

# 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.

## 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

---

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

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  $(u, s) \in R$ . (For instance,  $s$  is the pre-image of  $u$  under a one-way permutation.) We consider a family of hash functions  $\{H_{PK}\}$  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  $(u, s) \in R$ . We denote this public evaluation algorithm by  $\text{Pub}(PK, r)$  and the hash value by  $H_{PK}(u)$ .

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  $\text{Pub}(PK, u)$  without knowing either  $s$  or  $r$ . Specifically, there is a private evaluation algorithm  $\text{Priv}$  such that for all  $(u, s) \in R$ ,  $\text{Priv}(SK^*, u) = H_{PK}(u)$ .
- 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  $\text{Ext}$ , such that for all  $u, \tau$ :  $\text{Ext}(SK, u, \tau)$  outputs  $s$  satisfying  $(u, s) \in R$  iff  $\tau = H_{PK}(u)$ . 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

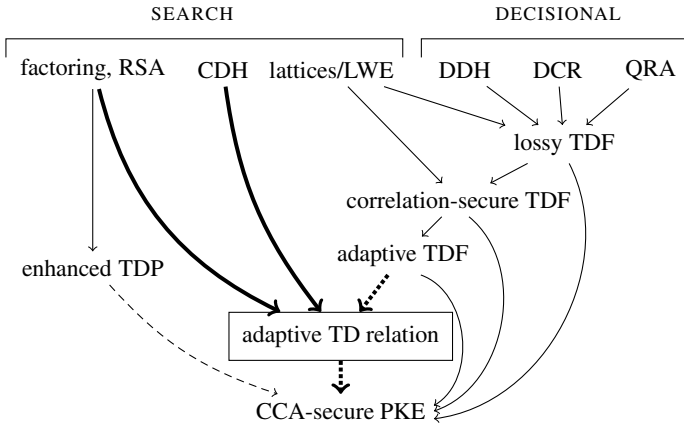
intermediate building block a somewhat richer cryptographic abstraction called *all-but-one extractable hash proofs* (which can be constructed generically from extractable hash proofs). The overall construction follows a variant of the Rackoff-Simon paradigm [37] (as opposed to the Naor-Yung double-encryption 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)  $(\text{Gen}, \text{Enc}, \text{Dec})$  with key-space  $\{0, 1\}^k$  consists three polynomial-time algorithms. Via  $(\text{PK}, \text{SK}) \leftarrow \text{Gen}(1^k)$  the randomized key-generation algorithm produces public/secret keys for security parameter  $1^k$ ; via  $(C, K) \leftarrow \text{Enc}(\text{PK})$ , the randomized encapsulation algorithm creates a uniformly distributed symmetric key  $K \in \{0, 1\}^k$ , together with a ciphertext  $C$ ; via  $K \leftarrow \text{Dec}(\text{SK}, C)$ , 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  $(C, K) \leftarrow \text{Enc}(\text{PK})$ , we have  $\Pr[\text{Dec}(\text{SK}, C) = K] = 1$ , where the probability is taken over the choice  $(\text{PK}, \text{SK}) \leftarrow \text{Gen}(1^k)$  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

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

with the restriction that  $\mathcal{A}$  is only allowed to query  $\text{Dec}(\text{SK}, \cdot)$  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  $\text{Adv}_{\text{KEM}}^{\mathcal{A}}(k)$  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  $R_{\text{pp}}$  indexed by a public parameter  $\text{pp}$ . We require that  $\text{pp}$  be efficiently samplable given a security parameter  $1^k$ , and assume that all algorithms are given  $\text{pp}$  as part of its input. We omit  $\text{pp}$  henceforth whenever the context is clear. We will also require that  $R_{\text{pp}}$  be efficiently verifiable (possibly given some trapdoor for  $\text{pp}$ ) and efficiently samplable, where the sampling algorithm is denoted by  $\text{SampR}$ .

