$, for positive numbers a, j, b without revealing them. This is equivalent to showing that $0 \leq (j-a)$ and $0 \leq (b-j)$. We only need to get these two sums reliably into commitments, and can then employ the standard techniques since the range (≥ 0) is now public. Using a group $\mathbf{G} = \langle \mathbf{g} \rangle$, where n is a special RSA modulus, $\mathbf{g}$ is a quadratic residue modulo n and $\mathbf{h} \in \mathbf{G}$. The prover commits to these values as $A = \mathbf{g}^a \mathbf{h}^{r_a}$, $J = \mathbf{g}^j \mathbf{h}^{r_j}$, and $B = \mathbf{g}^b \mathbf{h}^{r_b}$, for random values $r_a, r_j, r_b \in \{0,1\}^\ell$ where ℓ is a security parameter. The verifier next computes a commitment to $(j-a)$ as J/A and to $(b-j)$ as B/J. The prover and verifier then proceed with the standard public interval proofs with respect to these commitments, which for technical reasons require groups where Strong RSA holds.

6.1.3 Preliminaries

We now describe how to realize *stateful* credentials. The state records information about the user's *attributes* as well as her prior *access history*. We will consider two separate modes for "showing" a credential. In the first mode, the user exposes her portions of her state during the ProveCred protocol. This is useful for, say, a DRM application where the user's goal is to prove that her software is in a "licensed" state without revealing her name.

In mode two, the user uses her credential to gain access to resources *without* revealing her state. Specifically, we show how to tie this credential system to a number of protocols, such as adaptive oblivious transfer and blind signatures, where the user wants to hide *both* her name and the item she is requesting, while simultaneously proving that she has the credentials to obtain the item.

Camenisch-Lysyanskaya Signatures. Our constructions may be implemented with the Strong RSA signature scheme of Camenisch and Lysyanskaya [CL02], or with the LRSW-based signatures of [CL04]. Both schemes consist of the algorithms $(\mathsf{CLKeyGen}, \mathsf{CLSign}, \mathsf{CLVerify})$ as well as two protocols, which we describe below. We first define the algorithms:

$\mathsf{CLKeyGen}(1^\kappa)$. On input a security parameter, outputs a keypair (pk, sk).
$\mathsf{CLSign}(sk, M_1, \ldots, M_n)$. On input one or more messages and a secret signing key, outputs the signature σ.
$\mathsf{CLVerify}(pk, \sigma, M_1, \ldots, M_n)$. On input a signature, message(s) and public verification key, outputs 1 if the signature verifies, 0 otherwise.

Additionally, the scheme consists of two protocols: (1) a protocol for a user to obtain a signature on the value(s) in a Pedersen (or Fujisaki-Okamoto) commitment [Ped92, FO97] without the signer learning anything about the message(s), and (2) a proof of knowledge of a signature.

In §6.2.2 we will use RSA-based CL signatures in conjunction with bilinear groups, *e.g.,*to prove knowledge of a CL signature on a commitment set in a bilinear group. These proofs can be conducted efficiently using techniques described in [CHL05].

6.1.4 Basic Construction

Our construction begins with the anonymous credentials of Camenisch and Lysyanskaya [Lys02, CL02, CL04], where the state is embedded as a field in the signature. The core innovation here is a protocol for performing state updates, and a technique for "translating" a history-dependent update policy into a cryptographic representation that can be used as an input to this protocol.

The setup, credential granting, and credential update protocols are presented in Figure 6.1. We will now briefly describe the intuition behind them.

Setup. First, the credential provider $\mathcal{P}$ generates its keypair and identifies one or more access policies it wishes to enforce. Each policy — encoded as a graph — may be applied to one or more users. The provider next "translates" the graph into a cryptographic representation which consists of the graph description, in addition to a separate CL signature corresponding to each tag in the graph, embedding the graph id, start, and end states. The files are distributed to users via an authenticated broadcast channel (*e.g.,* by signing and publishing them on a website).

A Note on Efficiency. It is important to emphasize that the "translation" of policy graphs may be conducted offline, and thus the cost of the online protocols (executed between user and provider) is *constant* and independent of the size of the policy. Furthermore, if many users share the same policy, this will further amortize the cost. Thus, our scheme is practical even for extremely complex policies containing thousands of distinct states and transition rules.

Obtaining a Credential. When a user $\mathcal{U}$ wishes to obtain a credential, he first generates a keypair that the CA certifies. He then negotiates with the provider to select an update policy to which the credential will be bound, as well the credential's initial state. The user next engages in a protocol to blindly extract a CL signature— under the provider's secret key— binding the user's public key, the initial state, policy id, and two random nonces chosen by the user: an *update nonce* N_u and a *usage nonce* N_s. The update nonce is revealed when the user updates the credential and the usage nonce is revealed when the user show's her credential. This signature, as well as the nonce and state information, form the credential. While the protocol for obtaining a credential, as currently described, reveals the user's identity through the use of her public key, we can apply the techniques found in [CL01, CL02] to provide a randomized pseudonym rather than the public key.

Updating the Credential's State. When the user wishes to update a credential, she first identifies a valid tag within the credential's access policy. She then generates a new pair of nonces and a commitment embedding these values, as well as the new state. Next, the user sends the update nonce along with the commitment. The provider records this nonce and the commitment into a database — however, if the nonce is already in the database but associated with a *different* commitment, the provider aborts the protocol, which prevents the user from re-using an old version of a credential. By recording the nonce and commitment together, we allow the user to restart the protocol if it has failed as long as she uses the same commitment. Otherwise, the user and provider then interact to conduct zero-knowledge proof that: (1) the remainder of this information is identical to the current credential, (2) the user has knowledge of the secret key corresponding to this credential, and (3) the policy graph contains a signature on a tag from the previous to the new state. If these conditions are met, the user obtains a new credential embedding the new state.

Showing (or Privately Proving Possession of) a Credential. The approach to using a single-show credential (Figure 6.2) follows [CL02, CL04]. When a user wishes to prove possession of a $\mathcal{P}$ credential to $\mathcal{P}$, he first reveals the credential usage nonce and the current state of the credential. $\mathcal{P}$ must check that this nonce has not been used before. The user then proves knowledge of: (1) a CL signature embedding this state value and nonce formed under $\mathcal{P}$'s public key, and (2) a secret key that is consistent with the CL signature.

Single-show vs. multi-show. This is an example of a "single-show" credential. It can be shown only once, or the verifier will recognize the repeated usage nonce. To restore its anonymity, the user may return to $\mathcal{P}$ and execute the update protocol to replace the usage nonce. This update policy gives users a way to use a single credential multiple

$\mathsf{Setup}(\mathcal{U}(1^k), \mathcal{P}(1^k, \Pi_1, \ldots, \Pi_n))$**:** The provider $\mathcal{P}$ generates parameters for the CL signature, as well as for the Pedersen commitment scheme.

Party $\mathcal{P}$ runs CLKeyGen twice, to create the CL signature keypairs $(spk_{\mathcal{P}}, ssk_{\mathcal{P}})$ and $(gpk_{\mathcal{P}}, gsk_{\mathcal{P}})$. It retains $(pk_{\mathcal{P}}, sk_{\mathcal{P}}) = ((spk_{\mathcal{P}}, gpk_{\mathcal{P}}), (ssk_{\mathcal{P}}, gsk_{\mathcal{P}}))$ as its keypair. The provider's public key $pk_{\mathcal{P}}$ must be certified by a trusted CA.

Each party $\mathcal{U}$ selects $u \xleftarrow{\$} \mathbb{Z}_q$ and computes the keypair $(pk_{\mathcal{U}}, sk_{\mathcal{U}}) = (g^u, u)$. The user's public key $pk_{\mathcal{U}}$ must be certified by a trusted CA.

Next, for each policy graph Π, $\mathcal{P}$ generates a cryptographic representation $\Pi_{\mathcal{C}}$.

1. $\mathcal{P}$ parses Π to obtain a unique policy identifier pid.
2. For each tag $t = (\mathsf{pid}, S, T)$ in Π, $\mathcal{P}$ computes a signature $\sigma_{S \to T} \leftarrow \mathsf{CLSign}(gsk_{\mathcal{P}}, (\mathsf{pid}, S, T))$.
3. $\mathcal{P}$ sets $\Pi_{\mathcal{C}} \leftarrow \langle \Pi, \forall t : \sigma_{S \to T} \rangle$ and publishes this value via an authenticated channel.

$\mathsf{ObtainCred}(\mathcal{U}(pk_{\mathcal{P}}, sk_{\mathcal{U}}, \Pi_{\mathcal{C}}), \mathcal{P}(pk_{\mathcal{U}}, sk_{\mathcal{P}}, \Pi_{\mathcal{C}}, S))$**:** On input a graph Π and initial state S, $\mathcal{U}$ first obtains $\Pi_{\mathcal{C}}$. $\mathcal{U}$ and $\mathcal{P}$ then conduct the following protocol:

1. $\mathcal{U}$ picks random show and update nonces $N_s, N_u \in \mathbb{Z}_q$ and computes $A \leftarrow \mathrm{PedCom}(sk_{\mathcal{U}}, N_s, N_u)$.
2. $\mathcal{U}$ conducts an interactive proof to convince $\mathcal{P}$ that A correlates to $pk_{\mathcal{U}}$.
3. $\mathcal{U}$ and $\mathcal{P}$ run the CL signing protocol on committed values so that $\mathcal{U}$ obtains the state signature $\sigma_{\mathsf{state}} \leftarrow \mathsf{CLSign}(ssk_{\mathcal{P}}, (sk_{\mathcal{U}}, N_s, N_u, \mathsf{pid}, S))$ with pid, S contributed by $\mathcal{P}$.
4. $\mathcal{U}$ stores the credential $\mathsf{Cred} = (\Pi_{\mathcal{C}}, S, \sigma_{\mathsf{state}}, N_s, N_u)$.

Figure 6.1: Protocols for obtaining a stateful anonymous credential.

times. One can adapt this scheme to support k-times anonymous use by using the Dodis-Yampolskiy [DY05] pseudorandom function to generate the nonces from a common seed, as shown in [CHK+06].

Theorem 6.1.2 *When instantiated with the RSA (resp., bilinear) variant of CL signatures, the anonymous credential scheme above achieves user, provider, and verifier security (definition 6.1.1) under the strong RSA (resp., LRSW) assumption.*

Due to space constraints, we omit the proof of Theorem 6.1.2. However, the proof of Theorem 6.2.2 naturally includes the security of our credential system.

$\mathsf{ProveCred}(\mathcal{U}(pk_{\mathcal{P}}, sk_{\mathcal{U}}, \mathsf{Cred}), \mathcal{P}(pk_{\mathcal{P}}, E))$**:** User $\mathcal{U}$ proves knowledge of the Cred as follows:

1. $\mathcal{U}$ parses Cred as $(\Pi_{\mathcal{C}}, S, \sigma_{\mathsf{state}}, N_s, N_u)$, and sends its usage nonce N_s to $\mathcal{P}$ (who aborts if $N_s \in E$).
2. Otherwise, $\mathcal{U}$ continues with either:
   - (mode one) Sending her current credential state S to $\mathcal{P}$ in the clear.
   - (mode two) Sending a commitment to S.
3. $\mathcal{U}$ then conducts an interactive proof to convince $\mathcal{P}$ that it possesses a CL signature σ_{state} embedding N_s, S, and that it has knowledge of the secret key $sk_{\mathcal{U}}$.
4. $\mathcal{P}$ adds N_s to E.

$\mathsf{UpdateCred}(\mathcal{U}(pk_{\mathcal{P}}, sk_{\mathcal{U}}, \mathsf{Cred}, \Pi_{\mathcal{C}}, T), \mathcal{P}(sk_{\mathcal{P}}, \Pi_{\mathcal{C}}, D))$**:** Given a credential Cred currently in state S, $\mathcal{U}$ and $\mathcal{P}$ interact to update the credential to state T:

1. $\mathcal{U}$ parses $\mathsf{Cred} = (\Pi_{\mathcal{C}}, S, \sigma_{\mathsf{state}}, N_s, N_u)$ and identifies a signature $\sigma_{S \to T}$ in $\Pi_{\mathcal{C}}$ that corresponds to a transition from state S to T (if none exists, $\mathcal{U}$ aborts).
2. $\mathcal{U}$ selects $N'_s, N'_u \xleftarrow{\$} \mathbb{Z}_q$ and computes $A \leftarrow \mathrm{PedCom}(sk_{\mathcal{U}}, N'_s, N'_u, \mathsf{pid}, T)$.
3. $\mathcal{U}$ sends (N_u, A) to $\mathcal{P}$. $\mathcal{P}$ looks in the database D for a pair $(N_u, A' \neq A)$. If no such pair is found, then $\mathcal{P}$ adds (N_u, A) to D. Otherwise $\mathcal{P}$ aborts.
4. $\mathcal{U}$ proves to $\mathcal{P}$ knowledge of values $(sk_{\mathcal{U}}, \mathsf{pid}, S, T, N'_s, N'_u, N_s, \sigma_{\mathsf{state}}, \sigma_{S \to T})$ such that:
   (a) $A = \mathrm{PedCom}(sk_{\mathcal{U}}, N'_s, N'_u, \mathsf{pid}, T)$.
   (b) $\mathsf{CLVerify}(spk_{\mathcal{P}}, \sigma_{\mathsf{state}}, (sk_{\mathcal{U}}, N_s, N_u, \mathsf{pid}, S)) = 1$.
   (c) $\mathsf{CLVerify}(gpk_{\mathcal{P}}, \sigma_{S \to T}, (\mathsf{pid}, S, T)) = 1$
5. If these proofs do not verify, $\mathcal{P}$ aborts. Otherwise $\mathcal{U}$ and $\mathcal{P}$ run the CL signing protocol on committed values to provide $\mathcal{U}$ with $\sigma'_{\mathsf{state}} \leftarrow \mathsf{CLSign}(ssk_{\mathcal{P}}, A)$.
6. $\mathcal{U}$ stores the updated credential $\mathsf{Cred}' = (\Pi_{\mathcal{C}}, T, \sigma'_{\mathsf{state}}, N'_s, N'_u)$.

