Witness Encryption and its Applications

Sanjam Garg UCLA

Craig Gentry[*] IBM Watson

Amit Sahai[†] UCLA

Brent Waters [‡] U.T. Austin

Abstract

We put forth the concept of *witness encryption.* A witness encryption scheme is defined for an NP language L (with corresponding witness relation R). In such a scheme, a user can encrypt a message M to a particular problem instance x to produce a ciphertext. A recipient of a ciphertext is able to decrypt the message if x is in the language and the recipient knows a witness w where $R(x, w)$ holds. However, if x is not in the language, then no polynomial-time attacker can distinguish between encryptions of any two equal length messages. We emphasize that the encrypter himself may have no idea whether x is actually in the language.

Our contributions in this paper are threefold. First, we introduce and formally define witness encryption. Second, we show how to build several cryptographic primitives from witness encryption. Finally, we give a candidate construction based on the NP-complete Exact Cover problem and Garg, Gentry, and Halevi's recent construction of "approximate" multilinear maps.

Our method for witness encryption also yields the first candidate construction for an open problem posed by Rudich in 1989: constructing computational secret sharing schemes for an NP-complete access structure.

1 Introduction

When we encrypt a message using a public-key encryption scheme, we allow the receiver to learn our message only if he knows a secret key corresponding to his public key. What if we don't really care if he knows a secret key, but we do care if he knows a solution to a crossword puzzle that we saw in the *Times*? Or if he knows a short proof for the Goldbach conjecture? Or, in general, the solution to some NP search problem? In this paper, we ask the question:

[*] This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20202. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

[†] Research supported in part from a DARPA/ONR PROCEED award, NSF grants 1228984, 1136174, 1118096, 1065276, 0916574 and 0830803, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0389. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

[‡] Supported by NSF CNS-0915361 and CNS-0952692, CNS-1228599 DARPA through the U.S. Office of Naval Research under Contract N00014-11-1-0382, DARPA N11AP20006, Google Faculty Research award, the Alfred P. Sloan Fellowship, and Microsoft Faculty Fellowship, and Packard Foundation Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense or the U.S. Government.

Can we encrypt a message so that it can only be opened
by a recipient who knows a witness to an NP relation?

We introduce the concept of *witness encryption* for general NP languages. A witness encryption scheme is defined for an NP language L (with corresponding witness relation R). In such a scheme, a user can encrypt a message M to a particular problem instance x to produce a ciphertext. A recipient of a ciphertext is able to decrypt the message if x is in the language and the recipient knows a witness w where $R(x, w)$ holds. However, if x is not in the language, then no polynomial-time attacker can distinguish between encryptions of any two equal length messages[1]. We emphasize that the encrypter himself may have no idea whether x is actually in the language.

In this paper we construct, and explore the applications of, witness encryption for NP-complete problems. Targeting witness encryption for NP-complete problems is appealing. First, we can create encryption puzzles of the type mentioned above. There are multiple real life examples where a monetary award has been offered for the solution to a puzzle or problem including: the Clay Institute Millennium Prize Problems [Ins] and the Eternity Puzzle [Web]. For these challenges one could consider encoding the problem in terms of an NP-complete problem and encrypting the password to a bank account containing the funds. Witness encryption is especially well-suited to the situation where the encrypter may not be available (or even alive) at the time when a decrypter uses a witness to decrypt the ciphertext. This distinguishes the goal of witness encryption from the interactive setting, where general secure two-party computation protocols [Yao86, GMW87] may be used for this purpose [AIR01].

Witness Encryption is closely related to the notion of computational secret sharing for NP-complete access structures, first posed by Rudich in 1989 [Rud89] (see [Bei11]). For example, consider the NP-complete 3-Exact Cover problem (Proposition 2.25, [Gol08]), where an instance is defined by a set of subsets $T_1, \ldots, T_\ell$ of $[n]$ such that each $|T_i| = 3$, and the problem is to find an *exact cover* $T_{i_1}, \ldots, T_{i_t}$ such that each element of the universe $[n]$ is contained in exactly one set T_{i_j}. The corresponding secret sharing problem would identify each of the $\binom{n}{3}$ subsets T of $[n]$ of size 3 with a different party P_T. The secret sharing scheme would require a way to take a secret x and construct potential shares λ_T for each party P_T. The two guarantees needed would be: (1) *efficient recovery*: if a set of parties $P_{T_{i_1}}, \ldots, P_{T_{i_t}}$ knew of an exact cover among their sets, then these parties would be able to efficiently recover the secret x from their shares $\lambda_{T_{i_1}}, \ldots, \lambda_{T_{i_t}}$. Note that the monotonicity of recovery is maintained here – if a set of parties contains an exact cover, so must any other superset of these parties. (2) *privacy*: if a set of parties $P_{T_{i_1}}, \ldots, P_{T_{i_t}}$ does not contain an exact cover, then these parties should not be able to distinguish between secret sharings of distinct secrets x and x'.

It is easy to see that such a Rudich-type secret sharing scheme would imply a Witness Encryption scheme; the converse, however, is not clear. However, as we note below, our construction of Witness Encryption extends to yield a Rudich-type secret sharing scheme, as well, under the same computational assumption. This yields the first candidate construction for Rudich's open problem since its posing in 1989. Later in this introduction, we briefly compare Witness Encryption to other similar concepts that have appeared in the literature.

[1] We note that this formalization does not capture the requirement that the decrypter must have *knowledge* of the witness w. Formalizing knowledge requirements in cryptography is often quite problematic, especially in non-interactive settings (see, e.g., [HT98]). Indeed, we see this clean formulation of witness encryption without explicit discussion of knowledge as an important feature of our work. We defer exploring more complex knowledge-based formulations of witness encryption to future work.

Witness encryption is also a surprisingly useful tool for building cryptographic schemes. Indeed, we show witness encryption gives intriguing new solutions with novel properties for cryptographic primitives including public key encryption [DH76, GM84], Identity-Based Encryption [Sha84, BF03] and Attribute-Based Encryption [SW05] for circuits. (Our work can be seen as extending and refining the work of Rudich [Rud89, Bei11] who showed how Rudich-type secret sharing schemes can be used for constructing novel OT protocols.)

Our contributions in this paper are threefold. First, we introduce and formally define witness encryption. Second, we show how to build several cryptographic primitives from witness encryption. Finally, we give a candidate construction based on the NP-complete Exact Cover problem [Kar72] and Garg, Gentry, and Halevi's [GGH12] recent construction of "approximate" multilinear maps

We now provide an overview of how to build several cryptographic primitives from witness encryption, and then how to build the witness encryption scheme itself.

1.1 Building Cryptographic Primitives from Witness Encryption

We demonstrate the power of witness encryption as a flexible tool for building cryptographic primitives. We consider a progression of cryptographic applications, starting with the basic case of Public-Key Encryption, moving next to Identity-Based Encryption, and finally showing how to realize Attribute-Based Encryption for general circuits. Each new step in the progression is more challenging and requires new techniques.

We also point out interesting and unique features of each system that emerge from our use of witness encryption. For instance, existing public-key encryption schemes, like RSA, all have "heavy" key generation algorithms, that require non-trivial structured mathematical computations. For instance, RSA key generation involves choosing large random prime numbers. In contrast, our public-key encryption scheme based on witness encryption has a key generation algorithm whose complexity is *independent* of the complexity of the underlying witness encryption scheme: it requires only a single evaluation of a pseudo-random generator (PRG), a primitive whose existence can be based on one-way functions [HILL99]. This also gives rise to an intriguing possibility: if it were possible to build witness encryption from one-way functions, then this would yield public-key encryption from one-way functions. We emphasize this possibility not because we think it likely that witness encryption could be easily built from one-way functions, but because the use of witness encryption to build public-key encryption is inherently *non-black-box* in the one-way function. Thus, this route to achieving public-key encryption from one-way functions would not contradict the famed black-box impossibility result of Impagliazzo and Rudich [IR89].

Similarly, in our IBE system, private keys can be constructed from *any* unique signature scheme and are not tied to any specific algebraic structure or the complexity of the witness encryption scheme. Indeed, the setup and key generation algorithm are oblivious to what underlying witness encryption system is used. Suppose we built an IBE system of this nature with a particular witness encryption system. Now suppose that later on the community discovered a witness encryption system that was better in some way (e.g. had better performance or security assurances). This new witness encryption system could be swapped in *without* requiring any changes to the public parameters or private keys. Actually, the system is even more dynamic in that individual users can choose which witness encryption system they want to use on a per-ciphertext basis. We contrast this with contemporary IBE systems where the public parameters are intimately linked with either a certain choice of pairing friendly elliptic curve [BF03], choice of Learning with Error (LWE) parameters [CHKP10, ABB10], or RSA modulus [Coc01]. If, for example, a certain class of elliptic curves were later discovered to be vulnerable to attacks, any system using them would need to be completely rekeyed starting with the authority. [2]

Technical Overview of Applications. We now give an overview of the progression of ideas for our constructions, with the details following in Section 4.

Public-Key Encryption. We begin by showing how witness encryption and pseudorandom generators (PRGs) give rise to public-key encryption. We assume a length doubling PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\cdot\lambda}$, however, any PRG that expands by a super-logarithmic number of bits will suffice. To generate a key one simply chooses a random seed s as the secret key and lets the public key be the output of $G(s) \to t$. Encryption is simply a witness encryption that t is in the output space of the PRG. A user with the secret key s can prepare a witness and decrypt the message. We prove security using a simple hybrid technique. We first switch generating t honestly to choosing t as a uniformly random string in $\{0,1\}^{2\cdot\lambda}$; with very high probability is will not be in the range of G. By the security of the PRG, the attacker's advantage should remain the same. At this point, the NP statement is no longer be true and our witness encryption security definition directly applies.

Identity-Based Encryption. Moving on to Identity-Based Encryption, we must now be able to give out several secret keys. Our approach is to turn Naor's[3] observation that IBEs give rise to signature schemes on its head and derive IBE secret keys from essentially any signature scheme with unique signatures. As a first attempt, one might try to let a secret key for identity $\mathcal{I}$ be a signature on $\mathcal{I}$. Then we can create a ciphertext by using witness encryption on an NP statement that there exists a signature σ on $\mathcal{I}$. While our correctness property states that one can decrypt, security is harder to argue. The reason is that an attacker could know that the statement is true, that there exists a signature on $\mathcal{I}$, without having any clue what the signature actually is.

As a next iteration on the idea, we will try to modify the construction by showing that from any attacker that breaks our system, we can actually extract a forgery on the challenge identity $\mathcal{I}^*$. Our method is for the encrypter to choose randomness r and to create two witness encryptions: one for the statement that there exists a signature σ in $\mathcal{I}$ where the Goldreich-Levin [GL89] hard-core bit of σ, r is 0, and the other witness encryption for the statement where it is 1. A user with a signature σ will choose the appropriate one. This idea, however, actually hits another snag in that there could be several signatures which verify for any $\mathcal{I}$. For this reason, we will use *unique* signature schemes [GO92][4] where on honest setup there will be at most one verifying signature per message. Therefore only one of the two witness encryption statements can be true. At this point we can extract a forgery for any attacking user.

Attribute-Based Encryption for Circuits. We finally move to the most complex case of achieving Attribute-Based Encryption for general circuits. Until very recently [GGH$^+$13, GVW13], no solutions to this problem were known. In this setting, a private key corresponds to a (bounded-size) boolean circuit f that takes n bit inputs, and a ciphertext corresponds to an n bit input a. If $f(a) = 1$ then the user should be able to decrypt.

