Efficient Chosen-Ciphertext Security via Extractable Hash Proofs

Hoeteck Wee*

Queens College, CUNY
hoeteck@cs.qc.cuny.edu

Abstract. We introduce the notion of an extractable hash proof system. Essentially, this is a special kind of non-interactive zero-knowledge proof of knowledge system where the secret keys may be generated in one of two modes to allow for either simulation or extraction.

- We show how to derive efficient CCA-secure encryption schemes via extractable hash proofs in a simple and modular fashion. Our construction clarifies and generalizes the recent factoring-based cryptosystem of Hofheinz and Kiltz (Eurocrypt '09), and is reminiscent of an approach proposed by Rackoff and Simon (Crypto '91). We show how to instantiate extractable hash proof system for hard search problems, notably factoring and computational Diffie-Hellman. Using our framework, we obtain the first CCA-secure encryption scheme based on CDH where the public key is a constant number of group elements and a more modular and conceptually simpler variant of the Hofheinz-Kiltz cryptosystem (though less efficient).
- We introduce adaptive trapdoor relations, a relaxation of the adaptive trapdoor functions considered by Kiltz, Mohassel and O'Neil (Eurocrypt '10), but nonetheless imply CCA-secure encryption schemes. We show how to construct such relations using extractable hash proofs, which in turn yields realizations from hardness of factoring and CDH.

* Supported by NSF CAREER Award CNS-0953626 and PSC-CUNY Award # 6014939 40.

1 Introduction

The most basic security guarantee we require of a public key encryption scheme (PKE) is that of semantic security against chosen-plaintext attacks (CPA) [21]: it is infeasible to learn anything about the plaintext from the ciphertext. On the other hand, there is a general consensus within the cryptographic research community that in virtually every practical application, we require semantic security against adaptive chosen-ciphertext attacks (CCA) [37, 15], wherein an adversary is given access to decryptions of ciphertexts of her choice. So far, there have been two largely separate lines of works addressing the construction of CCA-secure encryption schemes: the first examines constructions from general assumptions starting with the beautiful works of Dolev, Dwork, Naor and Yung [15, 34, 37, 39, 31, 18, 36, 38, 33, 29] and related questions pertaining to minimal assumptions; the second examines practical and efficient constructions from specific number-theoretic assumptions, starting from those of Cramer and Shoup [11, 40, 12, 30, 2, 24, 9, 10, 25]. In recent years, two distinct trends have surfaced in each of these lines of works.

Practical CCA from Search Problems. Until very recently, all of the practical CCA-secure encryption schemes (namely the Cramer-Shoup encryption scheme and all its variants) inherently relied on decisional assumptions, e.g., the Decisional Diffie-Hellman (DDH) assumption or the quadratic residuosity assumption. In general, decisional assumptions are a much stronger class of assumptions than computational assumptions based on search problems, such as factoring, finding shortest vectors in lattices, or even the Computational Diffie-Hellman (CDH) problem. Indeed, there are groups, such as certain elliptic curve groups with bilinear pairing map, where the DDH assumption does not hold, but the Computational Diffie-Hellman (CDH) problem appears to be hard. As such, schemes based on search problems are generally preferred to those based on decisional assumptions. However, such schemes seem to be very hard to obtain.

Several years ago, Canetti, Halevi and Katz [9] proposed the first practical CCA-secure PKE based on a computational assumption, namely the Bilinear DH assumption in bilinear groups (BDH). Since then, a series of works have shown how to base CCA-secure encryption schemes on CDH [10, 22, 23] and on hardness of factoring [25]. However, there seems to be no overarching framework explaining these schemes. Partial progress towards a unifying approach was made recently by Cramer, Hofheinz and Kiltz [13]; their approach remains unsatisfactory in two ways: first, it does not encompass constructions from hardness of factoring (it does cover the RSA assumption, which is possibly a stronger assumption), and second, the ensuing schemes even with suitable algebraic optimizations, do not quite match the efficiencies obtained in preceding works (for instance, the public key in the RSA-based scheme contains a linear number of group elements, whereas that in the factoring-based scheme of Hofheinz and Kiltz [25] only requires a constant number of group elements).

CCA from weaker general assumptions. Since the breakthrough work of Peikert and Waters on lossy trapdoor functions [36], a series of works has identified successively weaker general assumptions from which we may realize CCA-secure encryption schemes [38, 29] (in a black-box way). The current state-of-the-art is the (tag-based) adaptive trapdoor functions of Kiltz, Mohassel and O'Neil [29]; roughly speaking, these are trapdoor functions that remain one-way even if the adversary is given access to a restricted inversion oracle that inverts the function on "most" inputs. In spite of the black-box separations indicating that adaptive trapdoor functions are strictly weaker than its predecessors [29, 41], all of the concrete (standard) assumptions from which we can realize adaptive trapdoor functions are not significantly different from those known to imply lossy trapdoor functions. Most notably, we do not know how to base adaptive

trapdoor functions on hardness of factoring (or the standard RSA assumption, and more generally, any hard *search* problem not related to lattices). On the other hand, we do know how to derive CCA-secure encryption schemes from enhanced trapdoor permutations, which may in turn be based on hardness of factoring [15, 16, 19].

1.1 Our Contributions

We introduce the notion of an *extractable hash proof system*, inspired in part by the Cramer-Shoup universal hash proof systems [12]. Informally, extractable hash proofs are like universal hash proofs in that they are a special kind of non-interactive zero-knowledge proofs [4], except we replace the soundness requirement (corresponding to smoothness) with a "proof of knowledge property" [37, 14]. That is, the secret keys may be generated in one of two modes to allow for either simulation or extraction. Using extractable hash proofs, we obtain new insights into the construction of CCA-secure encryption schemes, and obtain new results for both lines of works described earlier. Before we describe our results, we present an overview of extractable hash proofs.

Extractable Hash Proof Systems. Fix R to be a relation corresponding to some hard search problem – namely, R is efficiently samplable, but given a random u, it is hard to find an s such that $(u, s) \in \mathsf{R}$. (For instance, s is the pre-image of u under a one-way permutation.) We consider a family of hash functions $\{\mathsf{H}_{\mathrm{PK}}\}$ indexed by a public key PK which maps an input u to some value. (We clarify that the name is somewhat of a misnomer since the "hash function" will in fact be injective, and possibly even length-increasing.) Moreover, we require that the hash function be efficiently computable given PK and the coin tosses r used to sample $(u, s) \in \mathsf{R}$. We denote this public evaluation algorithm by $\mathsf{Pub}(\mathrm{PK}, r)$ and the hash value by $\mathsf{H}_{\mathrm{PK}}(u)$.

Associated with this family of functions is a set-up algorithm that generates the public key PK along with a secret key. The set-up algorithm operates in one of two modes. In both modes, the algorithm generates exactly the same distribution of public keys; however, the functionality afforded by the secret key depends on the mode:

- In the hashing mode, the secret key SK^* allows us to compute the hash value $\mathsf{Pub}(\mathrm{PK}, u)$ without knowing either s or r. Specifically, there is a private evaluation algorithm Priv such that for all $(u, s) \in \mathsf{R}$, $\mathsf{Priv}(\mathrm{SK}^*, u) = \mathsf{H}_{\mathrm{PK}}(u)$.

- In the extraction mode, the secret key SK allows us to verify whether a hash value is correctly computed and if so extract a witness s. More formally, there is an extraction algorithm Ext, such that for all u, τ: $\mathsf{Ext}(\mathrm{SK}, u, \tau)$ outputs s satisfying $(u, s) \in \mathsf{R}$ iff $\tau = \mathsf{H}_{\mathrm{PK}}(u)$. This implies efficient verification of the hash value (given SK) whenever R is efficiently computable.

Looking ahead, we will rely on the extraction mode for decryption in a CCA-secure encryption scheme, and on the hashing mode for the proof of security. This is opposite to the use of universal hash proofs in the Cramer-Shoup framework, where the hashing mode is used for decryption and the smoothness property (corresponding to soundness and thus extraction) is used to establish security. Moreover, unlike Cramer-Shoup hash proofs, extractable hash proofs are designed in tandem with families of relations, and are particularly well-suited for use with computationally hard search problems.

Practical CCA via Extractable Hash Proofs. We provide a generic construction of CCA-secure encryption schemes from extractable hash proofs. We use as an intermediate building block a somewhat richer cryptographic abstraction called *all-but-one extractable hash proofs* (which can be constructed generically from extractable hash proofs). The overall construction follows a variant of the Rackoff-Simon paradigm [37] (as opposed to the Naor-Yung double-encryption paradigm [34], also used in [13]): encrypt (or commit to) a one-time symmetric key (which is in turn used to encrypt the message, following the hybrid encryption paradigm), and then provide a zero-knowledge proof of knowledge of the key using an extractable hash proof. Indeed, such an approach was used implicitly in the afore-mentioned cryptosystems based on computational assumptions; however, the connection to the Rackoff-Simon paradigm has never been made explicit. Our framework may be viewed as a clarification and unification of all these constructions. We present extractable hash proofs related to hardness of factoring and CDH; in addition, we obtain the following new cryptosystems:

- a variant of the Hofheinz-Kiltz CCA-secure encryption scheme based on hardness of factoring (Fig 3), which is more modular and both conceptually and mathematically simpler, albeit less efficient — there is a linear blow-up in both ciphertext overhead and public key size over the previous scheme;
- a CCA-secure encryption scheme based on CDH where the public key comprises a constant number of group elements (Fig 5) and a linear ciphertext overhead; previous works all require a linear number of group elements [10, 22, 23] in the public key. Our construction offers a trade-off between public key size and ciphertext overhead when compared with the schemes in [22, 23]; such a trade-off may be preferable when encrypting very long messages via the hybrid encryption paradigm.

