On the Limitations of the Spread of an IBE-to-PKE Transformation

Eike Kiltz

CWI Amsterdam
The Netherlands
kiltz@cwi.nl
http://kiltz.net

Abstract. By a generic transformation by Canetti, Halevi, and Katz (CHK) every Identity-based encryption (IBE) scheme implies a chosen-ciphertext secure public-key encryption (PKE) scheme. In the same work it is claimed that this transformation maps the two existing IBE schemes to two *new and different* chosen-ciphertext secure encryption schemes, each with individual advantages over the other.

In this work we reconsider one of the two specific instantiations of the CHK transformation (when applied to the "second Boneh/Boyen IBE scheme"). We demonstrate that by applying further simplifications the resulting scheme can be proven secure under a weaker assumption than the underlying IBE scheme.

Surprisingly, our simplified scheme nearly converges to a recent encryption scheme due to Boyen, Mei, and Waters which itself was obtained from the other specific instantiation of the CHK transformation (when applied to the "first Boneh/Boyen IBE scheme"). We find this particularly interesting since the two underlying IBE schemes are completely different.

The bottom line of this paper is that the claim made by Canetti, Halevi, and Katz needs to be reformulated to: the CHK transformation maps the two known IBE schemes to nearly one single encryption scheme.

1 Introduction

Chosen-ciphertext secure encryption schemes. One of the main fields of interest in cryptography is the design and the analysis of the security of encryption schemes in the public-key setting. In this work we consider such schemes for which one can provide theoretical proofs of security (without relying on heuristics such as the random oracle), but which are also efficient and practical.

The notion of chosen-ciphertext security was introduced by Naor and Yung [13] and developed by Rackoff and Simon [14], and Dolev, Dwork, and Naor [9]. In a chosen ciphertext attack, the adversary is given access to a decryption oracle that allows him to obtain the decryptions of ciphertexts of his choosing. Intuitively, security in this setting means that an adversary obtains (effectively) no information about encrypted messages, provided the corresponding ciphertexts are never submitted to the decryption oracle. For different reasons, the notion of chosen-ciphertext security has emerged as the "right" notion of security for encryption schemes.

As an example of an encryption scheme that meets this strong security property in the standard model we have the scheme from Cramer and Shoup [7, 8] which was recently improved by Kurosawa and Desmedt [11]. Until 2004 the Cramer-Shoup scheme and its variants remained basically the only practical schemes with such strong security properties that could be proved secure in the standard model (under a reasonable complexity-theoretic assumption).

FROM IDENTITY-BASED ENCRYPTION TO CHOSEN-CIPHERTEXT SECURE ENCRYPTION. One of the recent celebrated applications of *identity-based encryption* (IBE) is the work due to Canetti, Halevi, and Katz [6, 2] showing an elegant black-box transformation from any IBE (plus a one-time signature) into an encryption scheme without giving up its efficiency. We will refer to this as the *CHK transformation*. If the IBE scheme is weakly (selective-identity) chosen-plaintext secure then the resulting encryption scheme is chosen-ciphertext secure. Efficient constructions of IBE schemes in the standard model were recently developed by Boneh and Boyen [1] so the CHK transformation provides further alternative instances of chosen-ciphertext secure encryption schemes in the standard model.

Boneh and Katz [4] later improve the efficiency of the CHK transformation by basically replacing the one-time signature by a message authentication code (MAC). The latter BK transformation results in shorter ciphertexts and more efficient encryption/decryption.

SPECIFIC INSTANTIATIONS OF THE CHK TRANSFORMATION. Until now there are only two different identity-based encryption schemes known, both due to Boneh and Boyen [1]. The CHK transformation maps each individual IBE scheme to a new chosen-ciphertext secure encryption scheme [6]. In particular, in Chapter 7 of [2] the following two encryption schemes are proposed:

1. IBE-to-PKE[BB1]: the first Boneh/Boyen IBE scheme [1] plugged into the CHK-transformation
2. IBE-to-PKE[BB2]: the second Boneh/Boyen IBE scheme [1] plugged into the CHK-transformation

It is claimed in [6, 4, 2] that the two encryption schemes have different properties. In particular, the second scheme offers more efficient decryption while relying on a stronger assumption.

REVISITING THE IBE-TO-PKE[BB1] SCHEME. Boyen, Mei, and Waters [5] recently revisited the IBE-to-PKE[BB1] scheme, i.e. the encryption scheme obtained from the CHK transformation instantiated with the first IBE scheme from [1]. By avoiding the CHK transformation they show how to make the resulting scheme more efficient in terms of computational time and ciphertext expansion. In particular, they come up with a chosen-ciphertext secure encryption scheme with security based on the *Bilinear Decisional Diffie-Hellman* (BDDH) assumption in the standard model.

1.1 Our Results

Revisiting the IBE-to-PKE[BB2] scheme. In this work we reconsider the IBE-to-PKE[BB2] scheme, i.e. the encryption scheme obtained by the CHK-transformation instantiated with the second IBE scheme from Boneh and Boyen. Similar to the work from [5] we obtain a direct construction avoiding the CHK transformation. The resulting scheme is again simple and practical.

We can prove security of the resulting encryption scheme with respect to a weaker assumption than the security assumption needed for the IBE scheme. In particular, our scheme can be proved secure under the new *square Bilinear Decisional Diffie-Hellman* (square-BDDH) assumption, whereas the original IBE scheme can only be proved secure under the *q-Bilinear Decisional Diffie-Hellman* (q-BDDHI) assumption.[1] (We stress that unfortunately our results do not imply that the underlying IBE scheme can be proved secure under this weaker assumption.)

Comparison with the encryption scheme from Boyen, Mei, and Waters. Surprisingly, our simplified IBE-to-PKE[BB2] encryption scheme turns out to be (nearly) equivalent to the encryption scheme from Boyen, Mei, and Waters [5] which itself was a simplification of the IBE-to-PKE[BB1] scheme.

Our **main result** can be formulated as follows: In contrast to what was claimed in [6, 4, 2] for the two different IBE schemes BB1 and BB2, we have

$$\text{IBE-to-PKE[BB1]} \approx \text{IBE-to-PKE[BB2]}\,,$$

where "$\approx$" reads "nearly converges to" and will be further explained below. In other words, the CHK IBE-to-PKE transformation does not seem to spread the IBE schemes well over all encryption schemes, i.e. the transformation maps the two different IBE schemes from Boneh and Boyen to nearly the same encryption scheme.