If we were to follow in the steps of our IBE solution, we might give a private key for circuit f as a unique signature on f, and for the proof try to extract a signature forgery from an attacker

[2] Obviously, our system cannot run away from this problem entirely in that if our underlying signature scheme were broken this would require rekeying of the system. The main point is that there is a strong separation between the security of our keys and the more complex witness encryption component.

[3] The observation was noted in [BF03].

[4] Goldwasser and Ostrovsky call this notion as *invariant* signatures.

that decrypts the (selectively chosen) challenge input a^*. The problem with this approach is that even if the underlying signature scheme is unique, there could exists many circuits that a^* satisfies. If an attacker "used" a different one each time it decrypted, we could not extract a single forgery from the Goldreich-Levin bits.

For this next step in our progression, we will have to develop a new technique. Intuitively, we will develop a new special type of signature scheme. In the real usage of our signature scheme the holder of the signature key can sign any message (thought of as a circuit) f. However, there is an alternative way to generate the public signature parameters that takes in an extra input a^*. In this alternative generation *there will not exist any valid signatures on a circuit f such that* $f(a^*) = 1$. Moreover, if no such signatures (signatures on f where $f(a^*) = 1$) are requested it is computationally hard to distinguish a normal set of parameters from an alternative set with input a^*.

With this special type of signature scheme, we can return to the approach of letting encryption for input a be a witness encryption of the NP statement that there is a signature on f where $f(a) = 1$. The proof of security will be a hybrid experiment where the first step is to change from a normal set of parameters to an alternative set for a^*. Our special signatures are realized from information theoretically sound Non-Interactive Witness Indistinguishable Proofs (NIWIs) and commitments with perfect soundness.

Fully Secure IBE. Finally, we extend the ideas for building ABE for circuits to get *adaptive* security for IBE (without complexity leveraging). Intuitively, we execute a "partitioning" strategy like [BF03, Wat05] where the reduction algorithm splits identities into two disjoint sets: those it can generate private keys for and those it can use as a valid challenge ciphertext. Again, we will use a version of a special signature scheme. In the real usage the holder can sign any message (thought of as an identity). The alternative parameter generation will take as input a PRF key s and a specified number of bits in the output range; the input of the PRF F' is a message. For this alternative parameter generation one can sign M if and only if $F'_K(M) \neq 0$.

In the reduction we do a hybrid proof where the (hidden) range of the PRF is set to be approximately the number of queries, q, made by the attacker. This means that approximately $1/q$ fraction of the identities will be useful as a challenge ciphertext and the other $1 - 1/q$ the reduction can make a private key for. Since are partitioning is tighter to the $1/q$ probability than [Wat05] we avoid the artificial abort issue that proof faced.

1.2 Building Witness Encryption Schemes

Conceptually, in our particular witness encryption scheme, a ciphertext consists of components – "puzzle pieces", if you will – that the decrypter puts together to compute the message-masking key. Our goal is to find the simplest manner to assemble such a "puzzle" for general NP relations using the framework of multilinear maps[5]. To this end, we identify the Exact Cover problem, one of Karp's original NP-complete problems (under Karp/Levin reductions). An instance of Exact Cover consists of a number n and a collection of subsets $T_1, \ldots, T_\ell \subset [n]$. A witness is a set $I \subseteq [\ell]$ such that $\{T_i : i \in I\}$ is a partition of $[n]$. When I is a witness, the puzzle pieces $\{T_i : i \in I\}$ fit together exactly to solve the puzzle $[n]$. There may be many witnesses, and therefore many sets of pieces that lead to a solution.

[5] Indeed, we also have a direct construction (omitted here) for the standard NP-complete problem of Satisfiability; however that construction is significantly more complex than the construction we provide here.

This "puzzle pieces" approach is reminiscent of what one sees in many previous schemes that use (cryptographic) *bilinear maps*, such as attribute-based encryption schemes [SW05]. However, bilinearity allows the decrypter to not only add, but also subtract, components. For this reason, as far as we know, attribute-based encryption schemes using bilinear maps can only enforce policies that are approximately equal in power to linear span programs. To prevent the "subtraction of puzzle pieces", we apparently need a stronger cryptographic tool. Accordingly, we use (cryptographic) *multilinear maps* [BS03], which allow the "multiplication of puzzle pieces", but not division of them, which will work just as well.

Specifically, suppose we have a n-multilinear group family consisting of a sequence of groups $\mathbb{G}_1, \ldots, \mathbb{G}_n$ of the same order p, together with generators $g_1, \ldots, g_n$ and a set of multilinear maps $e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j}$ for $i + j \leq n$ that satisfy $e_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab}$. For convenience, we collapse the multilinear maps into a single polymorphic function $e : \mathbb{G}_{i_1} \times \cdots \times \mathbb{G}_{i_t} \to \mathbb{G}_{i_1 + \cdots + i_t}$ for $i_1 + \cdots + i_t \leq n$ given by $e(g_{i_1}^{a_1}, \ldots, g_{i_t}^{a_t}) = (g_{i_1+\cdots+i_t})^{a_1 \cdots a_t}$. The multilinear map allows multiplication (in the exponent) up to degree n. There is no mechanism for division.

Now, given an Exact Cover instance $(n, T_1, \ldots, T_\ell)$ and a n-multilinear group family, our witness encryption scheme is quite simple. To encrypt $M \in \mathbb{G}_n$ (we assume that the message can be encoded as a group element), generate random scalars $a_1, \ldots, a_n \in \mathbb{Z}_p$, and send a ciphertext that consists of $C = M \cdot g_n^{a_1 \cdots a_n}$, and the "puzzle pieces" $C_i = (g_{|T_i|})^{\prod_{j \in T_i} a_j}$ for all $i \in [\ell]$. If the decrypter knows a witness $I = \{i_1, \ldots, i_t\} \subseteq [\ell]$ such that $\{T_i : i \in I\}$ is a partition of $[n]$, it can decrypt in the obvious way using the multilinear map. In particular, it computes $g_n^{a_1 \cdots a_n} = e(C_{i_1}, \ldots, C_{i_t})$, and divides this value from C to recover M.

Intuitively, the construction is secure since the only way to make $g_n^{a_1 \cdots a_n}$ is to find an exact cover of $[n]$. Formally, we base security on the assumed hardness of the "Decision Multilinear No-Exact-Cover Problem" – roughly, given an instance x of Exact Cover that has no solution, it is hard to distinguish the distribution $(C_1, \ldots, C_\ell, g_n^{a_1 \cdots a_n})$ from the distribution $(C_1, \ldots, C_\ell, g_n^r)$ where r is random and independent.

Unfortunately, this security assumption is intimately tied to the encrypter's particular NP relation instance. It would certainly be more satisfying to base security on a fixed, natural assumption that works for all instances. However, we prove that it is *impossible* to base the security of a restricted class of witness encryption scheme via an efficient black box reduction on a *simple* assumption – i.e., on a non-interactive hardness assumption that is *independent* of the hardness of deciding the specific NP problem instance being encrypted. The underlying assumption must either change with the NP relation instance, or the complexity of breaking the assumption must be greater than the complexity of deciding the relation. We leave open the problem of circumventing this impossibility result and constructing a witness encryption scheme based on a simple assumption by using *complexity leveraging*. We also show for the sake of completeness that it is impossible to construct a statistically-sound witness encryption scheme for NP unless the polynomial hierarchy collapses (such a result follows from known results in the statistical zero knowledge literature [IOS97, Rud89, Bei11]).

Of course, constructing the "pure" cryptographic multilinear map envisioned by Boneh and Silverberg [BS03] remains a long-standing open problem. However, Garg, Gentry and Halevi [GGH12] recently used ideal lattices to construct "approximate" or "noisy" cryptographic multilinear maps, which they call graded encoding systems. We show that their graded encoded systems suffice to construct our witness encryption scheme.

1.3 Other Related Work

As mentioned above, witness encryption, both as a notion and in terms of the applications that we envision, is interesting only as a *non-interactive* primitive. In the interactive setting, general secure two-party computation protocols [Yao86, GMW87] suffice for an interactive analog of witness encryption, where the decrypter essentially "commits" to his witness before the encrypter sends a message [AIR01]. From the completely different perspective of statistical zero-knowledge protocols, a concept similar in spirit to witness encryption has been studied under the heading of *instance-dependent commitments* (ID commitments, for short), starting as early as [TW87]. In an ID commitment, a party commits to a value m with respect to an instance x. Depending on whether $x \in L$ or not, the commitment is required to be statistically binding or statistically hiding. Both interactive [OV08, CCKV08] and non-interactive [TW87, IOS97, IS91, KMV07, CCKV08, GOVW12] variants of ID commitment schemes have been studied in the literature, with recent work also considering the notion of efficient extractability [GOVW12]. However, non-interactive primitives have been considered only in a setting where statistical hiding is desired. These works can be seen as establishing the existence of statistical witness encryption schemes for a few specific languages known to be in SZK (more specifically, languages that possess certain kinds of hash proof systems [CS02]). Indeed, it is impossible to construct such a statistical primitive for NP complete languages unless the polynomial hierarchy collapses [IOS97](for self-containment, we also provide a proof of this here).It is intriguing that these works in the statistical zero knowledge literature considered a notion related to ours, despite coming from a very different perspective. However, we stress that no prior work considered the notion of witness encryption for general NP languages, which is the focus of this work. Furthermore, no candidate constructions for NP-complete languages were contemplated prior to our work, explicitly or implicitly.

2 Preliminaries

In this section, we provide background on cryptographic multilinear maps [BS03] and Garg et al.'s lattice-based "approximate" multilinear maps (a.k.a. "graded encoding systems") [GGH12].

2.1 Cryptographic Multilinear Maps: Dream Version

It remains a long-standing open problem to construct a multilinear group family (defined below) over which natural problems, such as discrete log and higher-degree versions of Diffie-Hellman, are intractable. Garg, Gentry and Halevi (GGH) [GGH12] recently constructed "approximate" multilinear maps from ideal lattices, which are limited to polynomial degree, after which the "noisiness" of their encodings overwhelms the signal, somewhat like ciphertexts in somewhat homomorphic encryption schemes. Their approximate multilinear maps suffice to construct witness encryption, but, for clarity, we also describe our constructions using clean "exact" multilinear maps, described here.

Let $\mathsf{params} \leftarrow \mathcal{G}(1^\lambda, n)$ be a description of a *multilinear group family* $\mathbb{G}_1, \ldots, \mathbb{G}_n$, each of prime order $p = p(\lambda)$ for security parameter λ, with canonical generators $g_1, g_2, \ldots, g_n$, and multilinear map e. The multilinear map e is actually a set of bilinear maps $\{e_{i,j} : G_i \times G_j \to G_{i+j} \mid i, j \geq 1;\ i + j \leq n\}$, where $e_{i,j}$ satisfies the following relation:

$$e_{i,j}\left(g_i^a, g_j^b\right) = g_{i+j}^{ab} \ : \ \forall a, b \in \mathbb{Z}_p.$$

We observe that one consequence of this is that $e_{i,j}(g_i, g_j) = g_{i+j}$ for each valid i, j.

When the context is obvious, we will sometimes abuse notation drop the subscripts i, j. For example, we may simply write:

$$e\left(g_i^a, g_j^b\right) = g_{i+j}^{ab}.$$

For $i_1, \ldots, i_t \in [n]$ with $i_1 + \ldots + i_t \leq n$, we write:

$$e\left(g_{i_1}^{a_{i_1}}, \ldots, g_{i_t}^{b_{i_t}}\right) = (g_{i_1+\cdots+i_t})^{a_{i_1}\cdots a_{i_t}},$$

where the left side is computed iteratively by setting $B_1 = g_{i_1}^{a_{i_1}}$, then for $j = 2, \ldots, t$ setting $B_j = e(B_{j-1}, g_{i_j}^{a_{i_j}})$, and finally outputting B_t.