Our framework also encompasses a series of CCA-secure encryption schemes [9, 7, 27, 28] derived from the identity-based encryption schemes in [5, 8] whose security are based on decisional assumptions.

CCA from Adaptive Trapdoor Relations. We also propose a relaxation of adaptive trapdoor functions, which we call *adaptive trapdoor relations*. The relaxation here lies in the functionality requirement for evaluation: we only require that there exists an efficient sampling algorithm that generates a random input to the trapdoor function along with its image; the function itself need not be efficiently computable. It follows immediately from [29] (with essentially the same construction as that in [36, 38]) that adaptive trapdoor relations imply CCA-secure encryption schemes. Interestingly, the ensuing construction unlike previous constructions, is not witness-recovering (that is, the decryption algorithm does not completely recover the randomness used for encryption, c.f. [36, Section 1.1]).

Next, we show how to derive adaptive trapdoor relations from hardness of factoring and CDH. This partially answers an open problem posed in [29] on realizing adaptive trapdoor functions from hard search problems not related to lattices. (A comparison with previous works is shown in Fig 1.) Our construction relies on the use of extractable hash proofs and is very similar to our CCA-secure encryption schemes. Moreover, our adaptive trapdoor relations are fairly efficient and achieve parameters similar to the state-of-the-art lossy trapdoor functions based on DCR and DDH respectively [17].

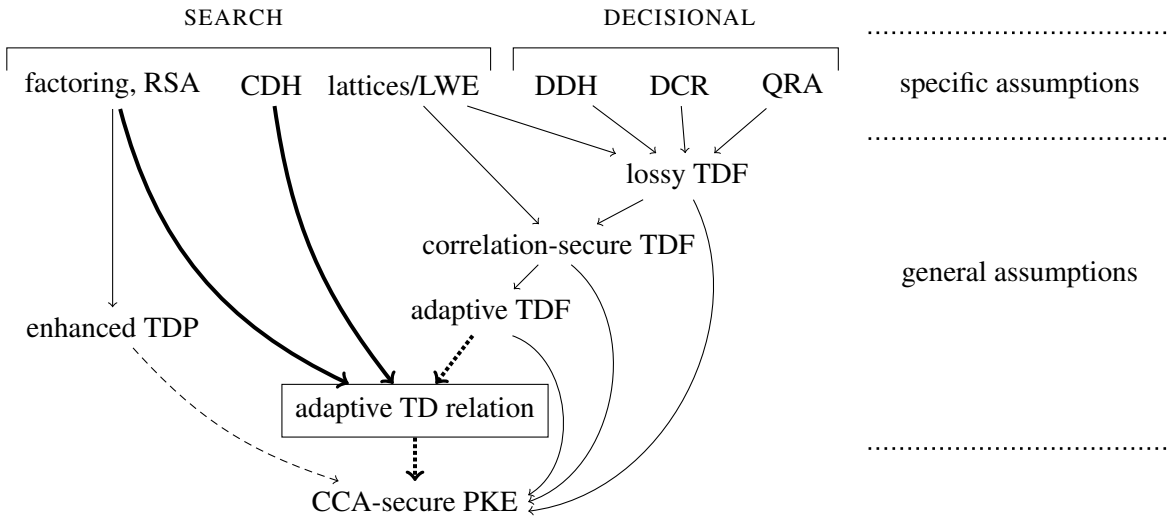

Fig. 1. Summary of CCA-secure PKEs from general assumptions, and how the latter relate to (standard) specific assumptions [15, 16, 19, 36, 38, 35, 29, 32, 17]. Here, lossy TDF and adaptive TDF refer to the respective all-but-one/tag-based variants. The bold lines denote our contributions (the dotted lines denote those that are straight-forward or follow readily from previous work). All of the constructions from general assumptions are black-box, except for the one marked with dashed lines. (Following current conventions, we do not regard hash proof systems [12] as a general assumption.)

2 Preliminaries and Definitions

2.1 Key Encapsulation Mechanisms

A *key encapsulation mechanism* (KEM) $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with key-space $\{0,1\}^k$ consists three polynomial-time algorithms.Via $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k)$ the randomized key-generation algorithm produces public/secret keys for security parameter 1^k; via $(C, K) \leftarrow \mathsf{Enc}(\mathrm{PK})$, the randomized encapsulation algorithm creates a uniformly distributed symmetric key $K \in \{0,1\}^k$, together with a ciphertext C; via $K \leftarrow \mathsf{Dec}(\mathrm{SK}, C)$, the possessor of secret key SK decrypts ciphertext C to get back a key K which is an element in $\{0,1\}^k$ or a special reject symbol $\perp$. For consistency, we require that for all k and all $(C, K) \leftarrow \mathsf{Enc}(\mathrm{PK})$, we have $\Pr[\mathsf{Dec}(\mathrm{SK}, C) = K] = 1$, where the probability is taken over the choice $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k)$ and the coins of all the algorithms in the expression above.

Chosen-Ciphertext Security. The common requirement for a KEM is indistinguishability against chosen-ciphertext attacks (IND-CCA) [12] where an adversary is allowed to adaptively query a decapsulation oracle with ciphertexts to obtain the corresponding session key. More formally, for an adversary $\mathcal{A}$, we define the advantage function

$$\mathsf{AdvCCA}^{\mathcal{A}}_{\mathrm{KEM}}(k) := \Pr\left[b = b' : \begin{array}{l} (\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Gen}(1^k); \\ (C, K_0) \leftarrow \mathsf{Enc}(\mathrm{PK}); K_1 \leftarrow_{\mathrm{R}} \{0,1\}^k; \\ b \leftarrow_{\mathrm{R}} \{0,1\}; \\ b' \leftarrow \mathcal{A}^{\mathsf{Dec}(\mathrm{SK},\cdot)}(\mathrm{PK}, K_b, C) \end{array} \right]$$

with the restriction that $\mathcal{A}$ is only allowed to query $\mathsf{Dec}(\mathrm{SK}, \cdot)$ on ciphertexts different from the challenge ciphertext C. A KEM is said to be *indistinguishable against chosen ciphertext attacks* (IND-CCA) if for all PTA adversaries $\mathcal{A}$, the advantage $\mathsf{AdvCCA}^{\mathcal{A}}_{\mathrm{KEM}}(k)$ is a negligible function in k.

It was shown in [12] that an IND-CCA secure KEM with a CCA-secure symmetric encryption scheme yields an IND-CCA secure hybrid encryption scheme.

2.2 Binary Relations for Search Problems

Fix a family of (binary) relations R_{PP} indexed by a public parameter PP. We require that PP be efficiently samplable given a security parameter 1^k, and assume that all algorithms are given PP as part of its input. We omit PP henceforth whenever the context is clear. We will also require that R_{PP} be efficiently verifiable (possibly given some trapdoor for PP) and efficiently samplable, where the sampling algorithm is denoted by SampR.

Intuitively, the relation R_{PP} corresponds to a hard search problem, that is, given a random u, it is hard to find s such $(u, s) \in \mathsf{R}_{\mathrm{PP}}$. More formally, we say that a binary relation R_{PP} is *one-way* if:

- with overwhelming probability over PP, for all u, there exists at most one s such that $(u, s) \in \mathsf{R}_{\mathrm{PP}}$; and
- there is an efficiently computable generator G such that $\mathsf{G}_{\mathrm{PP}}(s)$ is pseudorandom even against an adversary that gets PP, u and oracle access to R_{PP}, where $(u, s) \leftarrow_{\mathrm{R}} \mathsf{SampR}(\mathrm{PP})$. (We will also refer to G as extracting hard-core bits from s.)

For relations where computing s given u is hard on average, we may derive a generator G_{PP} with a one-bit output via the Goldreich-Levin hard-core bit $\mathsf{GL}(\cdot)$ [20] (with the randomness in PP). In many cases as we shall see shortly, we may derive a linear number of hard-core bits by either iterating a one-way permutation or relying on decisional assumptions. Next, we present one-way relations related to hardness of factoring and the Diffie-Hellman assumption.

Iterated Squaring. Fix a Blum integer $N = PQ$ for safe primes $P, Q \equiv 3 \pmod 4$ (such that $P = 2p+1$ and $Q = 2q+1$ for primes p, q). Following [26], we work over the cyclic group of signed quadratic residues, given by the quotient group $\mathbb{QR}^+_N := \mathbb{QR}_N / \pm 1$. $\mathbb{QR}^+_N$ is a cyclic group of order pq and is efficiently recognizable (by verifying that the Jacobi symbol is $+1$). In addition, the map $x \mapsto x^2$ is a permutation over $\mathbb{QR}^+_N$. Furthermore, assuming that factoring Blum integers are hard on average and that safe primes are dense, the family of permutations $x \mapsto x^2$ (indexed by N) acting on the groups $\mathbb{QR}^+_N$ is one-way.