Figure 6.2: Protocol for proving knowledge of and updating a single-show anonymous credential.

6.2 Oblivious Database Access Control

In this section we show how stateful anonymous credentials can be used to control access to *oblivious databases*. Recall that an oblivious database permits users to request data items without revealing their item choices to the database operator (*e.g.,* where the

item choices are sensitive as in a medical databases).

Although we possess efficient building blocks such as k-out-of-N Oblivious Transfer (OT), little progress has been made towards the deployment of practical oblivious databases. In part, this is due to a fundamental tension with the requirements of a database operator to provide some form of access control. In this section, we show that it is possible to embed flexible, history dependent access controls into an oblivious database, without compromising the user's privacy. Specifically, we show how to combine our stateful anonymous credential system with an adaptive Oblivious Transfer protocol to construct a multi-user oblivious database that supports complex access control policies. We show how to *efficiently* couple stateful credentials with the recent standard-model adaptive OT construction due to Camenisch, Neven and shelat [CNs07]. Our stateful credentials can also be efficiently coupled with the adaptive OT of Green and Hohenberger [GH08b].

Linking Policies to Database Items. To support oblivious database access, we extend our policy graphs to incorporate *tags* of the form $(id, S \rightarrow T, i)$, where id is the policy, $S \rightarrow T$ is the edge, and i is the message index that is allowed by that tag. Each edge in the graph may be associated with one or more tags, which correspond to the items that can be obtained from the database when traversing that edge. As described in Section 6.1, we place null transitions on each terminal state that allow the user to update her credential and access a predefined null message. The set of all tags, both legitimate and null, are signed by the database and published. Figure 6.3 shows an example policy for a small database. The interested reader can view a complete discussion of some of the non-trivial access control policies allowed by our credential system in Appendix B.

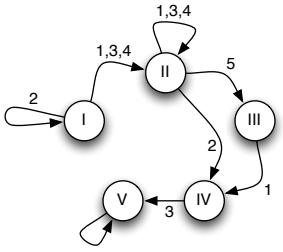

Figure 6.3: Sample access policy for a small oblivious database. The labels on each transition correspond to the database item indices that can be requested when a user traverses the edge, with null transitions represented by unlabeled edges.

6.2.1 Protocol Descriptions and Security Definitions for Oblivious Databases

Our oblivious database protocols combine the scheme of Section 6.1.4 with a multi-receiver oblivious transfer OT protocol. Each transaction is conducted between one of a

collection of users and a single database server $\mathcal{D}$. We now describe the protocol specifications.

$\mathsf{Setup}(\mathcal{U}(1^k), \mathcal{D}(1^k, \Pi_1, \ldots, \Pi_n)$**:** The database $\mathcal{D}$ generates parameters $params$ for the scheme. As in the basic credential scheme, it generates a cryptographic representation $\Pi_{\mathcal{C}}$ for each policy graph, and publishes those values via an authenticated channel. Each user $\mathcal{U}$ generates a keypair and requests that it be certified by a trusted CA.

$\mathsf{OTObtainCred}(\mathcal{U}(pk_{\mathcal{D}}, sk_{\mathcal{U}}, \Pi_{\mathcal{C}}), \mathcal{D}(pk_{\mathcal{U}}, sk_{\mathcal{D}}, \Pi_{\mathcal{C}}, S))$**:** $\mathcal{U}$ registers with the system and receives a credential Cred which binds her to a policy graph Π_{id} and starting state S.

$\mathsf{OTAccessAndUpdateCred}(\mathcal{U}(pk_{\mathcal{D}}, sk_{\mathcal{U}}, \mathsf{Cred}, t), \mathcal{D}(sk_{\mathcal{D}}, E))$**:** $\mathcal{U}$ requests an item at index i in the database from state S by selecting a tag $t = (id, S \to T, i)$ from the policy graph. The user then updates her credential Cred, in such a way that $\mathcal{D}$ does not learn her identity, her attributes, or her current state. Simultaneously, $\mathcal{U}$ obtains a message from the database at index i. At the end of a successful protocol, $\mathcal{U}$ updates the state information in Cred, and $\mathcal{D}$ updates a local datastore E.

Security. We informally describe the security properties of an oblivious database system. We then present the formal definition, which extends definition 6.1.1 by incorporating the concept of a message database $M_1, \ldots, M_N$ held by the database $\mathcal{D}$.

Database Security: No (possibly colluding) subset of corrupted users can obtain any collection of items that is not specifically permitted by the users' policies.

User Security: A malicious database controlling some collection of corrupted users cannot learn any information about a user's identity or her state in the policy graph, beyond what is available through auxiliary information from the environment.

Definition 6.2.1 (Security for Oblivious Databases with Access Controls) Security is defined according to the following experiments. As before, we do not explicitly specify auxiliary input to the parties, but this information can be provided in order to achieve sequential composition.

Real experiment. The real-world experiment $\mathbf{Real}_{\hat{\mathcal{D}},\hat{\mathcal{U}}_1,\ldots,\hat{\mathcal{U}}_\eta}(\eta, N, k, \Pi_1, \ldots, \Pi_\eta, M_1, \ldots, M_N, \Sigma)$ is modeled as k rounds of communication between a possibly cheating database $\hat{\mathcal{D}}$ and a collection of η possibly cheating users $\{\hat{\mathcal{U}}_1, \ldots, \hat{\mathcal{U}}_\eta\}$. In this experiment, $\hat{\mathcal{D}}$ is given the policy graph for each user $\Pi_1, \ldots, \Pi_\eta$, a message database $M_1, \ldots, M_N$ and the users are given an adaptive strategy Σ that, on input of the user's identity and current graph state, outputs the next action to be taken by the user.

At the beginning of the experiment, the database and users conduct the Setup and $\mathsf{OTObtainCred}$ protocols. At the end of this step, $\hat{\mathcal{D}}$ outputs an initial state S_1, and each user $\hat{\mathcal{U}}_i$ output state $U_{1,i}$. For each subsequent round $j \in [2, k]$, each user may interact with $\hat{\mathcal{D}}$ to request an item i as required by the strategy Σ. Following each round, $\hat{\mathcal{D}}$ outputs

S_j, and the users output $(U_{1,j}, \ldots, U_{\eta,j})$. At the end of the k^{th} round the output of the experiment is $(S_k, U_{1,k}, \ldots, U_{j,k})$.

We will define the *honest* database $\mathcal{D}$ as one that honestly runs its portion of Setup in the first round, honestly runs its side of the OTObtainCred and OTAccessAndUpdateCred protocols when requested by a user at round $j > 1$, and outputs $S_k = params$. Similarly, an honest user $\mathcal{U}_i$ runs the Setup protocol honestly in the first round, and executes the user's side of the OTObtainCred, OTAccessAndUpdateCred protocols, and eventually outputs the received value Cred along with all messages received.

Ideal experiment. In experiment $\mathbf{Ideal}_{\hat{\mathcal{D}}',\hat{\mathcal{U}}'_1,\ldots,\hat{\mathcal{U}}'_\eta}(\eta, N, k, \Pi_1, \ldots, \Pi\eta, M_1, \ldots, M_N, \Sigma)$ the possibly cheating database $\hat{\mathcal{D}}'$ sends the policy graphs to the trusted party $\mathcal{T}$. In each round $j \in [1, k]$, every user $\hat{\mathcal{U}}'$ (following strategy Σ) selects a message index $i \in [1, N+1]$ and sends a message containing the user's identity and (i, S, T) to $\mathcal{T}$. $\mathcal{T}$ then checks the policy graph corresponding to that user to determine if the action is permitted, and sends $\hat{\mathcal{D}}'$ a bit b_1 indicating the outcome of this test. $\hat{\mathcal{D}}'$ then returns a bit b_2 determining whether the transaction should succeed. If $b_1 \wedge b_2$, then $\mathcal{T}$ returns M_i to $\hat{\mathcal{U}}'_i$, otherwise it returns $\perp$. Following each round, $\hat{\mathcal{D}}'$ outputs P_j, and the users output $(U_{1,j}, \ldots, U_{\eta,j})$. At the end of the k^{th} round the output of the experiment is $(P_k, U_{1,k}, \ldots, U_{\eta,k})$.

Let $\ell(\cdot), c(\cdot)$ be polynomially-bounded functions. We now define database and user security in terms of the experiments above.

Database Security. A stateful anonymous credential scheme is database-secure if for every collection of real-world p.p.t. receivers $\hat{\mathcal{U}}_1, \ldots, \hat{\mathcal{U}}_\eta$ there exists a collection of p.p.t. ideal-world receivers $\hat{\mathcal{U}}'_1, \ldots, \hat{\mathcal{U}}'_\eta$ such that $\forall N = \ell(\kappa)$, $N = d(\kappa)$, $k \in c(\kappa)$, PF, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\mathcal{D},\hat{\mathcal{U}}_1,\ldots,\hat{\mathcal{U}}_\eta}(\eta, N, k, \Pi_1, \ldots, \Pi_\eta, M_1, \ldots, M_N, \Sigma) \stackrel{c}{\approx} \mathbf{Ideal}_{\mathcal{D},\hat{\mathcal{U}}'_1,\ldots,\hat{\mathcal{U}}'_\eta}(\eta, N, k, \Pi_1, \ldots, \Pi_\eta, M_1, \ldots, M_N, \Sigma)$$

User Security. A stateful anonymous credential scheme provides Receiver security if for every real-world p.p.t. database $\hat{\mathcal{D}}$ and collection of dishonest users, there exists a p.p.t. ideal-world sender $\hat{\mathcal{D}}'$ such that $\forall N = \ell(\kappa)$, $\eta = d(\kappa)$, $k \in c(\kappa)$, PF, Σ, and every p.p.t. distinguisher:

$$\mathbf{Real}_{\hat{\mathcal{D}},\mathcal{U}_1,\ldots,\mathcal{U}_\eta}(\eta, N, k, \Pi_1, \ldots, \Pi_\eta, M_1, \ldots, M_N, \Sigma) \stackrel{c}{\approx} \mathbf{Ideal}_{\hat{\mathcal{D}}',\mathcal{U}'_1,\ldots,\mathcal{U}'_\eta}(\eta, N, k, \Pi_1, \ldots, \Pi_\eta, M_1, \ldots, M_N, \Sigma)$$

6.2.2 Constructions

In our model, many users share access to a single database. To construct our protocols, we extend the basic credential scheme of Section 6.1.4 by linking it to an adaptive OT

protocol. The two protocols that we select for our constructions are (1) the random-oracle $\mathsf{OT}^N_{k\times 1}$ of §4.2.2, and (2) the standard-model $\mathsf{OT}^N_{k\times 1}$ protocol of Camenisch *et al.* [CNs07]. In both cases, the database operator commits to a collection of N messages, along with a special *null* message at index $N+1$. It them distributes these commitments (*e.g.,* via a website). Each user then registers with the database using the OTObtainCred protocol, and agrees to be bound by a policy that will control her ability to access the database.

To obtain items from the database, the user runs the OTAccessAndUpdateCred protocol, which proves (in zero knowledge) that its request is consistent with its policy. Provided the user does not violate policy, the user is assured that the database operator learns nothing about its identity, or the nature of its request. Figures 6.4 and 6.5 describes the protocol based on the Oblivious Transfer scheme of §4.2.2 (we implicitly specify a blind IBE scheme defined by SetupIBE, BlindExtract, Encrypt, Decrypt).

Figures 6.6 and 6.7 describe the protocol based on the Oblivious Transfer scheme of Camenisch *et al.* [CNs07]. We will now provide a security argument for this protocol, noting that the other protocol can be proven secure using the same arguments.

Theorem 6.2.2 *The scheme described in Figures 6.6 and 6.7 satisfies definition 6.2.1 under the q-PDDH, q-SDH, and Strong RSA assumptions.*

We now sketch a proof of Theorem 6.2.2. Our sketch will refer substantially to the original proof of Camenisch *et al.* [CNs07]. We note that our proof will consider two components: (1) the security of the underlying OT scheme (which is based on the proof of [CNs07]), and a separate proof of the anonymous credential scheme.

Proof sketch. Our sketch separately considers User and Database security.

User Security. Let us assume that an adversary has corrupted a database $\mathcal{D}$ and some subset of the users $\hat{\mathcal{U}}_1, \ldots, \hat{\mathcal{U}}_N$. In this model, corruptions will be static. We show that for every such adversary, we can construct a simulator such that the output of the ideal experiment conducted with the simulator will be indistinguishable from the output of the real experiment.