Our witness encryption scheme will rely on the intractability of the following Decision Multilinear No-Exact-Cover Problem. (Recall that the Exact Cover problem is one of Karp's original NP-complete problems. The instance is a number n and a collection of subsets $T_1, \ldots, T_\ell \subset [n]$. A witness is a set $I \subseteq [\ell]$ such that $\{T_i : i \in I\}$ is a partition of $[n]$.)

Definition 2.1 (Decision Multilinear No-Exact-Cover Problem)**.** *Let $x = \{T_i : T_i \subset [n], i \in [\ell]\}$ be an instance of* Exact Cover *that has no solution. Let $\mathsf{params} \leftarrow \mathcal{G}(1^\lambda, n)$ be a description of a multilinear group family with order prime $p = p(\lambda)$. Let $a_1, \ldots, a_n, r$ be uniformly random in $\mathbb{Z}_p$. For $i \in [\ell]$, let $h_i = (g_{|T_i|})^{\prod_{j \in T_i} a_j}$. Distinguish between the two distributions:*

$$(\mathsf{params}, h_1, \ldots, h_\ell, g_n^{a_1\cdots a_n}) \quad \text{and} \quad (\mathsf{params}, h_1, \ldots, h_\ell, g_n^r).$$

(The search version is: given $\mathsf{params}, h_1, \ldots, h_\ell$, output $g_n^{a_1\cdots a_n}$. However, we will not need it here.)

Definition 2.2 (Decision Multilinear No-Exact-Cover Assumption)**.** *The Decision Multilinear No-Exact-Cover Assumption is that for all instances x with no solution, all PPT algorithms $\mathcal{A}$ have negligible advantage against the Decision Multilinear No-Exact-Cover Problem.*

2.2 Graded Encoding Systems: Definition

Garg, Gentry and Halevi (GGH) [GGH12] defined an "approximate" version of a multilinear group family, which they call a *graded encoding system.* As a starting point, they view g_i^α in a multilinear group family as simply an *encoding* of α at "level-i". This encoding permits basic functionalities, such as equality testing (it is easy to check that two level-i encodings encode the same exponent), additive homomorphism (via the group operation in $\mathbb{G}_i$), and bounded multiplicative homomorphism (via the multilinear map e). They retain the notion of a somewhat homomorphic encoding with equality testing, but they use probabilistic encodings, and replace the multilinear group family with "less structured" sets of encodings related to lattices.

Abstractly, their n-graded encoding system for a ring R includes a system of sets $\mathcal{S} = \{S_i^{(\alpha)} \subset \{0,1\}^* : i \in [0, n], \alpha \in R\}$ such that, for every fixed $i \in [0, n]$, the sets $\{S_i^{(\alpha)} : \alpha \in R\}$ are disjoint (and thus form a partition of $S_i \overset{\text{def}}{=} \bigcup_\alpha S_i^{(\alpha)}$). The set $S_i^{(\alpha)}$ consists of the "level-i encodings of α". Moreover, the system comes equipped with efficient procedures, as follows:[6]

[6] Since GGH's realization of a graded encoding system uses "noisy" encodings over ideal lattices, the procedures incorporate information about the magnitude of the noise.

Instance Generation. The randomized $\mathsf{InstGen}(1^\lambda, 1^n)$ takes as input the security parameter λ and integer n. The procedure outputs $(\mathsf{params}, \mathbf{p}_{zt})$, where params is a description of an n-graded encoding system as above, and $\mathbf{p}_{zt}$ is a level-n "zero-test parameter".

Ring Sampler. The randomized $\mathsf{samp}(\mathsf{params})$ outputs a "level-zero encoding" $a \in S_0$, such that the induced distribution on α such that $a \in S_0^{(\alpha)}$ is statistically uniform.

Encoding. The (possibly randomized) $\mathsf{enc}(\mathsf{params}, i, a)$ takes $i \in [n]$ and a level-zero encoding $a \in S_0^{(\alpha)}$ for some $\alpha \in R$, and outputs a level-i encoding $u \in S_i^{(\alpha)}$ for the same α.

Re-Randomization. The randomized $\mathsf{reRand}(\mathsf{params}, i, u)$ re-randomizes encodings to the same level, as long as the initial encoding is under a given noise bound. Specifically, for a level $i \in [n]$ and encoding $u \in S_i^{(\alpha)}$, it outputs another encoding $u' \in S_i^{(\alpha)}$. Moreover for any two encodings $u_1, u_2 \in S_i^{(\alpha)}$ whose noise bound is at most some b, the output distributions of $\mathsf{reRand}(\mathsf{params}, i, u_1)$ and $\mathsf{reRand}(\mathsf{params}, i, u_2)$ are statistically the same.

Addition and negation. Given params and two encodings at the same level, $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, we have $\mathsf{add}(\mathsf{params}, u_1, u_2) \in S_i^{(\alpha_1+\alpha_2)}$, and $\mathsf{neg}(\mathsf{params}, u_1) \in S_i^{(-\alpha_1)}$, subject to bounds on the noise.

Multiplication. For $u_1 \in S_{i_1}^{(\alpha_1)}$, $u_2 \in S_{i_2}^{(\alpha_2)}$, we have $\mathsf{mult}(\mathsf{params}, u_1, u_2) \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$.

Zero-test. The procedure $\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, u)$ outputs 1 if $u \in S_n^{(0)}$ and 0 otherwise. Note that in conjunction with the procedure for subtracting encodings, this gives us an equality test.

Extraction. This procedure extracts a "canonical" and "random" representation of ring elements from their level-n encoding. Namely $\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u)$ outputs (say) $K \in \{0,1\}^\lambda$, such that:

(a) With overwhelming probability over the choice of $\alpha \in R$, for any two $u_1, u_2 \in S_n^{(\alpha)}$, $\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u_1) = \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u_2)$,

(b) The distribution $\{\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u) : \alpha \in R, u \in S_n^{(\alpha)}\}$ is statistically uniform over $\{0,1\}^\lambda$.

Remark 1. *We can extend* add *and* mult *to handle more than two encodings as inputs, by applying the binary versions of* add *and* mult *iteratively. Also, it is convenient to define a canonicalized encoding algorithm* $\mathsf{enc}^\dagger(\mathsf{params}, i, a) = \mathsf{reRand}(\mathsf{params}, i, \mathsf{enc}(\mathsf{params}, i, a))$ *which outputs encodings according to a "nice" distribution.*

It is straightforward to adapt the Decision Multilinear No-Exact-Cover Problem to the setting of graded encodings.

Definition 2.3 (Decision Graded Encoding No-Exact-Cover Problem)**.** *Let $x = \{T_i : T_i \subset [n], i \in [\ell]\}$ be an instance of* Exact Cover *that has no solution. Let* $(\mathsf{params}, \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^n)$ *be a description of a n-graded encoding system with $|R|$ prime, and a level-n zero-test parameter. Generate $a_1, \ldots, a_n, r$ via* $\mathsf{samp}(\mathsf{params})$*. For $i \in [\ell]$, let* $h_i \leftarrow \mathsf{enc}^\dagger(\mathsf{params}, |T_i|, \prod_{j \in T_i} a_j)$*. Distinguish between the two distributions:*

$$(h_1, \ldots, h_\ell, \mathsf{enc}^\dagger(\mathsf{params}, n, a_1 \cdots a_n)) \quad \text{and}(h_1, \ldots, h_\ell, \mathsf{enc}^\dagger(\mathsf{params}, n, r)).$$

Definition 2.4 (Decision Graded Encoding No-Exact-Cover Assumption)**.** *The Decision Graded Encoding No-Exact-Cover Assumption is that for all instances x with no solution, all PPT algorithms $\mathcal{A}$ have negligible advantage against the Decision Graded Encoding No-Exact-Cover Problem.*

2.3 Graded Encoding Systems: Realization

Concretely, GGH's n-graded encoding system works as follows. (This is a whirlwind overview; see [GGH12] for details.) The system uses three rings. First, it uses the ring of integers $\mathcal{O}$ of the m-th cyclotomic field. This ring is typically represented as the ring of polynomials $\mathcal{O} = \mathbb{Z}[x]/(\Phi_m(x))$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial, which has degree $N = \phi(m)$. Second, for some suitable integer modulus q, it uses the quotient ring $\mathcal{O}/(q) = \mathbb{Z}_q[x]/(\Phi_m(x))$, similar to the NTRU encryption scheme [HPS98]. The encodings live in $\mathcal{O}/(q)$. Finally, it uses the quotient ring $R = \mathcal{O}/\mathcal{I}$, where $\mathcal{I} = \langle g \rangle$ is a principal ideal of $\mathcal{O}$ that is generated by g and where $|\mathcal{O}/\mathcal{I}|$ is a large prime. This is the ring "R" referred to above; elements of R are what is encoded.

What does a GGH encoding look like? For a fixed random $z \in \mathcal{O}/(q)$, an element of $S_i^{(\alpha)}$ – that is, a level-i encoding of $\alpha \in R$ – has the form $e/z^i \in \mathcal{O}/(q)$, where $e \in \mathcal{O}$ is a "small" representative of the coset $\alpha + \mathcal{I}$ (it has coefficients that are very small compared to q). To add encodings $e_1/z^i \in S_i^{(\alpha_1)}$ and $e_2/z^i \in S_i^{(\alpha_2)}$, just add them in $\mathcal{O}/(q)$ to obtain $(e_1 + e_2)/z^i$, which is in $S_i^{(\alpha_1+\alpha_2)}$ if $e_1 + e_2$ is "small". To mult encodings $e_1/z^{i_1} \in S_{i_1}^{(\alpha_1)}$ and $e_2/z^{i_2} \in S_{i_2}^{(\alpha_2)}$, just multiply them in $\mathcal{O}/(q)$ to obtain $e_1 \cdot e_2/z^{i_1+i_2}$, which is in $S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$ if $e_1 \cdot e_2$ is "small". This smallness condition limits the GGH encoding system to degree polynomial in the security parameter. Intuitively, dividing encodings does not "work", since the resulting denominator has a nontrivial term that is not z.

The GGH params allow everyone to generate encodings of random (known) values. The params include a level-1 encoding of 1 (from which one can generate encodings of 1 at other levels), and (for each $i \in [n]$) a sufficient number of level-i encodings of 0 to enable re-randomization. To encode (say at level-1), run $\mathsf{samp}(\mathsf{params})$ to sample a small element a from $\mathcal{O}$, e.g. according to a discrete Gaussian distribution. For a Gaussian with appropriate deviation, this will induce a statistically uniform distribution over the cosets of $\mathcal{I}$. Then, multiply a with the level-1 encoding of 1 to get a level-1 encoding u of $a \in R$. Finally, run $\mathsf{reRand}(\mathsf{params}, 1, u)$, which involves adding a random Gaussian linear combination of the level-1 encodings of 0, whose noisiness (i.e., numerator size) "drowns out" the initial encoding.

