

Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model

Kristiyan Haralambiev^{1,*}, Tibor Jäger², Eike Kiltz^{3,**}, and Victor Shoup^{4,***}

¹ Dept. of Computer Science, New York University, Courant Institute,
251 Mercer Street, New York, NY 10012, USA
kkh@cs.nyu.edu

² Horst Görtz Institute for IT Security, Ruhr-University Bochum, Germany
tiber.jager@rub.de

³ Cryptology & Information Security Group, CWI, Amsterdam, The Netherlands
kiltz@cwi.nl

⁴ Dept. of Computer Science, New York University, Courant Institute,
251 Mercer Street, New York, NY 10012, USA
shoup@cs.nyu.edu

Abstract. This paper proposes practical chosen-ciphertext secure public-key encryption systems that are provably secure under the *computational* Diffie-Hellman assumption, in the standard model. Our schemes are conceptually simpler and more efficient than previous constructions. We also show that in bilinear groups the size of the public-key can be shrunk from n to $2\sqrt{n}$ group elements, where n is the security parameter.

1 Introduction

Security against chosen-ciphertext attack (CCA) is nowadays considered to be the standard security notion for public-key encryption. In this work we are interested in practical schemes with proofs of security under mild security assumptions (such as the computational Diffie-Hellman assumption), without relying on heuristics such as the random oracle model [2].

ELGAMAL ENCRYPTION. Let \mathbb{G} be a cyclic group generated by g . The ElGamal encryption scheme, described as a key-encapsulation mechanism ($\text{Gen}, \text{Enc}, \text{Dec}$), is as follows

$$\begin{aligned} \text{Gen} : sk = z, pk = Z = g^z, \quad \text{Enc}(pk) : C = g^r, K = Z^r, \\ \text{Dec}(sk, C) : K = C^z \in \mathbb{G}, \end{aligned}$$

where all appearing exponents are chosen at random. It can be proved one-way (OW-CPA) secure under the computational Diffie-Hellman (DH) assumption,

* Supported by NSF award number CNS-0716690.

** Supported by the research program Sentinels.

*** Supported by NSF award number CNS-0716690.

but its semantic (IND-CPA) security is equivalent to the stronger DDH assumption. To obtain an IND-CPA secure variant from the DH assumption one commonly uses the Goldreich-Levin [13] hard-core predicate $f_{\text{gl}}(\cdot, R)$ with randomness R to extract a pseudorandom bit from the Diffie-Hellman seed. By a standard randomness-reusing technique one obtains a scheme that encapsulates n -bit keys:

$$\begin{aligned} \text{Gen}_{\text{dh}} : sk_{\text{dh}} &= (z_1, \dots, z_n), \quad pk_{\text{dh}} = (Z_1 = g^{z_1}, \dots, Z_n = g^{z_n}), \\ \text{Enc}(pk) : C_{\text{dh}} &= g^r, \quad K_{\text{dh}} = (f_{\text{gl}}(Z_1^r, R), \dots, f_{\text{gl}}(Z_n^r, R)) \in \{0, 1\}^n, \end{aligned} \quad (1)$$

where decapsulation reconstructs the seed values Z_i^r by computing $Z_i^r = C_{\text{dh}}^{z_i}$. Combined with a one-time pad it yields an IND-CPA secure encryption scheme.

IND-CCA SECURITY FROM DECISIONAL ASSUMPTIONS. Whereas CPA-secure schemes can be constructed generically, building CCA-secure schemes seems more difficult and usually requires stronger hardness assumptions. The first practical CCA-secure encryption scheme (without random oracles) was proposed in a seminal paper by Cramer and Shoup [10]. Their construction was later generalized to hash proof systems [9]. However, the Cramer-Shoup encryption scheme and all its variants [22,7,20,21,16,17] inherently rely on *decisional assumption*, e.g., the Decisional Diffie-Hellman (DDH) assumption or the quadratic residuosity assumption. Moreover, there are groups, such as certain elliptic curve groups with bilinear pairing map, where the DDH assumption does not hold, but the DH problem appears to be hard.

IND-CCA SECURITY FROM COMPUTATIONAL ASSUMPTIONS. The DDH assumption has often been criticized as being too strong [3,12] and in general wrong in certain cryptographically relevant groups [19]. Schemes based on the DH assumption are preferred but, surprisingly, even with strong tools such as the Cramer Shoup framework [10] such schemes seem to be hard to obtain.

Canetti, Halevi and Katz [5] proposed the first practical public-key encryption scheme based on a computational assumption, namely the Bilinear DH assumption in bilinear groups. Later, as a general tool to construct secure cryptographic primitives against active attacks, Cash *et al.* [8] proposed the Twin Diffie-Hellman (2DH) assumption. Though seemingly a stronger assumption, the *interactive* Strong 2DH assumption (which is the 2DH assumption where the adversary is additionally given an oracle that solves the 2DH problem for fixed bases) is implied by the standard DH assumption. Building on “IBE techniques” [4,5], Cash *et al.* obtained the first practical encryption scheme which is CCA-secure assuming the strong 2DH assumption, and therefore also assuming the standard DH assumption. Here the decisional 2DH oracle provided by Strong 2DH assumption plays a crucial role in distinguishing consistent from non-consistent ciphertexts. However, to prove IND-CCA security, [8] had to add n group elements to the ciphertext of the scheme from Equation (1) which renders the scheme quite impractical. In independent work, Hanaoka and Kurosawa [14] used a different approach based on broadcast encryption, and could thereby reduce the number of group elements in the ciphertexts to a constant. According to [14], their approach is not based on the twinning framework.

Recently, Hofheinz and Kiltz gave a CCA-secure encryption scheme based on the factoring assumption [18].

1.1 Our Contributions

In this paper we propose a number of new encryption schemes that are CCA-secure assuming the standard DH assumption. We apply the Twin Diffie-Hellman framework from [8] to the CPA-secure scheme given in Equation (1). Therefore our schemes are simple and intuitive. As summarized in [15, Table 1], they improve efficiency of prior schemes from [8,14].

A SCHEME FROM STRONG DH. To illustrate our main ideas we first give a toy scheme that is IND-CCA secure assuming the *Strong DH assumption* [1] (The Strong DH assumption is that the DH assumption holds when the adversary is equipped with a (fixed-base) DDH oracle.) This is essentially the same scheme as ElGamal from Equation (1), but one more group element is added to the ciphertext.

$$\begin{aligned} \text{Gen}_{\text{sdh}} : \quad & sk = (sk_{\text{dh}}, x, x'), \quad pk = (pk_{\text{dh}}, X = g^x, X' = g^{x'}) \\ \text{Enc}_{\text{sdh}}(pk) : \quad & C = (C_{\text{dh}}, (X^t X')^r), \quad K = K_{\text{dh}}, \end{aligned} \quad (2)$$

where $t = T(C_{\text{dh}})$ is the output of a target collision resistant hash function. Decryption only returns K if the ciphertext $C = (C_0, C_1)$ is consistent, i.e., if $C_0^{xt+x'} = C_1$. In all other cases it rejects and returns \perp . The additional element $(X^t X')^r$ from the ciphertext is used as a handle for an all-but-one simulation technique (based on techniques from identity-based encryption [4]) to be able to simulate the decryption oracle for all ciphertexts, except the challenge ciphertext. The above simulation technique works only if consistent ciphertexts can be distinguished from inconsistent ones, which is why we need the DDH oracle provided by the Strong DH assumption.