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

- with overwhelming probability over  $\text{pp}$ , for all  $u$ , there exists at most one  $s$  such that  $(u, s) \in R_{\text{pp}}$ ; and
- there is an efficiently computable generator  $G$  such that  $G_{\text{pp}}(s)$  is pseudorandom even against an adversary that gets  $\text{pp}$ ,  $u$  and oracle access to  $R_{\text{pp}}$ , where  $(u, s) \leftarrow_{\text{R}} \text{SampR}(\text{pp})$ . (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  $G_{\text{pp}}$  with a one-bit output via the Goldreich-Levin hard-core bit  $\text{GL}(\cdot)$  [20] (with the randomness in  $\text{pp}$ ). 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  $\mathbb{QR}_N^{\pm} := \mathbb{QR}_N / \pm 1$ .  $\mathbb{QR}_N^{\pm}$  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  $\mathbb{QR}_N^{\pm}$ . 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  $\mathbb{QR}_N^{\pm}$  is one-way.

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

$$R_{\text{pp}}^{\text{isqr}} = \left\{ (u, s) \in \mathbb{QR}_N^{\pm} \times \mathbb{QR}_N^{\pm} : u = s^{2^k} \right\}$$

The associated sampling algorithm  $\text{SampR}$  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:

$$G_{\text{pp}}^{\text{bbs}}(s) := (\text{lsb}_N(s), \text{lsb}_N(s^2), \dots, \text{lsb}_N(s^{2^{k-1}}))$$

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

$$R_{\text{pp}}^{\text{dh}} = \left\{ (u, s) \in \mathbb{G} \times \mathbb{G} : s = u^\alpha \right\}$$

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

- The Strong DH assumption [1] asserts that computing  $g^{ab}$  given  $(g, g^a, g^b)$  is hard on average, even given oracle access to  $R_{(g, g^a)}(\cdot, \cdot)$  (note that in bilinear groups, this is equivalent to CDH). Under Strong DH, we may extract a single hard-core bit from  $s$  using  $\text{GL}(s)$ .
- 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:

$$G_{\text{pp}}^{\text{bddh}}(s) := e(s, g^\gamma) \quad \left( \Rightarrow G_{\text{pp}}^{\text{bddh}}(g^{\alpha r}) = e(g, g)^{\alpha \gamma r} \right)$$

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

Following the twinning framework [10], we will also consider the twin Diffie-Hellman relation given by:

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

where  $\text{PP}$  is given by  $(g, g^\alpha, g^\beta)$  for random  $\alpha, \beta \leftarrow_{\mathbb{R}} \mathbb{Z}_q$ , and  $\text{SampR}$  picks  $r \leftarrow_{\mathbb{R}} \mathbb{Z}_q$  and outputs  $(g^r, (g^{\alpha r}, g^{\beta r}))$ . As shown in [10, Theorem 9],  $\text{GL}(s_0)$  is a hard-core bit for the relation  $R_{\text{pp}}^{2\text{dh}}$  under CDH.

### 2.3 Extractable Hash Proofs

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

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

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

$$\tau = H_{\text{PK}}(u) \iff (u, \text{Ext}(\text{SK}, u, \tau)) \in R$$

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

$$\text{Priv}(\text{SK}^*, u) = H_{\text{PK}}(u)$$

(INDISTINGUISHABILITY.) The first outputs (namely  $\text{PK}$ ) of  $\text{SetupHash}(\text{PP})$  and  $\text{SetupExt}(\text{PP})$  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  $\text{PK}$ , that takes a tag as an additional input. More formally, an *all-but-one (ABO) extractable hash proof system* is a tuple of algorithms  $(\text{SetupExt}, \text{SetupABO}, \text{Pub}, \text{Ext}, \text{Ext}^*, \text{Priv})$  satisfying the following properties with overwhelming probability over  $\text{PP}$ :

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

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

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

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

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

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

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

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

## 2.4 Trapdoor Functions

Informally, trapdoor functions are a family of functions  $\{F_{\text{FID}}\}$  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 an 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  $F_{\text{FID}}$  be efficiently computable, we only require that we can efficiently sample from the



distribution  $(s, F_{\text{FID}}(\text{TAG}, s))$  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  $\{F_{\text{FID}}\}$  that satisfies the following properties:

(TRAPDOOR GENERATION.) There is an efficient randomized algorithm TDG that outputs a random  $(\text{FID}, \text{TID})$ .