To permit testing of whether a level-n encoding $u = e/z^n \in S_n$ encodes 0, GGH publishes a level-n zero-test parameter $\mathbf{p}_{zt} = hz^n/g$, where h is "somewhat small"[7] and g is the generator of $\mathcal{I}$. The procedure $\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, u)$ simply computes $\mathbf{p}_{zt} \cdot u$ and tests whether its coefficients are small modulo q. If u encodes 0, then $e \in \mathcal{I}$ and equals $g \cdot c$ for some (small) c, and thus $\mathbf{p}_{zt} \cdot u = h \cdot c$ has no denominator and is small modulo q. If u encodes something nonzero, $\mathbf{p}_{zt} \cdot u$ has g in the denominator and is not small modulo q. The $\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, u)$ procedure works by applying a strong extractor to the most significant bits of $\mathbf{p}_{zt} \cdot u$. For any two $u_1, u_2 \in S_n^{(\alpha)}$, we have (subject to noise issues) $u_1 - u_2 \in S_n^{(0)}$, which implies $\mathbf{p}_{zt}(u_1 - u_2)$ is small, and hence $\mathbf{p}_{zt} \cdot u_1$ and $\mathbf{p}_{zt} \cdot u_2$ have the same most significant bits (for an overwhelming fraction of α's).

Garg et al. provide an extensive cryptanalysis of the encoding system, which we will not review here. We remark that the underlying assumptions are stronger, but related to, the hardness

[7] Its coefficients are on the order of (say) $q^{2/3}$, while other terms – such as a numerator e or the principal ideal generator g – are much, much smaller.

assumption underlying the NTRU encryption scheme: that it is hard to distinguish a uniformly random element from $\mathcal{O}/(q)$ from a ratio of "small" elements – i.e., an element $u/v \in \mathcal{O}/(q)$ where $u, v \in \mathcal{O}/(q)$ both have coefficients that are on the order of (say) q^ϵ for small constant ϵ.

3 Witness Encryption

Definition 3.1. *A* witness encryption *scheme for an* **NP** *language L (with corresponding witness relation R) consists of the following two polynomial-time algorithms:*

> **Encryption.** *The algorithm Encrypt$(1^\lambda, x, M)$ takes as input a security parameter 1^λ, an unbounded-length string x, and a message $M \in \mathcal{M}$ for some message space $\mathcal{M}$, and outputs a ciphertext* CT.
>
> **Decryption.** *The algorithm Decrypt(CT, w) takes as input a ciphertext* CT *and an unbounded-length string w, and outputs a message M or the symbol $\perp$.*

These algorithms satisfy the following two conditions:

- ***Correctness.*** *For any security parameter λ, for any $M \in \mathcal{M}$, and for any $x \in L$ such that $R(x, w)$ holds, we have that*

$$\Pr\Big[\mathit{Decrypt}\big(\mathit{Encrypt}(1^\lambda, x, M), w\big) = M\Big] = 1$$

- ***Soundness Security.*** *For any $x \notin L$, for any PPT adversary A and messages $M_0, M_1 \in \mathcal{M}$, there exists a negligible function neg$(\cdot)$, such that:*

$$\Big|\Pr\left[A(\mathit{Encrypt}(1^\lambda, x, M_0)) = 1\right] - \Pr\left[A(\mathit{Encrypt}(1^\lambda, x, M_1)) = 1\right]\Big| < \mathit{neg}(\lambda)$$

Remark 2. *We stress that witness encryption does not require any setup algorithm.*

The Security-Correctness Gap. We remark that the correctness stipulates that an algorithm can decrypt if $x \in L$ it knows a witness w for the relation R. Security states that if $x \notin L$ then no polynomial-time algorithm can decrypt. However, our definition is (intentionally) silent on the case when $x \in L$, but the algorithm does not know a witness for the relation R.

4 Cryptographic Primitives from Witness Encryption

We turn to building cryptographic primitives from witness encryption. We show a progression of primitives starting with the basic case of Public-Key Encryption, then moving on to Identity-Based Encryption, and finally showing how to realize Attribute-Based Encryption for circuits. Each new step in the progression will be more challenging and require new techniques. We also give a construction of a fully secure IBE scheme.

We now formally describe our encryption systems. For building them we will assume the existence of a witness encryption scheme for an NP-Complete language L for which there exists a Karp-Levin reduction. We focus on presenting the constructions in this section and defer the proofs to Appendix A.

4.1 Public Key Encryption

We now describe our public key encryption system in terms of three algorithms.

***Setup*$_{\mathbf{PKE}}$**(1^λ)
The setup algorithm chooses a random PRG seed $s \in \{0,1\}^\lambda$. Next, it uses the PRG $G : \{0,1\}^\lambda \rightarrow \{0,1\}^{2 \cdot \lambda}$ to compute $G(s) \rightarrow t \in \{0,1\}^{2\lambda}$. The public key $\mathrm{PK} = (t, \lambda)$ is the output of the PRG and the security parameter. The secret key $\mathrm{SK} = s$ is the seed.

***Encrypt*$_{\mathbf{PKE}}$**$(\mathrm{PK} = (t, \lambda), M)$
To encrypt the algorithm prepares an instance x such that $x \in L$ if and only if t is in the range of G. It uses the Karp-Levin reduction to the NP-complete language L to do this. Next, it computes $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x, M) \rightarrow C$ to encrypt the message M for the instance x. The output ciphertext is $\mathrm{CT} = (x, C)$.

***Decrypt*$_{\mathbf{PKE}}$**$(\mathrm{SK} = s, \mathrm{CT} = (x, C))$
The decryption algorithm is given an instance x and witness encryption ciphertext C. If the ciphertext was formed properly, the algorithm can use its knowledge of s to obtain a witness w that $x \in L$. Next, it calls $\mathit{Decrypt}_{\mathrm{WE}}(C, w)$ to recover the encrypted message M.

4.2 Identity-Based Encryption

We now describe the four algorithms comprising of our IBE system. We assume the existence of a unique signature system, where on honest setup there will be exactly one signature that will verify. We also use $\mathrm{GL}(\sigma, r)$ to denote the Goldreich-Levin [GL89] hardcore bit of σ using randomness r. Recall, that the GL predicate is the bitwise inner product between σ and r.

***Setup*$_{\mathbf{IBE}}$**(1^λ)
The IBE setup algorithm runs *Setup-Signature*(1^λ) for the *unique* signature system. It sets the public parameters PP to be the signature verification key and the master secret key MSK to be the signature signing key.

***KeyGen*$_{\mathbf{IBE}}$**$(\mathrm{MSK}, \mathcal{I})$
The key generation algorithm simply computes a signature on the identity by calling $\mathit{Sign}(\mathrm{MSK}, \mathcal{I}) \rightarrow$ SK.

***Encrypt*$_{\mathbf{IBE}}$**$(\mathrm{PP}, \mathcal{I}, M)$
The encryption algorithm will actually prepare two witness encryption ciphertexts. Suppose that signatures in our underlying signature scheme are of length k. The algorithm chooses a random $r \in \{0,1\}^k$.

It prepares an instance x_0 such that $x_0 \in L$ if and only if there exists a signature σ where $\mathrm{GL}(\sigma, r) = 0$ and where $\mathit{Verify}(\mathrm{PP}, \sigma, \mathcal{I}) = \text{true}$. It computes $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x_0, M) \rightarrow C_0$. Next, it creates a ciphertext for the opposite condition. It prepares x_1 where $x_1 \in L$ if and only if there exists a signature σ where $\mathrm{GL}(\sigma, r) = 1$ and σ verifies on $\mathcal{I}$. It computes $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x_1, M) \rightarrow C_1$. The output ciphertext is $\mathrm{CT} = (\mathcal{I}, x_0, x_1, r, C_0, C_1)$.

Since we are using a unique signature scheme there will exist only one signature σ where *Verify*$(\mathrm{PP}, \sigma, M) = \text{true}$. Thus, either $x_0 \in L$ or $x_1 \in L$, but not both. For non-unique signature schemes, there might be multiple signatures that could verify and the above condition would not necessarily hold.

***Decrypt*$_{\mathbf{IBE}}$**$(\mathrm{SK} = \sigma, \mathrm{CT} = (\mathcal{I}, x_0, x_1, r, C_0, C_1))$
The decryption algorithm first computes the bit $b = \mathrm{GL}(\sigma, r)$. Then it uses its knowledge of σ to obtain a witness w that $x_b \in L$. Finally, it calls *Decrypt*$_{\mathrm{WE}}(C_b, w)$ to recover the encrypted message M.

4.3 Attribute-Based Encryption for Circuits

We now describe a construction of (Key-Policy) Attribute-Based Encryption for circuits. In this setting, a private key corresponds to a boolean circuit f that takes n bit inputs. A ciphertext corresponds to an n bit value a. If $f(a) = 1$ then the user should be able to decrypt.

Let Com be a *perfectly binding* non-interactive commitment scheme[8] and let (P, V) be a non-interactive zap (defined in Section B).

***Setup*$_{\mathbf{ABE}}$**(1^λ)
The ABE setup algorithm generates commitments $c_1 = \mathsf{Com}(0; r)$ and $c_2 = \mathsf{Com}(0^n; s)$. It sets the public parameters PP to be (c_1, c_2) and the master secret key MSK to be r.

***KeyGen*$_{\mathbf{ABE}}$**(MSK, f)
The key generation algorithm simply outputs (f, π_f) where $\pi_f = P(x_f, r)$. Here, r is the randomness used in generation of c_1 and x_f is the following NP-statement:

$$\exists (w_1, a, w_2) \text{ such that } c_1 = \mathsf{Com}(0; w_1) \vee (c_2 = \mathsf{Com}(a; w_2) \wedge f(a) = 0) \tag{1}$$

Note that f is the circuit for which the secret key is being issued. Also note that the proof π_f will have size specified by fixed polynomial in $\lambda, n, |f|$ denoted by $\mu(\lambda, n, |f|)$.

***Encrypt*$_{\mathbf{ABE}}$**(PP, a, M)
The encryption algorithm prepares an instance x' such that $x' \in L$ if and only if there exists a circuit g such that $|g| \leq \ell_{max}$ and a proof π_g (of size $\mu(\lambda, n, |g|)$) such that $V(x_g, \pi_g) = 1 \wedge g(a) = 1$ where x_g is the NP-statement as defined in Equation 1.

Next, it computes *Encrypt*$_{\mathrm{WE}}(1^\lambda, x', M) \to C$. The output ciphertext is $\mathrm{CT} = (a, x', C, \ell_{max})$.

***Decrypt*$_{\mathbf{ABE}}$**$((f, \pi_f), \mathrm{CT} = (a, x', C, \ell_{max}))$
If $f(a) = 1$ and $|f| \leq \ell_{max}$, the decryption algorithm uses π_f to obtain a witness w to the fact that $x' \in L$. Finally, it calls *Decrypt*$_{\mathrm{WE}}(C, w)$ to recover the encrypted message M.

Remark 3. *Note that in our scheme each secret key corresponds to a circuit. These circuits could be as large as desired (still polynomial sized though). However at the time of encryption the encrypter specifies an upper bound (ℓ_{max} above) on the size of the circuits corresponding to which*

[8] Such a commitment scheme can be constructed using any one-to-one one way function.

the secret keys can be used. This parameter could be a global system parameter fixed once and for all or could be a parameter that the encrypter fixes on a per encryption basis.

4.4 Fully Secure Identity-Based Encryption

Now, we describe a fully secure IBE scheme for identities of length n. Let Com be a *perfectly binding* non-interactive commitment scheme and let (P, V) be a non-interactive zap (defined in Section B). Let F be a PRF family from n bits to λ bits with seed length λ. Further let $F_s(y)$ denote the PRF output on input y with s. For any $t \in [\lambda]$, let $F'_{s,t}(y)$ denote the t least significant bits from $F_s(y)$.