FIRST SCHEME FROM DH. Our first scheme, which is secure under the (standard) DH assumption, applies the twinning framework to the above idea by adding an additional element $(Y^t Y')^r$ to the ciphertext.

$$\begin{aligned} \text{Gen}_{\text{dh1}} : \quad & sk = (sk_{\text{dh}}, x, x', y, y'), \\ & pk = (pk_{\text{dh}}, X = g^x, X' = g^{x'}, Y = g^y, Y' = g^{y'}) \\ \text{Enc}_{\text{dh1}}(pk) : \quad & C = (C_{\text{dh}}, (X^t X')^r, (Y^t Y')^r), \\ & K = K_{\text{dh}}. \end{aligned} \quad (3)$$

Again, decryption only returns K if the ciphertext is consistent, and \perp otherwise. By analogy to the scheme from Equation (2) it is IND-CCA secure under the Strong 2DH assumption which, by the Twinning theorem from [8], is implied by the standard DH assumption. Again, the Decisional 2DH oracle provided by the Strong DH assumption is crucial for distinguishing consistent from inconsistent ciphertexts in the reduction.

SECOND SCHEME FROM DH. Our second scheme from the DH assumption applies an “implicit rejection technique” to remove the second element from the ciphertext.

$$\begin{aligned}
\text{Gen}_{\text{dh2}} : \quad & sk = (sk_{\text{dh}}, x, x', y, y'), \\
& pk = (pk_{\text{dh}}, X = g^x, X' = g^{x'}, Y = g^y, Y' = g^{y'}) \\
\text{Enc}_{\text{dh2}}(pk) : \quad & C = (C_{\text{dh}}, (X^t X')^r), \\
& K = K_G \oplus K_{\text{dh}}, \text{ where } K_G = G((Y^t Y')^r),
\end{aligned} \tag{4}$$

where $G : \mathbb{G} \rightarrow \{0, 1\}^n$ is a secure pseudorandom generator. Decryption only returns K if the ciphertext $C = (C_0, C_1)$ is consistent, i.e., if $C_0^{x^t + x'} = C_1$. In that case K_G is computed as $K_G = G(C_0^{y^t + y'})$. Unfortunately, we are not able to show full CCA security of this KEM but, instead, we are able to prove the weaker constrained CCA (CCCA) security [16] under the DH assumption. A CCCA-secure KEM plus a symmetric authenticated encryption scheme (i.e., a MAC plus a one-time pad) yields CCA-secure encryption. The intuition behind the security is similar to the scheme from Equation (3) with the difference that, during the simulation, the values Y and Y' are set-up such that, if the ciphertext is inconsistent, then the simulated decryption will produce K_G that is uniform in the adversary's view and therefore $K = K_G \oplus K_{\text{dh}}$ is also uniform. Consequently, when combined with symmetric authenticated encryption such inconsistent decryption queries will get rejected by the symmetric cipher.

REDUCING THE SIZE OF THE PUBLIC-KEYS. Our schemes are quite practical, except for the large public-key which consists of $\approx n$ group elements. We also propose two methods to reduce the size of the public-key when our schemes are instantiated over bilinear groups. Most interestingly, we note that the public-key can be shrunk from n to $2\sqrt{n}$ elements by "implicitly defining" the n elements of pk_{dh} as $Z_{i,j} := \hat{e}(Z_i, Z'_j)$, for $i, j \in [1, \sqrt{n}]$. (Here $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a symmetric bilinear map.) Note that now only the $2\sqrt{n}$ elements Z_i, Z'_j need to be stored in the public-key.¹ Furthermore, in bilinear groups it is also possible to move the n values Z_1, \dots, Z_n from the public-key pk_{dh} into the system parameter that can be shared among many users. In that case the public-key only contains one group element, but the system parameters are still of size $\approx n$. We remark that the observation of putting public-key elements into the systems parameters is not new and has been made before, e.g., for Water's IBE scheme [24]. Finally, we also sketch how our ideas can be extended to construct an IBE scheme. All our bilinear constructions are CCA secure under the Bilinear DH (BDH) assumption.

2 Preliminaries

2.1 Notation

In the following we let $(\mathbb{G}_\kappa)_{\kappa \in \mathbb{N}}$ be a family of prime-order groups, indexed by security parameter κ . Occasionally we write \mathbb{G} shorthand for some group $\mathbb{G}_\kappa \in (\mathbb{G}_\kappa)_{\kappa \in \mathbb{N}}$, when the reference to the security parameter κ is clear. We denote with

¹ We remark that this is a generic technique that may also be applied to other Diffie-Hellman based constructions suffering from large public keys, such as the DDH-based lossy trapdoor functions in [23,11].

poly(κ) an unspecified positive integer-valued polynomial, and with $\text{negl}(\kappa)$ a negligible function in κ , that is, $|\text{negl}(\kappa)| < o(1/\kappa^c)$ for every positive integer c . For a positive integer n , we denote with $[n]$ the set $[n] = \{1, \dots, n\}$.

2.2 Key Encapsulation Mechanisms

Let $n = n(\kappa)$ be a polynomial. A *key-encapsulation mechanism* ($\text{Gen}, \text{Enc}, \text{Dec}$) with key-space $\{0, 1\}^n$ consists of three polynomial-time algorithms (PTAs). Via $(pk, sk) \leftarrow \text{Gen}(1^n)$ the randomized key-generation algorithm produces public/secret keys for security parameter $\kappa \in \mathbb{N}$; via $(C, K) \leftarrow \text{Enc}(pk)$ the randomized encapsulation algorithm creates an uniformly distributed symmetric key $K \in \{0, 1\}^n$, together with a ciphertext C ; via $K \leftarrow \text{Dec}(sk, C)$ the possessor of secret key sk decrypts ciphertext C to get back a key K which is an element in $\{0, 1\}^n$ or a special rejection symbol \perp . For consistency, we require that for all $\kappa \in \mathbb{N}$, and all $(C, K) \leftarrow \text{Enc}(pk)$ we have $\Pr[\text{Dec}(sk, C) = K] = 1$, where the probability is taken over the choice of $(pk, sk) \leftarrow \text{Gen}(1^n)$, 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) [10] 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

$$\text{AdvCCA}_{\text{KEM}_{\text{dh1}}}^{\mathcal{A}}(\kappa) := \Pr \left[b = b' : \begin{array}{l} (pk, sk) \leftarrow \text{Gen}(1^n) \\ (C, K_0) \leftarrow \text{Enc}(pk) \\ K_1 \leftarrow \{0, 1\}^n; b \leftarrow \{0, 1\} \\ b' \leftarrow \mathcal{A}^{\text{Dec}(\cdot)}(pk, K_b, C) \end{array} \right] - \frac{1}{2},$$

where oracle $\text{Dec}(C_i)$ returns $K_i \leftarrow \text{Dec}(sk, C_i)$. The restriction is that \mathcal{A} is only allowed to query $\text{Dec}(\cdot)$ on ciphertexts C_i different from the challenge ciphertext C . A key encapsulation mechanism is said to be *indistinguishable against chosen ciphertext attacks* (IND-CCA) if for all PTA adversaries \mathcal{A} , the advantage $\text{AdvCCA}_{\text{KEM}_{\text{dh1}}}^{\mathcal{A}}(\kappa)$ is a negligible function in κ .

It was proved in [10] that an IND-CCA secure KEM and a CCA-secure symmetric encryption scheme yields an IND-CCA secure hybrid encryption scheme.