Our simulator operates as follows. First, $\mathcal{D}$ outputs the parameters for the credential system, the cryptographic representation of each graph, and $pk, C_1, \ldots, C_N$. If these parameters are incorrectly formed, the simulator aborts. The simulator next generates a credential key for each uncorrupted user and negotiates with $\mathcal{D}$ to join the system under an appropriate policy. When $\mathcal{D}$ executes the proof of knowledge that $H = e(g,h)$ with some uncorrupted user, our simulator rewinds to extract the value h (this extraction succeeds with all but negligible probability). For $i = 1$ to N, the simulator decrypts C_i using h to obtain M_i. This collection of plaintexts is sent to the trusted party $\mathcal{T}$.

Whenever an uncorrupted user queries $\mathcal{T}$ to obtain message i (according to a state transition defined in their policy), $\mathcal{T}$ verifies that this request is permitted by policy and updates its view of the user's state. Next, it notifies our simulator which runs the

$\mathsf{Setup}(\mathcal{U}(1^k), \mathcal{D}(1^k))$**:** When the database operator $\mathcal{D}$ is initialized with a database of messages $M_1, \ldots, M_N$, it conducts the following steps:

1. $\mathcal{D}$ selects parameters for the OT scheme as $(params, msk) \leftarrow \mathsf{SetupIBE}(1^\kappa, c(\kappa))$. $\mathcal{D}$ generates two CL signing keypairs $(spk_\mathcal{D}, ssk_\mathcal{D})$ and $(gpk_\mathcal{D}, gsk_\mathcal{D})$, and $\mathcal{U}$ generates her keypair $(pk_\mathcal{U}, sk_\mathcal{U})$ as in the Setup protocol of Figure 6.1.
2. For $i = 1$ to $(N + 1)$, $\mathcal{D}$ computes a ciphertext $C_i = (A_i, B_i)$ as:
   (a) $W_i \xleftarrow{\$} \mathcal{M}$.
   (b) If $i \leq N$, then $A_i \leftarrow \mathsf{Encrypt}(params, j, W_i)$ and $B_i \leftarrow H(i||W_i) \oplus M_i$.
   (c) If $i = (N + 1)$, compute A_i as above and set $B_i = H(i||W_i)$.
3. For every graph Π to be enforced, $\mathcal{D}$ generates a cryptographic representation $\Pi_\mathcal{C}$ as follows:
   (a) $\mathcal{D}$ parses Π to obtain a unique policy identifier pid.
   (b) For each tag $t = (\mathsf{pid}, S, T, i)$ with $i \in [1, N + 1]$, $\mathcal{D}$ computes the signature $\sigma_{S \to T,i} \leftarrow \mathsf{CLSign}(gsk_\mathcal{P}, (\mathsf{pid}, S, T, i))$. Finally, $\mathcal{D}$ sets $\Pi_\mathcal{C} \leftarrow \langle \Pi, \forall t : \sigma_{S \to T,i} \rangle$.

$\mathcal{D}$ and each $\mathcal{U}$ in the system generate and certify keys. For each graph Π that $\mathcal{D}$ wishes to enforce, $\mathcal{D}$ constructs and publishes a cryptographic representation $\Pi_\mathcal{C}$.

$\mathsf{OTObtainCred}(\mathcal{U}(pk_\mathcal{D}, sk_\mathcal{U}, \Pi_\mathcal{C}), \mathcal{D}(pk_\mathcal{U}, sk_\mathcal{D}, \Pi_\mathcal{C}, S))$**:** When user $\mathcal{U}$ wishes to join the system, it negotiates with $\mathcal{D}$ to agree on a policy Π and initial state S, then:

1. $\mathcal{U}$ picks a random show nonce $N_s \in \mathbb{Z}_q$ and computes $A \leftarrow \mathrm{PedCom}(sk_\mathcal{U}, N_s)$.
2. $\mathcal{U}$ conducts an interactive proof to convince $\mathcal{D}$ that A correlates to $pk_\mathcal{U}$, and $\mathcal{D}$ conducts an interactive proof of knowledge to convince $\mathcal{U}$ that it knows msk.
3. $\mathcal{U}$ and $\mathcal{P}$ run the CL signing protocol on committed values so that $\mathcal{U}$ obtains the state signature $\sigma_\mathsf{state} \leftarrow \mathsf{CLSign}(ssk_\mathcal{P}, (sk_\mathcal{U}, N_s, \mathsf{pid}, S))$ with pid, S contributed by $\mathcal{P}$.
4. $\mathcal{U}$ stores the credential $\mathsf{Cred} = (\Pi_\mathcal{C}, S, \sigma_\mathsf{state}, N_s)$.

Figure 6.4: The global setup and user-initialization protocols for an access-controlled oblivious database based on the $\mathsf{OT}^N_{k\times 1}$ of §4.2.2.

OTAccessAndUpdateCred protocol on an arbitrary (uncorrupted) user's policy under index $N + 1$ (this is the "dummy" transition and is always permitted by the credential system). If

Once the Setup and OTObtainCred algorithms have been run, $\mathcal{U}$ can adaptively retrieve items from the database using the following protocol.

$\mathsf{OTAccessAndUpdateCred}(\mathcal{U}(pk_{\mathcal{D}}, sk_{\mathcal{U}}, \mathsf{Cred}, t), \mathcal{D}(pk_{\mathcal{D}}, E))$**:** When $\mathcal{U}$ wishes to obtain the message indexed by $i \in [1, N]$ (or conduct a dummy transaction, for which it sets $i = (N + 1)$), it first identifies a tag t in Π such that $t = (id, S \rightarrow T, i)$.

1. $\mathcal{U}$ parses $\mathsf{Cred} = (\Pi_{\mathcal{C}}, S, \sigma_{\mathsf{state}}, N_s)$, and further parses $\Pi_{\mathcal{C}}$ to find $\sigma_{S \rightarrow T,i}$.
2. $\mathcal{U}$ selects $N'_s \xleftarrow{\$} \mathbb{Z}_q$ and computes $A \leftarrow \mathrm{PedCom}(sk_{\mathcal{U}}, N'_s, \mathsf{pid}, T)$.
3. $\mathcal{U}$ then sends N_s to $\mathcal{D}$. $\mathcal{D}$ checks the database E for $(N_s, A' \neq A)$, and if it finds such an entry it aborts. Otherwise it adds (N_s, A) to E.
4. $\mathcal{U}$ runs the BlindExtract protocol with $\mathcal{D}$ on input i, and proves knowledge of $(i, sk_{\mathcal{U}}, \sigma_{S \rightarrow T,i}, \sigma_{\mathsf{state}}, id, S, T, N'_s)$ such that the following conditions hold:
   (a) $\mathcal{U}$'s input to BlindExtract is i.
   (b) $A = \mathrm{PedCom}(sk_{\mathcal{U}}, N'_s, \mathsf{pid}, T)$.
   (c) $\mathsf{CLVerify}(spk_{\mathcal{P}}, \sigma_{\mathsf{state}}, (sk_{\mathcal{U}}, N_s, \mathsf{pid}, S)) = 1$.
   (d) $\mathsf{CLVerify}(\mathcal{P}, \sigma_{S \rightarrow T,i}, (\mathsf{pid}, S, T, i)) = 1$.
5. If these proofs verify, $\mathcal{U}$ and $\mathcal{D}$ run the CL signing protocol on committed values such that $\mathcal{U}$ obtains $\sigma'_{\mathsf{state}} \leftarrow \mathsf{CLSign}(ssk_{\mathcal{D}}, A)$. $\mathcal{U}$ stores the updated credential $\mathsf{Cred}' = (\Pi_{\mathcal{C}}, T, \sigma'_{\mathsf{state}}, N'_s)$.
6. If the BlindExtract protocol succeeded, $\mathcal{U}$ obtains sk_i.

At the conclusion of this protocol, $\mathcal{U}$ parses C_i into (A_i, B_i) and outputs the message $M_i = H(i||\mathsf{Decrypt}(params, i, sk_i, A_i)) \oplus B_i$.

Figure 6.5: A protocol for an accessing data items based on the $\mathsf{OT}^N_{k \times 1}$ of §4.2.2.

this protocol succeeds, the simulator sends a bit 1 to $\mathcal{T}$ which returns M_i to the user.

Claim. The transcript produced by this simulator is indistinguishable from the transcript produced by the real experiment. This is true for following reasons:

1. The probability that the simulator *incorrectly* extracts h (or fails to extract it) is negligible.
2. The probability that the adversary distinguishes a protocol executed on an arbitrary user/dummy index is negligible: this is due to (a) the witness-indistinguishability property of the credential proofs of knowledge, and (b) the element V transmitted to $\mathcal{D}$ during OTAccessAndUpdateCred is indistinguishable from a random element.

Note that the we need not argue the unforgeability of the anonymous credential scheme here, since we consider only actions taken by the uncorrupted user.

Database Security. Let us assume that an adversary has corrupted some subset of the users $\hat{\mathcal{U}}_1, \ldots, \hat{\mathcal{U}}_N$ (corruptions are static). We show that for every such adversary, we can construct a simulator such that the output of the ideal experiment conducted with the simulator will be indistinguishable from the output of the real experiment.

Our simulator operates as follows. First, it generates the public and privacy parameters for the credential scheme along with the cryptographic representation of the policies provided by $\mathcal{T}$. It generates the parameters for the OT scheme pk, sk as normal, but sets the plaintext for each database element to a dummy value (the identity element) and produces ciphertexts $C_1, \ldots, C_N$ (and generates the dummy message $C_{(N+1)}$ as normal). It sends these parameters to each corrupted user, and to each user proves that $H = e(g, h)$.

Whenever a corrupted user initiates the $\mathsf{OTAccessAndUpdateCred}$ protocol with $\mathcal{D}$, the simulator verifies that the user's request (including ZK proofs) verifies, and that neither N_u or N_s has been seen before. If so, it rewinds and uses the extractors for the ZK proofs to learn the user's identity, the index of the message i being requested, the blinding factor v, and the user's current and previous credential state S, T. The server transmits the user's identity values (i, S, T) to $\mathcal{T}$ which verifies that they satisfy the policy (updating the policy state in the process). If $\mathcal{T}$ returns $\perp$, then $\mathcal{D}$ aborts the protocol with the user. Otherwise if $\mathcal{T}$ returns M_i, then the simulator parses $C_i = (A_i, B_i)$ and returns $U = (B_i^v)/M_i$. The simulator uses rewinding to simulate the proof and convince the user that U has been correctly formed.

Claim. The transcript produced by this simulator is indistinguishable from the transcript produced by the real experiment. This claim rests on the following points:

1. The false message collection $C_1, \ldots, C_{(N+1)}$ is indistinguishable from the real message by the semantic security of the encryption scheme, which holds under the q-PDDH assumption (see [CNs07] for the full argument).
2. The simulated proof of U's structure is indistinguishable from a correct real proof.
3. The simulator never queries $\mathcal{T}$ on a tuple (i, S, T) that violates the user's policy. This reduces to the unforgeability of the CL signature (which is in turn based on Strong RSA or LRSW). Specifically, to violate policy, a user must satisfy one of the following conditions:
   (a) Prove knowledge of a signature σ_δ that it was not given, or
   (b) Prove knowledge of a signature $\sigma_{S \to T}$ that it was not given. In either case, the simulator can use the extractor for the proof system to obtain the forged signature and win the CL signature forgery game.
   (c) Misuse the CL signing protocol such that it receives a signature that is not equivalent to a signature on the commitment A (or mispresent the structure of A).

□

6.2.3 On Universal Composability

In this chapter, we have focused our applications on fully-simulatable OT schemes. These are somewhat weaker than the UC-secure protocols of Chapter 5, since they do not allow for generic composition. This is primarily due to the mechanics of the underlying anonymous credential schemes, which depend on rewinding for their security proofs. However, it might be possible to adapt the protocols of Chapter 5 given some UC-secure credential scheme. We leave this as an open problem, noting that the recent non-interactive credentials of Belenkiy, Chase, Kolweiss and Lysyanskaya [BCKL08] might serve as a starting point for a fully UC-secure solution.

6.2.4 Extensions to Compact Access Policies in Practice

Extension #1: Equivalence Classes. In the scheme presented thus far, a tag in the policy graph must be defined on every item index in the database. However, there are cases where many items may have the same access rules applied, and therefore we can reduce the number tags used by referring to the entire group with a single tag. A simple solution is to replace specific item indices with general equivalence classes in the graph tags. The OT database can be easily re-organized to support this concept by renumbering the item indices (previously $[1, N]$) using values of the form $(c||i) \in \mathbb{Z}_q$ where c is the identity of the item class, and $||$ represents concatenation. During the OTAccessAndUpdateCred protocol, $\mathcal{U}$ can obtain any item $(c||i)$ by performing a zero-knowledge proof on the first half of the selection index, which shows that the user's selected tag contains the class c.

Extension #2: Encoding Contiguous Ranges. An alternative approach requires the database operator to arrange the identities of objects in the same class so that they fall in contiguous ranges. In this case, we will label the graph edges with *ranges* of items rather than single values. The credentials will also replace the value i with an upper and lower bound for the range that the holder of the credential is permitted to access. We make a slight change to the OTAccessAndUpdateCred protocol so that rather than proving equality between the requested object and the object present in the tag, the user now proves that the requested object lies in the range described in the user selected tag, as described by the hidden range proof technique in Section 6.1.2. Notice that while this approach requires that the database be reorganized such that classes of items remain in contiguous index ranges, it can be used to represent more advanced data structures, such as hierarchical classes.

