# Chosen Ciphertext Secure Public Key Encryption with a Simple Structure

Goichiro Hanaoka[1], Hideki Imai[1,2],
Kazuto Ogawa[3], and Hajime Watanabe[1]

[1] RCIS, AIST, Japan
[2] Chuo University, Japan
[3] Japan Broadcasting Corporation, Japan

**Abstract.** In this paper, we present a new public key encryption scheme with an easy-to-understand structure. More specifically, in the proposed scheme, for fixed group elements $g_1, \ldots, g_\ell$ in the public key a sender computes only $g_1^r, \ldots, g_\ell^r$ for encryption where $r$ is a single random number. Due to this simple structure, its security proof becomes very short (and one would easily understand the simulator's behavior for simultaneously dealing with embedding a hard problem and simulating a decryption oracle). Our proposed scheme is provably chosen-ciphertext secure under the gap Diffie-Hellman assumption (without random oracles). A drawback of our scheme is that its ciphertext is much longer than known practical schemes. We also propose a modification of our scheme with improved efficiency.

## 1 Introduction

### 1.1 Background

Chosen-ciphertext security (CCA-security, for short) [15, 31] is a standard notion of security for practical public key encryption (PKE) schemes. Furthermore, this security also implies universally composable security [9]. Among existing CCA-secure PKE schemes, ECIES [1] (see Appendix A) has a simple and interesting structure. Namely, (i) its encryption algorithm calculates only $g_1^r$ and $g_2^r$ with a common random $r$ where $g_1$ and $g_2$ are *fixed* bases which are contained in the public key, and (ii) its security can be proven with the *gap* Diffie-Hellman (GDH) assumption [28] (in the random oracle model [2]). Due to this simple structure, we can construct a practical PKE scheme, and more importantly, it is easy, especially for non-experts, to understand the basic mechanism for handling CCA adversaries (in the random oracle model).

The main motivation of this paper is to construct a PKE scheme which provides the above two properties in the standard model (i.e. without using random oracles).

### 1.2 Our Contribution

In this paper, we propose a novel PKE scheme which has similar properties to ECIES. Specifically, in the proposed scheme, for fixed group elements $g_1, \ldots, g_\ell$

K. Matsuura and E. Fujisaki (Eds.): IWSEC 2008, LNCS 5312, pp. 20–33, 2008.

in the public key a sender computes only $g_1^r, \ldots, g_\ell^r$ for encryption where $r$ is a random, and CCA-security of the proposed scheme can be proven under the GDH assumption.

Due to the simple structure of our proposed scheme (which is similar to ECIES), one can easily understand its essential idea for dealing with chosen-ciphertext attacks. For example, CCA-security of the proposed scheme is *optimally* (i.e. without any security loss) reduced to indistinguishability of a hardcore bit of the (gap) Diffie-Hellman key. In other words, in its security proof, the simulator can perfectly respond to any decryption query without any error probability. Especially, for starters our scheme would give an easy-to-understand insight for designing CCA-secure PKE schemes. Specifically, in the security proof, the simulator can perfectly respond to any decryption query by simply un-mask one of its components with an ordinary exponentiation (not, for example, the Boneh-Boyen-like technique [5]). Namely, since in the simulation one of components of a decryption query must form as $K^a$ where $K$ is the answer of the query and $a$ is a secret for the simulation, the simulator can easily extract the answer $K$ as $K = (K^a)^{\frac{1}{a}}$. It is also easy to see how the given instance of the underlying hard problem is embedded in the challenge ciphertext.

Unfortunately, this scheme is not practical as it has only one-bit plaintext space and its ciphertext consists of $k + 2$ group elements (plus one bit) where $k$ is a security parameter, and hence our scheme has nothing advantageous to existing practical PKE schemes in all practical aspects. In particular, the Kiltz scheme [25] is more efficient than our scheme in terms of both data sizes and computational costs with the same underlying assumption, i.e. the GDH assumption.[1] However, we stress that the main contribution of the proposed scheme is its easy-to-understand structure for protecting chosen-ciphertext attacks.