CONSTRAINED CHOSEN-CIPHERTEXT SECURITY. Chosen-ciphertext security can be relaxed to indistinguishability against constrained chosen-ciphertext attacks (IND-CCCA) [16]. Intuitively, one only allows the adversary to make a decapsulation query if it already has some “a priori knowledge” about the decapsulated key. This partial knowledge about the key is modeled implicitly by letting the adversary additionally provide an efficiently computable Boolean predicate $\text{pred} : \{0, 1\}^n \rightarrow \{0, 1\}$. If $\text{pred}(K) = 1$ then the decapsulated key K is returned, and \perp otherwise. The amount of uncertainty the adversary has about the session key (denoted as *plaintext uncertainty* $\text{uncert}_{\mathcal{A}}$) is measured by the fraction of keys for which the predicate evaluates to 1. We require this fraction to be negligible for every query, i.e. the adversary has to have a high a priori knowledge

about the decapsulated key when making a decapsulation query. More formally, for an adversary \mathcal{A} we define the advantage function

$$\text{AdvCCCA}_{\text{KEM}_{\text{dh2}}}^{\mathcal{A}}(\kappa) := \Pr \left[b = b' : \begin{array}{l} (pk, sk) \leftarrow \text{Gen}(1^n) \\ (C, K_0) \leftarrow \text{Enc}(pk) \\ K_1 \leftarrow \{0, 1\}^n; b \leftarrow \{0, 1\} \\ b' \leftarrow \mathcal{A}^{\text{CDec}(\cdot, \cdot)}(pk, K_b, C) \end{array} \right] - \frac{1}{2},$$

where oracle $\text{CDec}(pred_i, C_i)$ first computes $K_i \leftarrow \text{Dec}(sk, C_i)$. If $K_i = \perp$ or $pred_i(K_i) = 0$ then return \perp . Otherwise, return K_i . The restriction is that \mathcal{A} is only allowed to query $\text{CDec}(pred_i, C_i)$ on predicates $pred_i$ that are provided as PTA and on ciphertexts C_i different from the challenge ciphertext C .

To adversary \mathcal{A} in the above experiment we also associate \mathcal{A} 's plaintext uncertainty $uncert_{\mathcal{A}}(\kappa)$ when making Q decapsulation queries, measured by

$$uncert_{\mathcal{A}}(\kappa) := \frac{1}{Q} \sum_{1 \leq i \leq Q} \Pr_{K \in \{0, 1\}^n} [pred_i(K) = 1],$$

where $pred_i : \mathbb{G} \rightarrow \{0, 1\}$ is the predicate \mathcal{A} submits in the i th decapsulation query. Finally, a key encapsulation mechanism is said to be *indistinguishable against constrained chosen ciphertext attacks* (IND-CCCA) if for all PTA adversaries \mathcal{A} with negligible $uncert_{\mathcal{A}}(\kappa)$, the advantage $\text{AdvCCCA}_{\text{KEM}_{\text{dh2}}}^{\mathcal{A}}(n)$ is a negligible function in κ .

It was proved in [16] that an IND-CCCA secure KEM plus a symmetric encryption scheme secure in the sense of authenticated encryption yields an IND-CCA secure hybrid encryption scheme.

We refer to the full version [15, Appendix A] for other definitions of standard cryptographic primitives such as hash functions and pseudorandom generators.

2.3 Diffie-Hellman Assumptions

Let $\mathbb{G} = \mathbb{G}_{\kappa}$ be a cyclic group generated by g . Define

$$\text{dh}(A, B) := C, \quad \text{where } A = g^a, B = g^b, \text{ and } C = g^{ab}. \quad (5)$$

The problem of computing $\text{dh}(A, B)$ given random $A, B \in \mathbb{G}$ is the *computational Diffie-Hellman (DH) problem*. The *DH assumption* asserts that this problem is hard, that is, $\Pr[\mathcal{A}(A, B) = \text{dh}(A, B)] \leq \text{negl}(\kappa)$ for all probabilistic polynomial-time algorithms \mathcal{A} . The *DH predicate* is defined as

$$\text{dhp}(A, \hat{B}, \hat{C}) := \text{dh}(A, \hat{B}) \stackrel{?}{=} \hat{C}.$$

The Strong DH assumption states that it is hard to compute $\text{dh}(A, B)$, given random $A, B \in \mathbb{G}$, along with access to a *decision oracle* for the predicate $\text{dhp}(A, \cdot, \cdot)$, which on input (\hat{B}, \hat{C}) , returns $\text{dhp}(A, \hat{B}, \hat{C})$.

Let dh be defined as in (5). Define the function

$$\begin{aligned} 2\text{dh} : \quad & \mathbb{G}^3 \rightarrow \mathbb{G}^2 \\ & (A_1, A_2, B) \mapsto (\text{dh}(A_1, B), \text{dh}(A_2, B)). \end{aligned}$$

This function, introduced in [8], is called the *twin DH function*. One can also define a corresponding *twin DH predicate*:

$$2\text{dhp}(A_1, A_2, \hat{B}, \hat{C}_1, \hat{C}_2) := 2\text{dh}(A_1, A_2, \hat{B}) \stackrel{?}{=} (\hat{C}_1, \hat{C}_2).$$

The *twin Diffie-Hellman assumption* states it is hard to compute $2\text{dh}(A_1, A_2, B)$, given random $A_1, A_2, B \in \mathbb{G}$. The *strong twin DH assumption* states that it is hard to compute $2\text{dh}(A_1, A_2, B)$, given random $A_1, A_2, B \in \mathbb{G}$, along with access to a *decision oracle* for the predicate $2\text{dhp}(A_1, A_2, \cdot, \cdot, \cdot)$, which on input $(\hat{B}, \hat{C}_1, \hat{C}_2)$, returns $2\text{dhp}(A_1, A_2, \hat{B}, \hat{C}_1, \hat{C}_2)$. It is clear that the (strong) twin DH assumption implies the DH assumption.

We will make use of a result from [8], which essentially states that the DH assumption implies the *strong twin Diffie-Hellman assumption*.

Lemma 1 (Theorem 3 of [8]). *Let \mathbb{G} be a group of prime order p , $\log_2 p = \text{poly}(\kappa)$. Suppose \mathcal{A} is an adversary against the strong twin Diffie-Hellman problem in \mathbb{G} , running in polynomial-time in κ and having non-negligible success probability. Then there exists a polynomial-time adversary \mathcal{B} against the computational Diffie-Hellman problem in \mathbb{G} having non-negligible success probability.*

2.4 Hard-Core Functions

In the following we denote with $f_{\text{gl}} : \mathbb{G} \times \{0, 1\}^u \rightarrow \{0, 1\}^\nu$ a Goldreich-Levin hard-core function [13] for $\text{dh}(A, B)$ with randomness space $\{0, 1\}^u$ and range $\{0, 1\}^\nu$, where u and ν are suitable integers (depending on the given group representation).

The following lemma is from [8, Theorem 9].

Lemma 2. *Let $\mathbb{G} = \mathbb{G}_\kappa$ be a prime-order group generated by g . Let $A_1, A_2, B \stackrel{\$}{\leftarrow} \mathbb{G}$ be random group elements, $R \stackrel{\$}{\leftarrow} \{0, 1\}^u$, and let $K = f_{\text{gl}}(\text{dh}(A_1, B), R)$. Let $U_\nu \stackrel{\$}{\leftarrow} \{0, 1\}^\nu$ be uniformly random. Suppose there exists a probabilistic polynomial-time algorithm \mathcal{B} having access to an oracle computing $2\text{dhp}(A_1, A_2, \cdot, \cdot, \cdot)$ and distinguishing the distributions*

$$\Delta_{\text{dh}} = (g, A_1, A_2, B, K, R) \quad \text{and} \quad \Delta_{\text{rand}} = (g, A_1, A_2, B, U_\nu, R)$$

with non-negligible advantage. Then there exists a probabilistic polynomial-time algorithm computing $\text{dh}(A, B)$ on input (A, B) with non-negligible success probability.