6.3 Other Applications of Stateful Anonymous Credentials

Oblivious IBE Key Extraction. Identity-Based Encryption (*e.g.,* [BF01, Coc01]) is a form of public-key encryption where users can substitute an arbitrary string— for example,

a name or email address— in place of a traditional public key. In an IBE deployment, the corresponding decryption keys are generated by a trusted party known as the Private Key Generator (PKG).

Under normal circumstances, the user cannot hide its identity from the PKG. Indeed, this can be problematic, since the PKG must verify that a user is authorized to obtain a key for a given identity. In some anonymous communication scenarios, however, it can be desirable to anonymously grant temporary decryption keys to users without learning the user's identity.

Green and Hohenberger [GH08b] propose a means by which a user can *blindly* extract a decryption key from a PKG, such that the PKG does not learn the identity extracted. These techniques can also be extended to allow for partially-blind extraction, where a portion of the identity is known to the PKG, which is useful when keys also embed some known, restricted information, such as the time period during which they will be valid. Unfortunately, these techniques deprive the PKG of the ability to control which keys are given out. Using our stateful anonymous credential system, we can realize efficient solutions for blind, yet controlled, access to the IBE keys for the Boneh-Boyen IBE [BB04a] and the Waters IBE [Wat05].

Oblivious (Blind) Signatures. As observed by Moni Naor, there is a connection between decryption keys in IBE schemes and digital signatures.[2] Specifically, the decryption key corresponding to an identity id in any full-secure IBE scheme is a signature on the message id where the signature verification key is the master public key of the IBE scheme. Thus, the blind key extraction protocol for the Waters IBE [Wat05] is also a blind signature scheme for the Waters signature. Fortunately, we can put efficient access controls on top of this, as well.

Imagine several scenarios in which this is truly exciting: a signer can now specify a policy under which he is willing to blindly sign messages, and then can enforce this policy without violating any of the user's privacy or even learning her identity. This leads to practical data timestamping services (*e.g.,* [Ver05, Sur]) that do not learn anything about what a user is signing, or even who originated a specific request. Alternatively, blind signatures can be useful for forensic purposes: a device can be required to obtain a signature each time it undertakes a controversial action, and use these signatures to convince a later investigator that each action was in fact allowed by policy. Additionally, our access controls can also be placed onto the blind signing protocols of the Strong RSA [CL02] and bilinear [CL04] signatures of Camenisch and Lysyanskaya, as well as the short bilinear signatures of Boneh and Boyen [BB04b]. These are all schemes secure in the standard model.

Oblivious Keyword Search. IBE key extraction can also be used to implement public-key searchable encryption [OK04, BCOP04, WBDS04], which permits users to search a collection of encrypted files for those matching a particular keyword. For example, Waters *et al.* [WBDS04] describe a searchable encrypted audit log in which a third party auditor

[2]This observation was credited to Naor by Boneh and Franklin [BF01].

is granted the ability to independently search the encrypted log for specific keywords. In these schemes, the scope of the user's searches is generally limited by a trusted authority, which generates "search trapdoors" for particular words at the searcher's request. Unfortunately, this trusted party necessarily learns the details of each search term, which may be problematic in circumstances where the pattern of trapdoor requests reveals sensitive information. Using the blind key extraction techniques described above, Green and Hohenberger [GH08b] discuss how an authority can blindly deliver search trapdoors without learning which terms are being monitored. Again, our techniques can help regulate which key word searches are allowed.

$\mathsf{Setup}(\mathcal{U}(1^k), \mathcal{D}(1^k))$**:** When the database operator $\mathcal{D}$ is initialized with a database of messages $M_1, \ldots, M_N$, it conducts the following steps:

1. $\mathcal{D}$ selects parameters for the OT scheme as $\gamma = (q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathsf{BMsetup}(1^\kappa)$, $h \xleftarrow{\$} \mathbb{G}$, $x \xleftarrow{\$} \mathbb{Z}_q$, and $H \leftarrow e(g, h)$. $\mathcal{D}$ generates two CL signing keypairs $(spk_{\mathcal{D}}, ssk_{\mathcal{D}})$ and $(gpk_{\mathcal{D}}, gsk_{\mathcal{D}})$, and $\mathcal{U}$ generates her keypair $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ as in the Setup protocol of Figure 6.1.
2. For $i = 1$ to $(N + 1)$, $\mathcal{D}$ computes a ciphertext $C_i = (A_i, B_i)$ as:
   (a) If $i \leq N$, then $A_i = g^{\frac{1}{x+i}}$ and $B_i = e(h, A_i) \cdot M_i$.
   (b) If $i = (N + 1)$, compute A_i as above and set $B_i = e(h, A_i)$.
3. For every graph Π to be enforced, $\mathcal{D}$ generates a cryptographic representation $\Pi_{\mathcal{C}}$ as follows:
   (a) $\mathcal{D}$ parses Π to obtain a unique policy identifier pid.
   (b) For each tag $t = (\mathsf{pid}, S, T, i)$ with $i \in [1, N + 1]$, $\mathcal{D}$ computes the signature $\sigma_{S \to T,i} \leftarrow \mathsf{CLSign}(gsk_{\mathcal{P}}, (\mathsf{pid}, S, T, i))$. Finally, $\mathcal{D}$ sets $\Pi_{\mathcal{C}} \leftarrow \langle \Pi, \forall t : \sigma_{S \to T,i} \rangle$.
4. $\mathcal{D}$ sets $pk_{\mathcal{D}} = (spk_{\mathcal{D}}, gpk_{\mathcal{D}}, \gamma, H, g^x, C_1, \ldots, C_n)$ and $sk_{\mathcal{D}} = (ssk_{\mathcal{D}}, gsk_{\mathcal{D}}, h)$. $\mathcal{D}$ then publishes each $\Pi_{\mathcal{C}}$ and the OT parameters $pk_{\mathcal{D}}$ via an authenticated channel.

$\mathcal{D}$ and each $\mathcal{U}$ in the system generate and certify keys. For each graph Π that $\mathcal{D}$ wishes to enforce, $\mathcal{D}$ constructs and publishes a cryptographic representation $\Pi_{\mathcal{C}}$.

$\mathsf{OTObtainCred}(\mathcal{U}(pk_{\mathcal{D}}, sk_{\mathcal{U}}, \Pi_{\mathcal{C}}), \mathcal{D}(pk_{\mathcal{U}}, sk_{\mathcal{D}}, \Pi_{\mathcal{C}}, S))$**:** When user $\mathcal{U}$ wishes to join the system, it negotiates with $\mathcal{D}$ to agree on a policy Π and initial state S, then:

1. $\mathcal{U}$ picks a random show nonce $N_s \in \mathbb{Z}_q$ and computes $A \leftarrow \mathrm{PedCom}(sk_{\mathcal{U}}, N_s)$.
2. $\mathcal{U}$ conducts an interactive proof to convince $\mathcal{D}$ that A correlates to $pk_{\mathcal{U}}$, and $\mathcal{D}$ conducts an interactive proof of knowledge to convince $\mathcal{U}$ that $e(g, h) = H$. (This proof can be conducted efficiently in four rounds as in [CNs07].).
3. $\mathcal{U}$ and $\mathcal{P}$ run the CL signing protocol on committed values so that $\mathcal{U}$ obtains the state signature $\sigma_{\mathsf{state}} \leftarrow \mathsf{CLSign}(ssk_{\mathcal{P}}, (sk_{\mathcal{U}}, N_s, \mathsf{pid}, S))$ with pid, S contributed by $\mathcal{P}$.
4. $\mathcal{U}$ stores the credential $\mathsf{Cred} = (\Pi_{\mathcal{C}}, S, \sigma_{\mathsf{state}}, N_s)$.

Figure 6.6: The global setup and user-initialization protocols for an access-controlled oblivious database based on the $\mathsf{OT}^N_{k \times 1}$ of Camenisch, Neven and shelat [CNs07].

Once the Setup and OTObtainCred algorithms have been run, $\mathcal{U}$ can adaptively retrieve items from the database using the following protocol.

$\mathsf{OTAccessAndUpdateCred}(\mathcal{U}(pk_{\mathcal{D}}, sk_{\mathcal{U}}, \mathsf{Cred}, t), \mathcal{D}(pk_{\mathcal{D}}, E))$**:** When $\mathcal{U}$ wishes to obtain the message indexed by $i \in [1, N]$ (or conduct a dummy transaction, for which it sets $i = (N + 1)$), it first identifies a tag t in Π such that $t = (id, S \rightarrow T, i)$.

1. $\mathcal{U}$ parses $\mathsf{Cred} = (\Pi_{\mathcal{C}}, S, \sigma_{\mathsf{state}}, N_s)$, and further parses $\Pi_{\mathcal{C}}$ to find $\sigma_{S \rightarrow T,i}$.
2. $\mathcal{U}$ selects $N'_s \xleftarrow{\$} \mathbb{Z}_q$ and computes $A \leftarrow \mathrm{PedCom}(sk_{\mathcal{U}}, N'_s, \mathsf{pid}, T)$.
3. $\mathcal{U}$ then sends N_s to $\mathcal{D}$. $\mathcal{D}$ checks the database E for $(N_s, A' \neq A)$, and if it finds such an entry it aborts. Otherwise it adds (N_s, A) to E.
4. $\mathcal{U}$ parses $C_i = (A_i, B_i)$. It selects a random $v \leftarrow \mathbb{Z}_q$ and sets $V \leftarrow (A_i)^v$. It sends V to $\mathcal{D}$ and proves knowledge of $(i, v, sk_{\mathcal{U}}, \sigma_{S \rightarrow T,i}, \sigma_{\mathsf{state}}, id, S, T, N'_s)$ such that the following conditions hold:
   (a) $e(V, y) = e(g, g)^v e(V, g)^{-i}$.
   (b) $A = \mathrm{PedCom}(sk_{\mathcal{U}}, N'_s, \mathsf{pid}, T)$.
   (c) $\mathsf{CLVerify}(spk_{\mathcal{P}}, \sigma_{\mathsf{state}}, (sk_{\mathcal{U}}, N_s, \mathsf{pid}, S)) = 1$.
   (d) $\mathsf{CLVerify}(\mathcal{P}, \sigma_{S \rightarrow T,i}, (\mathsf{pid}, S, T, i)) = 1$.
5. If these proofs verify, $\mathcal{U}$ and $\mathcal{D}$ run the CL signing protocol on committed values such that $\mathcal{U}$ obtains $\sigma'_{\mathsf{state}} \leftarrow \mathsf{CLSign}(ssk_{\mathcal{D}}, A)$. $\mathcal{U}$ stores the updated credential $\mathsf{Cred}' = (\Pi_{\mathcal{C}}, T, \sigma'_{\mathsf{state}}, N'_s)$.
6. Finally, $\mathcal{D}$ returns $U = e(V, h)$ to $\mathcal{U}$ and interactively proves that U is correctly formed (this four-round proof is described in [CNs07]).

At the conclusion of this protocol, $\mathcal{U}$ obtains the message $M_i = B_i / U^{1/v}$.

Figure 6.7: A protocol for accessing data items based on the Camenisch, Neven and shelat protocol [CNs07].

Chapter 7

Conclusion and Open Problems

THIS work has proposed a number of building blocks constructing practical oblivious databases. By combining these building blocks, we believe that it is possible to construct highly efficient databases with strong security properties and flexible access control capabilities. To illustrate this, we showed how to combine the OT protocols of Chapter 4 with the access control protocols of Chapter 6.

Open Problems. This work leaves some open problems, which we will now briefly enumerate.

1. **Fully-simulatable $\mathsf{OT}^N_{k\times 1}$ from weaker assumptions.** In Chapter 4 we achieved a very efficient non-adaptive OT^N_k in the standard model using relatively weak security assumptions (DBDH). Unfortunately, we were only able to achieve *adaptive* $\mathsf{OT}^N_{k\times 1}$ in the random oracle model. While the UC-secure protocol of Chapter 5 offers one solution to that problem, it requires stronger, q-based security assumptions. Since we believe that adaptive security is critical for a practical oblivious database, we would like to achieve this under the weakest possible assumptions.

   Thus an open problem is to develop an efficient fully-simulatable (or UC-secure) $\mathsf{OT}^N_{k\times 1}$ secure in the standard model under assumptions as weak as (or weaker than) DBDH. One approach to this problem is to this problem would be to develop CCA-secure *blind decryption* [SY96] using the IBE techniques of §4.3.

2. **UC-secure Anonymous Credentials.** In Chapter 6 we used stateful anonymous credentials to implement access controls for oblivious databases. Unfortunately, CL-based credential schemes rely on rewinding for their proofs of security [CL04]; thus it was not possible to compose them with the UC-secure $\mathsf{OT}^N_{k\times 1}$ of Chapter 5. We leave as an open problem the development of a fully UC-secure (stateful) credential system that can be linked to our OT constructions.

3. **Committing and *Unique* Identity-Based Encryption Schemes.** The IBE-based protocols of Chapter 4 require an IBE scheme that is *committing — i.e.,* it is difficult for a PKG to generate two valid keys that open a given ciphertext to different values. As we noted in §4.3.3, this property may not hold for some IBE schemes, *e.g.,* that of Gentry [Gen06]. Thus, we believe that it is an interesting problem to identify other Blind IBE schemes that are either committing, or even *unique* (*i.e.,* there is at most one key per identity).

4. **Keyword Searches and Complex Queries.** The protocols in this work considered a very basic model of database access where the user already knows the index of the item to be requested. Practical databases applications typically require complex queries *e.g.,* keyword searches. In §4.4 we noted that *anonymous* blind IBE can be used to implement private searches on encrypted data. This is a first step. However, we believe that it is an open question to permit even more complex query types.