We stress that the equivalence is not obtained by "simplifying away" all possible differences between the two schemes. In fact, the "core" of the two schemes is the same and already the raw schemes IBE-to-PKE[BB1] and IBE-to-PKE[BB2] can be shown to be equivalent by removing the unnecessary overhead of the two respective decryption algorithms.

We now explain the meaning of the above "$\approx$". There is only a small difference between the two simplified schemes "hidden" in the respective key generation algorithms. Intuitively, in the BMW construction key generation involves the generation of one more independent random element (let's call it y), whereas our scheme "recycles" the randomness. More precisely, this value y contains some redundant information and therefore depends on some other element from the key.

Complexity Theoretic Assumptions. We study the relations between all mentioned assumptions, in particular showing the (assumption-wise) hierarchy q-BDDHI (for any $q \geq 1$) implies square-BDDH implies BDDH.

[1] Here q is an upper bound on the decryption queries made by an adversary attacking the choosen-ciphertext security of the scheme.

Discussion. We study the spread of the CHK transformation, i.e. how well the CHK transformation spreads different IBE schemes over the set of all encryption schemes. Our results indicate that the CHK transformation maps the two different IBE schemes to one single encryption scheme. Unfortunately these two IBE schemes are the only IBE schemes we know until today.

In light of the number of different encryption schemes secure against chosen-ciphertext attacks in the standard model the implication of our result is purely destructive. Due to its similarities we propose to "remove" the IBE-to-PKE[BB2] scheme from our toolbox of *different* practical encryption schemes: instead of two we only get one new scheme from identity-based techniques.

From a theoretical side we find it interesting that two completely different identity-based encryption schemes finally lead to very similar encryption schemes after applying the CHK transformation and some simplifications. Again we stress that this does not imply that the two different IBE schemes from [1] also converge to one (and there are reasons that they don't).

Presentation. To simplify our presentation all schemes will be described as key encapsulation mechanisms rather than full public-key encryption schemes. We remark that since a secure key encapsulation mechanism plus a secure symmetric encryption scheme implies secure public-key encryption this is a more general concept.

In Section 2 we formally define the concept of a key encapsulation mechanism. Next, in Section 3 we state all relevant complexity-theoretic assumptions and classify them by their strength. The two schemes, the original one by Canetti, Halevi, and Katz, and our proposed simplification are presented in Section 4. We conclude this paper with an efficiency comparison of the two schemes in Section 5.

2 Notation and Definitions

If x is a string, then $|x|$ denotes its length, while if S is a set then $|S|$ denotes its size. If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If S is a set then $s \stackrel{\$}{\leftarrow} S$ denotes the operation of picking an element s from S uniformly at random. We write $\mathcal{A}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and by $z \stackrel{\$}{\leftarrow} \mathcal{A}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and letting z be the output. We write $\mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ to indicate that $\mathcal{A}$ is an algorithm with inputs $x, y, \ldots$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$ and by $z \stackrel{\$}{\leftarrow} \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ we denote the operation of running $\mathcal{A}$ with inputs $(x, y, \ldots)$ and access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, and letting z be the output.

We now formally introduce the notions of a key-encapsulation mechanism together with a security definition.

2.1 Public Key Encapsulation Schemes

A *public-key encapsulation mechanism* (KEM) $\mathcal{KEM} = (\mathsf{KEMkg}, \mathsf{KEMencaps}, \mathsf{KEMdecaps})$ with key-space $\mathsf{KeySp}(k)$ consists of three polynomial-time algorithms. Via $(pk, sk) \xleftarrow{\$} \mathsf{KEMkg}(1^k)$ the randomized key-generation algorithm produces keys for security parameter $k \in \mathbb{N}$; via $(K, C) \xleftarrow{\$} \mathsf{KEMencaps}(pk)$ a key $K \in \mathsf{KeySp}(k)$ together with a corresponding ciphertext C is created; via $K \leftarrow \mathsf{KEMdecaps}(sk, C)$ the possessor of secret key sk decrypts ciphertext C to get back a key. For consistency, we require that for all $k \in \mathbb{N}$, and all $(K, C) \xleftarrow{\$} \mathsf{KEMencaps}(pk)$ we have $\Pr[\mathsf{KEMdecaps}(sk, C) = K] = 1$, where the probability is taken over the choice of $(pk, sk) \xleftarrow{\$} \mathsf{KEMkg}(1^k)$, and the coins of all the algorithms in the expression above.

Formally, we associate with an adversary $\mathcal{A}$ the following experiment:

Experiment $\mathbf{Exp}^{kem\text{-}\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}}(k)$

$(pk, sk) \xleftarrow{\$} \mathsf{KEMkg}(1^k)$

$K_0^* \xleftarrow{\$} \mathsf{KeySp}(k)$; $(K_1^*, C^*) \xleftarrow{\$} \mathsf{KEMencaps}(pk)$

$\delta \xleftarrow{\$} \{0,1\}$

$\delta' \xleftarrow{\$} \mathcal{A}^{\mathsf{Dec}}(pk, K_\delta^*, C^*)$

If $\delta \neq \delta'$ then return 0 else return 1

where the oracle $\mathsf{Dec}(C)$ returns $K \xleftarrow{\$} \mathsf{KEMdecaps}(sk, C)$ with the restriction that $\mathcal{A}$ is not allowed to query oracle $\mathsf{Dec}(\cdot)$ for the target ciphertext C^*. We define the advantage of $\mathcal{A}$ in the experiment as

$$\mathbf{Adv}^{kem\text{-}\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}}(k) \;=\; \left| \Pr\left[\mathbf{Exp}^{kem\text{-}\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}}(k) = 1 \right] - \frac{1}{2} \right| \;.$$

A KEM scheme $\mathcal{KEM}$ is said to be *secure against adaptively-chosen ciphertext attacks* if the advantage function $\mathbf{Adv}^{kem\text{-}\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}}(k)$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

2.2 Target Collision Resistant Hash Functions

Let $(\mathsf{CR}_s)_{s \in S}$ be a family of hash functions for security parameter k and with seed $s \in S = S(k)$. $\mathcal{F}$ is said to be *collision resistant* if, for a hash function $\mathsf{CR} = \mathsf{CR}_s$ (where the seed is chosen at random from S), it is infeasible for any polynomial-time adversary to find two distinct values $x \neq y$ such that $\mathsf{CR}(x) = \mathsf{CR}(y)$.