3 Chosen-Ciphertext Secure Key Encapsulation

In this section we build our first CCA-secure key-encapsulation mechanism whose security is based on the DH assumption.

Let $\mathbb{G} = \mathbb{G}_\kappa$ be a group of prime order p and let $n = n(\kappa)$ be a polynomial. Let $\mathsf{T}_s : \mathbb{G} \rightarrow \mathbb{Z}_p$ be a hash function with key s that is assumed to be target collision resistant (see [15, Appendix A] for a formal definition). Let $\text{KEM}_{\text{dh1}} = (\text{Gen}, \text{Enc}, \text{Dec})$ be defined as follows.

Gen(1^κ) Choose a random generator $g \xleftarrow{\$} \mathbb{G}$ and randomness $R \xleftarrow{\$} \{0, 1\}^u$ for f_{gl} . Choose a random seed s for the hash function T_s , choose random integers $x, x', y, y', z_1, \dots, z_n \xleftarrow{\$} \mathbb{Z}_p$, and set $X = g^x, X' = g^{x'}, Y = g^y, Y' = g^{y'}, Z_1 = g^{z_1}, \dots, Z_n = g^{z_n}$. Set

$$pk = (g, X, X', Y, Y', Z_1, \dots, Z_n, R, s) \text{ and } sk = (pk, x, x', y, y', z_1, \dots, z_n)$$

and return (pk, sk) .

Enc(pk) On input of public key pk , sample $r \xleftarrow{\$} \mathbb{Z}_p$. Set $C_0 = g^r, t = \mathsf{T}_s(C_0), C_1 = (X^t X')^r, C_2 = (Y^t Y')^r$, and

$$K = (f_{\text{gl}}(Z_1^r, R), \dots, f_{\text{gl}}(Z_n^r, R))$$

Return $((C_0, C_1, C_2), K)$.

Dec($sk, (C_0, C_1, C_2)$) Set $t = \mathsf{T}_s(C_0)$. If $C_1 \neq C_0^{xt+x'}$ or $C_2 \neq C_0^{yt+y'}$ then return \perp . Otherwise compute and return

$$K = (f_{\text{gl}}(C_0^{z_1}, R), \dots, f_{\text{gl}}(C_0^{z_n}, R)).$$

Theorem 1. *Let T_s be a target collision-resistant hash function and suppose that the computational Diffie-Hellman assumption holds in \mathbb{G} . Then KEM_{dh1} is IND-CCA secure.*

In the proof we use a trick from [4] to set up the public key and challenge ciphertext in a way to perform an all-but-one simulation. This enables the simulator to embed the given Diffie-Hellman challenge, while at the same time being able to decapsulate any ciphertext submitted by the adversary. We combine this technique with the twinning technique from [8], to be able to check for consistency of submitted ciphertexts.

PROOF. In the following we write (C_0^*, C_1^*, C_2^*) to denote the challenge ciphertext with corresponding key K_0^* , denote with K_1^* the random key chosen by the IND-CCA experiment, and set $t^* = \mathsf{T}_s(C_0^*)$.

We proceed in a sequence of games. We start with a game where the challenger proceeds like 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 random. Then we show that all games are computationally indistinguishable under the computational Diffie-Hellman assumption. Let W_i denote the event that \mathcal{A} outputs b' such that $b' = b$ in Game i .

Game 0. This is the standard IND-CCA game. By definition we have

$$\Pr[W_0] = \frac{1}{2} + \text{Adv}_{\text{CCA}_{\text{KEM}_{\text{dh1}}}}^{\mathcal{A}}(\kappa)$$

Game 1. We proceed as in Game 0, except that the challenger returns \perp if the adversary queries to decapsulate a ciphertext (C'_0, C'_1, C'_2) with $C'_0 = C_0^*$. Note that the probability that the adversary submits a ciphertext such that $C'_0 = C_0^*$ before seeing the challenge ciphertext is bounded by q/p , where q is the number of chosen-ciphertext queries issued by \mathcal{A} . Since $q = \text{poly}(\kappa)$, we have $q/p \leq \text{negl}(\kappa)$. Moreover, a ciphertext is inconsistent, thus gets rejected, if $C'_0 = C_0^*$ and $C'_1 \neq C_1^*$ or $C'_2 \neq C_2^*$, and is rejected by definition if $C'_1 = C_1^*$ and $C'_2 = C_2^*$. Therefore

$$|\Pr[W_1] - \Pr[W_0]| \leq \text{negl}(\kappa).$$

Game 2. We define Game 2 like Game 1, except for the following. Now the challenger aborts, if the adversary asks to decapsulate a ciphertext (C'_0, C'_1, C'_2) with $C'_0 \neq C_0^*$ and $\mathsf{T}_s(C'_0) = \mathsf{T}_s(C_0^*)$. By the target collision resistance of T_s , we have

$$|\Pr[W_2] - \Pr[W_1]| \leq \text{negl}(\kappa).$$

Game 3. We define Game 3 like Game 2, except that we sample $K_0^* \xleftarrow{\$} \{0, 1\}^{n\nu}$ uniformly random. Note that now both K_0^* and K_1^* are chosen uniformly random, thus we have

$$\Pr[W_3] = \frac{1}{2}.$$

We claim that

$$|\Pr[W_3] - \Pr[W_2]| \leq \text{negl}(\kappa)$$

under the computational Diffie-Hellman assumption. We prove this by a hybrid argument. To this end, we define a sequence of hybrid games H_0, \dots, H_n , such that H_0 equals Game 2 and H_n equals Game 3. Then we argue that hybrid H_i is indistinguishable from hybrid H_{i-1} for $i \in \{1, \dots, n\}$ under the computational Diffie-Hellman assumption. The claim follows, since $n = n(\kappa)$ is a polynomial. We define H_0 exactly like Game 2. Then, for i from 1 to n , in hybrid H_i we set the first $i\nu$ bits of K_0^* to independent random bits, and proceed otherwise exactly like in hybrid H_{i-1} . Thus, hybrid H_n proceeds exactly like Game 3.

Let E_i denote the event that \mathcal{A} outputs 1 in Hybrid i . Suppose

$$|\Pr[E_0] - \Pr[E_n]| = 1/\text{poly}_0(\kappa), \tag{6}$$

that is, the success probability of \mathcal{A} in Hybrid 0 is not negligibly close to the success probability in Hybrid n . Note that then there must exist an index i such that $|\Pr[E_{i-1}] - \Pr[E_i]| = 1/\text{poly}(\kappa)$ (since if $|\Pr[E_{i-1}] - \Pr[E_i]| \leq \text{negl}(\kappa)$ for all i , then we would have $|\Pr[E_0] - \Pr[E_n]| \leq \text{negl}(\kappa)$).

Suppose there exists an algorithm \mathcal{A} for which (6) holds. Then we can construct an adversary \mathcal{B} having access to a 2dhp oracle and distinguishing the distributions Δ_{dh} and Δ_{rand} , which by Lemma 2 is sufficient to prove security under the computational Diffie-Hellman assumption in \mathbb{G} . Adversary \mathcal{B} receives a challenge $\delta = (g, A_1, A_2, B, L, R)$ as input, and has access to an oracle

evaluating $2\text{dhp}(A_1, A_2, \cdot, \cdot, \cdot)$. \mathcal{B} guesses an index $i \in [n]$, which with probability at least $1/n$ corresponds to the index i such that $|\Pr[E_{i-1}] - \Pr[E_i]| = \max_i |\Pr[E_{i-1}] - \Pr[E_i]|$, and proceeds as follows.