Bibliography

[ACdM05] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *ACM Conference on Computer and Communications Security*, pages 92–101. ACM Press, 2005.

[AH99] L. Adleman and M. Huang. Function field sieve methods for discrete logarithms over finite fields. *Information and Computation*, 151:5—16, 1999.

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2001.

[Bak07] Stephen Baker. Google and the wisdom of clouds. *BusinessWeek*, December 2007. Available from `http://www.businessweek.com/magazine/content/07_52/b4064048925836.htm`.

[BB04a] Dan Boneh and Xavier Boyen. Efficient selective-ID secure Identity-Based Encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EU-*

ROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, volume 3027 of *Lecture Notes in Computer Science*, pages 382–400. Springer, 2004.

[BBDP01] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 566–582, London, UK, 2001. Springer-Verlag.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference*, volume 3152 of *Lecture Notes in Computer Science*, pages 45–55. Springer, 2004.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. Non-interactive anonymous credentials. In Ran Canetti, editor, *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 356–374. Springer, 2008.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.

[BCR86] Gilles Brassard, Claude Crépeau, and Jean-Marc Robert. All-or-nothing disclosure of secrets. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 234–238. Springer, 1986.

BIBLIOGRAPHY

[BDS+03] Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana Smetters, Jessica Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03)*, page 180, Washington, DC, USA, 2003. IEEE Computer Society.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[BGdMM05] Lucas Ballard, Matthew Green, Breno de Medeiros, and Fabian Monrose. Correlation-resistant storage from keyword searchable encryption. Johns Hopkins University, Technical Report., 2005. Available at `http://eprint.iacr.org/2005/417`.

[BGH07] Dan Boneh, Craig Gentry, and Michael Hamburg. Space-efficient identity based encryption without pairings. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 647–657. IEEE Computer Society, 2007. Available at `http://crypto.stanford.edu/~dabo/pubs.html`.

[BJ06] Michael Barbaro and Tom Zeller Jr. A Face is Exposed for AOL Searcher No. 4417749. *The New York Times*, August 2006. `http://www.nytimes.com/2006/08/09/technology/09aol.html`.

[BK04] I. F. Blake and V. Kolesnikov. Strong Conditional Oblivious Transfer and Computing on Intervals. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security*, volume 3329 of *Lecture Notes in Computer Science*, pages 515–529. Springer, 2004.

[BL88] D. Elliot Bell and Leonard J. LaPadula. Secure Computer System: Unified Exposition and Multics Interpretation. *Comm. of the ACM*, 1:271–280, 1988.

BIBLIOGRAPHY

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil Pairing. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001.

[BM89] Mihir Bellare and Silvio Micali. Non-interactive oblivious transfer and applications. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference*, volume 435 of *Lecture Notes in Computer Science*, pages 547–557. Springer, 1989.

[BMW05] Xavier Boyen, Qixiang Mei, and Brent Waters. Simple and efficient CCA2 security from IBE techniques. In Vijay Atluri, Catherine Meadows, and Ari Juels, editors, *ACM Conference on Computer and Communications Security*, pages 320–329. ACM, 2005.

[BN89] David F. C. Brewer and Michael J. Nash. The Chinese Wall Security Policy. In *IEEE Symposium on Security and Privacy*, pages 206–214. IEEE Computer Society Press, May 1989.

[Bol03] Alexandra Boldyreva. Threshold, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme. In Yvo Desmedt, editor, *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography*, volume 2567 of *Lecture Notes in Computer Science*, pages 31–46. Springer, 2003.

[Bos08] Martin H. Bosworth. Hackers Hit T.J.Maxx, Marshalls: Customer Data Exposed in Major Data Breach. *Consumer Affairs*, October 2008. `http://www.consumeraffairs.com/news04/2007/01/tj_maxx_data.html`.

[Bou00] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000,*

International Conference on the Theory and Application of Cryptographic Techniques, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer, 2000.

[BP97] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer, 1997.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security - CCS '93*, pages 62–73, Fairfax, VA, November 1993. ACM.

[BW06] Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.

[BW07] Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2007.

[Can01] Ran Canetti. Universally Composable Security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001*, pages 136–145, Las Vegas, Nevada, USA, October 2001. IEEE Computer Society. Available from `http://eprint.iacr.org/2000/067`.

BIBLIOGRAPHY

[Can08] Ran Canetti. Universally composable security: Towards the bare bones of trust. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security*, volume 5350 of *Lecture Notes in Computer Science*, pages 88–112. Springer, 2008.

[CCS07] L. Chen, Z. Cheng, and Nigel Smart. Identity-based key agreement protocols from pairings. *International Journal of Information Security*, 6:213–241, August 2007.

[CDM00] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography, Third International Workshop on Practice and Theory in Public Key Cryptography, PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–373. Springer, 2000.

[CDPW07] R. Canetti, Y. Dodis, R. Pass, and S. Walfish. Universally composable security with pre-existing setup. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007*, volume 4392 of *Lecture Notes in Computer Science*, pages 61–85. Springer, 2007.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference*, volume 2139 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 2001.

[CFGN96] Ran Canetti, Uri Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *Proc. of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 639–648, 1996.

[CFT98] Agnes Chan, Yair Frankel, and Yiannis Tsiounis. Easy come – easy go divisible cash. In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques*, volume 1403 of *Lecture Notes in Computer Science*, pages 561–575. Springer, 1998.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[CGH09] Scott Coull, Matthew Green, and Susan Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In Stanislaw Jarecki and Gene Tsudik, editors, *The International Conference on Theory and Practice of Public-Key Cryptography (PKC 2009)*, 2009. Available at `http://eprint.iacr.org/2008/563`.

[CH02] Jan Camenisch and Els Van Herreweghen. Design and implementation of the IDEMIX anonymous credential system. In *CCS '02*, pages 21–30. ACM, 2002.

[CH05] Craig Chatfield and Rene Hexel. User identity and ubiquitous computing: User selected pseudonyms. In *Workshop on UbiComp Privacy PRIVACY IN CONTEXT*, 2005.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO '82*, pages 199–203. Plenum Press, 1982.

[Cha85] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[Che06] Jung Hee Cheon. Security analysis of the strong diffie-hellman problem. In *EUROCRYPT '06*, volume 4004 of LNCS, pages 1–11, 2006.

BIBLIOGRAPHY

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from Identity Based Encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3027 of LNCS of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.

[CHK+06] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: Efficient periodic n-times anonymous authentication. In *ACM CCS '06*, pages 201–210, 2006.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *EUROCRYPT '05*, volume 3494 of LNCS, pages 302–321, 2005.

[CHL06] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Balancing accountability and privacy using e-cash. In *SCN '06*, volume 4116 of LNCS, pages 141–155, 2006.

[CKDS09] Jan Camenisch, Markulf Kohlweiss, Alfredo Rial Durán, and Caroline Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public-key encrypted data. In Stanislaw Jarecki and Gene Tsudik, editors, *The International Conference on Theory and Practice of Public-Key Cryptography (PKC 2009)*, Lecture Notes in Computer Science. Springer, 2009.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(5):965–981, 1998.

[CL01] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques*, volume 2045 of LNCS of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Security in Communication Networks '02*, volume 2576 of LNCS, pages 268–289, 2002.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO '04*, volume 3152 of LNCS, pages 56–72. Springer, 2004.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC '02*, pages 494–503. ACM Press, 2002.

[CM99] Jan Camenisch and Markus Michels. Proving in zero-knowledge that a number n is the product of two safe primes. In *EUROCRYPT '99*, volume 1592 of LNCS, pages 107–122, 1999.

[CNN05] Info on 3.9M Citigroup customers lost, 2005. `http://money.cnn.com/2005/06/06/news/fortune500/security_citigroup/`.

[CNs07] Jan Camenisch, Gregory Neven, and abhi shelat. Simulatable adaptive oblivious transfer. In *EUROCRYPT '07*, volume 4515 of LNCS, pages 573–590, 2007.

[Coc01] Clifford Cocks. An identity based encryption scheme based on Quadratic Residues. In *Cryptography and Coding, IMA International Conference*, volume 2260 of LNCS, pages 360–363, 2001.

[COR99] Giovanni Di Crescenzo, Rafail Ostrovsky, and S. Rajagopolan. Conditional oblivious transfer and time released encryption. In *EUROCRYPT '99*, volume 1592, pages 74–89, 1999.

[CR03] Ran Canetti and Tal Rabin. Universal composition with joint state. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference*, volume 2729 of *Lecture Notes in Computer Science*, pages 265–281. Springer, 2003.

[Cré87] Claude Crépeau. Equivalence between two flavours of oblivious transfer. In *CRYPTO '87*, volume 293 of LNCS, pages 350–354. Springer, 1987.

[CS97] Jan Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *CRYPTO '97*, volume 1296 of LNCS, pages 410–424, 1997.

[CS05] Sanjit Chatterjee and Palash Sarkar. Trading time for space: Towards an efficient IBE scheme with short(er) public parameters in the standard model. In *ICISC 2005*, volume 3935 of LNCS, pages 424–440, 2005.

[CS06] Sanjit Chatterjee and Palash Sarkar. HIBE with Short Public Parameters without Random Oracle. In *ASIACRYPT '06*, volume 4284 of LNCS, pages 145–160, 2006.

[CT05] Cheng-Kang Chu and Wen-Guey Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *PKC '05*, volume 3386 of LNCS, pages 172–183, 2005.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT No.2(6):644–654, November 1976.

[DHRS04] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In *TCC '04*, volume 2951 of LNCS, pages 446–472, 2004.

[DNO08] Ivan Damgård, Jesper Buus Nielsen, and Claudio Orlandi. Essentially optimal universally composable oblivious transfer. Cryptology ePrint Archive, Report 2008/220, 2008. To appear in the Proceedings of ICISC 2008. Available at `http://eprint.iacr.org/2008/220`.

[DoD85] Trusted Computer System Evaluation Criteria. Technical Report DoD 5200.28-STD, Department of Defense, December 1985.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A Verifiable Random Function with Short Proofs an Keys. In *Public Key Cryptography*, volume 3386 of LNCS, pages 416–431, 2005.

BIBLIOGRAPHY

[EGL82] Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. In Ernest F. Brickell, editor, *CRYPTO '82*, volume 740 of *Lecture Notes in Computer Science*, pages 205–210. Springer, 1982.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In *TCC '05*, volume 3378 of LNCS, pages 303–324, 2005.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO '97*, volume 1294 of LNCS, pages 16–30, 1997.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86*, volume 263 of LNCS, pages 186–194, 1986.

[Gen06] Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT '06*, volume 4004 of LNCS, pages 445–464, 2006.

[GH08a] Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. Cryptology ePrint Archive, Report 2008/383, 2008. Available at `http://eprint.iacr.org/2008/383`.

[GH08b] Matthew Green and Susan Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security*, volume 5350 of *Lecture Notes in Computer Science*. Springer, 2008.

[GH08c] Matthew Green and Susan Hohenberger. Universally composable adaptive oblivious transfer. In Josef Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security*, volume 5350 of *Lecture Notes in Computer Science*. Springer, 2008. Full version available at `http://eprint.iacr.org/163`.

BIBLIOGRAPHY

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal of Computing*, 18(1):186–208, 1989. First published at STOC 1985.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, STOC '87*, pages 218–229, New York, New York, USA, 1987. ACM.

[Gon06] Antone Gonsalves. AOL exposes search data on 658,000 people. *TechWeb*, August 2006. `http://www.techweb.com/wire/security/191801184`.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358. Springer, 2006.

[Goy07] Vipul Goyal. Reducing trust in the PKG in identity based cryptosystems. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference*, volume 4622 of *Lecture Notes in Computer Science*, pages 430–447. Springer, 2007.

[GPS06] S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. Available from `http://eprint.iacr.org/165`.

[GS02] Craig Gentry and Alice Silverberg. Hierarchical ID-Based cryptography. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002.

BIBLIOGRAPHY

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, 2008.

[HK07] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. Cryptology ePrint Archive, Report 2007/118, 2007. Originally appeared in EUROCRYPT '05. Available at `http://eprint.iacr.org/2007/118`.

[Hru08] Juel Hruska. Employees, not hackers, cause most corporate data loss, October 2008. Available from `http://arstechnica.com/`.

[JN01] Antoine Joux and Kim Nguyen. Separating decision diffie-hellman from diffie-hellman in cryptographic groups. Available from `http://eprint.iacr.org/2001/003/`, 2001.

[Jou00] Antoine Joux. A one-round protocol for tripartite Diffie-Hellman. In *Proceedings of ANTS-IV conference*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394, 2000.

[Kal05] Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. In *EUROCRYPT '05*, volume 3494 of LNCS, pages 78–95, 2005.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC '88*, pages 20–31, Chicago, Illinois, USA, 1988. ACM.

[Lam69] Butler W. Lampson. Dynamic Protection Structures. In *AFIPS Conference*, volume 35, pages 27–38, 1969.

[Lin08] Yehuda Lindell. Efficient fully-simulatable oblivious transfer. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at*

the RSA Conference 2008, volume 4964 of *Lecture Notes in Computer Science*. Springer, 2008. Available at `http://eprint.iacr.org/2008/035`.

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *SAC '99: Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography*, pages 184–199. Springer, 1999.

[Lys02] Anna Lysyanskaya. *Signature schemes and applications to cryptographic protocol design*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, September 2002.