In our constructions, the public parameter PP comprises (N, g), where N is a random $2k$-bit Blum integer and g is chosen uniformly from $\mathbb{QR}^+_N$. We will henceforth assume that g is a generator for $\mathbb{QR}^+_N$, which happens with probability $1 - O(1/\sqrt{N})$. We consider the relation:

$$\mathsf{R}^{\mathsf{isqr}}_{\mathrm{PP}} = \left\{ (u, s) \in \mathbb{QR}^+_N \times \mathbb{QR}^+_N : u = s^{2^k} \right\}$$

The associated sampling algorithm SampR picks a random $r \in [(N-1)/4]$ and outputs $(g^{2^k r}, g^r)$. Note that the output distribution is statistically close to the uniform distribution over $\mathbb{QR}^+_N$ whenever g is a generator. Using the Blum-Blum-Shub (BBS) pseudorandom generator [3], we may extract k hard-core bits from s that are pseudorandom even given u, that is:

$$\mathsf{G}^{\mathsf{bbs}}_{\mathrm{PP}}(s) := (\mathsf{lsb}_N(s), \mathsf{lsb}_N(s^2), \ldots, \mathsf{lsb}_N(s^{2^{k-1}}))$$

Diffie-Hellman Relation. We consider a family of groups $\mathbb{G}$ of prime order q. The public parameter PP is given by (g, g^α) for a random $g \leftarrow_R \mathbb{G}$ and a random $\alpha \leftarrow_R \mathbb{Z}_q$. We consider the Diffie-Hellman relation

$$\mathsf{R}_{\text{PP}}^{\mathsf{dh}} = \left\{(u, s) \in \mathbb{G} \times \mathbb{G} : s = u^\alpha\right\}$$

Note that $\mathsf{R}_{\text{PP}}^{\mathsf{dh}}$ is efficiently verifiable in bilinear groups (by computing a pairing) or if provided with α as a trapdoor. The associated sampling algorithm SampR picks a $r \leftarrow_R \mathbb{Z}_q$ and outputs $(g^r, g^{\alpha r})$. Next, we explain how to obtain hard-core bits for $\mathsf{R}_{\text{PP}}^{\mathsf{dh}}$ under various assumptions.

- The Strong DH assumption assumption [1] asserts that computing g^{ab} given (g, g^a, g^b) is hard on average, even given oracle access to $\mathsf{R}_{(g,g^a)}(\cdot, \cdot)$ (note that in bilinear groups, this is equivalent to CDH). Under Strong DH, we may extract a single hard-core bit from s using $\mathsf{GL}(s)$.
- The Bilinear DDH (BDDH) assumption [6] asserts that $e(g, g)^{abc}$ is pseudorandom given g, g^a, g^b, g^c where g, g^a, g^b, g^c are random elements of a bilinear group. Under BDDH, we may extract a linear number of hard-core bits from s using:

  $$\mathsf{G}_{\text{PP}}^{\mathsf{bddh}}(s) := e(s, g^\gamma) \quad \left(\Rightarrow \mathsf{G}_{\text{PP}}^{\mathsf{bddh}}(g^{\alpha r}) = e(g, g)^{\alpha\gamma r}\right)$$

  where PP is now given by (g, g^α, g^γ). In addition, we may improve efficiency by pre-computing the pairing and setting PP to be $(g, g^\alpha, e(g, g^\gamma))$ and computing $\mathsf{G}_{\text{PP}}^{\mathsf{bddh}}(g^r) := e(g, g^\gamma)^r$. This construction extends naturally to the Gap Hashed DH assumption [28].

Twin Diffie-Hellman Relation. As before, we consider a family of groups $\mathbb{G}$ of prime order q. The public parameter PP is now given by (g, g^α, g^β) for a random $g \leftarrow_R \mathbb{G}$ and random $\alpha, \beta \leftarrow_R \mathbb{Z}_q$. The Twin Diffie-Hellman relation [10] is given by:

$$\mathsf{R}_{\text{PP}}^{\mathsf{2dh}} = \left\{(u, (s_0, s_1)) \in \mathbb{G} \times \mathbb{G}^2 : (s_0, s_1) = (u^\alpha, u^\beta)\right\}$$

The associated sampling algorithm SampR picks $r \leftarrow_R \mathbb{Z}_q$ and outputs $(g^r, (g^{\alpha r}, g^{\beta r}))$. Note that $\mathsf{R}_{\text{PP}}^{\mathsf{2dh}}$ is efficiently verifiable given (α, β) as a trapdoor. Given random PP, g^r, the CDH assumption asserts that computing $g^{\alpha r}$ is hard on average, and the DDH assumption asserts that $g^{\alpha r}$ is pseudorandom. As shown in [10, Theorems 6 and 9], both problems remain hard even given oracle access to $\mathsf{R}_{\text{PP}}^{\mathsf{2dh}}(\cdot, (\cdot, \cdot))$ under the respective assumptions. This means that under CDH, we may extract a single hard-core bit from (s_0, s_1) by outputting $\mathsf{GL}(s_0)$ and under DDH, we may extract a linear number of hard-core bits by outputting s_0.

2.3 Extractable Hash Proofs

We consider a family of hash functions $\{\mathsf{H}_{\text{PK}}\}$ indexed by a public key PK. An *extractable hash proof system* associated with a one-way relation R_{PP} is a tuple of algorithms $(\mathsf{SetupExt}, \mathsf{SetupHash}, \mathsf{Pub}, \mathsf{Ext}, \mathsf{Priv})$ satisfying the following properties with overwhelming probability over PP:

(PUBLIC EVALUATION.) For all $(\text{PK}, \text{SK}) \leftarrow \mathsf{SetupExt}(\text{PP})$ and $(u, s) = \mathsf{SampR}(r)$: $\mathsf{Pub}(\text{PK}, r) = \mathsf{H}_{\text{PK}}(u)$.

(EXTRACTION MODE.) For all $(\text{PK}, \text{SK}) \leftarrow \mathsf{SetupExt}(\text{PP})$ and all (u, τ):

$$\tau = \mathsf{H}_{\text{PK}}(u) \quad \Longleftrightarrow \quad (u, \mathsf{Ext}(\text{SK}, u, \tau)) \in \mathsf{R}$$

(HASHING MODE.) For all $(\text{PK}, \text{SK}^*) \leftarrow \mathsf{SetupHash}(\text{PP})$ and all $(u, s) \in \mathsf{R}$,

$$\mathsf{Priv}(\text{SK}^*, u) = \mathsf{H}_{\text{PK}}(u)$$

(INDISTINGUISHABILITY.) The first outputs (namely PK) of $\mathsf{SetupHash}(\text{PP})$ and $\mathsf{SetupExt}(\text{PP})$ are statistically indistinguishable.

All-But-One Extractable Hash Proofs. For all of our applications, it is convenient to work with a richer abstraction, where as before, we consider a family of hash functions indexed by a public key PK, that takes a tag as an additional input. More formally, an *all-but-one (ABO) extractable hash proof system* is a tuple of algorithms $(\mathsf{SetupExt}, \mathsf{SetupABO}, \mathsf{Pub}, \mathsf{Ext}, \mathsf{Ext}^*, \mathsf{Priv})$ satisfying the following properties with overwhelming probability over PP:

(PUBLIC EVALUATION.) For all PK, TAG and $(u, s) = \mathsf{SampR}(r)$: $\mathsf{Pub}(\text{PK}, \text{TAG}, r) = \mathsf{H}_{\text{PK}}(\text{TAG}, u)$.

(EXTRACTION MODE.) For all $(\text{PK}, \text{SK}) \leftarrow \mathsf{SetupExt}(\text{PP})$ and all (TAG, u, τ):

$$\tau = \mathsf{H}_{\text{PK}}(\text{TAG}, u) \iff (u, \mathsf{Ext}(\text{SK}, \text{TAG}, u, \tau)) \in \mathsf{R}$$

(ALL-BUT-ONE MODE.) For all TAG^* and all $(\text{PK}, \text{SK}^*) \leftarrow \mathsf{SetupABO}(\text{PP}, \text{TAG}^*)$: for all $(u, s) \in \mathsf{R}$,

$$\mathsf{Priv}(\text{SK}^*, \text{TAG}^*, u) = \mathsf{H}_{\text{PK}}(\text{TAG}^*, u)$$

In addition, for all $\text{TAG} \neq \text{TAG}^*$ and all (u, τ):

$$\tau = \mathsf{H}_{\text{PK}}(\text{TAG}, u) \iff (u, \mathsf{Ext}^*(\text{SK}^*, \text{TAG}, u, \tau)) \in \mathsf{R}$$

(INDISTINGUISHABILITY.) For all TAG^*, the first outputs (namely PK) of $\mathsf{SetupABO}(\text{PP}, \text{TAG}^*)$ and $\mathsf{SetupExt}(\text{PP})$ are statistically indistinguishable.

2.4 Trapdoor Functions.

Informally, trapdoor functions are a family of functions $\{\mathsf{F}_{\text{FID}}\}$ that are easy to sample, compute and invert with trapdoor, and hard to invert without the trapdoor (in this work, we always assume that the functions are injective). In the tag-based setting, the function takes an additional input, namely the tag; also, the trapdoor is independent of the tag. A family of *adaptive trapdoor functions* [29] is one that remains one-way even if the adversary is given access to a inversion oracle, except the adversary cannot query the oracle on the same tag as that in the challenge.