Set-up of the public key. \mathcal{B} picks random integers $d, e, f \xleftarrow{\$} \mathbb{Z}_p$, and sets $X = A_1^e$, $X' = A_1^{-et^*} g^d$, $Y = A_2$, $Y' = A_2^{-t^*} g^f$, and $Z_i = A_1$, where $t^* = \mathsf{T}_s(B)$. R is used as randomness for $f_{\text{gl}}(\cdot, R)$, the rest of the public key is generated as in Game 0. Note that X, X', Y, Y', Z_i are independent and uniformly distributed group elements.

Handling decapsulation queries. When \mathcal{A} issues a decapsulation query $(C_0 = g^r, C_1, C_2)$, \mathcal{B} computes $t = \mathsf{T}_s(C_0)$, $\tilde{X} = (C_1/C_0^d)^{1/(et-et^*)}$, and $\tilde{Y} = (C_2/C_0^f)^{1/(t-t^*)}$. Assuming $t \neq t^*$ and that the ciphertext is formed correctly (that is, $C_0 = g^r$, $C_1 = (X^t X')^r$, and $C_2 = (Y^t Y')^r$) we have

$$\begin{aligned} \tilde{X} &= ((X^t X')^r / (g^r)^d)^{1/(et-et^*)} = (A_1^{er(t-t^*)} g^{rd} / g^{rd})^{1/(et-et^*)} \\ &= A_1^r = \text{dh}(A_1, C_0), \end{aligned}$$

and likewise $\tilde{Y} = A_2^r = \text{dh}(A_2, C_0)$. \mathcal{B} tests consistency of ciphertexts by querying $2\text{dhp}(A_1, A_2, C_0, \tilde{X}, \tilde{Y})$, which returns 1 if and only if $\tilde{X} = \text{dh}(A_1, C_0)$ and $\tilde{Y} = \text{dh}(A_2, C_0)$.

If this test is passed, then \mathcal{B} sets $K_0^* = (K_{0,1}^*, \dots, K_{0,n}^*)$ as $K_{0,i}^* = f_{\text{gl}}(\tilde{X}, R)$ and $K_{0,j}^* = f_{\text{gl}}(C_0^{z_j}, R)$ for $j \in [n] \setminus \{i\}$. Since by Game 2 we have $t \neq t^*$ for all queries issued by \mathcal{A} , \mathcal{B} can answer all decapsulation queries correctly.

Set-up of the challenge ciphertext. \mathcal{B} sets $C_0^* = B$, $C_1^* = B^d$, and $C_2^* = B^f$. Note that, by the set-up of X, X', Y, Y' , this is a consistent ciphertext, since we have

$$(X^{t^*} X')^{\log_g B} = ((A_1^e)^{t^*} A_1^{-et^*} g^d)^{\log_g B} = B^d$$

and (similarly) $(Y^{t^*} Y')^{\log_g B} = B^f$. Then \mathcal{B} samples $i-1$ uniformly random bits K_1, \dots, K_{i-1} , sets $K_i = L$, $K_j = f_{\text{gl}}((C_0^*)^{z_j}, R)$ for j from $i+1$ to n , and outputs the challenge $((C_0^*, C_1^*, C_2^*), (K_1, \dots, K_n))$.

Now, if $\delta \xleftarrow{\$} \Delta_{\text{dh}}$ then $L = f_{\text{gl}}(\text{dh}(B, Z_i), R)$. Thus \mathcal{A} 's view when interacting with \mathcal{B} is identical to Hybrid H_{i-1} . If $\delta \xleftarrow{\$} \Delta_{\text{rand}}$, then \mathcal{A} 's view is identical to Hybrid H_i . Thus \mathcal{B} can use \mathcal{A} to distinguish $\delta \in \Delta_{\text{dh}}$ from $\delta \in \Delta_{\text{rand}}$. \square

We remark that the same proof strategy can be used to prove that the KEM given in equation (2) (Section 1) is CCA-secure under the Strong DH assumption.

4 Constrained Chosen-Ciphertext Secure Key Encapsulation

In this section we build a more efficient variant of our first CCA-secure key-encapsulation mechanism, which we cannot prove CCA-secure. However, we can

prove that it is secure in the sense of constrained CCA security, which is sufficient to obtain CCA-secure hybrid encryption. Again the security is based on the DH assumption.

Let $\mathbb{G} = \mathbb{G}_\kappa$ be a group of prime order p and let $n = n(\kappa)$ be a polynomial. Let $\text{KEM}_{\text{dh2}} = (\text{Gen}, \text{Enc}, \text{Dec})$ be defined as follows.

Gen(1^κ) Choose a random generator $g \xleftarrow{\$} \mathbb{G}$ and randomness $R \xleftarrow{\$} \{0, 1\}^u$ for f_{gl} .

Choose a random seed s for the hash function $\text{T}_s : \mathbb{G} \rightarrow \mathbb{Z}_p$, choose random integers $x, x', y, y', z_1, \dots, z_n \xleftarrow{\$} \mathbb{Z}_p$, and set $X = g^x$, $X' = g^{x'}$, $Y = g^y$, $Y' = g^{y'}$, $Z_1 = g^{z_1}, \dots, Z_n = g^{z_n}$. Let $\text{G} : \mathbb{G} \rightarrow \{0, 1\}^n$ be a pseudorandom generator. Set

$$pk = (g, X, X', Y, Y', Z_1, \dots, Z_n, R, s, \text{G}) \text{ and } sk = (pk, x, x', y, y', z_1, \dots, z_n)$$

and return (pk, sk) .

Enc(pk) On input of public key pk , sample $r \xleftarrow{\$} \mathbb{Z}_p$. Set $C_0 = g^r$, $t = \text{T}_s(C_0)$, $C_1 = (X^t X')^r$, $K_{\text{G}} = \text{G}((Y^t Y')^r)$, and

$$K_{\text{dh}} = (f_{\text{gl}}(Z_1^r, R), \dots, f_{\text{gl}}(Z_n^r, R))$$

Set $K = K_{\text{G}} \oplus K_{\text{dh}}$ and return $((C_0, C_1), K)$.

Dec($sk, (C_0, C_1)$) Set $t = \text{T}_s(C_0)$. If $C_1 \neq C_0^{x^t + x'}$ then return \perp . Otherwise compute $K_{\text{G}} = \text{G}(C_0^{y^t + y'})$ and

$$K_{\text{dh}} = (f_{\text{gl}}(C_0^{z_1}, R), \dots, f_{\text{gl}}(C_0^{z_n}, R)),$$

and return $K = K_{\text{G}} \oplus K_{\text{dh}}$.

Theorem 2. *Let T_s be a target collision-resistant hash function, G be a pseudorandom generator, and suppose that the computational Diffie-Hellman assumption holds in \mathbb{G} . Then KEM_{dh2} is IND-CCCA secure.*

Since we removed one element from the ciphertext (which was crucial to apply the twinning technique from the proof of Theorem 1 to check for consistency of ciphertexts) we have to use different means to prove the constrained chosen-ciphertext security of KEM_{dh2} . Here we exploit the new set-up of the encapsulated key, which allows us to reject invalid ciphertexts “implicitly.” Due to space restrictions, the proof is deferred to the full version [15].