[Mar07] John Markoff. Software via the Internet: Microsoft in 'Cloud' Computing. *The New York Times*, September 2007. `http://www.nytimes.com/2007/09/03/technology/03cloud.html`.

[Men05] Alfred Menezes. An introduction to pairing-based cryptography. Notes from lectures given in Santander, Spain, 2005. Available at `http://www.math.uwaterloo.ca/~ajmeneze/publications/pairings.pdf`.

[Mil04] Victor Miller. The weil pairing, and its efficient calculation. *Journal of Cryptology*, 17:235—261, 2004.

[MS98] Shingo Miyazaki and Kouichi Sakurai. A more efficient untraceable e-cash system with partially blind signatures based on the discrete logarithm problem. In Rafael Hirschfeld, editor, *Financial Cryptography, Second International Conference, FC '98*, volume 1465 of *Lecture Notes in Computer Science*, pages 296–308. Springer, 1998.

[MVO91] Alfred Menezes, Scott Vanstone, and Tatsuaki Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 80–89, New York, NY, USA, 1991. ACM.

BIBLIOGRAPHY

[Nac05] David Naccache. Secure and *practical* identity-based encryption. Cryptology ePrint Archive, Report 2005/369, 2005. `http://eprint.iacr.org/`.

[NP99a] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, STOC '99*, pages 245–254, Atlanta, Georgia, USA, 1999. ACM.

[NP99b] Moni Naor and Benny Pinkas. Oblivious transfer with adaptive queries. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference*, volume 1666 of LNCS of *Lecture Notes in Computer Science*, pages 573–590. Springer, 1999.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, SODA '01*, pages 448–457, Washington, DC, USA, January 2001. ACM/SIAM.

[OK04] Wakaha Ogata and Kaoru Kurosawa. Oblivious keyword search. *Journal of Complexity, special issue on coding and cryptography*, 20(2-3):356–371, 2004.

[Oka06] Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99. Springer, 2006.

[Par95] The European Parliament. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, 1995. `http://www.cdt.org/privacy/eudirective/EU_Directive_.html`.

[Pas08] Rafael Pass and abhi shelat. *A Course in Cryptography*. (manuscript), 2008. Available from `http://www.cc.gatech.edu/~atk/teaching/notes-crypto-spring08.pdf`.

[Ped92] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '92*, volume 576 of LNCS, pages 129–140, 1992.

[Pol78] J. M. Pollard. Monte Carlo methods for index computation $(\mathrm{mod} p)$. *Mathematics of Computation*, 32(143):918–924, July 1978.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[Rab81] Michael Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[Sch91] Claus-Peter Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):239–252, 1991.

[Sco02] Mike Scott. Authenticated ID-based key exchange and remote log-in with simple token and PIN number, 2002. Available at `http://eprint.iacr.org/2002/164`.

[sec08] QKD network demonstration and conference. `http://www.secoqc.net/html/conference/`, 2008.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology, Proceedings of CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Tech-*

niques, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266. Springer, 1997.

[Sur] Surety, LLC. Surety LLC. `http://www.surety.com/`.

[SY96] K. Sakurai and Y. Yamane. Blind decoding, blind undeniable signature and their application to privacy protection. In Ross J. Anderson, editor, *Information Hiding, First International Workshop*, volume 1174 of *Lecture Notes in Computer Science*, pages 257–264. Springer, 1996.

[TFS04] I. Teranishi, J. Furukawa, and K. Sako. k-Times Anonymous Authentication. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security*, volume 3329 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2004.

[Uni96] United States Congress. Health Insurance Portability and Accountability Act (HIPAA), 1996. Available at `http://aspe.hhs.gov/admnsimp/pl104191.htm`.

[Ver04] Eric R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *Journal of Cryptology*, 17:277–296, 2004.

[Ver05] Verisign. Verisign Code Signing for Microsoft Authenticode Technology. `http://www.verisign.com/static/030999.pdf`, 2005.

[Wat05] Brent Waters. Efficient Identity-Based Encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.

[WBDS04] Brent R. Waters, Dirk Balfanz, Glenn Durfee, and D. K. Smetters. Building an encrypted and searchable audit log. In *Proceedings of the Network and*

Distributed System Security Symposium, NDSS 2004. The Internet Society, 2004.

[Wei83] Steven Weisner. Conjugate coding. *SIGACT News*, 15:78–88, 1983.

[Yao86] Andrew Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science, FOCS '86*, pages 162–167, Toronto, Canada, October 1986. IEEE Computer Society.

[Zel06] Tom Jr. Zeller. Your life as an open book. *The New York Times*, August 2006. `http://www.nytimes.com/2006/08/12/technology/12privacy.html`.

Appendix A

Additional Material

A.1 An Alternate UC-Secure Construction from the Uniform Hidden q-SDH and q-SDLIN Assumptions

In this section we describe a second adaptive UC-secure oblivious transfer construction, which can be used as an alternative to the algorithms specified in §5.2. This construction uses an alternative set of assumptions in the *symmetric* bilinear map setting, including the SXDH assumption (see §3.3). The security of this second scheme is based on the following additional hardness assumptions:

Definition A.1.1 (Uniform q-Hidden Strong Diffie-Hellman (q-HSDH) [BW07, BCKL08]) *Let* $\mathsf{BMsetup}(1^\kappa) \rightarrow (p, \mathbb{G}, \mathbb{G}_T, e, g) = \gamma$. *For all p.p.t. adversaries* Adv*, the following probability is strictly less than* $1/\mathrm{poly}(\kappa)$*:*

$$\Pr[h \xleftarrow{\$} \mathbb{G}; x, c_1, \ldots, c_q \xleftarrow{\$} \mathbb{Z}_q; (A, B, C) \leftarrow \mathsf{Adv}(\gamma, g, g^x, h, (g^{1/(x+c_1)}, g^{c_1}, h^{c_1}) \in \mathbb{G}^3, \ldots, (g^{1/(x+c_q)}, g^{c_q}, h^{c_q}) \in \mathbb{G}^3) : (A, B, C) = (g^{1/(x+c)}, g^c, h^c) \wedge c \notin \{c_1, \ldots, c_q\}].$$

Boyen and Waters did not specify the distribution for sampling the c_i values in q-HSDH [BW07]. Following Belenkiy *et al.* [BCKL08], we explicitly require that they be sampled uniformly from $\mathbb{Z}_q$.

Definition A.1.2 (q-Strong Decision Linear (q-SDLIN)) *Let* $\mathsf{BMsetup}(1^\kappa) \rightarrow (p, \mathbb{G}, \mathbb{G}_T, e, g) = \gamma$. *Let* u, v, h *be random elements in* $\mathbb{G}$ *and* x_1, x_2, r_i, s_i *be random values in* $\mathbb{Z}_q$*, then given the values* $(\gamma, u, v, h, u^{x_1}, u^{x_2}, \{u^{r_i}, v^{s_i}, u^{1/(x_1+r_i)}, v^{1/(x_2+s_i)}\}_{i\in[1,q]})$*, no p.p.t. adversary* Adv *can distinguish* $\{h^{r_i+s_i}\}_{i\in[1,q]}$ *from* q *random values in* $\mathbb{G}$ *with non-negligible advantage.*

A.1.1 The Construction

This $\mathsf{OT}^N_{k\times 1}$ fits within the framework described in Figure 5.1, but uses an alternative set of algorithms ($\mathsf{OTGenCRS}$, $\mathsf{OTInitialize}$, $\mathsf{OTRequest}$, $\mathsf{OTRespond}$, $\mathsf{OTComplete}$), which we will now describe:

$\mathsf{OTGenCRS}(1^\kappa)$. Given security parameter κ, generate parameters for a bilinear mapping $\gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathsf{BMsetup}(1^\kappa)$. Compute $GS_S \leftarrow \mathsf{GSSetup}(\gamma)$ and $GS_R \leftarrow \mathsf{GSSetup}(\gamma)$. Choose random values $g_1, g_2, h \in \mathbb{G}$ and output $\mathsf{crs} = (\gamma, GS_S, GS_R, g_1, g_2, h)$.

$\mathsf{OTInitialize}(\mathsf{crs}, m_1, \ldots, m_N)$. This algorithm is executed by the Sender. On input a collection of N messages and the crs, it outputs a commitment to the database, T, for publication to the Receiver, together with a Sender secret key, sk. We treat messages as elements of $\mathbb{G}$, since there exist efficient mappings between strings in $\{0,1\}^\ell$ and elements in $\mathbb{G}$ (e.g., [BF01, ACdM05]).

1. Choose random values $x_1, x_2, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{Z}_q$.
2. Set $(u_1, u_2) \leftarrow (h^{1/x_1}, h^{1/x_2})$, and $pk \leftarrow (u_1, u_2, u_1^{\alpha_1}, u_2^{\alpha_2}, g_2^{\alpha_3})$.
3. For $j = 1, \ldots, N$ encrypt each message m_j as:
   (a) Select random $r, s, t \in \mathbb{Z}_q$.
   (b) Set $C_j \leftarrow \Big(u_1^r,\ u_2^s,\ g_1^r,\ g_2^s,\ m_j \cdot h^{r+s},\ u_1^{1/(\alpha_1+r)},\ u_2^{1/(\alpha_2+s)},\ g_2^t,\ (u_1^r u_2^s h)^t g_1^{\alpha_3}\Big)$.
4. Set $T \leftarrow (pk, C_1, \ldots, C_N)$ and $sk \leftarrow (x_1, x_2)$. Output (T, sk).

Notice that the value T has a structure that can be publicly verified. Represent pk as $(p_1, \ldots, p_5)$. Parse each ciphertext C_i as $(c_1, \ldots, c_9)$ and check that the following conditions hold:

$$\begin{aligned} e(p_1, c_3) = e(c_1, g_1) \quad &, \quad e(p_2, c_4) = e(c_2, g_2) \\ e(c_6, p_3 \cdot c_1) = e(p_1, p_1) \quad &, \quad e(c_7, p_4 \cdot c_2) = e(p_2, p_2) \\ &, \quad e(g_2, c_9)/e(c_8, c_1 \cdot c_2 \cdot h) = e(g_1, p_5). \end{aligned}$$

$\mathsf{OTRequest}(\mathsf{crs}, T, \sigma)$. This algorithm is executed by a Receiver. On input T generated by the Sender, along with an item index σ, generates a query Q for transmission to the Sender.

1. Parse T as $(pk, C_1, \ldots, C_N)$, and ensure that it is correctly formed (see above). If T is not correctly formed, abort the protocol. This check need be done only once.
2. Parse crs as $(\gamma, GS_S, GS_R, g_1, g_2, h)$, pk as $(p_1, \ldots, p_5)$, and C_σ as $(c_1, \ldots, c_9)$.

3. Select random $v_1, v_2 \in \mathbb{Z}_q$ and set $d_1 \leftarrow (c_1 \cdot p_1^{v_1}), d_2 \leftarrow (c_2 \cdot p_2^{v_2})$, $t_1 \leftarrow h^{v_1}$, $t_2 \leftarrow h^{v_2}$.
4. Use the Groth-Sahai techniques and reference string GS_R to compute the Witness-Indistinguishable proof of values pertaining to the ciphertext C_σ (which the Receiver wishes to have the Sender help him open) and blinding values:

$$\begin{aligned}\pi = NIWI_{GS_R}\{(c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9, t_1, t_2) : \\ e(c_6, p_3 \cdot c_1) = e(p_1, p_1) \ \wedge \ e(p_1, c_3)e(c_1, g_1^{-1}) = 1 \ \wedge \\ e(c_7, p_4 \cdot c_2) = e(p_2, p_2) \ \wedge \ e(p_2, c_4)e(c_2, g_2^{-1}) = 1 \ \wedge \\ e(d_1 \cdot c_1^{-1}, h)e(p_1^{-1}, t_1) = 1 \ \wedge \ e(d_2 \cdot c_2^{-1}, h)e(p_2^{-1}, t_2) = 1 \ \wedge \\ e(g_2, c_9)e(c_8, c_1 \cdot c_2 \cdot h)^{-1} = e(g_1, p_5)\}\end{aligned}$$

   To explain what is happening in this statement, first observe that the second and fourth equations ensure that (p_1, g_1, c_1, c_3) and (p_2, g_2, c_2, c_4) are both DDH tuples. Thus, for some values of $r, s \in \mathbb{Z}_q$, we have that $p_1^r = c_1$, $g_1^r = c_3$, $p_2^s = c_2$ and $g_2^s = c_4$. Under this characterization of (c_1, c_2) and with $(p_1, \ldots, p_5)$ all public, the first and third equations ensure that $c_6 = p_1^{1/(\alpha_1 + r)}$ and $c_7 = p_2^{1/(\alpha_2 + s)}$, where $p_3 = p_1^{\alpha_1}$ and $p_4 = p_2^{\alpha_2}$ for some values $\alpha_1, \alpha_2 \in \mathbb{Z}_q$. The next two equations guarantee that if we view $d_1 = p_1^{v_1 + r}$ and $d_2 = p_2^{v_2 + s}$, for some values $v_1, v_2 \in \mathbb{Z}_q$, then $t_1 = h^{v_1}$ and $t_2 = h^{v_2}$. Finally, the last equation ensures that if we represent $c_8 = g_2^t$ and $p_5 = g_2^{\alpha_3}$ for some t, α_3, then $c_9 = (c_1 c_2 h)^t \cdot g_1^{\alpha_3}$. These checks guarantee that the witness used by the Receiver, and thus the decryption request being made, corresponds to one of the N ciphertexts published by the Sender.
5. Set request $Q \leftarrow (d_1, d_2, \pi)$, and private state $Q_{priv} \leftarrow (Q, \sigma, v_1, v_2)$. Output (Q, Q_{priv}).