A weaker notion is that of *target collision resistant hash functions*. Here it should be infeasible for a polynomial-time adversary to find, given a randomly chosen element x and a randomly drawn hash function $\mathsf{TCR} = \mathsf{TCR}_s$, a distinct element $y \neq x$ such that $\mathsf{TCR}(x) = \mathsf{TCR}(y)$. (In collision resistant hash functions the value x may be chosen by the adversary.) Such hash functions are also called *universal one-way hash functions* [12] and can be built from arbitrary one-way functions [12, 15]. We define

$$\mathbf{Adv}^{\mathrm{hash\text{-}tcr}}_{\mathsf{TCR},\mathcal{A}}(k) = \Pr[\mathcal{A} \text{ finds a collision}].$$

Hash function family TCR is said to be a *target collision resistant* if the advantage function $\mathbf{Adv}^{\mathrm{hash\text{-}tcr}}_{\mathsf{TCR},\mathcal{A}}$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}$.

3 Assumptions

In this section we give a parameter generating algorithm for bilinear groups and pairings and state our complexity assumptions.

3.1 Parameter generation algorithms for Bilinear Groups.

The scheme will be parameterized by a *bilinear parameter generator*. This is a polynomial-time algorithm $\mathsf{BilinGen}$ that on input 1^k returns the description of a multiplicative cyclic group $\mathbb{G}_1$ of prime order p, where $2^k < p < 2^{k+1}$, the description of a multiplicative cyclic group $\mathbb{G}_T$ of the same order, a random element g that generates $\mathbb{G}_1$, and a bilinear pairing $\hat{e}$: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. This bilinear pairing should be efficiently computable and satisfy the conditions below.

Bilinear: For all $g, h \in \mathbb{G}_1, x, y \in \mathbb{Z}$, $\hat{e}(g^x, h^y) = \hat{e}(g, h)^{xy}$
Non-degenerate: $\hat{e}(g, g) \neq 1_{\mathbb{G}_2}$

We use $\mathbb{G}_1^*$ to denote $\mathbb{G}_1 \setminus \{0\}$, i.e. the set of all group elements except the neutral element. Throughout the paper we use $\mathcal{BG} = (\mathbb{G}_1, \mathbb{G}_T, p, \hat{e}, g)$ (obtained by running $\mathsf{BilinGen}$) as shorthand for the description of bilinear groups.

3.2 The square BDDH assumption

Let $\mathcal{BG}$ be the description of bilinear groups and let $g \in \mathbb{G}_1$ be a random element from group $\mathbb{G}_1$. Consider the following problem: Given $(g, g^a, g^b, W) \in \mathbb{G}_1^3 \times \mathbb{G}_T$ as input, output yes if $W = \hat{e}(g, g)^{a^2 b}$ and no otherwise. More formally we associate with an adversary $\mathcal{B}$ the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{sbddh}}_{\mathsf{BilinGen},\mathcal{B}}(1^k)$
$\mathcal{BG} \xleftarrow{\$} \mathsf{BilinGen}(1^k)$
$a, b, w \xleftarrow{\$} \mathbb{Z}_p^*$
$\gamma \xleftarrow{\$} \{0, 1\}$; if $\gamma = 0$ then $W \leftarrow \hat{e}(g, g)^{a^2 b}$ else $W \leftarrow \hat{e}(g, g)^w$
$\gamma' \xleftarrow{\$} \mathcal{B}(1^k, \mathcal{BG}, g, g^a, g^b, W)$
If $\gamma \neq \gamma'$ then return 0 else return 1

We define the advantage of $\mathcal{B}$ in the above experiment as

$$\mathbf{Adv}^{\mathrm{sbddh}}_{\mathsf{BilinGen},\mathcal{B}}(k) = \left| \Pr\left[\mathbf{Exp}^{\mathrm{sbddh}}_{\mathsf{BilinGen},\mathcal{B}}(1^k) = 1 \right] - \frac{1}{2} \right| .$$

We say that the *Square Bilinear Decision Diffie-Hellman (square BDDH) assumption relative to generator* $\mathsf{BilinGen}$ holds if $\mathbf{Adv}^{\mathrm{sbddh}}_{\mathsf{BilinGen},\mathcal{B}}$ is a negligible function in k for all polynomial-time adversaries $\mathcal{B}$.

3.3 The BDDH assumption

Let $\mathcal{BG}$ be the description of bilinear groups and let $g \in \mathbb{G}_1$ be a random element from group $\mathbb{G}_1$. Consider the following problem formalized by Boneh and Franklin [3]: Given $(g, g^a, g^b, g^c, W) \in \mathbb{G}_1^4 \times \mathbb{G}_T$ as input, output yes if $W = \hat{e}(g, g)^{abc}$ and no otherwise. The corresponding BDDH assumption can be formalized the same way as the square BDDH assumption in the last paragraph.

3.4 The q-BDDHI assumption

Let $\mathcal{BG}$ be as above and let $z \in \mathbb{G}_1$ be a random element from group $\mathbb{G}_1$. For a function $q = q(k) \geq 1$ polynomial in the security parameter k consider the following problem introduced by Boneh and Boyen [1]: Given $(z, z^y, z^{(y^2)}, \ldots, z^{(y^q)}, W) \in \mathbb{G}_1^{q+1} \times \mathbb{G}_T$ as input, output yes if $W = \hat{e}(z,z)^{1/y}$ and no otherwise.

3.5 Relation between the Assumptions

The next lemma classifies the strength of the different assumptions we introduced. Here "A $\leq$ B" means that assumption B implies assumption A, i.e. assumption B is a stronger assumption than A.

Lemma 1. *BDDH* $\leq$ *square BDDH* $\leq$ 1-*BDDHI* $\leq$ 2-*BDDHI* ...

In partiuclar this means that square BDDH is a stronger assumption than BDDH, but weaker than q-BDDHI (for any $q \geq 1$). The simple proof of Lemma 1 is postponed until Appendix B.

4 Key Encapsulation based on the second Boneh/Boyen IBE scheme

In this section we revisit the encryption scheme from [6, 4] obtained by applying the CHK transformation to the second Boneh/Boyen IBE scheme from [1]. As already mentioned in the Introduction the scheme is presented as a key encapsulation mechanism (KEM) instead of an encryption scheme as in the original paper. After reminding the reader of the original scheme we then move on to present our simplifications.