5 Reducing the Size of the Public Key

Let $(\mathbb{G}, \mathbb{G}_T)$ be a bilinear group that is equipped with an efficiently computable pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. (See, e.g., [6,4].) In this section we show that by instantiating our scheme from Equation (2) (Section 1) in bilinear groups we are able to reduce the size of the public-key considerably.

5.1 Bilinear Diffie-Hellman Assumption

Let

$$\text{bdh}(A, B, C) := D, \quad \text{where } A = g^a, B = g^b, C = g^c, \text{ and } D = \hat{e}(g, g)^{abc}. \quad (7)$$

The problem of computing $\text{bdh}(A, B, C)$ given random $A, B, C \in \mathbb{G}$ is the *computational Bilinear Diffie-Hellman (DH) problem*. The *BDH assumption* [6] asserts that this problem is hard, that is, $\Pr[\mathcal{A}(A, B, C) = \text{bdh}(A, B, C)] \leq \text{negl}(\kappa)$ for all probabilistic polynomial-time algorithms \mathcal{A} .

In the bilinear setting, the Goldreich-Levin theorem [13] gives us the following lemma for a $f_{\text{gl}} : \mathbb{G}_T \times \{0, 1\}^u \rightarrow \{0, 1\}^\nu$.

Lemma 3. *Let $\mathbb{G} = \mathbb{G}_\kappa$ be a prime-order group generated by g equipped with a pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Let $A, B, C \xleftarrow{\$} \mathbb{G}$ be random group elements, $R \xleftarrow{\$} \{0, 1\}^u$, and let $K = f_{\text{gl}}(\text{bdh}(A, B, C), R)$. Let $U_\nu \xleftarrow{\$} \{0, 1\}^\nu$ be uniformly random. Suppose there exists a probabilistic polynomial-time algorithm \mathcal{B} distinguishing the distributions*

$$\Delta_{\text{bdh}} = (g, A, B, C, K, R) \quad \text{and} \quad \Delta_{\text{rand}} = (g, A, B, C, U_\nu, R)$$

with non-negligible advantage. Then there exists a probabilistic polynomial-time algorithm computing $\text{bdh}(A, B, C)$ on input (A, B, C) with non-negligible success probability, hence breaking the BDH assumption.

5.2 Public-Key Encryption with Public Keys of Size $\mathcal{O}(1)$

Our first idea is a variant where the elements $\text{sys} = (g, X, X', Z_1, \dots, Z_n) \in \mathbb{G}^{n+3}$ can be put into the system parameters (that can be shared among many users) and the public-key to contain only one single group element Y . Our encryption scheme can be viewed as a BDH-variant of a Decisional BDH scheme from [7,20]. It is defined as follows.

Gen(1^κ) Given the system parameters sys choose a random integer $y \xleftarrow{\$} \mathbb{Z}_p$, and set $Y = g^y$. Set

$$pk = Y \quad \text{and} \quad sk = y$$

and return (pk, sk) .

Enc(pk) On input of public key pk , sample $r \xleftarrow{\$} \mathbb{Z}_p$. Set $C_0 = g^r$, $t = \mathsf{T}(C_0)$, $C_1 = (X^t X')^r$, and $K = (K_1, \dots, K_n)$, where

$$K_i = f_{\text{gl}}(\hat{e}(Y^r, Z_i), R), \quad \text{for } i \in [1, n].$$

Return $((C_0, C_1), K)$.

Dec($sk, (C_0, C_1)$) If $\hat{e}(C_0, X^t X') \neq \hat{e}(g, C_1)$ then return \perp . Otherwise, compute, for each $i \in [1, n]$,

$$K_i = f_{\text{gl}}(\hat{e}(C_0^y, Z_i), R)$$

and return $K = (K_1, \dots, K_n) \in \{0, 1\}^{n\nu}$.

Note that the consistency of the ciphertext is publicly verifiable, i.e., anyone could verify a ciphertext being consistent or not.

Theorem 3. *Let T be a target collision-resistant hash function and suppose that the computational Bilinear Diffie-Hellman assumption holds in \mathbb{G} . Then the above scheme is an IND-CCA secure KEM.*

PROOF. We proceed in a sequence of games similarly to Theorem 1.

As before, we write (C_0^*, C_1^*) to denote the challenge ciphertext with corresponding key K_0^* , denote with K_1^* the random key chosen by the IND-CCA experiment, and set $t^* = \mathsf{T}_s(C_0^*)$.

We start with a game where the challenger proceeds like 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 random. Then we show that all games are computationally indistinguishable under the computational Bilinear Diffie-Hellman assumption. Let W_i denote the event that \mathcal{A} outputs b' such that $b' = b$ in Game i .

Game 0. This is the standard IND-CCA game. By definition we have

$$\Pr[W_0] = \frac{1}{2} + \text{AdvCCA}_{\text{KEM}_{\text{bdh}}}^{\mathcal{A}}(\kappa)$$

Game 1. We proceed as in Game 0, except that the challenger aborts, if the adversary queries to decapsulate a ciphertext (C'_0, C'_1) with $C'_0 = C_0^*$. Note that the probability that the adversary submits a ciphertext such that $C'_0 = C_0^*$ before seeing the challenge ciphertext is bounded by q/p , where q is the number of chosen-ciphertext queries issued by \mathcal{A} . Since $q = \text{poly}(\kappa)$, we have $q/p \leq \text{negl}(\kappa)$. Moreover, a ciphertext is inconsistent, thus gets rejected, if $C'_0 = C_0^*$ and $C'_1 \neq C_1^*$, and is rejected by definition if $C'_0 = C_0^*$ and $C'_1 = C_1^*$. Therefore

$$|\Pr[W_1] - \Pr[W_0]| \leq \text{negl}(\kappa).$$

Game 2. We define Game 2 like Game 1, except for the following. Now the challenger aborts, if the adversary asks to decapsulate a ciphertext (C'_0, C'_1) with $C'_0 \neq C_0^*$ and $\mathsf{T}_s(C'_0) = \mathsf{T}_s(C_0^*)$. By the target collision resistance of T_s , we have

$$|\Pr[W_2] - \Pr[W_1]| \leq \text{negl}(\kappa).$$

Game 3. We define Game 3 like Game 2, except that we sample $K_0^* \xleftarrow{\$} \{0, 1\}^{n\kappa}$ uniformly random. Note that now both K_0^* and K_1^* are chosen uniformly random, thus we have

$$\Pr[W_3] = \frac{1}{2}.$$

We claim that

$$|\Pr[W_3] - \Pr[W_2]| \leq \text{negl}(\kappa)$$

under the computational Bilinear Diffie-Hellman assumption. We prove this by a hybrid argument. To this end, we define a sequence of hybrid games H_0, \dots, H_n ,

such that H_0 equals Game 2 and H_n equals Game 3. Then we argue that hybrid H_i is indistinguishable from hybrid H_{i-1} for $i \in \{1, \dots, n\}$ under the computational Bilinear Diffie-Hellman assumption. The claim follows, since $n = n(\kappa)$ is a polynomial. We define H_0 exactly like Game 2. Then, for i from 1 to n , in hybrid H_i we set the first $i\nu$ bits of K_0^* to independent random bits, and proceed otherwise exactly like in hybrid H_{i-1} . Thus, hybrid H_n proceeds exactly like Game 3.