$\mathsf{OTRespond}(\mathsf{crs}, T, sk, Q)$. This algorithm is executed by the Sender. If the Sender does not wish to answer any more requests for the Receiver, then the Sender outputs the message "reject". Otherwise, the Sender processes the Receiver's request Q as:

1. Parse crs as $(\gamma, GS_S, GS_R, g_1, g_2, h)$, T as $(pk, C_1, \ldots, C_N)$, and sk as (x_1, x_2).
2. Parse pk (from T) as $(p_1, \ldots, p_5)$.
3. Parse Q as (d_1, d_2, π) and verify proof π using GS_R. Abort if verification fails.
4. Set $a_1 \leftarrow d_1^{x_1}$, $a_2 \leftarrow d_2^{x_2}$, and $s \leftarrow a_1 \cdot a_2$.
5. Use the Groth-Sahai techniques and reference string GS_S to formulate a zero-knowledge proof that the decryption value s is properly computed:

$$\begin{aligned}\delta = NIZK_{GS_S}\{(a_1, a_2, a_3) : e(a_1, p_1)e(d_1^{-1}, a_3) = 1 \wedge e(a_2, p_2)e(d_2^{-1}, a_3) = 1 \wedge \\ e(s, a_3)e(a_1 \cdot a_2, h^{-1}) = 1 \ \wedge \ e(g, a_3) = e(g, h)\}\end{aligned}$$

Observe that the last equation ensures that $a_3 = h$. The third equation ensures that $s = a_1 \cdot a_2$, while the first two, since the values (p_1, d_1, p_2, d_2, h) are known to both parties, ensure that $a_1 = d_1^{x_1}$ and $a_2 = d_2^{x_2}$. This guarantees that s is correctly formed.

6. Output $R \leftarrow (s, \delta)$.

$\mathsf{OTComplete}(\mathsf{crs}, T, R, Q_{priv})$. This algorithm is executed by the Receiver. On input R generated by the Sender in response to a request Q, along with state Q_{priv}, outputs a message m or $\perp$. If R is the message "reject", then the Receiver outputs $\perp$. Otherwise, the Receiver does:

1. Parse crs as $(\gamma, GS_S, GS_R, g_1, g_2, h)$, T as $(pk, C_1, \ldots, C_N)$, R as (s, δ), and Q_{priv} as (Q, σ, v_1, v_2).
2. Verify proof δ using GS_S. If verification fails, output $\perp$.
3. Parse C_σ as $(c_1, c_2, c_3, c_4, c_5, \ldots)$ and output $m = c_5/(s \cdot h^{-v_1} \cdot h^{-v_2})$.

A.1.2 Efficiency Analysis

When the protocol in Figure 5.1 is implemented using the algorithms described above, we obtain a $(k + 1/2)$-round protocol with communications cost $O(N + k)$, where $k \leq N$. More concretely, the crs is comprised of 15 elements in $\mathbb{G}$, the Sender's public key contains 5 elements in $\mathbb{G}$, and each of the N ciphertexts in T requires 9 elements in $\mathbb{G}$. Moreover, each item transfer involves transmission of 95 elements of $\mathbb{G}$ from Receiver to Sender, and then 46 elements of $\mathbb{G}$ from Sender to Receiver.

The message space of our OT protocol is elements in $\mathbb{G}$, which will be sufficient for transferring a symmetric encryption key to unlock a file of arbitrary size.

A.1.3 Security Analysis

Theorem A.1.3 *Instantiated with the above algorithms,* OTA *securely realizes the functionality* $\mathcal{F}_{OT}^{N\times 1}$ *in the* $\mathcal{F}_{CRS}$-hybrid *model under the* q*-Strong Decision Linear and uniform* q*-HSDH assumptions.*

Let us now provide some intuition behind this proof. When either the Sender or the Receiver is corrupted, we wish to describe a simulator $\mathcal{S}$ such that it can interact with the ideal functionality $\mathcal{F}_{OT}^{N\times 1}$ (which we'll denote simply as $\mathcal{F}$) and the environment $\mathcal{Z}$ appropriately; i.e., $\mathsf{IDEAL}_{\mathcal{F},\mathcal{S},\mathcal{Z}} \overset{c}{\approx} \mathsf{EXEC}_{\mathsf{OTA},\mathcal{A},\mathcal{Z}}$.

We begin with the case where the real world adversary $\mathcal{A}$ corrupts the Sender, and thus $\mathcal{S}$ must interact with $\mathcal{F}$ as the ideal Sender and with (an internal copy of) $\mathcal{A}$ as a real-world Receiver. Here $\mathcal{S}$ does the following:

1. Ask $\mathcal{A}$ to begin an OT protocol, and set the crs *for these two parties* by running $\gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathsf{BMsetup}(1^\kappa)$, $GS_S \leftarrow \mathsf{GSSetup}(\gamma)$, $GS_R \leftarrow \mathsf{GSSetup}(\gamma)$, and selecting random elements $h \in \mathbb{G}$ and $a_1, a_2 \in \mathbb{Z}_q$. Set $\mathsf{crs} = (\gamma, GS_S, GS_R, h^{a_1}, h^{a_2}, h)$. When the parties query $\mathcal{F}_{CRS}$, return $(\mathsf{sid}, \mathsf{crs})$.
2. Obtain the database commitment T from $\mathcal{A}$. Verify that T is well-formed, abort if not. Otherwise, use a_1, a_2 to decrypt each ciphertext $C_i = (c_1, \ldots, c_9)$ as $m_i = c_5/(c_3^{1/a_1} \cdot c_4^{1/a_2})$. Map each element $m_i \in \mathbb{G}$ to a string in $\{0,1\}^\ell$ [ACdM05]. Send $(\mathsf{sid}, \mathbf{S}, m_1, \ldots, m_N)$ to $\mathcal{F}$.
3. Upon receiving $(\mathsf{sid}, \mathsf{request})$ from $\mathcal{F}$, choose a random index $\sigma \in [1, N]$ and return $\mathsf{OTRequest}(\mathsf{crs}, T, \sigma)$ to $\mathcal{A}$. This response includes two random values d_1, d_2 and a non-interactive witness indistinguishable proof with respect to $GS_R \in \mathsf{crs}$ that d_1, d_2 correspond to a ciphertext C_σ. This proof can be performed honestly and without rewinding.
4. If $\mathcal{A}$ issues a "reject" message or responds with anything other than a value in $\mathbb{G}$ and a valid NIZK proof, then $\mathcal{S}$ tells $\mathcal{F}$ to fail the request by sending message $(\mathsf{sid}, 0)$. Otherwise, $\mathcal{S}$ sends the message $(\mathsf{sid}, 1)$ to $\mathcal{F}$.

The indistinguishability argument here follows from the indistinguishability of the crs (from a real crs), the perfect extraction of the messages m_i, and the WI proof during each request phase, which guarantees that $\mathcal{A}$ (the corrupt Sender) cannot be selectively choosing to fail based on the Receiver's choices. Thus, $\mathcal{S}$ can adequately mimic its response pattern.

Next, we consider the case where the real world adversary $\mathcal{A}$ corrupts the Receiver, and thus $\mathcal{S}$ must interact with $\mathcal{F}$ as the ideal Receiver and with (and internal copy of) $\mathcal{A}$ as real-world Receiver. This case requires that the $q = N$ for the uniform q-HSDH assumption. Here $\mathcal{S}$ does the following:

1. Ask $\mathcal{A}$ to begin an OT protocol, and set the crs *for these two parties* by running $\gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathsf{BMsetup}(1^\kappa)$, $(GS_S, td_{sim}) \leftarrow \mathsf{GSSimulateSetup}(\gamma)$ and $(GS_R, td_{ext}) \leftarrow \mathsf{GSExtractSetup}(\gamma)$. Select random $g_1, g_2, h \in \mathbb{G}$. Set $\mathsf{crs} \leftarrow (\gamma, GS_S, GS_R, g_1, g_2, h)$. When the parties query $\mathcal{F}_{CRS}$, return $(\mathsf{sid}, \mathsf{crs})$.
2. $\mathcal{S}$ must commit to a database of messages for $\mathcal{A}$ *without* knowing the messages $m_1, \ldots, m_N$. Thus, $\mathcal{S}$ simply commits to arbitrary junk messages, and sends the corresponding T to $\mathcal{A}$.
3. When $\mathcal{A}$ makes a transfer request, $\mathcal{S}$ uses td_{ext} to extract the witness corresponding to the index σ from the NIWI proof. (This extraction is done via opening perfectly-binding commitments which are includes in the WI proof and does not require any rewinding.)
4. $\mathcal{S}$ now sends $(\mathsf{sid}, \mathbf{R}, \sigma)$ to $\mathcal{F}$ to obtain the real m_σ message.
5. Now, $\mathcal{S}$ returns a response to $\mathcal{A}$ which opens C_σ to m_σ and then uses td_{sim} to simulate an NIZK proof that this opening is correct. The NIZK proof here is designed in such a way that simulation is always possible and no rewinding is necessary.

The indistinguishability argument here follows from the indistinguishability of the crs (from a real crs), the indistinguishability of the "fake" database T, the ability to extract

witnesses from the NIWI proofs, and the zero-knowledge property of "fake" NIZK proofs. Notice that $\mathcal{S}$ is never *both* simulating and extracting via the same (subsection of the) common reference string; indeed, we do not require that the proofs be simulation-sound.

Appendix B

Access Control Models

A number of access control models can be used to describe access permissions for resources through the use of our stateful credential system and its extensions. The most widely used form of access controls are discretionary access control systems [DoD85], where access permissions are applied arbitrarily as they are needed. For instance, a systems administrator can describe a list of resources that a given user can access, otherwise known as a capabilities list [Lam69]. Such an access model is trivially achieved in our credential system as a separate graph for each user with a single state and a self loop with tags for each of the contiguous ranges of resources that the user can access.

Mandatory access control models, however, are far more interesting because of their use of the user's access history to enforce an access policy. These history-dependent access controls are difficult to capture with typical capabilities or access list implementations due to their dynamic nature. Here, we describe two non-trivial access control models used in real-world systems, the Brewer-Nash [BN89] and Bell-LaPadula [BL88] models, and provide an example policy graph for each in Figures B.1 and B.2, respectively.

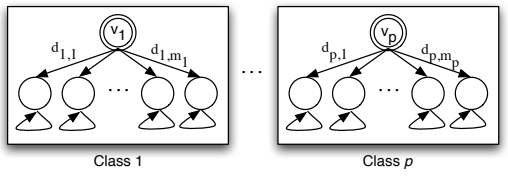

Figure B.1: Example access graphs for the Brewer-Nash model. The user receives one access graph per class, where each access graph allows access to at most one of the datasets $d_{i,j}$ for the associated conflict of interest class i.

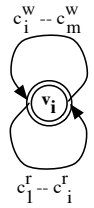

Figure B.2: Example access graph for a user with security level i in the Bell-LaPadula model. The graph allows read access to all resources in classes c_1^r through c_i^r and write access to all objects in classes c_i^w to c_m^w.

Brewer-Nash Model. The Brewer-Nash model [BN89], otherwise known as the Chinese Wall, is a mandatory access control model that is widely used in the financial services industry to prevent an employee from working on the finances of two companies that are in competition with one another. Intuitively, the resources in the system are first divided into groups based on the company they are associated with, called *datasets*. These datasets are further grouped into *conflict of interest classes* such that all of the companies that are in competition with one another have their datasets in the same class. The model ensures that once a user chooses an object from a dataset in a given class, that user has unrestricted access to all objects in the selected dataset, but no access to objects in any other dataset in that class. In Figure B.1, we denote the j^{th} dataset in class i as $d_{i,j}$, which we can succinctly represent in our access graphs using either the class label extension, or hidden range proof extension from Section 6.2.4.

Bell-LaPadula Model. Another well-known mandatory access control model is the Bell-LaPadula model [BL88], which is a Multilevel Security model. The Bell-LaPadula model is designed with the intent of maintaining data confidentiality in a classified computer system, and it is typically used in high security environments. In this security model, resources, and users are labeled with a security level (*e.g.,* top secret, secret, etc.). The security level labels are strictly ordered and provide a hierarchy that describes the sensitivity of information. The two basic properties of the Bell-LaPadula model state that a user cannot read a resource with a security level greater than her own, and she cannot write to resources with a security level less than her own. Therefore, the model ensures that information from highly sensitive objects cannot be written to low security objects by using the user as an intermediary. In Figure B.2, we denote the security levels as the integers $1, \ldots, m$. Furthermore, we split the access tags into separate read and write access controls through the use of separate indices. Therefore, a user with security level i gets a graph with tags $c_i^w, \ldots, c_m^w$ that allow her to write to any resource with a higher security level, and tags $c_1^r, \ldots, c_i^r$ that allow her to read any resource with a lower security level. Again, these ranges of resources can be succinctly represented by the extensions of Section 6.2.4.