For both schemes let the global system parameters be $\mathcal{BG} = (\mathbb{G}_1, \mathbb{G}_T, p, \hat{e}, g)$, a random bilinear group obtained by running $\mathsf{BilinGen}(1^k)$.

4.1 CHK2: the original scheme from [6]

In this construction, we use a one-time signature scheme $\mathcal{OTS} = (\mathsf{Skg}, \mathsf{Sign}, \mathsf{Vfy})$. The key generation algorithm Skg is run to obtain a random pair of verification/signing keys $(v, s) \stackrel{\$}{\leftarrow} \mathsf{Skg}(1^k)$; the signing key s is used to sign a message M to obtain a signature $\sigma \stackrel{\$}{\leftarrow} \mathsf{Sign}_s(M)$ on a message M; using the public verification key v, a signature σ can be verified by running $\mathsf{Vfy}_v(M, \sigma)$. We require that this scheme be secure in the sense of *strong unforgeability*, see [6] for exact definitions and constructions (details can be skipped here).

The key encapsulation mechanism proposed by Canetti, Halevi, and Katz [6] which we will denote by CHK2 is given in Fig. 1 (in order to simplify the comparison, compared to [6] we made some slight change of variables). It is straightforward to verify the correctness of the scheme. In terms of security the following theorem was derived in [6]:

$\mathsf{KEMkg}(1^k)$
$\quad x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$
$\quad h_1 \leftarrow g^{x_1}\ ;\ h_2 \leftarrow g^{x_2}\ ;\ z \leftarrow \hat{e}(h_2, h_2)$
$\quad pk \leftarrow (h_1, h_2, z)\ ;\ sk \leftarrow (x_1, x_2)$
$\quad$ Return (pk, sk)

$\mathsf{KEMencaps}(pk)$
$\quad r \xleftarrow{\$} \mathbb{Z}_p^*;\ c_1 \leftarrow g^r$
$\quad (v, s) \xleftarrow{\$} \mathsf{Skg}(1^k)\ ;\ c_2 \leftarrow h_1^r \cdot h_2^{v \cdot r}$
$\quad K \leftarrow z^r$
$\quad \sigma \xleftarrow{\$} \mathsf{Sign}_s(c_1 || c_2)$
$\quad C \leftarrow (c_1, c_2, v, \sigma)$
$\quad$ Return (K, C)

$\mathsf{KEMdecaps}(sk, C)$
$\quad$ Parse C as (c_1, c_2, v, σ)
$\quad$ If $\mathsf{Vfy}_v(c_1 || c_2, \sigma) = \mathtt{reject}$
$\quad\quad$ then return $\mathtt{reject}$.
$\quad$ Else
$\quad\quad r' \xleftarrow{\$} \mathbb{Z}_p$
$\quad\quad K \leftarrow \hat{e}\left(c_1^{r'} c_2, h_2^{1/(v + x_1/x_2 + r'/x_2)}\right)$

Fig. 1. The original CHK2 scheme.

Theorem 2. *Assuming the q-BDDHI assumption holds relative to the generator* BilinGen*,* $\mathcal{OTS}$ *is a strong, one-time signature scheme, then the KEM from Fig. 1 is chosen-ciphertext secure. Here* $q = q(k)$ *is an upper bound on the decapsulation queries made by an adversary attacking the scheme.*

4.2 CHK2': An equivalent Depcapsulation Algorithm

A closer inspection of the decapsulation algorithm of the CHK2 scheme from Fig. 1 shows that it *implicitly rejects* inconsistent ciphertexts (i.e., ciphertexts that were not obtained running the encapsulation algorithm with the correct public key) by returning a random session key in that case. Once consistency of the ciphertext is established, recovering the session key can be greatly improved.

For a value $v \in \mathbb{Z}_p$ we have

$$\begin{aligned} c_1^{x_1 + x_2 v} = c_2 &\Leftrightarrow \hat{e}(g, c_1^{x_1 + x_2 v}) = \hat{e}(g, c_2) \\ &\Leftrightarrow \hat{e}(g^{x_1 + x_2 v}, c_1) = \hat{e}(g, c_2) \\ &\Leftrightarrow \hat{e}(h_1 h_2^v, c_1) = \hat{e}(g, c_2). \end{aligned}$$

Therefore it can be publicly verified (using the public key only) if $c_1^{x_1 + x_2 v} = c_2$ by checking if $\hat{e}(h_1 h_2^v, c_1) = \hat{e}(g, c_2)$. A tuple (c_1, c_2) meeting this property is dubbed to be *consistent with* v. Note that any tuple (c_1, c_2) correctly generated by the encapsulation algorithm is always consistent with its verification key v. (A correctly generated ciphertext has the form $C = (c_1, c_2, v, \sigma) = (g^r, h_1^r \cdot h_2^{vr}, v, \sigma)$. Therefore $c_1^{x_1 + x_2 v} = (g^r)^{x_1 + x_2 v} = (g^{x_1})^r (g^{x_2 v})^r = h_1^r \cdot h_2^{rv}$.)

An equivalent way to compute the session key K, given that the signature was successfully verified, is as follows: First, a random key K is returned if (c_1, c_2) is not consistent with v, i.e. if $c_1^{x_1 + x_2 \cdot v} \neq c_2$ which can be checked as bescribed above. Otherwise, the key is recovered as $K = \hat{e}(h_2^{x_2}, c_1)$.

We claim that this decapsulation algorithm is equivalent to the one from CHK2 (Fig. 1). It is easy to verify that

$$\hat{e}(c_1^{r'} c_2, h_2^{1/(v+x_1/x_2+r'/x_2)}) = \hat{e}(h_2^{x_2}, c_1)^{\Delta(r')},$$

where $\Delta(r') = (r' + \log_{c_1} c_2)/(r' + x_1 + v \cdot x_2)$ is a random element from $\mathbb{Z}_p$ if $c_1^{x_1+x_2 v} \neq c_2$ (i.e., if (c_1, c_2) is not consistent with v) and $\Delta(r') = 1$ otherwise. We have seen that if (c_1, c_2) is consistent with v decapsulation computes the key K as

$$\begin{aligned} &= \hat{e}(h_2^{x_2}, g^r)^1 \\ &= \hat{e}(h_2, g^{x_2})^r \\ &= \hat{e}(h_2, h_2)^r = z^r \ , \end{aligned}$$

as the key computed in the encapsulation algorithm. This shows correctness.

We note that, equivalently, instead of returning a random key K the decapsulation algorithm could as well reject the ciphertext.