Adaptive Trapdoor Relations. In this work, we consider a relaxation of the functionality guarantee for adaptive trapdoor functions, that is, instead of requiring that F_{FID} be efficiently computable, we only require that we can efficiently sample from the distribution $(s, \mathsf{F}_{\text{FID}}(\text{TAG}, s))$ for a random s given FID, TAG. More precisely, a family of (tag-based) *adaptive trapdoor relations* is given by a family of injective functions $\{\mathsf{F}_{\text{FID}}\}$ that satisfies the following properties:

(TRAPDOOR GENERATION.) There is an efficient randomized algorithm TDG that outputs a random (FID, TID).

(PUBLIC SAMPLING.) There is an efficient randomized algorithm PSamp that on input (FID, TAG), outputs $(s, \mathsf{F}_{\text{FID}}(\text{TAG}, s))$ for a random s.[1]

[1] This is essentially the only distinction from the adaptive trapdoor functions in [29]; there, they require that F_{FID} be efficiently computable.

(TRAPDOOR INVERSION.) There is an efficient algorithm TdInv such that for all $(\text{FID}, \text{TID}) \leftarrow \mathsf{TDG}$ and for all TAG, y, computes $\mathsf{TdInv}(\text{TID}, \text{TAG}, y) = \mathsf{F}_{\text{FID}}^{-1}(\text{TAG}, y)$.[2]

(ADAPTIVE ONE-WAYNESS.) For all efficient stateful adversaries $\mathcal{A}$, the following quantity is negligible:

$$\Pr\left[s = s' : \begin{array}{l} \text{TAG}^* \leftarrow \mathcal{A}(1^k); \\ (\text{FID}, \text{TID}) \leftarrow_{\text{R}} \mathsf{TDG}(1^k); \\ (s, y) \leftarrow_{\text{R}} \mathsf{PSamp}(\text{FID}, \text{TAG}^*); \\ s' \leftarrow \mathcal{A}^{\mathsf{F}_{\text{FID}}^{-1}(\cdot,\cdot)}(\text{FID}, y) \end{array}\right]$$

where $\mathcal{A}$ is allowed to query $\mathsf{F}_{\text{FID}}^{-1}(\cdot, \cdot)$ on any tag different from TAG^*.

It follows immediately from [29, Theorem 2] that adaptive trapdoor relations imply IND-CCA secure encryption.

3 Generic Constructions from Extractable Hash Proofs

In this section, we show that starting from an extractable hash proof, we may derive (1) a IND-CPA secure encryption scheme (as a simple warm-up exercise); (2) an ABO-extractable hash proof; (3) an ABO-extractable hash proof with multiple hard-core bits; and finally, (4) a IND-CCA secure KEM.

3.1 CPA-Secure Encryption

Starting from an extractable hash proof $(\mathsf{SetupExt}, \mathsf{SetupHash}, \mathsf{Pub}, \mathsf{Ext}, \mathsf{Ext}^*, \mathsf{Priv})$ for a one-way relation R_{PP} with an associated generator G_{PP}, we may derive a IND-CPA secure bit encryption scheme as follows:

- $\mathsf{Gen}(\text{PP})$: same as $\mathsf{SetupExt}(\text{PP})$.
- $\mathsf{Enc}(\text{PK}, b)$: sample $(u, s) := \mathsf{SampR}(r)$ and output $(u, \mathsf{Pub}(\text{PK}, r), \mathsf{G}(s) \oplus b)$.
- $\mathsf{Dec}(\text{SK}, (u, \tau, c))$: compute $s := \mathsf{Ext}(\text{SK}, u, \tau)$ and return $\mathsf{G}(s) \oplus c$.

Observe that correctness of the encryption scheme follows readily from correctness of the extraction mode. To establish IND-CPA security, we consider an intermediate game where we generate (PK, SK^*) using $\mathsf{SetupHash}(\text{PP})$ and computes $\mathsf{H}_{\text{PK}}(u)$ in the ciphertext using $\mathsf{Priv}(\text{SK}^*, u)$. Any adversary that can distinguish between encryptions of 0 and 1 in this game yields a distinguisher that given PP, u distinguishes $\mathsf{G}(s)$ from random.

3.2 From Extractable to ABO-Extractable

Starting from an extractable hash proof for a relation R_{PP}, we may derive a ABO-extractable hash proof $(\mathsf{SetupExt}', \mathsf{SetupABO}', \mathsf{Pub}', \mathsf{Ext}', \mathsf{Ext}'^*, \mathsf{Priv}')$ for the same relation and tag space $\{0,1\}^\ell$ via a construction analogous to those in [34, 15, 36, 38]:

[2] Since F_{FID} is not necessarily efficiently computable given FID, it is crucial here that we quantify over all y and that TdInv outputs $\perp$ if y does not have a pre-image under $\mathsf{F}_{\text{FID}}(\text{TAG}, \cdot)$. In our constructions, it will be the case F_{FID} is efficiently computable given TID.

- $\mathsf{SetupExt}'(\mathrm{PP})$: run $\mathsf{SetupExt}(\mathrm{PP})$ to obtain $(\mathrm{PK}_{i,0}, \mathrm{SK}_{i,0}), (\mathrm{PK}_{i,1}, \mathrm{SK}_{i,1})$, $i = 1, \ldots, \ell$; output $\widetilde{\mathrm{PK}} = (\mathrm{PK}_{i,0}, \mathrm{PK}_{i,1})_{i\in[\ell]}$ and $\widetilde{\mathrm{SK}} = (\mathrm{SK}_{i,0}, \mathrm{SK}_{i,1})_{i\in[\ell]}$.
- $\mathsf{Pub}'(\widetilde{\mathrm{PK}}, \mathrm{TAG}, r)$: parse $\mathrm{TAG} = (\mathrm{TAG}_1, \ldots, \mathrm{TAG}_\ell)$ and output $(\mathsf{Pub}(\mathrm{PK}_{i,\mathrm{TAG}_i}, r))_{i\in[\ell]}$.
- $\mathsf{Ext}'(\widetilde{\mathrm{SK}}, \mathrm{TAG}, u, (\tau_1, \ldots, \tau_\ell))$: compute $s_i := \mathsf{Ext}(\mathrm{SK}_{i,\mathrm{TAG}_i}, u, \tau_i)$ for $i = 1, \ldots, \ell$, and output s_1 if all ℓ values agree, and $\perp$ otherwise.
- $\mathsf{SetupABO}'(\mathrm{PP}, \mathrm{TAG}^*)$: run $\mathsf{SetupHash}(\mathrm{PP})$ to generate $(\mathrm{PK}_{i,\mathrm{TAG}_i^*}, \mathrm{SK}_{i,\mathrm{TAG}_i^*})$ and $\mathsf{SetupExt}(\mathrm{PP})$ to generate $(\mathrm{PK}_{i,1-\mathrm{TAG}_i^*}, \mathrm{SK}_{i,1-\mathrm{TAG}_i^*})$, for $i = 1, \ldots, \ell$; output $\widetilde{\mathrm{PK}} = (\mathrm{PK}_{i,0}, \mathrm{PK}_{i,1})_{i\in[\ell]}$ and $\widetilde{\mathrm{SK}}^* = (\mathrm{SK}_{i,0}, \mathrm{SK}_{i,1})_{i\in[\ell]}$.
- $\mathsf{Priv}'(\widetilde{\mathrm{PK}}, \mathrm{TAG}, u)$: output $(\mathsf{Priv}(\mathrm{SK}_{i,\mathrm{TAG}_i}, u))_{i\in[\ell]}$.
- $\mathsf{Ext}'^*(\widetilde{\mathrm{SK}}^*, \mathrm{TAG}, u, (\tau_1, \ldots, \tau_\ell))$: first, check that $\tau_i = \mathsf{Priv}(\mathrm{SK}_{i,\mathrm{TAG}_i}, u)$ for all i such that $\mathrm{TAG}_i^* = \mathrm{TAG}_i$ and if not, output $\perp$; next, compute $s_i := \mathsf{Ext}(\mathrm{SK}_{i,\mathrm{TAG}_i}, u, \tau_i)$ for all i such that $\mathrm{TAG}_i^* \neq \mathrm{TAG}_i$; output the common value if all these values agree and $\perp$ otherwise.