$\boldsymbol{Setup}_{\mathbf{IBE}}(1^\lambda)$
The IBE setup algorithm generates commitments $c_1 = \mathsf{Com}(0; r)$, $c_2 = \mathsf{Com}(0^\lambda; R)$ and $c_3 = \mathsf{Com}(0^{\log(\lambda)}; R')$. It sets the public parameters PP to be (c_1, c_2, c_3) and the master secret key MSK to be r.

$\boldsymbol{KeyGen}_{\mathbf{IBE}}(\mathrm{MSK}, \mathcal{I})$
The key generation algorithm simply outputs $(\mathcal{I}, \pi_\mathcal{I})$ where $\pi_\mathcal{I} = P(x_\mathcal{I}, r)$. Here, r is the randomness used in generation of c_1 and $x_\mathcal{I}$ is the following NP-statement:

$$\exists(w_1, s, t, w_2, w_3) \text{ such that } c_1 = \mathsf{Com}(0; w_1) \vee (c_2 = \mathsf{Com}(s; w_2) \wedge c_3 = \mathsf{Com}(t; w_3) \wedge F_{s,t}(\mathcal{I}) \neq 0) \quad (2)$$

Note that $\mathcal{I}$ is the identity for which the secret key is being issued. Also note that the size of the proof $\pi_\mathcal{I}$ will be some fixed polynomial in $\lambda, |\mathcal{I}|$.

$\boldsymbol{Encrypt}_{\mathbf{IBE}}(\mathrm{PP}, \mathcal{I}, M)$
It prepares an instance x' such that $x' \in L$ if and only if there exists a proof $\pi_\mathcal{I}$ (of appropriate size) such that $V(x_\mathcal{I}, \pi_\mathcal{I}) = 1$ where $x_\mathcal{I}$ is the NP-statement as defined in Equation 2.

Next, it computes $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x', M) \to C$. The output ciphertext is $\mathrm{CT} = (\mathcal{I}, x', C)$.

$\boldsymbol{Decrypt}_{\mathbf{IBE}}((\mathcal{I}, \pi_\mathcal{I}), \mathrm{CT} = (\mathcal{I}', x', C))$
If $\mathcal{I} = \mathcal{I}'$ then it uses its knowledge of $\pi_\mathcal{I}$ to obtain a witness w to the fact that $x' \in L$. Finally, it calls $\mathit{Decrypt}_{\mathrm{WE}}(C, w)$ to recover the encrypted message M.

Security Proof. The key idea in proving the security of the above scheme is to be able to execute a partitioning strategy for the secret keys of the adaptive IBE. This is very similar to [Wat05] except that we do not have to deal with the issue of artificial aborts. Next we will give the complete proof.

5 Our Construction

Our witness encryption scheme is surprisingly simple. We use the EXACT COVER problem, one of Karp's original NP-complete problems. The instance is a number n and a collection of subsets $T_1, \ldots, T_\ell \subset [n]$. A witness is a set $I \subseteq [\ell]$ such that $\{T_i : i \in I\}$ is a partition of $[n]$. For clarity, we first present our construction using the "dream-version" of multilinear maps. Next, we present an instantiation of the scheme using the GGH graded encoding system.

5.1 Witness Encryption Using a Multilinear Group Family

Encrypt$(1^\lambda, x, M)$
The algorithm takes as input an Exact Cover instance x and a message M. It generates $\mathsf{params} \leftarrow \mathcal{G}(1^\lambda, n)$, which include a multilinear group family $\mathbb{G}_1, \ldots, \mathbb{G}_n$ of prime order $p = p(\lambda)$ with canonical generators $g_1, g_2, \ldots, g_n$ and multilinear map e. It chooses random $a_1, \ldots, a_n \in \mathbb{Z}_p$. For $M \in \mathbb{G}_n$ (we assume that the message can be encoded as a group element), the ciphertext CT consists of:

$$C = M \cdot g_n^{a_1 \cdots a_n} \text{ and } \forall i \in [\ell] \; C_i = (g_{|T_i|})^{\prod_{j \in T_i} a_j}$$

as well as params and a description of the exact cover instance x.

Decrypt$(\mathrm{CT}, w = I)$
The algorithm takes as input a ciphertext and a witness set $I = \{j_1, j_2, \ldots, j_{|I|}\} \subseteq [\ell]$ associated to a partition of $[n]$. The algorithm outputs

$$M = C / e(C_{j_1}, \ldots, C_{j_{|I|}}).$$

Correctness. Since I is associated to a partition of $[n]$, $e(C_{j_1}, \ldots, C_{j_{|I|}})$ is precisely $g_n^{a_1 \cdots a_n}$.

Security. Intuitively, the construction is secure since the only way to make $g_n^{a_1 \cdots a_n}$ is to find an exact cover of $[n]$. Formally, we base security on the Decision Multilinear No-Exact-Cover Assumption.

Theorem 5.1. *The scheme above is a sound witness encryption scheme under the Decision Multilinear No-Exact-Cover Assumption.*

Proof. Immediate. □

5.2 Witness Encryption Using a Graded Encoding System

The GGH graded encoding system is probabilistic and does not offer a bijection between encodings and a message space. Therefore, we present our witness encryption scheme as a key encapsulation mechanism (KEM). In a KEM, one encrypts a random key rather than a message. Then, the random key is used to encrypt the message – e.g., using a symmetric encryption scheme. In a sound witness encryption KEM, for any $x \notin L$, a PPT adversary should not be able to distinguish between the actual KEM key and a random string of the same length.

Encrypt$(1^\lambda, x, M)$
The algorithm takes as input an Exact Cover instance x. The algorithm runs the $\mathsf{InstGen}$ algorithm to generate $(\mathsf{params}, \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^n)$, where params is a description of a n-graded encoding system and $\mathbf{p}_{zt}$ is a level-n zero-test parameter. The algorithm samples level-zero encodings $a_i \leftarrow \mathsf{samp}(\mathsf{params})$ for $i \in [n]$. The ciphertext CT consists of:

$$\forall i \in [\ell] \; C_i \leftarrow \mathsf{enc}^\dagger(\mathsf{params}, |T_i|, \prod_{j \in T_i} a_j)$$

as well as params, $\mathbf{p}_{zt}$, and a description of the exact cover instance x. The KEM key K is:

$$K \leftarrow \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, \mathsf{enc}(\mathsf{params}, n, a_1 \cdots a_n)).$$

Decrypt$(\mathrm{CT}, w = I)$
The algorithm takes as input a ciphertext and a witness set $I = \{j_1, j_2, \ldots, j_{|I|}\} \subseteq [\ell]$ associated to a partition of $[n]$. The algorithm sets

$$B \leftarrow \mathsf{mult}(\mathsf{params}, C_{j_1}, \ldots, C_{j_{|I|}}).$$

The algorithm uses the extraction routine to derive the KEM key, $K \leftarrow \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, B)$.

Correctness. Assuming an appropriate choice of parameters, we note that since I is associated to a partition of $[n]$, $\mathsf{mult}(\mathsf{params}, C_{j_1}, \ldots, C_{j_{|I|}})$ is a level-n encoding of $a_1 \cdots a_n$'s coset in R. Moreover, the extraction algorithm is designed so that, with overwhelming probability over the choice of $\alpha \in R$, for any two valid level-n encodings B_1, B_2 of α, $\mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, B_1) = \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, B_2)$.

Regarding parameters, GGH show that one can achieve correctness and 2^λ security against known attacks while using the ring of integers for the m-th cyclotomic field for $m = \tilde{O}(n\lambda^2)$. As long as there is no circular dependence between the witness encryption scheme itself (its underlying ring of integers, etc.) and the encrypter's NP relation, the encrypter can choose m after n and λ to satisfy this requirement. See [GGH12] for additional guidance about setting parameters.

Remark 4. *Note that the construction of Witness Encryption described above can be easily adapted to the realize the stronger notion of computational secret sharing for an NP-complete access structure, first posed by Rudich in 1989 [Rud89] (see [Bei11]). More specifically, in the setting of the 3-Exact Cover problem (it is NP-complete, see Proposition 2.25 in [Gol08]) consider we could identify each of the $\binom{n}{3}$ subsets T of $[n]$ such that $|T| = 3$ with a different party P_T. The secret sharing scheme would require a way to take a secret x and construct potential shares λ_T for each party P_T. The two guarantees needed would be: (1)* efficient recovery*: if a set of parties $P_{T_{i_1}}, \ldots, P_{T_{i_t}}$ knew of an exact cover among their sets, then these parties would be able to efficiently recover the secret x from their shares $\lambda_{T_{i_1}}, \ldots, \lambda_{T_{i_t}}$. Note that the monotonicity of recovery is maintained here – if a set of parties contains an exact cover, so must any other superset of these parties. (2)* privacy*: if a set of parties $P_{T_{i_1}}, \ldots, P_{T_{i_t}}$ does not contain an exact cover, then these parties should not be able to distinguish between secret sharing of distinct secrets x and x'.*

Our construction of Witness Encryption extends to yield a Rudich-type secret sharing scheme, as well, under the same computational assumption. This can be done by setting the secret share λ_T provided to party P_T as $\mathsf{enc}^\dagger(\mathsf{params}, |T|, \prod_{j \in T} a_j)$. Parties having access to the right shares will be able to reconstruct the shared secret using the decryption procedure of the scheme. On the other hand privacy can be argued just like the soundness of the witness encryption scheme.

5.3 Security for Witness Encryption Using a Graded Encoding System

Intuitively, the construction is secure since the only way to obtain K is to apply the extraction algorithm to a level-n encoding of $a_1 \cdots a_n$, and the only way to obtain the latter is to find an exact cover of $[n]$. Formally, we base security on the Decision Graded Encoding No-Exact-Cover Assumption.

Theorem 5.2. *Our graded-encoding-system witness encryption construction is a sound KEM under the Decision Graded Encoding No-Exact-Cover Assumption.*

Proof. Suppose that a PPT adversary $\mathcal{A}$ can distinguish K (as output by *Encrypt*) from a uniformly random string with non-negligible advantage when the EXACT COVER instance x has no witness.

Then, there is an algorithm $\mathcal{B}$, with complexity polynomially related to $\mathcal{A}$, that solves the Decision Graded Encoding No-Exact-Cover Problem with non-negligible advantage (a contradiction).

Specifically, given an instance $(h_1, \dots, h_\ell, B)$ of the Decision Graded Encoding No-Exact-Cover Problem where $B = \mathsf{enc}^\dagger(\mathsf{params}, n, s)$ where s is either $a_1 \cdots a_n$ or r (random), $\mathcal{B}$ constructs a ciphertext CT that includes $C_i = h_i$ for $i \in [\ell]$, params, $\mathbf{p}_{zt}$, and a description of the exact cover instance x. It sets $K^* \leftarrow \mathsf{ext}(\mathsf{params}, \mathbf{p}_{zt}, B)$. It sends (CT, K^*) to $\mathcal{A}$. Note that if $B = \mathsf{enc}^\dagger(\mathsf{params}, n, a_1 \cdots a_n)$, then K^* has (statistically) the same distribution as a proper key, whereas if $B = \mathsf{enc}^\dagger(\mathsf{params}, n, r)$, then K^* is (statistically) random and independent by the randomness property of the sampling procedure and the properties of ext. Therefore, by assumption, $\mathcal{A}$ can distinguish whether K is well-formed or random with non-negligible advantage, and $\mathcal{B}$ can use $\mathcal{A}$'s output to solve the Decision Graded Encoding No-Exact-Cover Problem with non-negligible advantage. □

6 Impossibilities

In this section, we argue that it is unlikely that the hardness assumptions underlying the security of witness encryption schemes can be simplified significantly. In particular, we give two impossibility results.