4.3 CHK2": our simplification

In this section we show how to avoid the one-time signature scheme by replacing it with a (determinstic) target collision resistant hash function applied to parts of the ciphertext. We note that the usage of the hash function is somewhat reminiscent of the Cramer/Shoup scheme [7].

Let $\mathsf{TCR} : \mathbb{G}_1 \to \mathbb{Z}_p$ be a target collision resistant hash function. Our simplification of the above construction is depicted in Fig. 2. Correctness of decapsulation follows from the correctness of the last scheme.

$\mathsf{KEMkg}(1^k)$
- $x_1, x_2 \xleftarrow{\$} \mathbb{Z}_p^*$
- $h_1 \leftarrow g^{x_1}$; $h_2 \leftarrow g^{x_2}$; $z \leftarrow \hat{e}(h_2, h_2)$
- $pk \leftarrow (h_1, h_2, z)$; $sk \leftarrow (x_1, x_2, \underline{y = h_2^{x_2}})$
- Return (pk, sk)

$\mathsf{KEMencaps}(pk)$	$\mathsf{KEMdecaps}(sk, C)$
$r \xleftarrow{\$} \mathbb{Z}_p^*$; $c_1 \leftarrow g^r$	Parse C as (c_1, c_2)
$v \leftarrow \mathsf{TCR}(c_1)$; $c_2 \leftarrow h_1^r \cdot h_2^{v \cdot r}$	$v \leftarrow \mathsf{TCR}(c_1)$
$K \leftarrow z^r$	If $c_1^{x_1+x_2 \cdot v} \neq c_2$ then `reject`
$C \leftarrow (c_1, c_2)$	Else $K \leftarrow \hat{e}(y, c_1)$
Return (C, K)	Return K

Fig. 2. CHK2": Our simplification of CHK2.

Let $C = (c_1, c_2)$ be an arbitrary ciphertext and let $v = \mathsf{TCR}(c_1)$. We call C *consistent* if it passes the verification check in the decapsulation algorithm, i.e. if $c_1^{x_1+x_2 v} = c_2$. By the discussion above we note that our KEM allows for public verification of the consistency of a ciphertext by testing if $\hat{e}(h_1 h_2^v, c_1) = \hat{e}(g, c_2)$. This public consistency check will play a crucial role in the proof of security. We note that the original CHK2 scheme from Fig. 1 already has a similar public verification property (using the one-time signature scheme).

4.4 Security

Theorem 3. *Assume* TCR *is a target collision resistant hash function. Under the square BDDH assumption relative to the generator* $\mathsf{BilinGen}$ *the KEM from Fig. 2 is secure against chosen-ciphertext attacks.*

The security reduction is tight. The proof of Theorem 3 is given in Appendix A. We will try to provide some intuition instead.

Security of the scheme CHK2'. Let $C^* = (c_1^*, c_2^*, v^*, \sigma^*)$ be the challenge ciphertext output by the simulator in the security experiment. It is clear that, without any decryption oracle queries, the value of the bit δ remains hidden to the adversary. This is so because (c_1^*, c_2^*) comes from a chosen-plaintext secure encryption scheme, v^* is independent of the message, and σ^* is the result of applying the one-time signing algorithm to $c_1^*||c_2^*$.

We claim that decryption oracle queries cannot further help the adversary in guessing the value of δ. Consider an arbitrary ciphertext query $(c_1, c_2, v, \sigma) \neq (c_1^*, c_2^*, v^*, \sigma^*)$ made by the adversary during the experiment. If $v = v^*$ then $(c_1, c_2, \sigma) \neq (c_1^*, c_2^*, \sigma^*)$ and the decryption oracle will answer `reject` since the adversary is unable to forge a new valid signature σ with respect to v^*. Now let $v \neq v^*$. Intuitively, a query with $v \neq v^*$ does not help the adversary since the underlying IBE scheme is *selective-identity secure*. In a nutshell, this IBE security property exactly translates to what we need here. I.e, any decryption query made for the "identity" v distinct from "target identity" v^* (which is is completely independent of the adversary's view until it sees the target ciphertext; therefore the simulator may as well choose v^* in the beginning of the experiment) does not help the adversary further. Details will be given in the proof.

Security of the scheme CHK2". To argue for security we again claim that decryption oracle queries cannot further help the adversary in guessing the value of δ. If $v \neq v^*$ we can still argue as in the CHK2' scheme. If $v = v^*$ then by the target collision resistance of TCR we may assume $c_1 = c_1^*$. In this case consistency implies $c_2^* = c_2$ and therefore $C^* = C$.

5 Comparison and Efficiency

5.1 Relation between CHK2 and CHK2"

In terms of functionality of the CHK2" scheme we note that the element $y = h_2^{x_2}$ is contained in the secret key for the sole reason of improving efficiency of decapsulation when recovering the key as $K = \hat{e}(h_2^{x_2}, c_1) = \hat{e}(y, c_1)$. Apart from that, key-generation is equivalent to the CHK2 scheme from Section 4.1.

The value y gives rise to a tradeoff between the length of the secret key and decryption speed. In particular, the secret value $y = h_2^{x_2}$ can always be reconstructed by the owner of the secret key on-line during decapsulation. This variant makes the secret-key one element shorter with the drawback of one more exponentiation during decapsulation.

Every IBE scheme can be viewed as a more general concept, a *tag-based encryption* (TBE) scheme. It was recently shown [10] that TBE is already sufficient for the CHK transformation to obtain a chosen-ciphertext secure encryption scheme. We note that the TBE scheme implied by the BB2 IBE scheme already can be proved secure under the (weaker) square BDDH assumption meaning that the original CHK2 scheme is also secure under square BDDH. To be more precise, in the transformation chain IBE $\Rightarrow$ TBE $\Rightarrow$ "chosen-ciphertext secure encryption", the security improvement is already obtained after the first implication.

5.2 Relation between CHK1" and CHK2"

As we instantly notice, our CHK2" scheme from Section 4.3 is very similar to the scheme from Boyen, Mei, and Waters [5] which we will refer to as CHK1". (For completeness we remind the reader of CHK1" in Appendix C.) Let us point out the differences.

The only difference is that the key generation algorithm of CHK1" chooses (in an information theoretical sense) a new independent secret value y. In contrast, our scheme derives the secret value $y = h_2^{x_2}$ from h_2 and x_2, i.e. the secret key contains some redundant information. (The sole reason the value y is included in our scheme is to save one exponentiation in the decapsulation algorithm.) This dependence of y is the reason why we need a stronger assumption to prove security. Performance of the two KEMs is exactly the same.