Let E_i denote the event that \mathcal{A} outputs 1 in Hybrid i . Suppose that

$$|\Pr[E_0] - \Pr[E_n]| = 1/\text{poly}_0(\kappa), \quad (8)$$

that is, the success probability of \mathcal{A} in Hybrid 0 is not negligibly close to the success probability in Hybrid n . Note that then there must exist an index i such that $|\Pr[E_{i-1}] - \Pr[E_i]| = 1/\text{poly}(\kappa)$ (since if $|\Pr[E_{i-1}] - \Pr[E_i]| \leq \text{negl}(\kappa)$ for all i , then we would have $|\Pr[E_0] - \Pr[E_n]| \leq \text{negl}(\kappa)$).

Suppose that there exists an algorithm \mathcal{A} for which (8) holds. Then we can construct an adversary \mathcal{B} distinguishing the distributions Δ_{bdh} and Δ_{rand} , which by Lemma 3 is sufficient to prove security under the computational Bilinear Diffie-Hellman assumption in \mathbb{G} . Adversary \mathcal{B} receives a challenge $\delta = (g, A, B, C, L, R)$ as input, guesses an index $i \in [n]$, which with probability at least $1/n$ corresponds to the index i such that $|\Pr[E_{i-1}] - \Pr[E_i]| = \max_i |\Pr[E_{i-1}] - \Pr[E_i]|$, and proceeds as follows:

Set-up of the system parameters. \mathcal{B} picks random integers $d, e, f \xleftarrow{\$} \mathbb{Z}_p$, and sets $X = A^e$, $X' = A^{-et^*} g^d$, and $Z_i = A$, where $t^* = \mathsf{T}(C)$. The rest of the public key is generated as in Game 0. Note that C, X, X', Z_i are independent and uniformly distributed group elements.

Set-up of the public key. \mathcal{B} sets $Y = B$.

Handling decapsulation queries. When \mathcal{A} issues a decapsulation query $(C_0 = g^r, C_1)$, \mathcal{B} computes $t = \mathsf{T}_s(C_0)$ and tests the consistency of the ciphertext by verifying

$$\hat{e}(C_0, X^t X') \stackrel{?}{=} \hat{e}(g, C_1).$$

If the equality holds, then \mathcal{B} sets $K = (K_1, \dots, K_n)$ as $K_j = f_{\text{gl}}(\hat{e}(C_0^{z_j}, Y), R)$ for $j \in [n] \setminus \{i\}$ and $K_i = f_{\text{gl}}(\hat{e}(\tilde{X}, Y), R)$, where $\tilde{X} := (C_1/C_0^d)^{1/(et-et^*)}$. Note that

$$\begin{aligned} \tilde{X} &= ((X^t X')^r / (g^r)^d)^{1/(et-et^*)} = (A^{r(et-et^*)} g^{rd} / g^{rd})^{1/(et-et^*)} \\ &= A^r = \text{dh}(A, C_0). \end{aligned}$$

Since by Game 2 we have $t \neq t^*$, \mathcal{B} can answer all decapsulation queries correctly for all queries issued by \mathcal{A} .

Set-up of the challenge ciphertext. \mathcal{B} sets $C_0^* = C$ and $C_1^* = C^d$. Note that, by the set-up of X, X' , this is a consistent ciphertext, since we have

$$(X^{t^*} X')^{\log_g C} = ((A_1^e)^{t^*} A_1^{-et^*} g^d)^{\log_g C} = C^d$$

Then \mathcal{B} samples $i - 1$ uniformly random groups of ν bits K_1^*, \dots, K_{i-1}^* , sets $K_i^* = L$, $K_j^* = f_{\text{gl}}(\hat{e}(C_0^*, Y)^{z_j}, R)$ for j from $i + 1$ to n , and outputs the challenge $((C_0^*, C_1^*), (K_1^*, \dots, K_n^*))$.

Now, if $\delta \stackrel{\$}{\leftarrow} \Delta_{\text{bdh}}$ then we have $L = f_{\text{gl}}(\text{bdh}(A, B, C), R)$. Thus \mathcal{A} 's view when interacting with \mathcal{B} is identical to Hybrid H_{i-1} . If $\delta \stackrel{\$}{\leftarrow} \Delta_{\text{rand}}$, then \mathcal{A} 's view is identical to Hybrid H_i . Thus \mathcal{B} can use \mathcal{A} to distinguish $\delta \in \Delta_{\text{bdh}}$ from $\delta \in \Delta_{\text{rand}}$. \square

5.3 Public-Key Encryption with Public-Key of Size $\mathcal{O}(\sqrt{n})$

Our second idea reduces the size of the public-key from $\approx n$ to $\approx 2\sqrt{n}$ group elements (and no systems parameters). Assume n is a square and set $\eta := \sqrt{n}$. The public key contains elements $Z_1, Z'_1, \dots, Z_\eta, Z'_\eta \in \mathbb{G}$ which implicitly define $\eta^2 = n$ distinct elements $Z_{i,j} = \hat{e}(Z_i, Z'_j)$ in the target group \mathbb{G}_T . In our new scheme these elements can be used in place of Z_1, \dots, Z_n .

Gen(1^κ) Choose a random generator $g \stackrel{\$}{\leftarrow} \mathbb{G}$ and randomness $R \stackrel{\$}{\leftarrow} \{0, 1\}^u$ for f_{gl} . Choose a random seed s for the hash function \mathbb{T}_s , choose random integers $x, x', z_1, z'_1, \dots, z_\eta, z'_\eta \stackrel{\$}{\leftarrow} \mathbb{Z}_p$, and set $X = g^x$, $X' = g^{x'}$, $Z_1 = g^{z_1}$, $Z'_1 = g^{z'_1}$, \dots , $Z_\eta = g^{z_\eta}$, $Z'_\eta = g^{z'_\eta}$. Set

$$pk = (g, X, X', Z_1, Z'_1, \dots, Z_\eta, Z'_\eta, R, s) \text{ and } sk = (pk, x, x', z_1, z'_1, \dots, z_\eta, z'_\eta)$$

and return (pk, sk) .

Enc(pk) On input of public key pk , sample $r \stackrel{\$}{\leftarrow} \mathbb{Z}_p$. Set $C_0 = g^r$, $t = \mathbb{T}_s(C_0)$, $C_1 = (X^t X')^r$, and $K = (K_{1,1}, \dots, K_{\eta,\eta})$, where

$$K_{i,j} = f_{\text{gl}}(\hat{e}(Z_i^r, Z'_j), R), \text{ for } i, j \in [1, \eta].$$

Return $((C_0, C_1), K)$.

Dec($sk, (C_0, C_1)$) First reject if $\hat{e}(C_0, X^t X') \neq \hat{e}(g, C_1)$. Otherwise, for each $i, j \in [1, \eta]$ compute

$$K_{i,j} = f_{\text{gl}}(\hat{e}(C_0^{z_i}, Z'_j), R).$$

and return $K = (K_{1,1}, \dots, K_{\eta,\eta}) \in \{0, 1\}^{n\nu}$.

Like in the previous scheme, the consistency of the ciphertext is publicly verifiable. Furthermore, decryption can alternatively check consistency of the ciphertext by testing if $C_0^{x_t+x'} = C_1$.