(PUBLIC SAMPLING.) There is an efficient randomized algorithm PSamp that on input  $(\text{FID}, \text{TAG})$ , outputs  $(s, F_{\text{FID}}(\text{TAG}, s))$  for a random  $s$ .<sup>1</sup>

(TRAPDOOR INVERSION.) There is an efficient algorithm TdInv such that for all  $(\text{FID}, \text{TID}) \leftarrow \text{TDG}$  and for all TAG,  $y$ , computes  $\text{TdInv}(\text{TID}, \text{TAG}, y) = F_{\text{FID}}^{-1}(\text{TAG}, y)$ .<sup>2</sup>

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

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

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

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

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

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

<sup>2</sup> Since  $F_{\text{FID}}$  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  $F_{\text{FID}}(\text{TAG}, \cdot)$ . In our constructions, it will be the case  $F_{\text{FID}}$  is efficiently computable given TID.

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  $(\text{PK}, \text{SK}^*)$  using  $\text{SetupHash}(\text{PP})$  and computes  $H_{\text{PK}}(u)$  in the ciphertext using  $\text{Priv}(\text{SK}^*, u)$ . Any adversary that can distinguish between encryptions of 0 and 1 in this game yields a distinguisher that given  $\text{PP}, u$  distinguishes  $G(s)$  from random.

### 3.2 From Extractable to ABO-Extractable

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

- $\text{SetupExt}'(\text{PP})$ : run  $\text{SetupExt}(\text{PP})$  to obtain  $(\text{PK}_{i,0}, \text{SK}_{i,0}), (\text{PK}_{i,1}, \text{SK}_{i,1})$ ,  $i = 1, \dots, \ell$ ; output  $\widetilde{\text{PK}} = (\text{PK}_{i,0}, \text{PK}_{i,1})_{i \in [\ell]}$  and  $\widetilde{\text{SK}} = (\text{SK}_{i,0}, \text{SK}_{i,1})_{i \in [\ell]}$ .
- $\text{Pub}'(\widetilde{\text{PK}}, \text{TAG}, r)$ : parse  $\text{TAG} = (\text{TAG}_1, \dots, \text{TAG}_\ell)$  and output  $(\text{Pub}(\text{PK}_{i,\text{TAG}_i}, r))_{i \in [\ell]}$ .
- $\text{Ext}'(\widetilde{\text{SK}}, \text{TAG}, u, (\tau_1, \dots, \tau_\ell))$ : compute  $s_i := \text{Ext}(\text{SK}_{i,\text{TAG}_i}, u, \tau_i)$  for  $i = 1, \dots, \ell$ , and output  $s_1$  if all  $\ell$  values agree, and  $\perp$  otherwise.
- $\text{SetupABO}'(\text{PP}, \text{TAG}^*)$ : run  $\text{SetupHash}(\text{PP})$  to generate  $(\text{PK}_{i,\text{TAG}_i}^*, \text{SK}_{i,\text{TAG}_i}^*)$  and  $\text{SetupExt}(\text{PP})$  to generate  $(\text{PK}_{i,1-\text{TAG}_i}^*, \text{SK}_{i,1-\text{TAG}_i}^*)$ , for  $i = 1, \dots, \ell$ ; output  $\widetilde{\text{PK}} = (\text{PK}_{i,0}, \text{PK}_{i,1})_{i \in [\ell]}$  and  $\widetilde{\text{SK}}^* = (\text{SK}_{i,0}, \text{SK}_{i,1})_{i \in [\ell]}$ .
- $\text{Priv}'(\widetilde{\text{PK}}, \text{TAG}, u)$ : output  $(\text{Priv}(\text{SK}_{i,\text{TAG}_i}, u))_{i \in [\ell]}$ .
- $\text{Ext}'^*(\widetilde{\text{SK}}^*, \text{TAG}, u, (\tau_1, \dots, \tau_\ell))$ : first, check that  $\tau_i = \text{Priv}(\text{SK}_{i,\text{TAG}_i}, u)$  for all  $i$  such that  $\text{TAG}_i^* = \text{TAG}_i$  and if not, output  $\perp$ ; next, compute  $s_i := \text{Ext}(\text{SK}_{i,\text{TAG}_i}, u, \tau_i)$  for all  $i$  such that  $\text{TAG}_i^* \neq \text{TAG}_i$ ; 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  $R_{\text{PP}}$ , we may derive a ABO-extractable hash proof  $(\text{SetupExt}', \text{SetupABO}', \text{Pub}', \text{Ext}', \text{Ext}'^*, \text{Priv}')$  for the  $k$ -wise direct product  $R_{\text{PP}}^{\otimes k}$  of  $R_{\text{PP}}$ . This allows us to obtain more hard-core bits by using the  $k$ -wise direct product  $G_{\text{PP}}^{\otimes k}$  of  $G_{\text{PP}}$ . The construction is as follows:

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

### 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 (Gen, Enc, Dec) as follows:

- Gen(PP): same as SetupExt(PP).
- Enc(PK): sample  $(u, s) := \text{SampR}(r)$ , compute  $\text{TAG} := \text{TCR}(u)$ ,  $\tau := \text{Pub}(\text{PK}, \text{TAG}, r)$ , and return  $(C, K) := ((u, \tau), G(s))$ .
- Dec(SK,  $(u, \tau)$ ): compute  $\text{TAG} := \text{TCR}(u)$  and  $s := \text{Ext}(\text{SK}, \text{TAG}, u, \tau)$ ; if  $(u, s) \in R_{pp}$ , return  $G(s)$ , 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 (Gen, Enc, 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  $(u^*, s^*) = \text{SampR}(r)$ ,  $C^* = (u^*, \tau^*)$ ,  $K_0^*, K_1^*$  to denote the challenge ciphertext and keys chosen by the IND-CCA experiment, and we set  $\text{TAG}^*$  to denote the tag  $\text{TCR}(u^*)$  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  $G(s)$  is pseudorandom even given  $u$ .

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

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

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

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

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

On input  $(u, \tau)$ : compute  $\text{TAG} = \text{TCR}(u)$ ;

- if  $\text{TAG} = \text{TAG}^*$  return  $\perp$ .
- if  $\text{TAG} \neq \text{TAG}^*$ , compute  $s = \text{Ext}^*(\text{SK}^*, \text{TAG}, u, \tau)$ . If  $(u, s) \in \text{R}_{\text{PP}}$ , return  $G(s)$ , 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  $(\text{PK}, K_b^*, C^*)$  in Games 1 and 2 are statistically indistinguishable. As such, it suffices to show that for all  $\text{PK}$ ,  $\text{Dec}'(\text{SK}, \cdot)$  and  $\text{Dec}^*(\text{SK}^*, \cdot)$  agree on all inputs  $(u, \tau)$ . Let  $s$  denote the unique value such that  $(u, s) \in \text{R}_{\text{PP}}$  (if no such  $s$  exists, then both  $\text{Dec}'$  and  $\text{Dec}^*$  return  $\perp$ ) and let  $\text{TAG} = \text{TCR}(u)$ . We consider three cases:

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

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

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

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

**GAME 4: REPLACING  $G(s^*)$  WITH RANDOM.** We generate  $K_0^*$  at random from  $\{0, 1\}^k$  instead of using  $G(s^*)$  (recall here that  $(u^*, s^*) = \text{SampR}(r)$ ). 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  $G(s^*)$ , given  $\text{PP}$ ,  $u^*$  and oracle access to  $\text{R}_{\text{PP}}$  (the latter to simulate  $\text{Dec}^*$ ).