5.3 Relation between CHK1 and CHK2

We denote by CHK1 the scheme obtained by plugging the first Boneh/Boyen IBE scheme into the CHK transformation. We note that the CHK1 scheme (which for completeness is also presented in Appendix C) is already equivalent to the CHK2 scheme.

Similar to our scheme CHK2' between CHK2 and CHK2" from Section 4.4 (which was equivalent to CHK2) we can also build a scheme CHK1' between CHK1 and simple CHK1 scheme that still uses the one-time signature but simplifies decryption by equivalently replacing the original randomized decryption by a consistency check plus a deterministic computation of the key. Again this scheme CHK1' can be shown to be equivalent to CHK1.

Both schemes, CHK1' and CHK2' already give nearly the same schemes with the same small difference as the two schemes CHK1" and CHK2".

5.4 Efficiency

We summarize our results and present a quick efficiency comparison of our proposed scheme with the original scheme from Canetti, Halevi, and Katz [6].

The scheme CHK2 is the scheme obtained from the second Boneh/Boyen IBE scheme plugged into the CHK transformation from Section 4.1. We give the performance values for the more MAC-based BK transformation [4]. The scheme CHK2" from Section 4.3 is our simplified version of CHK2. For comparison the schemes CHK1 and its simplified variant CHK1" are given in Appendix C. For comparison we borrowed some figures from [2, 5]. Ciphertext overhead represents the difference (in bits) between the ciphertext length and the message length, and $|p|$ is the length of a group element.

Scheme	Assumption	Encapsulation	Decapsulation	Ciphertext	Keysize
		#pairings + #[multi,reg]-exp + ...		Overhead	(pk, sk)
CHK2"	square-BDDH	$0 + [1, 2] + \mathsf{TCR}$	$1 + [0, 1] + \mathsf{TCR}$	$2\|p\|$	$(3, 3)$
CHK2	q-BDDHI	$0 + [1, 2] + \mathsf{MAC}$	$1 + [0, 2] + \mathsf{MAC}$	$2\|p\| + 768$	$(3, 2)$
CHK1" [5]	BDDH	$0 + [1, 2] + \mathsf{TCR}$	$1 + [0, 1] + \mathsf{TCR}$	$2\|p\|$	$(3, 3)$
CHK1	BDDH	$0 + [1, 2] + \mathsf{MAC}$	$1 + [1, 0] + \mathsf{MAC}$	$2\|p\| + 768$	$(3, 3)$

5.5 Conclusion

We have shown that, after removing an unnecessary decryption overhead, CHK1 is nearly the same scheme as CHK2. Furthermore, their respective simplifications CHK1" [5] and CHK2" are also nearly the same. This contradicts the statement from [6, 4, 2] that the two schemes are different schemes, with different performance and security properties. In our point of view the fact that the CHK IBE-to-PKE transformation maps two different IBE schemes to nearly the same encryption scheme is very surprising.

For any new IBE scheme, even though it seems to be very different from the two known IBE schemes, care should be taken when claiming that the CHK transformation applied to it yields a new encryption scheme.

5.6 Acknowledgments

We thank Ronald Cramer for proposing the title and the anonymous PKC referees for their detailed comments. This research was supported by the research program Sentinels (`http://www.sentinels.nl`). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

References

1. D. Boneh and X. Boyen. Efficient selective-id secure identity based encryption without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer-Verlag, May 2004.

2. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. Accepted to *SIAM Journal on Computing*, January 2006.
3. D. Boneh and M. K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
4. D. Boneh and J. Katz. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In A. Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 87–103. Springer-Verlag, Feb. 2005.
5. X. Boyen, Q. Mei, and B. Waters. Simple and efficient CCA2 security from IBE techniques. In *ACM Conference on Computer and Communications Security—CCS 2005*, pages 320–329. New-York: ACM Press, 2005.
6. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer-Verlag, May 2004.
7. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer-Verlag, Aug. 1998.
8. R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.
9. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *23rd ACM STOC*, pages 542–552. ACM Press, May 1991.
10. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 581–600. Springer-Verlag, Mar. 2006.
11. K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442. Springer-Verlag, Aug. 2004.
12. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43. ACM Press, May 1989.
13. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*. ACM Press, May 1990.
14. C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer-Verlag, Aug. 1991.
15. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM STOC*, pages 387–394. ACM Press, May 1990.

A Proof of Theorem 3

Suppose there exists a polynomial time adversary $\mathcal{A}$ that breaks the chosen-ciphertext security of the encapsulation scheme with (non-negligible) advantage $\mathbf{Adv}^{kem\text{-}\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}}(k)$. We show that there exists an adversary $\mathcal{B}$ that runs in about the same time as $\mathcal{A}$ and runs adversary $\mathcal{A}$ as a subroutine to solve a random instance of the square BDDH problem with advantage

$$\mathbf{Adv}^{\mathrm{sbddh}}_{\mathcal{G},\mathcal{B}}(k) \geq \mathbf{Adv}^{kem\text{-}\mathrm{cca}}_{\mathcal{KEM},\mathcal{A}}(k) - \mathbf{Adv}^{\mathrm{hash\text{-}tcr}}_{\mathsf{TCR},\mathcal{H}}(k) \ . \tag{1}$$

Now Eqn. (1) proves the Theorem.

We now give the description of adversary $\mathcal{B}$. Adversary $\mathcal{B}$ inputs an instance of the square BDDH problem, i.e. $\mathcal{B}$ inputs the values $(1^k, \mathcal{BG}, g, g^a, g^b, W)$. $\mathcal{B}$'s goal is to determine whether $W = \hat{e}(g,g)^{a^2 b}$ or W is a random element in $\mathbb{G}_T$. Adversary $\mathcal{B}$ runs adversary $\mathcal{A}$ simulating its view as in the original KEM security experiment as follows:

Key Generation & Challenge. Initially adversary $\mathcal{B}$ picks a random value $d \in \mathbb{Z}_p^*$ and defines the target ciphertext

$$C^* = (\ c_1^* = g^b, \quad c_2^* = (g^b)^d\ ) \ . \tag{2}$$

and the challenge key as $K^* = W$. We denote $v = \mathsf{TCR}(c_1^*)$ as the target tag (associated with the target ciphertext). The public key $pk = (h_1, h_2)$ is defined as

$$pk = (\ h_1 = (g^a)^{-v^*} \cdot g^d), \quad h_2 = g^a, \quad z = \hat{e}(g^a, g^a)\ ) \ . \tag{3}$$

This implicitly defines the secret key $sk = (x_1, x_2, v)$ as $x_2 = a$, $x_1 = \log_g(h_1) = -v^* a + d$, and $v = h_2^{x_2} = g^{(a^2)}$ where x_1, x_2 and v are not known to adversary $\mathcal{B}$. Note that the public key is identically distributed as in the original KEM.