First we show that existence of a statistically secure variant of a witness encryption scheme (Section 3) implies the collapse of the polynomial hierarchy. This result appeared in [IOS97, Rud89, Bei11] and we present it here for the sake of completeness. Next we turn back to the computational variant of witness encryption (defined in Section 3) and show that a witness encryption scheme with some specific restrictions in the following scenario is impossible.

- The security of scheme is proved based on a *fixed* assumption, that does not depend on the specific instance of the language used in encryption.
- The security of our witness encryption scheme is lower than the hardness of deciding the instance x.

6.1 Impossibility of statistically sound witness encryption scheme

We start by defining the notion of statistical soundness. Next we recall some of the preliminary notation and background needed for proving our claim. Finally we will give our impossibility result.

A witness encryption scheme is said to be *statistically sound* if it has the following property:

- **Statistical Soundness Security.** For any $x \notin L$, for any (even unbounded) adversary A and messages $M_0, M_1 \in \mathcal{M}$, there exists a negligible function $neg(\cdot)$, such that:

$$\left| \Pr\left[A(\mathit{Encrypt}(1^\lambda, x, M_0)) = 1\right] - \Pr\left[A(\mathit{Encrypt}(1^\lambda, x, M_1)) = 1\right] \right| < neg(\lambda)$$

Notation. A *promise problem* Π consists of two disjoint sets Π_Y and Π_N, where Π_Y is the set of *YES instances* and Π_N is the set of *NO instances.* A promise problem Π is associated with the following computational problem: Given an input which is "promised" to lie in $\Pi_Y \cup \Pi_N$, decide whether it comes from Π_Y or Π_N. Note that languages are a special case of promise problems.

We say that a promise problem Π reduces to promise problem Γ if there is a polynomial-time computable function f such that:

$$x \in \Pi_Y \Rightarrow f(x) \in \Gamma_Y$$

$$x \in \Pi_N \Rightarrow f(x) \in \Gamma_N$$

If C is a circuit mapping m-bit strings to n-bit strings, then choosing an input u uniformly at random from $\{0,1\}^m$ defines a probability distribution on $\{0,1\}^n$ given by $C(u)$. For notational convenience, we will denote this probability distribution by C.

For probability distributions X and Y on a discrete set D, the *statistical difference* between X and Y is defined to be

$$||X - Y|| = \max_{S \subset D} |\Pr[X \in S] - \Pr[Y \in S]|.$$

The complexity class SZK consists of promise problems which have an interactive proof system with soundness error less that a small constant and the view of any malicious verifier can be simulated up to $neg(\lambda)$ statistical error. Sahai and Vadhan [SV03] demonstrated that SZK consists exactly of the problems that involve deciding whether two efficiently samplable distributions are either far apart or close together. More formally the following promise problem `Statistical Difference`:

$$SD_Y = \left\{ (C_0, C_1) : ||C_0 - C_1|| > \frac{2}{3} \right\}$$

$$SD_N = \left\{ (C_0, C_1) : ||C_0 - C_1|| < \frac{1}{3} \right\}$$

is SZK complete. In the above description C_0 and C_1 are circuits and define probability distributions as pointed out earlier.

Impossibility. Now we give our impossibility result. We will argue that any NP language for which we can construct a statistically sound witness encryption scheme has to be in SZK. Hence a witness encryption scheme for an NP-complete language implies that $\mathsf{NP} \subseteq \mathsf{SZK}$ which then implies the collapse of the polynomial hierarchy.

Lemma 6.1. *Let L be any NP language and further let ($Encrypt, Decrypt$) be a statistically sound witness encryption scheme for the language L. Then we have that $L \in \mathsf{SZK}$.*

Proof. We will give the proof by giving a reduction. Given a string $x \in \{0,1\}^*$ we will construct circuits C_0 and C_1 such that $||C_0 - C_1|| > \frac{2}{3}$ if $x \in L$ and $||C_0 - C_1|| < \frac{1}{3}$ if $x \notin L$. The construction of the circuits is very simple. The circuit C_b for both $b = 0$ and $b = 1$ corresponds to the distribution $Encrypt(1^\lambda, x, b)$. Now, by the correctness of the witness encryption scheme, given the witness for $x \in L$, one can almost always (i.e., except with negligible probability) tell whether the encrypted value is 0 or 1. In other words, the distributions C_0 and C_1 are almost disjoint.

On the other hand, in the case when $x \notin L$, by statistical soundness we have that the distributions C_0 and C_1 are statistically close. This concludes the proof. □

6.2 Impossibility of restricted witness encryption scheme under simple assumptions

Consider an *NP-complete* language L and a corresponding witness encryption scheme (*Encrypt*, *Decrypt*) for the language L. We will restrict ourselves to witness encryption schemes with the following properties:

- A ciphertext c is said to be a valid for statement x if $\exists M, r$ such that $c = \mathit{Encrypt}(1^\lambda, x, M; r)$. For the purposes of the impossibility result presented in this section we will restrict ourselves to witness encryption schemes for which *validity* of a ciphertext can be checked in polynomial time.

- Secondly, we will restrict ourselves to witness encryption schemes that come equipped with an *extraction function*, denoted as Ext. This extraction function given a ciphertext c, runs in time $T(\lambda)$ (an appropriate exponential function, say 2^λ) and extracts the encrypted message M, without knowledge of the witness or the fact that x is in L or not. Note that this is a significant restriction as $|x|$ can be significantly larger than λ and hence $T(\lambda) \ll 2^x$.

Next we will argue that no such restricted witness encryption scheme whose security can be reduced to a *simple* assumption exists. Our impossibility result for such restricted witness encryption scheme, only provides a partial argument against the impossibility of a witness encryption scheme under a simple assumption because the two restrictions are artificial and in fact aren't even satisfied by our own positive construction. However we still find value in this partial impossibility result as it highlights some of the technical challenges that need to be solved in order to construct a witness encryption scheme from a simple assumption. In particular this impossibility highlights that some form of complexity leveraging might be essential for realizing witness encryption under simple assumptions.

Definition 6.2 (Non-Interactive assumption). *A non-interactive assumption* $\mathsf{Ass} = (V, W, c)$ *is defined by efficient random systems* V *and* W*, modeling the* challenger*, and a constant* $c \in [0, 1)$*. On security parameter* λ*, the challenger generates a problem instance* $\mathcal{P} \leftarrow V(1^\lambda)$ *and provides it to the attacker* $A(1^\lambda, \mathcal{P})$*, which outputs the value* out*. The challenger then executes* $W(\mathsf{out})$ *which outputs a bit* b*. The* advantage *of the attacker* A *is defined as*

$$\mathsf{Adv}^{A}_{\mathsf{Ass}}(\lambda) = \Pr[\ W(A(1^\lambda, V(1^\lambda))) = 1\ ] - c.$$

The assumption Ass *is said to be* secure *if for all PPT attackers* A*, the advantage* $\mathsf{Adv}^{A}_{\mathsf{Ass}}(\lambda)$ *is negligible. An adversary* A *is said to* break *the assumption* Ass *if* $\mathsf{Adv}^{A}_{\mathsf{Ass}}(\lambda)$ *is a non-negligible function of* λ*.*

Lemma 6.3. *Let* (*Encrypt*, *Decrypt*) *be a restricted (as explained above) witness encryption scheme for some NP-complete language* L *with the corresponding* T*-time extraction function* Ext*. Further let* $\mathsf{Ass} = (V, W, c)$ *be a non-interactive assumption. In such a scenario assuming* $q \cdot T + \mathsf{poly}(\lambda)$*-hard one-way functions exist (one way functions secure against adversaries running in time* $q \cdot T + \mathsf{poly}(\lambda)$*) we claim that there does not exist any PPT reduction* $\mathcal{R}^{A(x,\cdot)}(x, \mathcal{P})$ *that on input an instance* x *of the language* L*, an instance* $\mathcal{P} \leftarrow V(1^\lambda)$ *and* q *black-box queries to an adversary* A *(that breaks soundness of the witness encryption scheme for ciphertexts generated corresponding to the instance* x*) breaks the assumption* Ass*.*

Proof. Let us start by assuming that there exists such a reduction $\mathcal{R}$ that uses (in a black-box manner) an adversary A (breaking the soundness of the encryption scheme) and breaks the underlying non-interactive assumption Ass. Now for our proof we will construct a meta-reduction $\mathcal{M}$ that uses this reduction $\mathcal{R}$, simulates A for it and breaks the assumption on its own. We will prove this by a sequence of hybrids.

First we will specify some notation. Let $f : \{0,1\}^n \to \{0,1\}^{3n}$ be any $q \cdot T + \mathsf{poly}(\lambda)$-hard PRG (can be constructed using $q \cdot T + \mathsf{poly}(\lambda)$-hard OWF) where n is an appropriate polynomial in λ. Let L' be the language such that $y \in L'$ if $\exists w$ such that $f(w) = y$. Further let g be an NP-reduction such that $g(y) \in L$ if and only if $y \in L'$.

- H_0: Execute $\mathcal{R}$ with inputs $x, \mathcal{P}$ where $x, \mathcal{P}$ are sampled as follows. Sample a string y uniformly in $\{0,1\}^{3n}$ and set $x = g(y)$. With overwhelming probability y with not be pseudorandom and hence $x \notin L$. $\mathcal{P}$ is an instance of the assumption Ass sampled using $V(1^\lambda)$. $\mathcal{R}$'s calls to the adversary are simulated as follows. If the queried ciphertext is valid then output the output of the extractor function Ext on input that ciphertext, outputting $\perp$ otherwise. Finally feed the output of $\mathcal{R}$, to W as input. Output the output of W as the output of the experiment.

  By soundness of the witness encryption scheme we have that $\mathcal{R}$ breaks the assumption Ass. Therefore, based on the assumption on reduction $\mathcal{R}$ we have that the probability that the above experiment outputs 1 is non-negligibly greater than c. The running time of this hybrid is $q \cdot T + \mathsf{poly}(\lambda)$.

- H_1: This hybrid is same as the previous hybrid except that y is sampled to be a pseudorandom string. Now we will have $x \in L$ where $x = g(y)$.

  The indistinguishability of H_0 and H_1 follows from the $q \cdot T + \mathsf{poly}(\lambda)$-hardness of the PRG.

- H_2: This is the same as the previous hybrid except that instead of simulating the adversary using the function Ext (that runs in T-time) we extract the encrypted message using the *Decrypt* procedure using the witness corresponding to the NP-statement x. This hybrid runs in time $\mathsf{poly}(\lambda)$.

  Indistinguishability between H_1 and H_2 follows based on the perfect correctness of the witness encryption scheme and the fact that response is given only when the queried ciphertext is valid.

Note that in the hybrid H_2 the simulation of the adversary is done locally in polynomial time. Hence the reduction $\mathcal{R}$ can be used to break the assumption on its own. This is a contradiction. This concludes the proof. □

Acknowledgements

We are grateful to the STOC reviewers for making us aware of the work of Rudich [Rud89, Bei11], and for their excellent and helpful comments. We thank Mihir Bellare for pointing an issue in our definition of correctness of witness encryption.

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT*, pages 119–135, 2001.

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In *IWCC*, pages 11–46, 2011.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. extended abstract in Crypto 2001.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 299–315, 2003.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003.