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  $1/2$ .  $\square$

## 4 Instantiations from Hardness of Factoring

We present a simple extractable hash proof for the iterated squaring relation from Section 2.2, namely  $R_{pp}^{\text{isqr}} := \{(u, s) \in \mathbb{QR}_N^+ \times \mathbb{QR}_N^+ : u = s^{2^k}\}$  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,  $PP = (N, g)$ ,  $PK \in \mathbb{QR}_N^+$ , and  $\text{SampR}(r) := (g^{2^k r}, g^r)$ , where  $r \in [(N-1)/4]$ . We define

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

PUBLIC EVALUATION / EXTRACTION.

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

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

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

HASHING MODE.

- SetupHash:  $PK = g^{2^k \cdot SK^* - 1}$ ,  $SK^* \leftarrow_R [(N-1)/4]$
- Priv( $SK^*, u$ ) =  $u^{SK^*}$

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

$$H_{PK}(u) = (g^{2^k \cdot SK + 1})^r = (g^{2^k \cdot SK^*})^r = u^{SK^*}$$

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.

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

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

## 4.2 Efficient ABO-Extractable Hash Proof

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

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

PUBLIC EVALUATION / EXTRACTION.

- SetupExt:  $\text{PK} = g^{2^{k+\ell} \cdot \text{SK}}, \text{SK} \leftarrow_{\text{R}} [(N-1)/4]$
- Pub( $\text{PK}, \text{TAG}, r$ ) =  $(\text{PK} \cdot g^{\text{TAG}})^r$
- Ext( $\text{SK}, \text{TAG}, u, \tau$ ): check that  $u, \tau \in \mathbb{QR}_N^+$  and that  $\tau^{2^{\ell+k}} = u^{\text{TAG}+2^{\ell+k} \cdot \text{SK}}$  and output  $\perp$  otherwise. Compute  $a, b, c \in \mathbb{Z}$  such that  $2^c = \gcd(\text{TAG}, 2^{\ell+k}) = a \cdot \text{TAG} + b2^{\ell+k}$  and then output  $(\tau^a \cdot u^{b-a \cdot \text{SK}})^{2^{\ell-c}}$ .

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

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

Moreover, if this holds, we have that  $g^{r \cdot \text{TAG}} = \tau \cdot u^{-\text{SK}}$  and together with  $u = g^{r2^{\ell+k}}$ , we may compute  $g^{r \cdot \gcd(\text{TAG}, 2^{\ell+k})} = g^{r2^c}$  from which we may compute  $s = g^{r2^\ell}$  since  $\gcd(\text{TAG}, 2^{\ell+k}) \leq 2^\ell$ .

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

- SetupABO:  $\text{PK} = g^{2^{k+\ell} \cdot \text{SK}^* - \text{TAG}^*}, \text{SK}^* \leftarrow_{\text{R}} [(N-1)/4]$
- Priv( $\text{SK}^*, u$ ) =  $u^{\text{SK}^*}$
- Ext\*( $\text{SK}^*, \text{TAG}, u, \tau$ ): check that  $u, \tau \in \mathbb{QR}_N^+$  and that  $\tau^{2^{\ell+k}} \neq u^{\text{TAG} - \text{TAG}^* + 2^{\ell+k} \cdot \text{SK}^*}$  and output  $\perp$  otherwise. Compute  $a, b, c \in \mathbb{Z}$  such that  $2^c = \gcd(\text{TAG} - \text{TAG}^*, 2^{\ell+k}) = a(\text{TAG} - \text{TAG}^*) + b2^{\ell+k}$  and then output  $(\tau^a \cdot u^{b-a \cdot \text{SK}^*})^{2^{\ell-c}}$ .

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

## 5 Instantiations from Diffie-Hellman Assumptions

We present an ABO-extractable hash proof for the Diffie-Hellman relation from Section 2.2, namely  $\mathbb{R}_{\text{pp}}^{\text{dh}} = \{(u, s) \in \mathbb{G} \times \mathbb{G} : s = u^\alpha\}$  where  $\mathbb{G}$  is a group of order  $q$  and  $\text{PP} = (g, g^\alpha)$ . 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  $\mathbb{G}_{\text{pp}}^{\text{bdh}}(\cdot)$ , we obtain a variant of the BDDH-based IND-CCA KEM in [7, 27] (see Fig 4).