3.3 Obtaining Multiple Hard-Core Bits

Starting from an ABO-extractable hash proof for a relation R_{PP}, we may derive a ABO-extractable hash proof $(\mathsf{SetupExt}', \mathsf{SetupABO}', \mathsf{Pub}', \mathsf{Ext}', \mathsf{Ext}'^*, \mathsf{Priv}')$ for the k-wise direct product $\mathsf{R}_{\mathrm{PP}}^{\otimes k}$ of R_{PP}. This allows us to obtain more hard-core bits by using the k-wise direct product $\mathsf{G}_{\mathrm{PP}}^{\otimes k}$ of G_{PP}. The construction is as follows:

- $\mathsf{SampG}'(r_1, \ldots, r_k) = (\mathsf{SampG}(r_1), \ldots, \mathsf{SampG}(r_k))$
- $\mathsf{SetupExt}'$ and $\mathsf{SetupABO}'$ are the same as $\mathsf{SetupExt}$ and $\mathsf{SetupABO}$ respectively.
- $\mathsf{Pub}'(\mathrm{PK}, \mathrm{TAG}, (r_1, \ldots, r_k))$: output $(\mathsf{Pub}(\widetilde{\mathrm{PK}}, \mathrm{TAG}, r_i))_{i\in[k]}$.
- $\mathsf{Ext}'(\mathrm{SK}, \mathrm{TAG}, (u_1, \ldots, u_k), (\tau_1, \ldots, \tau_\ell))$: compute $s_i := \mathsf{Ext}(\mathrm{SK}, u_i, \tau_i)$ for $i = 1, \ldots, \ell$, and output $(s_1, \ldots, s_k)$.
- $\mathsf{Priv}'(\widetilde{\mathrm{PK}}, \mathrm{TAG}, (u_1, \ldots, u_k))$: output $(\mathsf{Priv}(\mathrm{SK}, u_i))_{i\in[\ell]}$.
- $\mathsf{Ext}'^*(\mathrm{SK}, \mathrm{TAG}, (u_1, \ldots, u_k), (\tau_1, \ldots, \tau_\ell))$: output $(\mathsf{Ext}(\mathrm{SK}, u_i, \tau_i))_{i\in[k]}$.

3.4 CCA-Secure Encryption

Starting from an ABO-extractable hash proof for a one-way relation R_{PP} along with a target collision-resistant hash function TCR, we may derive a IND-CCA KEM $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

- $\mathsf{Gen}(\mathrm{PP})$: same as $\mathsf{SetupExt}(\mathrm{PP})$.
- $\mathsf{Enc}(\mathrm{PK})$: sample $(u, s) := \mathsf{SampR}(r)$, compute $\mathrm{TAG} := \mathsf{TCR}(u), \tau := \mathsf{Pub}(\mathrm{PK}, \mathrm{TAG}, r)$, and return $(C, K) := ((u, \tau), \mathsf{G}(s))$.
- $\mathsf{Dec}(\mathrm{SK}, (u, \tau))$: compute $\mathrm{TAG} := \mathsf{TCR}(u)$ and $s := \mathsf{Ext}(\mathrm{SK}, \mathrm{TAG}, u, \tau)$; if $(u, s) \in \mathsf{R}_{\mathrm{PP}}$, return $\mathsf{G}(s)$, else return $\perp$.

We assume here that G_{PP} has linear output length; if not, we first apply the transformation in Section 3.3.

Theorem 1. *If* R_{PP} *is a one-way relation, then the above* KEM $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* IND-CCA *secure.*

Proof. Observe that correctness of the encryption scheme follows readily from correctness of the extraction mode. We proceed to establish IND-CCA security. In the following, we write $(u^*, s^*) = \mathsf{SampR}(r), C^* =$

$(u^*, \tau^*), K_0^*, K_1^*$ to denote the challenge ciphertext and keys chosen by the IND-CCA experiment, and we set TAG^* to denote the tag $\mathsf{TCR}(u^*)$ used in computing C^*. We proceed via a sequence of games. We start with Game 0, where the challenger proceeds like in the standard IND-CCA game (i.e, K_0^* is a real key and K_1^* is a random key) and end up with a game where both K_0^* and K_1^* are chosen uniformly at random. Then, we show that all games are indistinguishable under the assumption that $\mathsf{G}(s)$ is pseudorandom even given u.

GAME 1: ELIMINATING COLLISIONS. We replace the decapsulation mechanism Dec with Dec' that outputs $\perp$ on inputs (u, τ) such that $\mathsf{TCR}(u) = \text{TAG}^*$ but otherwise proceeds like Dec. We show that Games 0 and 1 are computationally indistinguishable, by arguing that Dec and Dec' essentially agree on all inputs (u, τ). We consider three cases:

- case 1: $\mathsf{TCR}(u) \neq \text{TAG}^*$. Here, Dec and Dec' agree by definition.
- case 2: $u \neq u^*$ but $\mathsf{TCR}(u) = \mathsf{TCR}(u^*) = \text{TAG}^*$. This only occurs with negligible probability, by target collision-resistance of TCR.
- case 3: $u = u^*$ but $\tau \neq \tau^*$. This means $\tau \neq \mathsf{H}_{\text{PK}}(\text{TAG}^*, u)$ and therefore Dec returns $\perp$ and agrees with Dec'.

GAME 2: DECAPSULATION WITH $\mathsf{SetupABO}$. We modify the IND-CCA experiment from Game 1, we generate the keys (PK, SK^*) using $\mathsf{SetupABO}$ instead of $\mathsf{SetupExt}$ and we answer decapsulation queries using SK^* instead of SK. More precisely, the IND-CCA experiment proceeds as follows:

$$
\begin{aligned}
&(u^*, s^*) \leftarrow \mathsf{SampR}(r); \text{TAG}^* := \mathsf{TCR}(u^*);\\
&(\text{PK}, \text{SK}^*) \leftarrow \mathsf{SetupABO}(\text{PP}, \text{TAG}^*);\\
&C^* := (u^*, \mathsf{Pub}(\text{PK}, \text{TAG}^*, r)); K_0^* := \mathsf{G}(s^*); K_1^* \leftarrow_{\text{R}} \{0,1\}^k;\\
&b \leftarrow_{\text{R}} \{0,1\};\\
&b' \leftarrow \mathcal{A}^{\mathsf{Dec}^*(\text{SK}^*, \cdot)}(\text{PK}, K_b^*, C^*)
\end{aligned}
$$

and where we replace $\mathsf{Dec}'(\text{SK}, \cdot)$ from Game 1 with $\mathsf{Dec}^*(\text{SK}^*, \cdot)$ which is defined as follows:

On input (u, τ): compute $\text{TAG} = \mathsf{TCR}(u)$;
- if $\text{TAG} = \text{TAG}^*$ return $\perp$.
- if $\text{TAG} \neq \text{TAG}^*$, compute $s = \mathsf{Ext}^*(\text{SK}^*, \text{TAG}, u, \tau)$. If $(u, s) \in \mathsf{R}_{\text{PP}}$, return $\mathsf{G}(s)$, else return $\perp$.

We claim a stronger statement, namely that for all r, the outputs of Games 1 and 2 are statistically indistinguishable. First, indistinguishability of the two modes imply that the view (PK, K_b^*, C^*) in Games 1 and 2 are statistically indistinguishable. As such, it suffices to show that for all PK, $\mathsf{Dec}'(\text{SK}, \cdot)$ and $\mathsf{Dec}^*(\text{SK}^*, \cdot)$ agree on all inputs (u, τ). Let s denote the unique value such that $(u, s) \in \mathsf{R}_{\text{PP}}$ (if no such s exists, then both Dec' and Dec^* return $\perp$) and let $\text{TAG} = \mathsf{TCR}(u)$. We consider three cases:

- case 1: $\text{TAG} = \text{TAG}^*$. Both Dec' and Dec^* output $\perp$ by definition.
- case 2: $\text{TAG} \neq \text{TAG}^*$. Here, Dec' always agrees with Dec by definition. By correctness of the extraction mode, $\mathsf{Ext}(\text{SK}, \text{TAG}^*, \tau)$ returns s iff $\tau = \mathsf{H}_{\text{PK}}(\text{TAG}, u)$. Similarly, by correctness of the all-but-one mode, $\mathsf{Ext}^*(\text{SK}^*, \text{TAG}^*, \tau)$ returns s iff $\tau = \mathsf{H}_{\text{PK}}(\text{TAG}, u)$. It follows that both Dec (and thus Dec') and Dec^* return $\mathsf{G}(s)$ if $\tau = \mathsf{H}_{\text{PK}}(\text{TAG}, u)$ and $\perp$ otherwise.

GAME 3: ENCAPSULATION WITH Priv. We compute $\mathsf{H}_{\text{PK}}(\text{TAG}^*, u^*)$ in C^* using Priv instead of Pub; that is, in the IND-CCA experiment from Game 2, we set

$$C^* := (u^*, \mathsf{Priv}(\text{SK}^*, \text{TAG}^*, u^*))$$

Games 2 and 3 are identically distributed by correctness of the all-but-one mode.

GAME 4: REPLACING $\mathsf{G}(s^*)$ WITH RANDOM. We generate K_0^* at random from $\{0,1\}^k$ instead of using $\mathsf{G}(s^*)$ (recall here that $(u^*, s^*) = \mathsf{SampR}(r)$). Observe that in Game 3, we never use knowledge of the witness s^* or randomness r associated with u^*. It follows from the pseudorandomness of G that Games 3 and 4 are computationally indistinguishable. Specifically, we may transform any distinguisher for Games 3 and 4 into a distinguisher K_0^* and $\mathsf{G}(s^*)$, given PP, u^* and oracle access to R_{PP} (the latter to simulate Dec^*).