Appendix C

Other Security Proofs

C.1 Proof of Theorem 4.3.4 (Boyen-Waters Anonymous IBE)

Unlike the BlindExtract protocols for the BB scheme, the protocol proposed above does *not* reduce generically to the security of the BW cryptosystem. Instead we must slightly modify the reduction. (In point of fact, the BW scheme has multiple reductions, for the separate properties of semantic security and anonymity. Our changes are compatible with each.) As we only make changes to the key generation algorithm, we do not quote the entire proof here.

To implement blind extraction, we define a new "helper" algorithm, which we refer to as ModExtract. We show that the scheme retains its IND-sID-CPA security even when the adversary has oracle access to this algorithm *as well as* the normal extraction algorithm. We then show leak-freeness for the BlindExtract protocol by using the ModExtract algorithm to help respond to protocol initiations.

ModExtract(msk, id, v). For user-specified $id, v \in \mathbb{Z}_q$, this algorithm is equivalent to calling the standard key extraction algorithm after replacing ω in msk with (ω/v). Thus, keys returned have the structure:

$$\left[g^{r_1 t_1 t_2 + r_2 t_3 t_4}, g^{\frac{-\omega}{v} t_2} (g_0 g_1^{id})^{-r_1 t_2}, g^{\frac{-\omega}{v} t_1} (g_0 g_1^{id})^{-r_1 t_1}, (g_0 g_1^{id})^{-r_2 t_4}, (g_0 g_1^{id})^{-r_2 t_3}\right]$$

Semantic Security. The IND-sID-CPA security of the BW scheme is based on the DBDH assumption. The full simulation is described in the original paper. The only portion of the simulation that needs to be modified is the key simulation of key extraction, which we replace with a simulations of the ModExtract algorithm (clearly, selecting $v = 1$

makes ModExtract equivalent to using the standard extraction algorithm.) To answer a ModExtract query, compute d_3, d_4 as in the original simulation. Compute the first three elements d_0, d_1, d_2 as:

$$\left[\left((g^{z_2})^{\frac{-1}{(id-id^*)v}} g^{r_1}\right)^{t_1 t_2} g^{r_2 t_3 t_4}, \left((g^{z_2})^{\frac{y}{(id-id^*)v}} (g_0 g_1^{id})^{r_1}\right)^{-t_2}, \left((g^{z_2})^{\frac{y}{(id-id^*)v}} (g_0 g_1^{id})^{r_1}\right)^{-t_1}\right]$$

The remainder of the simulation remains unchanged. The value t_1 is not included in the first element. Note that for the randomly distributed exponent $\tilde{r}_1 = r_1 - z_2/(id - id^*)v$ these elements have the correct form:

$$\left[g^{\tilde{r}_1 t_1 t_2 + r_2 t_3 t_4}, g^{-\omega t_2} (g_0 g_1^{id})^{-\tilde{r}_1 t_2}, g^{-\omega t_1} (g_0 g_1^{id})^{-\tilde{r}_1 t_1}\right]$$

Anonymity. The anonymity of the BW scheme is based on DLIN. Modifying the reduction in this case is extremely simple, since the parameter ω is chosen by the simulation. Thus the ModExtract queries are answered as in the original simulation, except that the simulator computes (ω/v) and uses this value in place of ω during each query. The rest of the simulation remains unchanged.

Security of BlindExtract**.** With this algorithm in place, we prove security of BlindExtract as follows. Let $\mathcal{A}$ be an adversary that receives $params$ and subsequently conducts instantiations of the BlindExtract protocol. We show that it is possible to answer these queries using only oracle access to the ModExtract algorithm (*i.e.,* passing chosen values id, v). Our simulation works as follows:

1. When $\mathcal{A}$ initiates a blind extraction query by submitting h and conducting $PoK\{(id, v) : h = g_0^v g_1^{v \cdot id}\}$, use the knowledge extractor for PoK to obtain id, v.
2. Next, issue a ModExtract query on id, v to obtain the secret key $(d_0, d_1, d_2, d_3, d_4)$.
3. Return to $\mathcal{A}$ the tuple $(d_0, d_1^v, d_2^v, d_3^v, d_4^v)$.

These responses are correctly distributed. Note that we do not address the committing property, or selective-failure blindness for this scheme.

C.2 Generic Group Proof of Hidden LRSW Assumption

We provide evidence to that the q-Hidden LRSW assumption may be hard. In the generic group model, elements of the bilinear groups $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are encoded as unique random strings. Thus, the adversary cannot directly test any property other than equality. Oracles are assumed to perform operations between group elements, such as performing the

group operations in $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$. The opaque encoding of the elements of $\mathbb{G}_1$ is defined as the function $\xi_1 : \mathbb{Z}_p \to \{0,1\}^*$, which maps all $a \in \mathbb{Z}_p$ to the string representation $\xi_1(a)$ of $g^a \in \mathbb{G}_1$. Likewise, we have $\xi_2 : \mathbb{Z}_p \to \{0,1\}^*$ for $\mathbb{G}_2$ and $\xi_T : \mathbb{Z}_p \to \{0,1\}^*$ for $\mathbb{G}_T$. The adversary Adv communicates with the oracles using the ξ-representations of the group elements only.

Theorem C.2.1 (Hidden LRSW is Hard in Generic Groups) *Let* Adv *be an algorithm that solves the q-Hidden LRSW problem in the generic group model. Let q_G be the number of queries* Adv *makes to the oracles computing the group action and pairing. If ξ_1, ξ_2, ξ_T are chosen at random, then the probability ϵ that* $\mathsf{Adv}(p, \xi_1(1), \xi_2(1), \xi_2(S), \xi_2(T), \{\xi_1(X_i), \xi_1(A_i), \xi_2(A_i), \xi_1(A_iX_i), \xi_1(A_iX_iT), \xi_1(A_i(S+STX_i))\}_{i\in[1,q]})$ *outputs a tuple* $(\xi_1(X), \xi_1(A), \xi_2(A), \xi_1(AX), \xi_1(AXT), \xi_1(A(S+STX)))$ *for some A, X where $A \neq 0$, $X \neq 0$ and $X \notin \{X_i\}$, is bounded by*

$$\epsilon \leq \frac{(q_G + 6q + 4)^2 \cdot 5}{p}.$$

Proof. Consider an algorithm $\mathcal{B}$ that interacts with Adv in the following game.

$\mathcal{B}$ maintains three lists of pairs $L_1 = \{(F_{1,i}, \xi_{1,i}) : i = 0, \ldots, \tau_1 - 1\}$, $L_2 = \{(F_{2,i}, \xi_{2,i}) : i = 0, \ldots, \tau_2 - 1\}$, $L_T = \{(F_{T,i}, \xi_{T,i}) : i = 0, \ldots, \tau_T - 1\}$, such that, at step τ in the game, we have $\tau_1 + \tau_2 + \tau_T = \tau + 4 + 6q$. Let the $F_{1,i}, F_{2,i}$ and $F_{T,i}$ be multivariate polynomials in $\mathbb{Z}_p[S, T, A_i, X_i]$. The $\xi_{1,i}$, $\xi_{2,i}$, and $\xi_{T,i}$ are set to unique random strings in $\{0,1\}^*$. We start the Hidden LRSW game at step $\tau = 0$ with $\tau_1 = 1 + 5q$, $\tau_2 = 3 + q$, and $\tau_T = 0$. These correspond to the polynomials $F_{1,0} = F_{2,0} = 1$, $F_{2,1} = S$, $F_{2,2} = T$, $F_{1,1} = X_1, F_{1,2} = A_1, F_{2,3} = A_1, F_{1,3} = A_1X_1, F_{1,4} = A_1X_1T$ and $F_{1,5} = A_1(S+STX_1)$, etc.

$\mathcal{B}$ begins the game with Adv by providing it with the random strings $\xi_{1,0}, \ldots, \xi_{1,5q}, \xi_{2,0}, \ldots, \xi_{2,q+2}$. Now, we describe the oracles Adv may query.

Group action: Adv inputs two group elements $\xi_{1,i}$ and $\xi_{1,j}$, where $0 \leq i, j < \tau_1$, and a request to multiply/divide. $\mathcal{B}$ sets $F_{1,\tau_1} \leftarrow F_{1,i} \pm F_{1,j}$. If $F_{1,\tau_1} = F_{1,u}$ for some $u \in \{0, \ldots, \tau_1 - 1\}$, then $\mathcal{B}$ sets $\xi_{1,\tau_1} = \xi_{1,u}$; otherwise, it sets ξ_{1,τ_1} to a random string in $\{0,1\}^* \setminus \{\xi_{1,0}, \ldots, \xi_{1,\tau_1-1}\}$. Finally, $\mathcal{B}$ returns ξ_{1,τ_1} to Adv, adds $(F_{1,\tau_1}, \xi_{1,\tau_1})$ to L_1, and increments τ_1. Group actions for $\mathbb{G}_2$ and $\mathbb{G}_T$ are handled the same way.

Pairing: Adv inputs two group elements $\xi_{1,i}$ and $\xi_{2,j}$, where $0 \leq i < \tau_1$ and $0 \leq j < \tau_2$. $\mathcal{B}$ sets $F_{T,\tau_T} \leftarrow F_{1,i} \cdot F_{2,j}$. If $F_{T,\tau_T} = F_{T,u}$ for some $u \in \{0, \ldots, \tau_T - 1\}$, then $\mathcal{B}$ sets $\xi_{T,\tau_T} = \xi_{T,u}$; otherwise, it sets ξ_{T,τ_T} to a random string in $\{0,1\}^* \setminus \{\xi_{T,0}, \ldots, \xi_{T,\tau_T-1}\}$. Finally, $\mathcal{B}$ returns ξ_{T,τ_T} to Adv, adds $(F_{T,\tau_T}, \xi_{T,\tau_T})$ to L_T, and increments τ_T.

We assume SXDH holds in $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ and therefore no ismorphism oracles exist.

Eventually Adv stops and outputs a tuple of elements $(\xi_{1,a}, \xi_{1,b}, \xi_{2,f}, \xi_{1,c}, \xi_{1,d}, \xi_{1,e})$, where $0 \leq a, b, c, d, e < \tau_1$ and $0 \leq f < \tau_2$.

Analysis of Adv's Output. We now argue that it is *impossible* for Adv's output to *always* be correct. Each output polynomial must be some linear combination of polynomials corresponding to elements available to Adv in the respective groups. Consider the polynomials $F_{1,e}$ and $F_{2,f}$.

$$F_{1,e} := e_0 + e_{1,i}X_i + e_{2,i}A_i + e_{3,i}A_iX_i + e_{4,i}A_iX_iT + e_{5,i}A_i(S + STX_i) \quad \text{(C.1)}$$

$$F_{2,f} := f_0 + f_1S + f_2T + f_{3,i}A_i \quad \text{(C.2)}$$

where a_iA_i is shorthand for $\sum_{i=1}^{q} a_iA_i$. For Adv's answer to be correct, we know their relationship must be, for some X:

$$P := F_{1,e} - F_{2,f}(S + STX) \equiv 0 \mod p.$$

By substituting in equations C.1 and C.2, we get:

$$P = e_0 + e_{1,i}X_i + e_{2,i}A_i + e_{3,i}A_iX_i + e_{4,i}A_iX_iT + e_{5,i}A_i(S + STX_i) - f_0(S + STX) - f_1S(S + STX) - f_2T(S + STX) - f_{3,i}A_i(S + STX)$$

Looking at the unique terms of this polynomial, we can immediately see that for $P \equiv 0$, it must be the case that for all i:

$$e_0 = 0\ ,\ e_{1,i} = 0\ ,\ e_{2,i} = 0\ ,\ e_{3,i} = 0\ ,\ e_{4,i} = 0,\ f_0 = 0\ ,\ f_1 = 0\ ,\ f_2 = 0$$

Thus, we are left with $P = e_{5,i}A_i(S + STX_i) - f_{3,i}A_i(S + STX)$. Since $F_{2,f} \neq 0$, we know that $f_{3,i} \neq 0$ (for at least one i) and thus $e_{5,j} \neq 0$ (for at least one j). It is easy to see that $e_{5,j}$ cannot be non-zero for more than one value, since it will not be possible to cancel both corresponding terms. Thus, the only resolution is for $X = X_j$, which is a contradiction. We conclude that Adv's success depends *solely* on his luck when the variables are instantiated.

Analysis of $\mathcal{B}$'s Simulation. At this point $\mathcal{B}$ chooses random values to instantiate the variables $s, t, x_i, a_i \in \mathbb{Z}_p$. We know that the chance of choosing a random assignment that hits the root of any given polynomial is bounded from above by the Schwartz-Zippel theorem by the degree of the polynomial divided by p. The maximum total degree of any polynomial here is 5. Taking all pairs of polynomials into consideration, we can bound the probability that a collision causes $\mathcal{B}$'s simulation to fail as $\leq \binom{q_G+6q+4}{2}5/p \leq (q_G + 6q + 4)^2 5/p$. □

Vita

Matthew Green is a Ph. D. candidate in the Johns Hopkins University Information Security Institute. His research includes the development cryptographic techniques for maintaining users privacy, as well as the deployment of privacy-friendly protocols for database access. He spent several years as a staff member at AT&T Labs/Research, and is also a co-founder of Independent Security Evaluators (ISE), a custom security evaluation firm with a global client base.

In 2005 he worked with a team at Johns Hopkins and RSA Laboratories to identify flaws in the Texas Instruments Digital Signature Transponder (DST), a cryptographically-enabled RFID device used in the Exxon Speedpass payment system and in millions of vehicle immobilizers. He is a recipient of the PET Award for contributions to the field of privacy enhancing technologies.