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

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

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

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

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

PUBLIC EVALUATION / EXTRACTION.

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

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

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

ABO-EXTRACTION MODE.

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

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

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

## 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, by considering:

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

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.

Gen(PP), PP = $(g, e(g, g^\gamma))$ :	Enc(PK):	Dec(SK, C):
SK := $(\alpha, \widetilde{SK}_1) \leftarrow_{\mathbb{R}} \mathbb{Z}_q^2$	$u := g^r, r \leftarrow_{\mathbb{R}} \mathbb{Z}_q$	parse C as $(u, \tau)$
$(h, \widetilde{PK}) := (g^\alpha, g^{\widetilde{SK}_1})$	$t := \text{TCR}(u), \tau := (\widetilde{PK} \cdot h^t)^r$	$t := \text{TCR}(u)$
PK := $(h, \widetilde{PK})$	$C := (u, \tau), K := e(g, g^\gamma)^r$	check $\tau = u^{\alpha t + \widetilde{SK}_1}$
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])

Gen(PP), PP = $(g, R)$ :	Enc(PK):	Dec(SK, C):
$(\alpha, \beta, SK_0, SK_1) \leftarrow_{\mathbb{R}} \mathbb{Z}_q^4$	for $i = 1, \dots, k$ :	parse C as $(u_i, \tau_i^0, \tau_i^1)_{i \in [k]}$
$(h_0, h_1) := (g^\alpha, g^\beta)$	$u_i := g^{r_i}, r_i \leftarrow_{\mathbb{R}} \mathbb{Z}_q$	$t := \text{TCR}(u_1, \dots, u_k)$
$(PK_0, PK_1) := (g^{SK_0}, g^{SK_1})$	$t := \text{TCR}(u_1, \dots, u_k)$	for $i = 1, \dots, k$ :
PK := $(h_0, h_1, PK_0, PK_1)$	for $i = 1, \dots, k$ , for $b = 0, 1$ :	check $\tau_i^0 = u_i^{\alpha t + SK_0}$
SK := $(\alpha, \beta, SK_0, SK_1)$	$\tau_i^b := (PK_b \cdot h_b^t)^{r_i}$	check $\tau_i^1 = u_i^{\beta t + SK_1}$
return (PK, SK)	$C := (u_i, \tau_i^0, \tau_i^1)_{i \in [k]}$	return $(GL_R(u_i^{\alpha}))_{i \in [k]}$
	$K := (GL_R(h_0^{r_i}))_{i \in [k]}$	
	return $(C, K)$	

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

## 6 Adaptive Trapdoor Relations

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

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

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  $\epsilon$ , we may construct an adversary  $\mathcal{B}$  given  $(PP, u)$  and oracle access to  $R_{PP}$ , computes  $s$  with probability roughly  $\epsilon$ :



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

**Fig. 6.** An adaptive trapdoor relation based on factoring

<p>TDG(PP), PP = (g):</p> <p>TID := <math>(\alpha, \widetilde{\text{SK}}) \leftarrow_{\mathcal{R}} \mathbb{Z}_q^2</math></p> <p>FID := <math>(h, \widetilde{\text{PK}}) := (g^\alpha, g^{\widetilde{\text{SK}}})</math></p> <p>return (FID, TID)</p>	<p>PSamp(FID, TAG; r):</p> <p>return <math>(h^r, (g^r, (\widetilde{\text{PK}} \cdot h^{\text{TAG}})^r))</math></p>	<p>TdInv(SK, TAG, (u, <math>\tau</math>)):</p> <p>if <math>\tau = u^{\alpha \cdot \text{TAG} + \widetilde{\text{SK}}}</math>:</p> <p>return <math>u^\alpha</math>, else <math>\perp</math></p>
--	--	--

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

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

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

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 thank the anonymous Crypto 2010 reviewers for pointing out that our framework applies to the constructions in [7, 27, 28], for suggesting the name “adaptive trapdoor relations”, and for many helpful comments. I also thank Payman Mohassel for helpful discussions.