We conclude by observing that in Game 4, both K_0^* and K_1^* are identically distributed, so the probability that $b' = b$ is exactly $1/2$. □

4 Instantiations from Hardness of Factoring

We present a simple extractable hash proof for the iterated squaring relation from Section 2.2, namely $\mathsf{R}_{\text{PP}}^{\mathsf{isqr}} := \left\{(u,s) \in \mathbb{QR}_N^+ \times \mathbb{QR}_N^+ : u = s^{2^k}\right\}$ where N is a Blum integer. We also present an efficient ABO-extractable hash proof for iterated squaring that avoids the linear blow-up incurred by the transformation in Section 3.2. Both of these extractable hash proofs appear implicitly in the Hofheinz-Kiltz cryptosystem [25, 26].

Applying the generic transformations in Section 3 to the first hash proof, we obtain (i) a simple factoring-based IND-CPA encryption scheme shown in Fig 2 where decryption does not require knowing the factorization of the modulus; and (ii) a simple factoring-based IND-CCA encryption shown in Fig 3. Applying the transformation in Section 3.4 to the efficient ABO-extractable hash proof, we recover the original Hofheinz-Kiltz cryptosystem.

4.1 A Simple Extractable Hash Proof

SYSTEM PARAMETERS. Here, $\text{PP} = (N, g), \text{PK} \in \mathbb{QR}_N^+$. and $\mathsf{SampR}(r) := (g^{2^k r}, g^r)$, where $r \in [(N-1)/4]$. We define

$$\mathsf{H}_{\text{PK}}(u) := (\text{PK} \cdot g)^r \text{ where } u = g^{2^k r}.$$

PUBLIC EVALUATION / EXTRACTION.

- $\mathsf{SetupExt}$: $\text{PK} = g^{2^k \cdot \text{SK}}, \text{SK} \leftarrow_{\text{R}} [(N-1)/4]$
- $\mathsf{Pub}(\text{PK}, r) = (\text{PK} \cdot g)^r$
- $\mathsf{Ext}(\text{SK}, u, \tau)$: output $\tau \cdot u^{-\text{SK}}$ if $u, \tau \in \mathbb{QR}_N^+$ and $\perp$ otherwise

Correctness of the extraction mode follows from the following simple calculation:

$$\tau = \mathsf{H}_{\text{PK}}(u) = s^{2^k \cdot \text{SK}+1} = u^{\text{SK}} \cdot s \iff \tau \cdot u^{-\text{SK}} = s$$

HASHING MODE.

– $\mathsf{SetupHash}$: $\mathrm{PK} = g^{2^k \cdot \mathrm{SK}^* - 1}$, $\mathrm{SK}^* \leftarrow_{\mathrm{R}} [(N-1)/4]$

– $\mathsf{Priv}(\mathrm{SK}^*, u) = u^{\mathrm{SK}^*}$

Correctness of the hashing mode follows from the observation that $2^k \cdot \mathrm{SK}^* = 2^k \cdot \mathrm{SK} + 1 \pmod{\phi(N)/4}$ and thus

$$\mathsf{H}_{\mathrm{PK}}(u) = (g^{2^k \cdot \mathrm{SK}+1})^r = (g^{2^k \cdot \mathrm{SK}^*})^r = u^{\mathrm{SK}^*}$$

To establish indistinguishability, observe that the distributions of PK in both modes are identical if we sample SK and SK^* uniformly at random from $\mathbb{Z}_{\phi(N)/4}$ instead of $[(N-1)/4]$; moreover, sampling SK and SK^* this way only changes the distributions by a negligible quantity.

Remark 1. This construction generalizes quite naturally to the RSA assumption (by replacing 2^k with an RSA exponent and $\mathbb{QR}_N^+$ with $\mathbb{Z}_N^*$); we omit the details since the ensuing construction is less efficient while relying on a stronger computational assumption.

4.2 Efficient ABO-Extractable Hash Proof

SYSTEM PARAMETERS. As before, $\mathrm{PP} = (N, g)$ and $\mathrm{PK} \in \mathbb{QR}_N^+$. The tag space is $\mathbb{Z}_{2^\ell}$ and $\mathsf{SampR}(r) := (g^{2^{k+\ell} r}, g^{2^\ell r})$, where $r \in [(N-1)/4]$. We define

$$\mathsf{H}_{\mathrm{PK}}(\mathrm{TAG}, u) := (\mathrm{PK} \cdot g^{\mathrm{TAG}})^r \text{ where } u = g^{2^{k+\ell} r}.$$

PUBLIC EVALUATION / EXTRACTION.

– $\mathsf{SetupExt}$: $\mathrm{PK} = g^{2^{k+\ell} \cdot \mathrm{SK}}$, $\mathrm{SK} \leftarrow_{\mathrm{R}} [(N-1)/4]$

– $\mathsf{Pub}(\mathrm{PK}, \mathrm{TAG}, r) = (\mathrm{PK} \cdot g^{\mathrm{TAG}})^r$

– $\mathsf{Ext}(\mathrm{SK}, \mathrm{TAG}, u, \tau)$: check that $u, \tau \in \mathbb{QR}_N^+$ and that $\tau^{2^{\ell+k}} = u^{\mathrm{TAG} + 2^{\ell+k} \cdot \mathrm{SK}}$ and output $\perp$ otherwise. Compute $a, b, c \in \mathbb{Z}$ such that $2^c = \gcd(\mathrm{TAG}, 2^{\ell+k}) = a \cdot \mathrm{TAG} + b 2^{\ell+k}$ and then output $(\tau^a \cdot u^{b - a \cdot \mathrm{SK}})^{2^{\ell - c}}$.

Correctness of the extraction mode follows from the calculations: write $u = s^{2^k}$ and $s = g^{2^\ell \cdot r}$. Then,

$$\tau = \mathsf{H}_{\mathrm{PK}}(\mathrm{TAG}, s^{2^k}) = g^{r \cdot (\mathrm{TAG} + 2^{k+\ell} \cdot \mathrm{SK})} \iff \tau^{2^{\ell+k}} = u^{\mathrm{TAG} + 2^{\ell+k} \cdot \mathrm{SK}}$$

Moreover, if this holds, we have that $g^{r \cdot \mathrm{TAG}} = \tau \cdot u^{-\mathrm{SK}}$ and together with $u = g^{r 2^{\ell+k}}$, we may compute $g^{r \cdot \gcd(\mathrm{TAG}, 2^{\ell+k})} = g^{r 2^c}$ from which we may compute $s = g^{r 2^\ell}$ since $\gcd(\mathrm{TAG}, 2^{\ell+k}) \leq 2^\ell$.

ABO-EXTRACTION MODE. We may write $2^{k+\ell} \cdot \mathrm{SK}^* = 2^{k+\ell} \cdot \mathrm{SK} + \mathrm{TAG}^*$

– $\mathsf{SetupABO}$: $\mathrm{PK} = g^{2^{k+\ell} \cdot \mathrm{SK}^* - \mathrm{TAG}^*}$, $\mathrm{SK}^* \leftarrow_{\mathrm{R}} [(N-1)/4]$

– $\mathsf{Priv}(\mathrm{SK}^*, u) = u^{\mathrm{SK}^*}$

– $\mathsf{Ext}^*(\mathrm{SK}^*, \mathrm{TAG}, u, \tau)$: check that $u, \tau \in \mathbb{QR}_N^+$ and that $\tau^{2^{\ell+k}} \neq u^{\mathrm{TAG} - \mathrm{TAG}^* + 2^{\ell+k} \cdot \mathrm{SK}^*}$ and output $\perp$ otherwise. Compute $a, b, c \in \mathbb{Z}$ such that $2^c = \gcd(\mathrm{TAG} - \mathrm{TAG}^*, 2^{\ell+k}) = a(\mathrm{TAG} - \mathrm{TAG}^*) + b 2^{\ell+k}$ and then output $(\tau^a \cdot u^{b - a \cdot \mathrm{SK}^*})^{2^{\ell - c}}$.

Correctness of the ABO-extraction mode is similar to that for the extraction mode.

$\mathsf{Gen}(\mathrm{PP})$, $\mathrm{PP} = (N, g)$:	$\mathsf{Enc}(\mathrm{PK}, b)$:	$\mathsf{Dec}(\mathrm{SK}, C)$:
$\mathrm{PK} := g^{2\mathrm{SK}}, \mathrm{SK} \leftarrow_{\mathrm{R}} [(N-1)/4]$	$r \leftarrow_{\mathrm{R}} [(N-1)/4]$	parse C as (u, τ, ψ)
return $(\mathrm{PK}, \mathrm{SK})$	return $(g^{2r}, (\mathrm{PK} \cdot g)^r, \mathsf{lsb}(g^r) \oplus b)$	return $\mathsf{lsb}(\tau \cdot u^{-\mathrm{SK}}) \cdot \psi$