With each ciphertext $C = (c_1, c_2)$ we associate a tag $v = \mathsf{TCR}(c_1)$. Recall that we call a ciphertext consistent (i.e., it passes the consistency test in the decapsulation algorithm) if $c_1^{x_1 + x_2 \cdot v} = c_2$. Note that the way the keys are setup this condition can be rewritten as

$$c_2 = c_1^{x_1 + x_2 v} = c_1^{x_2 v - v^* x_2 + d} = (c_1^{x_2})^{v - v^*} \cdot c_1^d \ . \tag{4}$$

Given a consistent ciphertext $C = (c_1, c_2)$ with associated tag $v \neq v^*$ the session key $K = \hat{e}(y, c_1)$ can alternatively be computed by Eqn. (4) as

$$K = \hat{e}(y, c_1) = \hat{e}(h_2^{x_2}, c_1) = \hat{e}(h_2, c_1^{x_2}) = \hat{e}(h_2, c_2 / c_1^d)^{(v - v^*)^{-1}} \ . \tag{5}$$

By Eqn. (4) and since $v^* = \mathsf{TCR}(c_1^*)$ the challenge ciphertext $C^* = (c_1^*, c_2^*) = (g^b, (g^b)^d) = (c_1^*, (c_1^*)^d)$ is consistent. If $W = \hat{e}(g,g)^{a^2 b}$ then it follows by Eqn. (3) (since $x_2 = a$ and $h_2 = g^{x_2}$) that $C^* = (g^b, (g^b)^d)$ is a correct ciphertext of key $K^* = W = \hat{e}(g,g)^{a^2 b} = \hat{e}(g^a, g^a)^b = z^b$, distributed as in the original experiment. On the other hand, when W is uniform and independent in $\mathbb{G}_T$ then C^* is independent of $K^* = W$ in the adversary's view.

Adversary $\mathcal{B}$ runs $\mathcal{A}$ on input (pk, K^*, C^*) answering to its queries as follows:

Decryption queries. The KEM decapsulation queries are simulated by $\mathcal{B}$ as follows: Let $C = (c_1, c_2)$ be an arbitrary ciphertext submitted to the decapsulation oracle $\mathsf{Dec}(\cdot)$. First $\mathcal{B}$ performs a consistency check as explained in Section 4.3, i.e. it checks if $\hat{e}(h_1 h_2^v, c_1) = \hat{e}(g, c_2)$ using the bilinear map from $\mathcal{BG}$. If C is not consistent then $\mathcal{B}$ returns reject. Otherwise, if the ciphertext is consistent $\mathcal{B}$ computes $v = \mathsf{TCR}(c_1)$ and distinguishes the following three cases:

Case 1: $v = v^*$ and $c_1 = c_1^*$: adversary $\mathcal{B}$ rejects the query. In this case consistency (c.f. Eqn. (4)) implies $c_2 = c_1^d = (c_1^*)^d = c_2^*$ and hence $C = C^*$ and the query made by $\mathcal{A}$ is illegal. Therefore it may be rejected by $\mathcal{B}$.

Case 2: $v = v^*$ and $c_1 \neq c_1^*$: adversary $\mathcal{B}$ found a collision $c_1 \neq c_1^*$ in TCR with $\mathsf{TCR}(c_1) = \mathsf{TCR}(c_1^*)$. In that case $\mathcal{B}$ returns the collision and aborts.

Case 3: $v \neq v^*$: adversary $\mathcal{B}$ computes the correct session key by Eqn. (5) as $K \leftarrow \hat{e}(h_2, c_2/c_1^d)^{(v-v^*)^{-1}}$.

This completes the description of the decapsulation oracle.

We have shown that unless $\mathcal{B}$ finds a collision in TCR (Case 2) the simulation of the decapsulation oracle is always perfect, i.e. the output of oracle $\mathsf{Dec}(C)$ is identically distributed as the output of $\mathsf{KEMdecaps}(sk, C)$.

Guess. Eventually, $\mathcal{A}$ outputs a guess $\delta' \in \{0, 1\}$ where $\delta' = 1$ means that K^* is the correct key. Algorithm $\mathcal{B}$ concludes its own game by outputting $\gamma' = \delta'$ where $\gamma' = 1$ means that $W = \hat{e}(g, g)^{a^2 b}$ and $\gamma' = 0$ means that W is random.

This completes the description of adversary $\mathcal{B}$.

Analysis. We have shown that as long as there is no hash collision in TCR found by $\mathcal{B}$, adversary $\mathcal{A}$'s view in the simulation is identically distributed to its view in the real attack game.

Note that c_1^* is a random element from $\mathbb{G}_1$ (provided from outside of $\mathcal{B}$'s view), therefore finding a value c_1 with $\mathsf{TCR}(c_1) = \mathsf{TCR}(c_1^*)$ really contradicts to the security property of the *target* collision resistant hash function. The probability that $\mathcal{B}$ finds a collision in the hash function TCR is bounded by $\mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)$, where $\mathcal{H}$ is an adversary against the target collision resistance of TCR, running in about the same time as $\mathcal{B}$.

Define "$\mathcal{B}$ wins" to be the event that $\mathcal{B}$ wins its square BDDH game, i.e. it outputs $\delta' = 1$ if $W = \hat{e}(g, g)^{a^2 b}$ and $\delta' = 0$ if W is random in $\mathbb{G}_T$. Assume there was no hash collision found by $\mathcal{B}$. On the one hand, if W is uniform and independent in $\mathbb{G}_T$ then the challenge ciphertext C^* is independent of $K^* = W$ in the adversary's view. In that case we have $\Pr[\mathcal{B} \text{ wins}] = \Pr[\delta' = 0] = \frac{1}{2}$. On the other hand, when $W = \hat{e}(g, g)^{a^2 b}$ then C^* is a correct ciphertext of the challenge key K^*, distributed as in the original experiment. Then, by our assumption, $\mathcal{A}$ must make a correct guess $\delta' = 1$ with advantage at least $\mathbf{Adv}^{kem\text{-cca}}_{\mathcal{KEM},\mathcal{A}}(k)$ and we have $|\Pr[\mathcal{B} \text{ wins}] - \frac{1}{2}| = |\Pr[\delta' = 1] - \frac{1}{2}| \geq \mathbf{Adv}^{kem\text{-cca}}_{\mathcal{KEM},\mathcal{A}}(k)$.