## References

[1] Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Heidelberg (2001)

[2] Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005)

- [3] Blum, L., Blum, M., Shub, M.: Comparison of two pseudo-random number generators. In: CRYPTO, pp. 61–78 (1982)
- [4] Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications. In: STOC, pp. 103–112 (1988)
- [5] Boneh, D., Boyen, X.: Efficient selective-id secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
- [6] Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. *SIAM J. Comput.* 32(3), 586–615 (2003)
- [7] Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM CCS, pp. 320–329 (2005)
- [8] Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
- [9] Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
- [10] Cash, D., Kiltz, E., Shoup, V.: The Twin Diffie-Hellman problem and applications. *J. Cryptology* 22(4), 470–504 (2009)
- [11] Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
- [12] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
- [13] Cramer, R., Hofheinz, D., Kiltz, E.: A twist on the Naor-Yung paradigm and its application to efficient CCA-secure encryption from hard search problems. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 146–164. Springer, Heidelberg (2010)
- [14] De Santis, A., Persiano, G.: Zero-knowledge proofs of knowledge without interaction. In: FOCS, pp. 427–436 (1992)
- [15] Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. *SIAM J. Comput.* 30(2), 391–437 (2000)
- [16] Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. *SICOMP* 29(1), 1–28 (1999)
- [17] Freeman, D.M., Goldreich, O., Kiltz, E., Rosen, A., Segev, G.: More constructions of lossy and correlation-secure trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 279–295. Springer, Heidelberg (2010)
- [18] Gertner, Y., Malkin, T., Myers, S.: Towards a separation of semantic and CCA security for public key encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 434–455. Springer, Heidelberg (2007)
- [19] Goldreich, O.: *Foundations of Cryptography: Volume II, Basic Applications*. Cambridge University Press, Cambridge (2004)
- [20] Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: STOC, pp. 25–32 (1989)
- [21] Goldwasser, S., Micali, S.: Probabilistic encryption. *J. Comput. Syst. Sci.* 28(2), 270–299 (1984)
- [22] Hanaoka, G., Kurosawa, K.: Efficient chosen ciphertext secure public key encryption under the computational Diffie-Hellman assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)

- [23] Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and efficient public-key encryption from computational Diffie-Hellman in the standard model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010)
- [24] Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
- [25] Hofheinz, D., Kiltz, E.: Practical chosen ciphertext secure encryption from factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
- [26] Hofheinz, D., Kiltz, E.: The group of signed quadratic residues and applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009)
- [27] Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
- [28] Kiltz, E.: Chosen-ciphertext secure key-encapsulation based on gap hashed diffie-hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)
- [29] Kiltz, E., Mohassel, P., O’Neil, A.: Adaptive trapdoor functions and chosen ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010)
- [30] Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
- [31] Lindell, Y.: A simpler construction of CCA2-secure public-key encryption under general assumptions. *J. Cryptology* 19(3), 359–377 (2006)
- [32] Mol, P., Yilek, S.: Chosen-ciphertext security from slightly lossy trapdoor functions. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 296–311. Springer, Heidelberg (2010)
- [33] Myers, S., Shelat, A.: Bit encryption is complete. In: FOCS, pp. 607–616 (2009)
- [34] Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC, pp. 427–437 (1990)
- [35] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC, pp. 333–342 (2009)
- [36] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC, pp. 187–196 (2008)
- [37] Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
- [38] Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009)
- [39] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS, pp. 543–553 (1999)
- [40] Shoup, V.: Using hash functions as a hedge against chosen ciphertext attack. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 275–288. Springer, Heidelberg (2000)
- [41] Vahlis, Y.: Two is a crowd? a black-box separation of one-wayness and security under correlated inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 165–182. Springer, Heidelberg (2010)