[CCKV08] André Chailloux, Dragos Florin Ciocan, Iordanis Kerenidis, and Salil P. Vadhan. Interactive and noninteractive zero knowledge are equivalent in the help model. In *TCC*, pages 501–534, 2008.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, pages 523–552, 2010.

[Coc01] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, pages 45–64, 2002.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *FOCS*, pages 283–293, 2000.

[GGH12] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices and applications. Cryptology ePrint Archive, Report 2012/610, 2012. `http://eprint.iacr.org/`.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. Cryptology ePrint Archive, Report 2013/128, 2013. `http://eprint.iacr.org/`.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Jour. of Computer and System Science*, 28(2):270–299, 1984.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

[GO92] Shafi Goldwasser and Rafail Ostrovsky. Invariant signatures and non-interactive zero-knowledge proofs are equivalent (extended abstract). In *CRYPTO*, pages 228–245, 1992.

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge University Press, New York, NY, USA, 1 edition, 2008.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In *Proceedings of Eurocrypt 2006, volume 4004 of LNCS*, pages 339–358. Springer, 2006.

[GOVW12] Sanjam Garg, Rafail Ostrovsky, Ivan Visconti, and Akshay Wadia. Resettable statistical zero knowledge. In *TCC*, pages 494–511, 2012.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits. In *STOC*, 2013.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In *CRYPTO*, pages 408–423, 1998.

[Ins] Clay Mathematics Institute. Millennium prize problems. `http://www.claymath.org/millennium/`.

[IOS97] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *J. Cryptology*, 10(1):37–50, 1997.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61, 1989.

[IS91] Toshiya Itoh and Kouichi Sakurai. On the complexity of constant round zkip of possession of knowledge. In *ASIACRYPT*, pages 331–345, 1991.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[KMV07] Bruce M. Kapron, Lior Malka, and Srinivasan Venkatesh. A characterization of non-interactive instance-dependent commitment-schemes (nic). In *ICALP*, pages 328–339, 2007.

[OV08] Shien Jin Ong and Salil P. Vadhan. An equivalence between zero knowledge and commitments. In *TCC*, pages 482–500, 2008.

[Rud89] Steven Rudich. Unpublished, 1989.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[SV03] Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2):196–249, 2003.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[TW87] Martin Tompa and Heather Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *FOCS*, pages 472–482, 1987.

[Wat05] Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[Web] Eternity Puzzle Website. Eternity puzzle. `http://www.eternity-puzzle.com/`.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

A Proofs for Section 4

Now we give security proofs for the constructions described in Section 4.

A.1 Public Key Encryption

Recall that the scheme presented in Section 4 relied on a PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\cdot\lambda}$ and our witness encryption scheme $(\mathit{Encrypt}_{\mathrm{WE}}, \mathit{Decrypt}_{\mathrm{WE}})$.

Lemma A.1. *Assuming that G is a PRG and $(\mathit{Encrypt}_{WE}, \mathit{Decrypt}_{WE})$ is a witness encryption scheme we have that $(\mathit{Setup}_{PKE}, \mathit{Encrypt}_{PKE}, \mathit{Decrypt}_{PKE})$ is a* semantically secure *public-key encryption scheme.*

Proof. Let us start by assuming that the encryption scheme is not semantically secure. In other words there exists an adversary $\mathcal{A}$ such that its advantage in the semantic security game (defined in Section C) is non-negligible. We will reach a contradiction by considering the following sequence of hybrids.

- H_0: This hybrid corresponds to the actual semantic security game as defined in Section C.
- H_1: Recall the public key generation process proceeds by sampling a random PRG seed $s \in \{0,1\}^\lambda$. Next, it uses the PRG $G : \{0,1\}^\lambda \to \{0,1\}^{2\cdot\lambda}$ to compute $G(s) \to t \in \{0,1\}^{2\lambda}$. The public key $\mathrm{PK} = (t, \lambda)$ is the output of the PRF and the security parameter. The hybrid H_1 is same as hybrid H_0 except that instead of sampling t according to the above process, we just generate a sample a random string in $\{0,1\}^{2\lambda}$ and use that to generate the public key.

  The indistinguishability of H_0 and H_1 follows from the security of the PRG.

Now note that in hybrid H_1 we use the witness encryption algorithm to encrypt to an instance x such that $x \in L$ if and only if t is in the range of G. Further note that since t is a randomly chosen string we can claim that t is not in the range of G except with negligible probability. This implies that $x \notin L$ except with negligible probability. Hence the advantage of $\mathcal{A}$ in hybrid H_1 can be directly reduced to soundness security of the underlying witness encryption scheme. This concludes the proof. □

A.2 Identity-Based Encryption

Recall that the IBE scheme presented in Section 4 relied on the Goldreich-Levin [GL89] hardcore bit $\mathrm{GL}(\sigma, r)$ of σ using randomness r, a unique signature scheme (*Setup-Signature*, *Sign*, *Verify*) and our witness encryption scheme ($\mathit{Encrypt}_{\mathrm{WE}}, \mathit{Decrypt}_{\mathrm{WE}}$).

Lemma A.2. *Assuming that (Setup-Signature, Sign, Verify) is a unique signature scheme existentially unforgeable under chosen message attack and* ($\mathit{Encrypt}_{WE}, \mathit{Decrypt}_{WE}$) *is a witness encryption scheme we have that* (Setup_{IBE}, $\mathit{Encrypt}_{IBE}, \mathit{Decrypt}_{IBE}$) *is a selectively secure IBE scheme.*

Proof. Let us start by assuming that the IBE scheme is not selectively secure. In other words there exists an adversary $\mathcal{A}$ such that its advantage in the selective security game (defined in Section C) is non-negligible. We will reach a contradiction by considering the following sequence of hybrids.

- H_0: This hybrid corresponds to the actual selective security game as defined in Section C.

- H_1: Recall that the ciphertext generation process in the scheme proceeds as follows. The encrypter generates instance x_0 (resp., x_1) such that $x_0 \in L$ (resp., $x_1 \in L$) if and only if there exists a signature σ where $\mathrm{GL}(\sigma, r) = 0$ (resp., $\mathrm{GL}(\sigma, r) = 1$) and where $\mathit{Verify}(\mathrm{PP}, \sigma, \mathcal{I}^*) =$ true. Then, it computes $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x_0, M_\beta) \to C_0$ and $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x_1, M_\beta) \to C_1$. Note that since the signature scheme used has unique signatures we can conclude that exactly only one of the instances x_0 and x_1 will be in L. Recall that M_0 and M_1 are the messages generated by the adversary, β is the random bit chosen by the challenger and $\mathcal{I}^*$ is the selective identity that the attacker is trying to attack.

  Now we describe hybrid H_1. Hybrid H_1 is same as the hybrid H_0 except that we change either C_0 if $x_0 \notin L$ or C_1 if $x_1 \notin L$. Whether $x_0 \notin L$ or $x_1 \notin L$ can be easily found by generating the signature σ and checking if $\mathrm{GL}(\sigma, r)$ is 0 or 1. We will describe the case in which $x_0 \notin L$. The other case in analogous. Unlike hybrid H_0 where C_0 was generated as $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x_0, M_\beta)$, in hybrid H_1 we generate C_0 as $\mathit{Encrypt}_{\mathrm{WE}}(1^\lambda, x_0, M_{1-\beta})$.

  Indistinguishability between H_0 and H_1 follows from the soundness security of the witness encryption scheme.

Now note that the adversary's ability to guess the encrypted bit in H_1 can be used to to directly guess the hardcore bit $\mathrm{GL}(\sigma_{\mathcal{I}^*}, r)$ of $\sigma_{\mathcal{I}^*}$ using randomness r. This is because the encrypted bit changes depending on the hardcore bit. Hence we can use this adversary to extract the unique signature on the identity $\mathcal{I}^*$. This leads to a contradiction as the signature scheme is assumed to be existentially unforgeable. Note that the adversary in addition to the challenge ciphertext also expects to receive secret keys for identities of its choice (except the challenge identity). In the reduction these can be provided by relying on the signing oracle. This concludes the proof. □

A.3 Attribute-Based Encryption for Circuits

Recall that the (Key-Policy) Attribute-Based Encryption scheme for circuits presented in Section 4 relied on Com a *perfectly binding* non-interactive commitment scheme, (P, V) a non-interactive zap (defined in Section B) and our witness encryption scheme $(\mathit{Encrypt}_{\mathrm{WE}}, \mathit{Decrypt}_{\mathrm{WE}})$.

Lemma A.3. *Assuming that* Com *is perfectly binding and computationally hiding commitment scheme,* (P, V) *is a non-interactive zap and* $(\mathit{Encrypt}_{WE}, \mathit{Decrypt}_{WE})$ *is a witness encryption scheme, we have that* $(\mathit{Setup}_{ABE}, \mathit{Encrypt}_{ABE}, \mathit{Decrypt}_{ABE})$ *is a selectively secure attribute based encryption scheme.*

Proof. Let us start by assuming that the ABE scheme is not selectively secure. In other words there exists an adversary $\mathcal{A}$ such that its advantage in the selective security game (defined in Section C) is non-negligible. We will reach a contradiction by considering the following sequence of hybrids.

- H_0: This hybrid corresponds to the actual selective security game as defined in Section C.
- H_1: This hybrid is same as the hybrid H_0 except that instead of committing to a zero string in c_2 we start by committing to the challenge value a^* that the attacker specifies.

  Indistinguishability follows based on the computational hiding property of the commitment scheme Com.
- H_2: This hybrid is same as the hybrid H_1 except that instead of generating the zap proof using the randomness used in the generation of the commitment c_1 we now use the randomness used in generation of the commitment c_2. Note that since the adversary is only allowed to query for secret keys corresponding to functions f such that $f(a^*) = 0$ we will always be able to answer the adversary's secret key queries.

  Indistinguishability follows based on the computational witness indistinguishability property of the zap.
- H_3: Same as the previous hybrid except that instead of generating c_1 as a commitment of 0 we generate it as a commitment to 1.

  Indistinguishability follows based on the computational hiding property of the commitment scheme Com.

Finally, note that in hybrid H_3 we used the witness encryption algorithm to encrypt to an instance x' such that $x' \in L$ if and only if there exists a circuit g such that $|g| \leq \ell_{max}$ and a proof π_g (of size $\mu(\lambda, n, |g|)$) such that $V(x_g, \pi_g) = 1 \wedge g(a) = 1$ where x_g is the NP-statement:

$$\exists (w_1, a, w_2) \text{ such that } c_1 = \mathsf{Com}(0; w_1) \vee (c_2 = \mathsf{Com}(a; w_2) \wedge g(a) = 0)$$

Note that c_1 is a commitment to 1 and c_2 is a commitment to a^*. Therefore, for every function h such that $h(a^*) = 1$ we will have x_h is unsatisfiable. Hence from the perfect soundness of zaps we can conclude that x' is unsatisfiable. Hence the ability of the adversary to correctly guess the value encrypted in H_3 can be directly reduced to the soundness security of the underlying witness encryption scheme. This concludes the proof. □

A.4 Fully Secure Identity-Based Encryption

Recall that the Fully Secure Identity-Based Encryption presented in Section 4 relied on Com a *perfectly binding* non-interactive commitment scheme, (P, V) a non-interactive zap (defined in Section B) and our witness encryption scheme $(\mathit{Encrypt}_{\mathrm{WE}}, \mathit{Decrypt}_{\mathrm{WE}})$.