Fig. 2. An IND-CPA bit encryption scheme based on hardness of factoring

$\mathsf{Gen}(\mathrm{PP})$, $\mathrm{PP} = (N, g)$:	$\mathsf{Enc}(\mathrm{PK})$:	$\mathsf{Dec}(\mathrm{SK}, C)$:
for $i = 1, \ldots, k$, for $b = 0, 1$:	$r \leftarrow_{\mathrm{R}} [(N-1)/4]$	parse C as $(u, \tau_1, \ldots, \tau_k)$
$\mathrm{SK}_{i,b} \leftarrow_{\mathrm{R}} [(N-1)/4]$	$u := g^{2^k r}, t := \mathsf{TCR}(u)$	check $u, \tau_1, \ldots, \tau_k \in \mathbb{QR}_N^+$
$\mathrm{PK}_{i,b} := g^{2^k \mathrm{SK}_{i,b}}$	for $i = 1, \ldots, k$:	$t := \mathsf{TCR}(u)$
$\mathrm{PK} := (\mathrm{PK}_{i,0}, \mathrm{PK}_{i,1})_{i \in [k]}$	$\tau_i := (\mathrm{PK}_{i,t_i} \cdot g)^r$	for $i = 1, \ldots, k$:
$\mathrm{SK} := (\mathrm{SK}_{i,0}, \mathrm{SK}_{i,1})_{i \in [k]}$	$C := (u, \tau_1, \ldots, \tau_k)$	check $\tau_i^{2^k} = u^{2^k \mathrm{SK}_{i,t_i}+1}$
return $(\mathrm{PK}, \mathrm{SK})$	return $(C, \mathsf{G}_{\mathrm{PP}}^{\mathsf{bbs}}(g^r))$	return $\mathsf{G}_{\mathrm{PP}}^{\mathsf{bbs}}(\tau_1 \cdot u^{-\mathrm{SK}_{1,t_1}})$

Fig. 3. An IND-CCA KEM based on hardness of factoring

5 Instantiations from Diffie-Hellman Assumptions

We present an ABO-extractable hash proof for the Diffie-Hellman relation from Section 2.2, namely $\mathsf{R}_{\mathrm{PP}}^{\mathsf{dh}} = \left\{(u, s) \in \mathbb{G} \times \mathbb{G} : s = u^{\alpha}\right\}$ where $\mathbb{G}$ is a group of order q and $\mathrm{PP} = (g, g^{\alpha})$. The construction is implicit in [5] and also [7, 27, 28, 23]. Applying the transformation in Section 3.4 to this hash proof system and the generator $\mathsf{G}_{\mathrm{PP}}^{\mathsf{bddh}}(\cdot)$, we obtain a variant of the BDDH-based IND-CCA KEM in [7, 27] (see Fig 4).

5.1 ABO-Extractable Hash Proof for the Diffie-Hellman Relation

SYSTEM PARAMETERS. Here, $\mathrm{PP} = (g, g^{\alpha}), \mathrm{SP} = \alpha$; the tag space is $\mathbb{Z}_q$; $\mathsf{SampR}(r) := (g^r, g^{\alpha r})$ where $r \in \mathbb{Z}_q$. We define

$$\mathsf{H}_{\mathrm{PK}}(u) := (g^{\alpha \cdot \mathrm{TAG}} \cdot \mathrm{PK})^r \text{ where } u = g^r.$$

PUBLIC EVALUATION / EXTRACTION.

- $\mathsf{SetupExt}$: $\mathrm{PK} = g^{\mathrm{SK}}, \mathrm{SK} \leftarrow_{\mathrm{R}} \mathbb{Z}_q$
- $\mathsf{Pub}(\mathrm{PK}, \mathrm{TAG}, r) = (g^{\alpha \cdot \mathrm{TAG}} \cdot \mathrm{PK})^r$
- $\mathsf{Ext}(\mathrm{SK}, \mathrm{TAG}, u, \tau) = (\tau \cdot u^{-\mathrm{SK}})^{\mathrm{TAG}^{-1}}$

Correctness of the extraction mode follows from the following simple calculation:

$$\tau = \mathsf{H}_{\mathrm{PK}}(\mathrm{TAG}, u) = u^{\alpha \cdot \mathrm{TAG} + \mathrm{SK}} \iff (\tau \cdot u^{-\mathrm{SK}})^{\mathrm{TAG}^{-1}} = u^{\alpha}$$

ABO-EXTRACTION MODE.

- $\mathsf{SetupABO}$: $\mathrm{PK} = g^{\mathrm{SK}^*} \cdot (g^{\alpha})^{-\mathrm{TAG}^*}, \mathrm{SK}^* \leftarrow_{\mathrm{R}} \mathbb{Z}_q$
- $\mathsf{Priv}(\mathrm{SK}^*, u) = u^{\mathrm{SK}^*}$
- $\mathsf{Ext}^*(\mathrm{SK}^*, \mathrm{TAG}, u, \tau) = (\tau \cdot u^{-\mathrm{SK}^*})^{(\mathrm{TAG} - \mathrm{TAG}^*)^{-1}}$

Correctness of the ABO-extraction mode follows from the fact that $\mathrm{SK}^* = \alpha \cdot \mathrm{TAG}^* + \mathrm{SK}$ and thus

$$\tau = \mathsf{H}_{\mathrm{PK}}(\mathrm{TAG}, u) = u^{\alpha(\mathrm{TAG} - \mathrm{TAG}^*)} \cdot u^{\mathrm{SK}^*} \iff (\tau \cdot u^{-\mathrm{SK}^*})^{(\mathrm{TAG} - \mathrm{TAG}^*)^{-1}} = u^{\alpha}$$

$\mathsf{Gen}(\text{PP})$, $\text{PP} = (g, e(g, g^\gamma))$:	$\mathsf{Enc}(\text{PK})$:	$\mathsf{Dec}(\text{SK}, C)$:
$\text{SK} := (\alpha, \widetilde{\text{SK}}) \leftarrow_{\text{R}} \mathbb{Z}_q^2$	$u := g^r, r \leftarrow_{\text{R}} \mathbb{Z}_q$	parse C as (u, τ)
$(h, \widetilde{\text{PK}}) := (g^\alpha, g^{\widetilde{\text{SK}}})$	$t := \mathsf{TCR}(u), \tau := (\widetilde{\text{PK}} \cdot h^t)^r$	$t := \mathsf{TCR}(u)$
$\text{PK} := (h, \widetilde{\text{PK}})$	$C := (u, \tau), K := e(g, g^\gamma)^r$	check $\tau = u^{\alpha t + \widetilde{\text{SK}}}$
return (PK, SK)	return (C, K)	return $e(u^\alpha, g^\gamma)$

Fig. 4. An IND-CCA KEM based on BDDH (variant of [7, 27])

$\mathsf{Gen}(\text{PP})$, $\text{PP} = (g, R)$:	$\mathsf{Enc}(\text{PK})$:	$\mathsf{Dec}(\text{SK}, C)$:
$(\alpha, \beta, \text{SK}_0, \text{SK}_1) \leftarrow_{\text{R}} \mathbb{Z}_q^4$	for $i = 1, \ldots, k$:	parse C as $(u_i, \tau_i^0, \tau_i^1)_{i \in [k]}$
$(h_0, h_1) := (g^\alpha, g^\beta)$	$\quad u_i := g^{r_i}, r_i \leftarrow_{\text{R}} \mathbb{Z}_q$	$t := \mathsf{TCR}(u_1, \ldots, u_k)$
$(\text{PK}_0, \text{PK}_1) := (g^{\text{SK}_0}, g^{\text{SK}_1})$	$t := \mathsf{TCR}(u_1, \ldots, u_k)$	for $i = 1, \ldots, k$:
$\text{PK} := (h_0, h_1, \text{PK}_0, \text{PK}_1)$	for $i = 1, \ldots, k$, for $b = 0, 1$:	$\quad$ check $\tau_i^0 = u_i^{\alpha t + \text{SK}_0}$
$\text{SK} := (\alpha, \beta, \text{SK}_0, \text{SK}_1)$	$\quad \tau_i^b := (\text{PK}_b \cdot h_b^t)^{r_i}$	$\quad$ check $\tau_i^1 = u_i^{\beta t + \text{SK}_1}$
return (PK, SK)	$C := (u_i, \tau_i^0, \tau_i^1)_{i \in [k]}$	return $(\mathsf{GL}_R(u_i^\alpha))_{i \in [k]}$
	$K := (\mathsf{GL}_R(h_0^{r_i}))_{i \in [k]}$	
	return (C, K)	

Fig. 5. An IND-CCA KEM based on CDH

5.2 Constructions for the Twin Diffie-Hellman Relation

The construction in the previous section extends naturally to yield an ABO-extractable hash proof for the twin Diffie-Hellman relation $\mathsf{R}_{\text{PP}}^{\mathsf{2dh}}$ (c.f. Section 2.2), by considering:

$$\mathsf{H}_{\text{PK}_0, \text{PK}_1}(u) := ((g^{\alpha \cdot \text{TAG}} \cdot \text{PK}_0)^r, (g^{\beta \cdot \text{TAG}} \cdot \text{PK}_1)^r) \text{ where } u = g^r.$$

We may then apply the transformations from Sections 3.3 and 3.4 to obtain a CDH-based IND-CCA KEM, shown in Fig 5. The public key comprises 5 group elements and the ciphertext comprises $O(k)$ group elements. We may also apply the transformation in Section 3.4 to obtain a DDH-based IND-CCA KEM, which is the same as that in [10, Section 6.2].