We also give some extensions of the proposed scheme for enhancing efficiency. By using these ideas, we can significantly reduce data sizes, and especially the ciphertext overhead becomes only two group elements which is the same as [25] (however, key sizes are still much longer than [25]).

### 1.3 Related Works

The first CCA-secure PKE scheme was proposed by Dolev, Dwork, and Naor [15] by extending the Naor-Yung paradigm [27] which is only non-adaptively CCA-secure. However, this scheme has a complicated structure primarily due to the use of *non-interactive zero knowledge* (NIZK) proof [4].

Cramer and Shoup [13] proposed the first practical CCA-secure scheme under the DDH assumption. This scheme was further improved by Shoup [33] and Kurosawa and Desmedt [26]. Furthermore, Hofheinz and Kiltz [23] showed a variant of the Cramer-Shoup scheme with a weaker assumption, i.e. the *n-linear* DDH assumption.

[1] In [25], security of the Kiltz scheme is discussed mainly based on the gap hashed Diffie-Hellman (GHDH) assumption instead of the standard GDH assumption. However, as (slightly) mentioned in [25], this scheme is also provably CCA-secure under the GDH assumption if the plaintext is only one-bit long.

Canetti, Halevi, and Katz [11] proposed a generic method for converting an (selectively secure) identity-based encryption scheme [6, 32] into a CCA-secure PKE scheme, and Boneh and Katz [7] improved its efficiency. Kiltz [24] discussed a more relaxed condition for achieving CCA-security. Boyen, Mei, and Waters [8] proposed practical CCA-secure schemes by using the basic idea of the Canetti-Halevi-Katz paradigm and specific properties of [35] and [5].

The above schemes, i.e. [11, 13] and their extensions, utilize powerful cryptographic tools such as subset membership problems [14] or identity-based encryption [6, 32] for *efficiently* achieving CCA-security. Therefore, in these schemes minimum functionality for achieving CCA-security seems unclear.

Kiltz [25] also proposed another practical CCA-secure scheme whose security is proven under the *gap hashed Diffie-Hellman* (GHDH) assumption. With a slight modification (by using the hard-core bit), this scheme is also provably secure under the GDH assumption. Our proposed scheme is considered as a redundant version of this scheme with an easier-to-understand structure.

In the random oracle methodology [2], it is possible to construct more efficient CCA-secure schemes, e.g. [1, 3, 17, 18, 29] (though this methodology is known as problematic [10]). Among these schemes, ECIES [1] has a very simple structure on which one can easily understand its essential mechanism for protecting chosen-ciphertext attacks (in the random oracle model). Namely, in ECIES it is impossible to properly encrypt a plaintext without submitting its corresponding Diffie-Hellman key to a random oracle, and therefore, a simulator can respond to any decryption query by simply picking it from random oracle queries (and one can correctly choose the valid Diffie-Hellman key with the help of the DDH oracle). See Appendix A for ECIES. Similarly to this, in our proposed scheme, we can easily understand that for generating a valid ciphertext a CCA adversary has to (implicitly) input the corresponding Diffie-Hellman key into redundant components of the ciphertext, and that the simulator can extract it from such redundant components. See also Sec. 6 for a more detailed description on this observation.

## 2 Definitions

Here, we give definitions for CCA-security of PKE schemes and some number theoretic assumptions, e.g. the GDH assumption. See also Appendix C for target collision resistant hash functions and data encapsulation mechanisms (DEMs).

### 2.1 Public Key Encryption

**The Model.** A public key encryption (PKE) scheme consists of the following three algorithms:

**Setup**$(1^k)$**.** Takes as input the security parameter $1^k$ and outputs a decryption key $dk$ and a public key $PK$.