Lemma A.4. *Assuming that* Com *is perfectly binding and computationally hiding commitment scheme,* (P, V) *is a non-interactive zap,* F *is a pseudorandom function family and* $(\mathit{Encrypt}_{WE}, \mathit{Decrypt}_{WE})$ *is a witness encryption scheme, we have that* $(\mathit{Setup}_{IBE}, \mathit{Encrypt}_{IBE}, \mathit{Decrypt}_{IBE})$ *is a fully secure IBE scheme.*

Proof. Let us start by assuming that the IBE scheme is not fully secure. In other words there exists an adversary $\mathcal{A}$ such that its advantage in the full security game (defined in Section C) is non-negligible. We will reach a contradiction by considering the following sequence of hybrids. Without loss of generality, let us assume that the number of secret key queries that the adversary makes (denoted as q) is a power of 2. (One can always pad the number of secret key queries to a power of 2 in case the number is not an exact power of 2.)

- H_0: This hybrid corresponds to the actual full security game as defined in Section C.

- H_1: This hybrid is same as the hybrid H_0 except that instead of always completing the game we will sometimes abort the game and make a random guess on behalf of the attacker. We will argue that if the adversary's advantage in hybrid H_0 was non-negligible then it continues to be so in hybrid H_1.

  More specifically, the challenger runs the real game, but in parallel samples a random function[9] $G : \{0,1\}^\lambda \rightarrow \{0,1\}^t$ where $t = \log q$. At the end of the game we check (we refer to this condition as Bad) to see if:

  $$\left(\exists i \in [q]\ G(\mathcal{I}_i) = 0^t\right) \vee \left(G(\mathcal{I}^*) \neq 0^t\right)$$

  where $\mathcal{I}^*$ is the challenge identity. If Bad is true then we abort the game and generate a random guess on behalf of the adversary. Otherwise if the Bad is false then the adversary wins if he guesses the correct bit β (where the challenger encrypted M_β).

  Observe that the advantage of the adversary in hybrid H_1 will be $O(\frac{1}{q})$ times the advantage of the adversary in hybrid H_0.

- H_2: This hybrid is same as the previous hybrid except that instead of using the random function G we use the PRF F. More specifically, at the start of the execution we sample a random seed $s \leftarrow \{0,1\}^\lambda$ and use the function $F'_{s,t}(\cdot)$ instead of G.

  Indistinguishability between hybrids H_2 and H_1 follows from the pseudorandomness property of the PRF F.

- H_3: This hybrid is same as the previous hybrid except that instead of postponing to check Bad to the very end. We check the Bad condition every time a secret key query is made and when the challenge identity is specified. If the Bad condition is ever true then we immediately abort the interaction with the adversary and output a random guess on behalf of the adversary.

  The advantage of the adversary in H_3 is identical to the advantage of the adversary in H_2.

[9] Since we will evaluate the function G only on polynomially many inputs. We can sample the function on those inputs on the fly.

- H_4: This hybrid is same as the previous hybrid except that we generate $c_2 = \mathsf{Com}(s; R)$ and $c_3 = \mathsf{Com}(t, R')$.

  Indistinguishability between hybrids H_4 and H_3 follows based on the computational hiding property of the commitment scheme Com.

- H_5: This hybrid is same as the hybrid H_4 except that instead of generating the zap proof using the randomness used in the generation of the commitment c_1 we now use the randomness used in the generation of the commitments c_2 and c_3. Note that since the adversary at this point only makes secret key queries such that Bad is false we will have that for every $i \in [q]$, $F_{s,t}(\mathcal{I}_i) \neq 0$ and hence we will always be able to answer the adversary's secret key queries.

  Indistinguishability between hybrids H_5 and H_4 follows based on the computational witness indistinguishability property of the zap.

- H_6: Same as the previous hybrid except that instead of generating c_1 as a commitment of 0 we generate it as a commitment to 1.

  Indistinguishability between hybrids H_6 and H_5 follows based on the computational hiding property of the commitment scheme Com.

Finally, note that in hybrid H_6 we used the witness encryption algorithm to encrypt to an instance x' such that $x' \in L$ if and only if there exists a proof $\pi_{\mathcal{I}^*}$ (of appropriate size) such that $V(x_{\mathcal{I}^*}, \pi_{\mathcal{I}^*}) = 1$ where $x_{\mathcal{I}^*}$ is the NP-statement:

$$\exists(w_1, s, t, w_2, w_3) \text{ such that } c_1 = \mathsf{Com}(0; w_1) \vee (c_2 = \mathsf{Com}(s; w_2) \wedge c_3 = \mathsf{Com}(t; w_3) \wedge F_{s,t}(\mathcal{I}^*) \neq 0)$$

However, since Com is perfectly binding and Bad is false we have that $F_{s,t}(\mathcal{I}^*) = 0$ and so $x_{\mathcal{I}^*}$ is unsatisfiable. Next from the perfect soundness of zaps we can conclude that x' is unsatisfiable. Hence the ability of the adversary to correctly guess the value encrypted in H_6 can be directly reduced to the soundness security of the underlying witness encryption scheme. This concludes the proof. □

B Non-interactive Zaps

In 2000, Dwork and Naor [DN00] proved that there exist "zaps", two-round witness-indistinguishable (WI) proofs in the plain model without a common reference string, where the verifier asks a single question and the prover sends back a single answer. Furthermore, [DN00] showed that their constructions allowed for the first message (from verifier to prover) to be reused – so that between a particular pair of prover and verifier, only one message from verifier to prover is required even if many statements are to be proven.

Barak, Ong, and Vadhan [BOV03] constructed the first non-interactive zaps for any NP relation by applying derandomization techniques to the construction of Dwork and Naor, based on trapdoor permutations and the assumption that (very good) Hitting Set Generators (HSG) against co-nondeterministic circuits exist. It is known that such HSG's can be built if there is a function in E that requires exponential-size *nondeterministic* circuits – *i.e.* the assumption states that some uniform exponential deterministic computations can (only) be sped up by at most a constant power (Time 2^{cn} becomes $2^{\varepsilon n}$), when given the added power of nondeterminism and advice specific to the length of the input. Subsequently, Groth, Ostrovsky and Sahai [GOS06] gave a much more efficient construction for non-interactive zaps based on bilinear-maps.

Let R be an efficiently computable binary relation. For pairs $(x, w) \in R$ we call x the statement and w the witness. Let L be the language consisting of statements in R.

A non-interactive zap for a relation R consists of a prover P and a verifier V. We require that they all be probabilistic polynomial time algorithms, *i.e.*, we are looking at *efficient prover* proofs. The prover takes as input (x, w) and produces a proof π. The verifier takes as input (x, π) and outputs 1 if the proof is acceptable and 0 if rejecting the proof. We call (P, V) a non-interactive zap for R if it has the completeness, soundness and witness indistinguishable properties described below.

Perfect completeness. A proof system is complete if an honest prover with a valid witness can convince an honest verifier. For all adversaries $\mathcal{A}$ we have

$$\Pr\Big[(x, w) \leftarrow \mathcal{A}; \pi \leftarrow P(x, w) : V(x, \pi) = 1 \text{ if } (x, w) \in R\Big] = 1.$$

Perfect soundness. A proof system is sound if it is infeasible to convince an honest verifier when the statement is false. For all $x \notin L$ and all adversaries $\mathcal{A}$ we have

$$\Pr\Big[\pi \leftarrow \mathcal{A}(x) : V(x, \pi) = 1\Big] = 0.$$

Witness-indistinguishability. Witness-indistinguishability means that proof does not reveal which witness the prover used. For all non-uniform polynomial time interactive adversaries $\mathcal{A}$ we have

$$\begin{aligned} & \Pr\Big[(x, w_0, w_1) \leftarrow \mathcal{A}(1^k); \pi \leftarrow P(1^k, x, w_0) : \mathcal{A}(\pi) = 1 \text{ and } (x, w_0), (x, w_1) \in R\Big] \\ \approx\ & \Pr\Big[(x, w_0, w_1) \leftarrow \mathcal{A}(1^k); \pi \leftarrow P(1^k, x, w_1) : \mathcal{A}(\pi) = 1 \text{ and } (x, w_0), (x, w_1) \in R\Big]. \end{aligned}$$

A hybrid argument shows that this definition of witness-indistinguishability is equivalent to a definition where we give the adversary access to multiple proofs using either witness w_0 or witness w_1.

C Security Definitions

We will recall the definition of semantic security for a public key encryption scheme and the definitions of selective and full security for IBE and ABE schemes.

Semantic Security. Semantic security is described by a security game between a challenger and an attacker. The game proceeds as follows.

- **Setup:** The challenger runs the *Setup* algorithm and gives the public key PK to the attacker.
- **Challenge:** The attacker declares two equal-length messages M_0 and M_1 to the challenger. The challenger flips a random coin $\beta \in \{0, 1\}$, and encrypts M_β, producing ciphertext CT^* which it gives to the adversary. We call this ciphertext the *challenge ciphertext*.
- **Guess:** The attacker outputs a guess β' for β.

The advantage of an attacker in this game is defined to be $\Pr[\beta = \beta'] - \frac{1}{2}$.

Selective Security. Selective security is described by a security game between a challenger and an attacker. We will describe the definition in the context of IBE. It can be adapted for the setting of ABE in a natural way. The game proceeds as follows.

- **Challenge Identity:** The attacker starts by declaring the challenge identity $\mathcal{I}^*$ that it wants to attack.
- **Setup:** The challenger runs the *Setup* algorithm and gives the public parameters PP to the attacker.
- **Query Phase 1:** The attacker queries the challenger for private keys corresponding to identities $\mathcal{I}_1, \ldots, \mathcal{I}_{q_1}$. (For all $i \in [q_1]$ we require that $\mathcal{I}_i \neq \mathcal{I}^*$)
- **Challenge:** The attacker declares two equal-length messages M_0 and M_1. The challenger flips a random coin $\beta \in \{0,1\}$, and encrypts M_β under the identity $\mathcal{I}^*$, producing ciphertext CT^* which it gives to the adversary.
- **Phase 2:** The attacker queries the challenger for private keys corresponding to the identities $\mathcal{I}_{q_1+1}, \ldots, \mathcal{I}_q$. (For all $i \in [q]\backslash[q_1]$ we require that $\mathcal{I}_i \neq \mathcal{I}^*$)
- **Guess:** The attacker outputs a guess β' for β.

The advantage of an attacker in this game is defined to be $\Pr[\beta = \beta'] - \frac{1}{2}$.

Full Security. Again we will describe the definition for the context of IBE. It can be adapted for the setting of ABE in a natural way. The game proceeds as follows.

- **Setup:** The challenger runs the *Setup* algorithm and gives the public parameters PP to the attacker.
- **Query Phase 1:** The attacker queries the challenger for private keys corresponding to identities $\mathcal{I}_1, \ldots, \mathcal{I}_{q_1}$.
- **Challenge:** The attacker declares two equal-length messages M_0 and M_1 and a challenge identity $\mathcal{I}^*$. We require that $\mathcal{I}^* \neq \mathcal{I}_i$ for all $i \in [q_1]$. The challenger flips a random coin $\beta \in \{0,1\}$, and encrypts M_β for the identity $\mathcal{I}^*$, producing ciphertext CT^* which it gives to the adversary.
- **Phase 2:** The attacker queries the challenger for private keys corresponding to the identities $\mathcal{I}_{q_1+1}, \ldots, \mathcal{I}_q$. (For all $i \in [q]\backslash[q_1]$ we require that $\mathcal{I}_i \neq \mathcal{I}^*$)
- **Guess:** The attacker outputs a guess β' for β.

The advantage of an attacker in this game is defined to be $\Pr[\beta = \beta'] - \frac{1}{2}$.