6 Adaptive Trapdoor Relations

Starting from an extractable hash proof $(\mathsf{SetupExt}, \mathsf{SetupABO}, \mathsf{Pub}, \mathsf{Ext}, \mathsf{Ext}^*, \mathsf{Priv})$ for a one-way relation R_{PP}, we may derive an adaptive trapdoor relation as follows:

- FID is (PP, PK) and for all $(u, s) \in \mathsf{R}_{\text{PP}}$, $\mathsf{F}_{\text{FID}}(\text{TAG}, s) := (u, \mathsf{H}_{\text{PK}}(\text{TAG}, u))$.
- $\mathsf{TDG}(1^k)$: computes $(\text{PK}, \text{SK}) \leftarrow \mathsf{SetupExt}(\text{PP})$ for a random PP and returns $\text{FID} := (\text{PP}, \text{PK})$ and $\text{TID} := \text{SK}$
- $\mathsf{PSamp}(\text{FID}, \text{TAG}; r)$: computes $(u, s) := \mathsf{SampR}(r), y := (u, \mathsf{Pub}(\text{PK}, \text{TAG}, r))$ and return (s, y).
- $\mathsf{TdInv}(\text{TID}, \text{TAG}, (u, \tau))$: computes $s := \mathsf{Ext}(\text{SK}, \text{TAG}, u)$ and returns s if $(u, s) \in \mathsf{R}_{\text{PP}}$ and $\perp$ otherwise.

$\mathsf{TDG}(\text{PP}), \text{PP} = (N, g)$:	$\mathsf{PSamp}(\text{FID}, \text{TAG}; r)$:	$\mathsf{TdInv}(\text{SK}, \text{TAG}, (u, \tau))$:
$\text{TID} \leftarrow_{\text{R}} [(N-1)/4]$	$(s, u) := (g^{2^\ell r}, g^{2^{k+\ell} r})$	check $u, \tau \in \mathbb{QR}_N^+$
$\text{FID} := g^{2^{k+\ell} \cdot \text{TID}}$	$\tau := (\text{FID} \cdot g^{\text{TAG}})^r$	check $\tau^{2^{\ell+k}} = u^{\text{TAG}+2^{\ell+k} \cdot \text{SK}}$
return (FID, TID)	return $(s, (u, \tau))$	find $a, b, c \in \mathbb{Z}$: $2^c = a \cdot \text{TAG} + b2^{\ell+k}$
		return $(\tau^a \cdot u^{b - a \cdot \text{SK}})^{2^{\ell-c}}$

Fig. 6. An adaptive trapdoor relation based on factoring

$\mathsf{TDG}(\text{PP}), \text{PP} = (g)$:	$\mathsf{PSamp}(\text{FID}, \text{TAG}; r)$:	$\mathsf{TdInv}(\text{SK}, \text{TAG}, (u, \tau))$:
$\text{TID} := (\alpha, \widetilde{\text{SK}}) \leftarrow_{\text{R}} \mathbb{Z}_q^2$	return $(h^r, (g^r, (\widetilde{\text{PK}} \cdot h^{\text{TAG}})^r))$	if $\tau = u^{\alpha \cdot \text{TAG} + \widetilde{\text{SK}}}$:
$\text{FID} := (h, \widetilde{\text{PK}}) := (g^\alpha, g^{\widetilde{\text{SK}}})$		return u^α, else $\perp$
return (FID, TID)		

Fig. 7. An adaptive trapdoor relation based on Strong DH

From an adaptive trapdoor relation, we may derive a one-bit IND-CCA encryption scheme following the construction in [29, Theorem 2], or a more efficient k-bit IND-CCA scheme by using the construction with multiple hard-core bits from Section 3.3.

Theorem 2. *If R_{PP} is a one-way relation, then the above construction yields an adaptive trapdoor relation.*

Proof (sketch). Trapdoor generation, public sampling and trapdoor inversion are straight-forward, so we only sketch the reduction for establishing adaptive one-wayness, which is very similar to that for our IND-CCA KEM in Section 3.4. Given an adversary $\mathcal{A}$ that breaks adaptive one-wayness with probability ϵ, we may construct an adversary $\mathcal{B}$ given (PP, u) and oracle access to R_{PP}, computes s with probability roughly ϵ:

- runs $\mathcal{A}(1^k)$ to get a tag TAG^*;
- computes $(\text{PK}, \text{SK}^*) \leftarrow_{\text{R}} \mathsf{SetupABO}(\text{PP}, \text{TAG}^*)$;
- computes $\text{FID} := (\text{PP}, \text{PK})$ and $\tau := \mathsf{Priv}(\text{SK}^*, \text{TAG}^*, u)$
- computes and outputs $s' \leftarrow \mathcal{A}(\text{FID}, (u, \tau))$, by simulating $\mathsf{F}_{\text{FID}}^{-1}(\cdot, \cdot)$ as follows:
  on input $(\text{TAG}, (u', \tau'))$ where $\text{TAG} \neq \text{TAG}^*$, compute $s' := \mathsf{Ext}^*(\text{SK}^*, \text{TAG}, u')$;
  output s' if $(u', s') \in \mathsf{R}_{\text{PP}}$ and $\perp$ otherwise.

It is easy to check that $\Pr[\mathcal{B}^{\mathsf{R}_{\text{PP}}(\cdot)}(\text{PP}, u) = s : (u, s) \leftarrow_{\text{R}} \mathsf{SampR}(\text{PP})] \approx \epsilon$, which contradicts the one-wayness of R_{PP}. □

Instantiating this construction with the ABO-extractable hash proofs in Sections 4.2 and 5.1, we derive the adaptive trapdoor relations shown in Fig 6 and 7, whose security are based on hardness of factoring and Strong DH respectively. By using the ABO-extractable hash proof in Section 5.2, we may also obtain an adaptive trapdoor relation based on CDH.

Acknowledgments. I am especially grateful to Eike Kiltz for pointing out that our framework applies to the constructions in [7, 27, 28], and to Payman Mohassel for sending me a draft of [29] and for helpful discussions. In addition, I would like to thank the anonymous Crypto 2010 reviewers for suggesting the name "adaptive trapdoor relations" and for many helpful comments.

References

[1] M. Abdalla, M. Bellare, and P. Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In *CT-RSA*, pages 143–158, 2001.

[2] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In *EUROCRYPT*, pages 128–146, 2005.

[3] L. Blum, M. Blum, and M. Shub. Comparison of two pseudo-random number generators. In *CRYPTO*, pages 61–78, 1982.

[4] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *STOC*, pages 103–112, 1988.

[5] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[6] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[7] X. Boyen, Q. Mei, and B. Waters. Direct chosen ciphertext security from identity-based techniques. In *ACM CCS*, pages 320–329, 2005.

[8] R. Canetti, S. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT*, pages 255–271, 2003.

[9] R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *EUROCRYPT*, pages 207–222, 2004.

[10] D. Cash, E. Kiltz, and V. Shoup. The Twin Diffie-Hellman problem and applications. *J. Cryptology*, 22(4):470–504, 2009.

[11] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO*, pages 13–25, 1998.

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

[13] R. Cramer, D. Hofheinz, and E. Kiltz. A twist on the Naor-Yung paradigm and its application to efficient CCA-secure encryption from hard search problems. In *TCC*, pages 146–164, 2010.

[14] A. De Santis and G. Persiano. Zero-knowledge proofs of knowledge without interaction. In *FOCS*, pages 427–436, 1992.

[15] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[16] U. Feige, D. Lapidot, and A. Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SICOMP*, 29(1):1–28, 1999.

[17] D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. In *PKC*, 2010. to appear.

[18] Y. Gertner, T. Malkin, and S. Myers. Towards a separation of semantic and CCA security for public key encryption. In *TCC*, pages 434–455, 2007.

[19] O. Goldreich. *Foundations of Cryptography: Volume II, Basic Applications*. Cambridge University Press, 2004.

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

[21] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[22] G. Hanaoka and K. Kurosawa. Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In *ASIACRYPT*, pages 308–325, 2008.

[23] K. Haralambiev, T. Jager, E. Kiltz, and V. Shoup. Simple and efficient public-key encryption from computational Diffie-Hellman in the standard model. In *PKC*, 2010. to appear.

[24] D. Hofheinz and E. Kiltz. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO*, pages 553–571, 2007.

[25] D. Hofheinz and E. Kiltz. Practical chosen ciphertext secure encryption from factoring. In *EUROCRYPT*, pages 313–332, 2009.

[26] D. Hofheinz and E. Kiltz. The group of signed quadratic residues and applications. In *CRYPTO*, pages 637–653, 2009.

[27] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, pages 581–600, 2006.

[28] E. Kiltz. Chosen-ciphertext secure key-encapsulation based on gap hashed diffie-hellman. In *Public Key Cryptography*, pages 282–297, 2007.

[29] E. Kiltz, P. Mohassel, and A. O'Neil. Adaptive trapdoor functions and chosen ciphertext security. In *EUROCRYPT*, 2010. to appear.

[30] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO*, pages 426–442, 2004.

[31] Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *J. Cryptology*, 19(3): 359–377, 2006.

[32] P. Mol and S. Yilek. Chosen-ciphertext security from slightly lossy trapdoor functions. In *PKC*, 2010. to appear.
[33] S. Myers and A. Shelat. Bit encryption is complete. In *FOCS*, pages 607–616, 2009.
[34] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.
[35] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342, 2009.
[36] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196, 2008.
[37] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO*, pages 433–444, 1991.
[38] A. Rosen and G. Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.
[39] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.
[40] V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In *EUROCRYPT*, pages 275–288, 2000.
[41] Y. Vahlis. Two is a crowd? a black-box separation of one-wayness and security under correlated inputs. In *TCC*, pages 165–182, 2010.