Theorem 4. *Let \mathbb{T}_s be a target collision-resistant hash function and suppose that the computational Bilinear Diffie-Hellman assumption holds in \mathbb{G} . Then the above scheme is an IND-CCA secure KEM.*

PROOF. The proofs goes analogously to that of Theorem 3 with Game 3 defining hybrid games $H_{1,0}, H_{1,1}, H_{1,2}, \dots, H_{1,\eta}, H_{2,1}, H_{2,2}, \dots, H_{2,\eta}, H_{3,1}, \dots, H_{\eta,\eta}$ (for convenience, we denote with $H_{i,j}^-$ the game preceding $H_{i,j}$ in this ordering,

e.g. $H_{3,1}^- = H_{2,\eta}$). Assuming that each two consecutive hybrid games are indistinguishable by \mathcal{A} , Game 2 (which is the same as $H_{1,0}$) is indistinguishable from $H_{\eta,\eta}$ (which is the same as Game 3). But when both K_0^* and K_1^* are chosen uniformly random then we have

$$\Pr[W_3] = \frac{1}{2}.$$

So all we have to show is that indeed the hybrid games are indistinguishable.

Suppose that there exists an algorithm \mathcal{A} for which

$$|\Pr[E_{\eta,\eta}] - \Pr[E_{1,0}]| = 1/\text{poly}_0(\kappa), \quad (9)$$

where $E_{i,j}$ denotes the event that \mathcal{A} outputs 1 in $H_{i,j}$. Then there are $i^*, j^* \in \{1 \dots \eta\}$ such that $\Pr[E_{i^*,j^*}] - \Pr[E_{i^*,j^*}^-] = 1/\text{poly}(\kappa)$, where $E_{i,j}^-$ denotes the event that \mathcal{A} outputs 1 in $H_{i,j}^-$. (If no such indices exist and the difference is negligible for all (i,j) , then $|\Pr[E_{\eta,\eta}] - \Pr[E_{1,0}]| = \text{negl}(\kappa)$.)

Then we can construct an adversary \mathcal{B} distinguishing the distributions Δ_{bdh} and Δ_{rand} , which by Lemma 3 is sufficient to prove security under the computational Bilinear Diffie-Hellman assumption in \mathbb{G} . Adversary \mathcal{B} receives a challenge $\delta = (g, A, B, C, L, R)$ as input, guesses indices $i, j \in [\eta]$, which with probability at least $1/\eta^2$ correspond to the indices i^*, j^* such that $|\Pr[E_{i^*,j^*}^-] - \Pr[E_{i^*,j^*}]| = \max_{i,j} |\Pr[E_{i,j}^-] - \Pr[E_{i,j}]|$, and proceeds as follows:

Set-up of the public-key. \mathcal{B} picks random integers $d, e, f \xleftarrow{\$} \mathbb{Z}_p$, and sets $X = A^e$, $X' = A^{-et^*} g^d$, $Z_{i^*} = A$, and $Z'_{j^*} = B$, where $t^* = \text{T}_s(C)$. The rest of the public key is generated as in scheme definition. Note that $C, X, X', Z_{i^*}, Z'_{j^*}$ are independent and uniformly distributed group elements.

Handling decapsulation queries. When \mathcal{A} issues a decapsulation query $(C_0 = g^r, C_1)$, \mathcal{B} computes $t = \text{T}_s(C_0)$ and tests the consistency of the ciphertext by verifying

$$\hat{e}(C_0, X^t X') \stackrel{?}{=} \hat{e}(g, C_1).$$

If the equality holds, then \mathcal{B} sets $K = (K_{1,1}, \dots, K_{\eta,\eta})$ as:

- $K_{i,j} = f_{\text{gl}}(\hat{e}(C_0, Z'_j)^{z_i}, R)$ for $i \in [\eta] \setminus \{i^*\}$ and $j \in [\eta]$,
- $K_{i^*,j} = f_{\text{gl}}(\hat{e}(C_0, Z_{i^*})^{z'_j}, R)$ for $j \in [\eta] \setminus \{j^*\}$, and
- $K_{i^*,j^*} = f_{\text{gl}}(\hat{e}(\tilde{X}, B), R)$, where $\tilde{X} := (C_1/C_0^d)^{1/(et-et^*)}$.

Note that

$$\begin{aligned} \tilde{X} &= ((X^t X')^r / (g^r)^d)^{1/(et-et^*)} = (A^{r(et-et^*)} g^{rd} / g^{rd})^{1/(et-et^*)} \\ &= A^r = \text{dh}(A, C_0). \end{aligned}$$

Since by Game 2 we have $t \neq t^*$, \mathcal{B} can answer all decapsulation queries correctly for all queries issued by \mathcal{A} .

Set-up of the challenge ciphertext. \mathcal{B} sets $C_0^* = C$ and $C_1^* = C^d$. Note that, by the set-up of X, X' , this is a consistent ciphertext, since we have

$$(X^{t^*} X')^{\log_g C} = ((A_1^e)^{t^*} A_1^{-et^*} g^d)^{\log_g C} = C^d$$

Then \mathcal{B} sets the key $K^* = (K_{1,1}^*, K_{1,2}^*, \dots, K_{i^*,j^*}^*, \dots, K_{\eta,\eta}^*)$ accordingly:

- the bits before K_{i^*,j^*}^* uniformly at random;
 - $K_{i^*,j^*}^* = L$;
 - and $K_{i,j}^* = f_{\text{gl}}(\text{bdh}(C, Z_i, Z_j'), R)$ for the remaining ν -bit blocks $K_{i,j}^*$, i.e. $i > i^*$ or $(i = i^* \wedge j > j^*)$, which is possible because \mathcal{B} knows z_i or z_j' ;
- and outputs the challenge $((C_0^*, C_1^*), K^*)$.

Now, if $\delta \stackrel{\$}{\leftarrow} \Delta_{\text{bdh}}$ then we have $L = f_{\text{gl}}(\text{bdh}(A, B, C), R)$. Thus \mathcal{A} 's view when interacting with \mathcal{B} is identical to Hybrid H_{i^*,j^*}^- . If $\delta \stackrel{\$}{\leftarrow} \Delta_{\text{rand}}$, then \mathcal{A} 's view is identical to Hybrid $H_{i,j}$. Thus \mathcal{B} can use \mathcal{A} to distinguish $\delta \in \Delta_{\text{bdh}}$ from $\delta \in \Delta_{\text{rand}}$. \square

We remark that the above construction also extends to a Boneh-Boyen-style [4] identity-based encryption scheme selective-identity secure under the computational Bilinear Diffie-Hellman assumption. The IBE scheme has the same parameters as the above scheme, a user secret key for an identity id contains $2n$ group elements of the form $(g^{z_i z_j'} \cdot (X^{id} X')^{s_{i,j}}, g^{s_{i,j}}) \in \mathbb{G}^2$.

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. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press, New York (November 1993)
3. Boneh, D.: The decision Diffie-Hellman problem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 48–63. Springer, Heidelberg (1998)
4. 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)
5. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM Journal on Computing 36(5), 915–942 (2006)
6. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: ACM CCS 2005, pp. 320–329. ACM Press, New York (November 2005)
8. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)

9. 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)
10. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing* 33(1), 167–226 (2003)
11. 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. 282–298. Springer, Heidelberg (2010)
12. Goldreich, O.: *Foundations of Cryptography: Basic Applications*, vol. 2. Cambridge University Press, Cambridge (2004)
13. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: 21st ACM STOC, pp. 25–32. ACM Press, New York (May 1989)
14. 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)
15. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and efficient public-key encryption from Computational Diffie-Hellman in the standard model. *Cryptology ePrint Archive, Report 2010/033* (2010), <http://eprint.iacr.org/>
16. 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)
17. 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)
18. 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)
19. Joux, A.: A one round protocol for tripartite Diffie-Hellman. *Journal of Cryptology* 17(4), 263–276 (2004)
20. 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)
21. 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)
22. 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)
23. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: 40th ACM STOC, pp. 187–196. ACM Press, New York (2008)
24. Waters, B.R.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)