Therefore, adversary $\mathcal{B}$'s advantage in the square BDDH game is bounded by $\mathbf{Adv}^{\text{sbddh}}_{\mathcal{G},\mathcal{B}}(k) \geq \mathbf{Adv}^{kem\text{-cca}}_{\mathcal{KEM},\mathcal{A}}(k) - \mathbf{Adv}^{\text{hash-tcr}}_{\mathsf{TCR},\mathcal{H}}(k)$ which proves Eqn. (1) and completes the proof of the theorem.

B Proof of Lemma 1

The implications BDDH $\leq$ square BDDH and 1-BDDHI $\leq$ 2-BDDHI $\leq$ 3-BDDHI $\leq \ldots$ are obvious. To prove "square BDDH assumption $\leq$ 1-BDDHI assumption", assume there exists a polynomial-time adversary $\mathcal{A}$ that breaks the square BDDH assumption with non-negliglible probability of success. We show that then there exists a polynomial-time adversary $\mathcal{B}$ with oracle access to $\mathcal{A}$ that breaks the 1-BDDHI assumption. Let $(h, h^z, W) \in \mathbb{G}_1^2 \times \mathbb{G}_T$ be an input instance of the 1-BDDHI problem given to $\mathcal{B}$. $\mathcal{B}$'s goal is to decide whether $W = \hat{e}(h,h)^{1/z}$ or W is random. $\mathcal{B}$ picks two random values x_0, y_0 and define its output as the bit $\gamma := \gamma'$, where γ' is input from $\mathcal{A}$ as

$$\gamma' \leftarrow \mathcal{A}(h^z, h^{x_0}, h^{y_0}, W' = W^{x_0^2 y_0}).$$

Defining $g = h^z$ (and hence $h = g^{1/z}$), $x = x_0/z$, and $y = y_0/z$ we have $(h^z, h^{x_0}, h^{y_0}) = (g, (g^{1/z})^{x_0}, (g^{1/z})^{y_0}) = (g, g^x, g^y)$. If $W = \hat{e}(h,h)^{1/z}$ then

$$W' = W^{x_0^2 y_0} = \hat{e}(h,h)^{1/z \cdot x_0^2 y_0} = \hat{e}(g,g)^{1/z^3 \cdot x_0^2 y_0} = \hat{e}(g,g)^{x^2 y}.$$

If W is a random element, so is W'. Therefore $\mathcal{B}$ solves 1-BDDHI with the same success probability as $\mathcal{A}$ solves square BDDH, which was non-negligible by assumption. This proves the lemma.

C The schemes CHK1 and CHK1"

For completeness we include the complete description of the schemes CHK1 [6] and CHK1" [5] in Fig. 3 and Fig. 4, respectively.

$\mathsf{KEMkg}(1^k)$
- $x, x_1, x_2, \xleftarrow{\$} \mathbb{Z}_p^*$
- $h_1 \leftarrow g^{x_1}$; $h_2 \leftarrow g^{x_2}$; $y \leftarrow g^x$; $z \leftarrow \hat{e}(g,y)$
- $pk \leftarrow (h_1, h_2, z)$; $sk \leftarrow (x_1, x_2, x)$
- Return (pk, sk)

$\mathsf{KEMencaps}(pk)$
- $r \xleftarrow{\$} \mathbb{Z}_p^*$; $c_1 \leftarrow g^r$
- $(v, s) \xleftarrow{\$} \mathsf{Skg}(1^k)$; $c_2 \leftarrow h_1^r \cdot h_2^{v \cdot r}$
- $K \leftarrow z^r$
- $\sigma \xleftarrow{\$} \mathsf{Sign}_s(c_1 || c_2)$
- $C \leftarrow (c_1, c_2, v, \sigma)$
- Return (K, C)

$\mathsf{KEMdecaps}(sk, C)$
- Parse C as (c_1, c_2, v, σ)
- If $\mathsf{Vfy}_v(c_1 || c_2, \sigma) = \texttt{reject}$
  - then return $\texttt{reject}$.
- Else
  - $r' \xleftarrow{\$} \mathbb{Z}_p$
  - $K \leftarrow \hat{e}\left(c_1^{x + r'(x_1 + x_2 \cdot v)} \cdot c_2^{-r'}, g\right)$

Fig. 3. The CHK1 scheme from [6].

Theorem 4 ([6]). *Assume $\mathcal{OTS}$ is a strong, one-time signature scheme. Under the BDDH assumption relative to generator $\mathcal{G}$, the CHK1 scheme from Fig. 3 is secure against chosen-ciphertext attacks.*

Theorem 5 ([5]). *Under the BDDH assumption relative to generator $\mathcal{G}$, the CHK1" scheme from Fig. 4 is secure against chosen-ciphertext attacks.*

$\mathsf{KEMkg}(1^k)$

$x_1, x_2, x \xleftarrow{\$} \mathbb{Z}_p^*$

$h_1 \leftarrow g^{x_1}$; $h_2 \leftarrow g^{x_2}$; $y \leftarrow g^x$; $z \leftarrow \hat{e}(g, y)$

$pk \leftarrow (h_1, h_2, z)$; $sk \leftarrow (x_1, x_2, y)$

Return (pk, sk)

$\mathsf{KEMencaps}(pk)$

$r \xleftarrow{\$} \mathbb{Z}_p^*$; $c_1 \leftarrow g^r$

$v \leftarrow \mathsf{TCR}(c_1)$; $c_2 \leftarrow h_1^r \cdot h_2^{v \cdot r}$

$K \leftarrow z^r$

$C \leftarrow (c_1, c_2)$

Return (C, K)

$\mathsf{KEMdecaps}(sk, C)$

Parse C as (c_1, c_2)

$v \leftarrow \mathsf{TCR}(c_1)$

If $c_1^{x_1 + x_2 \cdot v} \neq c_2$ then `reject`

Else $K \leftarrow \hat{e}(y, c_1)$

Return K

Fig. 4. The CHK1" scheme from [5].