**Encrypt**$(PK, M)$**.** Takes as input a public key $PK$ and a plaintext $M \in \mathcal{M}$, and outputs a ciphertext $\psi$.

**Decrypt**$(dk, \psi, PK)$**.** Takes as input the decryption key $dk$, a ciphertext $\psi$, and the public key $PK$, and outputs the plaintext $M \in \mathcal{M}$ or a special symbol "$\perp$".

We require that if $(dk, PK) \stackrel{R}{\leftarrow}$ **Setup**$(1^k)$ and $\psi \stackrel{R}{\leftarrow}$ **Encrypt**$(PK, M)$ then **Decrypt**$(dk, \psi, PK) = M$.

**Chosen-Ciphertext Security.** CCA-security of a PKE scheme is defined using the following game between an attack algorithm $\mathsf{A}$ and a challenger. Both the challenger and $\mathsf{A}$ are given $1^k$ as input.

**Setup.** The challenger runs **Setup**$(1^k)$ to obtain a decryption key $dk$ and a public key $PK$, and gives $PK$ to $\mathsf{A}$.

**Query I.** Algorithm $\mathsf{A}$ adaptively issues decryption queries $\psi_1, \ldots, \psi_m$. For query $\psi_i$, the challenger responds with **Decrypt**$(dk, \psi_i, PK)$.

**Challenge.** At some point, $\mathsf{A}$ submits a pair of plaintexts $(M_0, M_1) \in \mathcal{M}^2$. Then, the challenger picks a random $b \in \{0, 1\}$, runs algorithm **Encrypt** to obtain the challenge ciphertext $\psi^\star \stackrel{R}{\leftarrow}$ **Encrypt**$(PK, M_b)$, and gives $\psi^\star$ to $\mathsf{A}$.

**Query II.** Algorithm $\mathsf{A}$ continues to adaptively issue decryption queries $\psi_{m+1}$, $\ldots, \psi_{q_D}$. For query $\psi_i (\neq \psi^\star)$, the challenger responds as **Query I**.

**Guess.** Algorithm $\mathsf{A}$ outputs its guess $b' \in \{0, 1\}$ for $b$ and wins the game if $b = b'$.

Let $\mathsf{AdvPKE}_\mathsf{A}$ denote the probability that $\mathsf{A}$ wins the game.

**Definition 1.** We say that a PKE scheme is $(\tau, \epsilon, q_D)$ *CCA-secure* if for all $\tau$-time algorithms $\mathsf{A}$ who make a total of $q_D$ decryption queries, we have that $|\mathsf{AdvPKE}_\mathsf{A} - 1/2| < \epsilon$.

### 2.2 Number Theoretic Assumptions

**The Gap (Hashed) Diffie-Hellman Assumption.** Let $\mathbb{G}$ be a multiplicative group with prime order $p$. Then, the *gap Diffie-Hellman* (GDH) problem in $\mathbb{G}$ is stated as follows. Let $\mathsf{A}$ be an algorithm, and we say that $\mathsf{A}$ has advantage $\epsilon$ in solving the GDH problem in $\mathbb{G}$ if

$$\Pr[\mathsf{A}^{\mathcal{O}}(g, g^\alpha, g^\beta) = g^{\alpha\beta}] \geq \epsilon,$$

where the probability is over the random choice of generators $g$ in $\mathbb{G}$, the random choice of $\alpha$ and $\beta$ in $\mathbb{Z}_p$, and the random bits consumed by $\mathsf{A}$. The oracle $\mathcal{O}$ denotes the DDH oracle which on input $(g_1, g_2, g_3, g_4) \in \mathbb{G}^4$, answers whether $\log_{g_1} g_3 = \log_{g_2} g_4$ or not, and $\mathsf{A}$ is allowed to access $\mathcal{O}$ in any time and any order.

**Definition 2.** *We say that the $(\tau, \epsilon)$-GDH assumption holds in $\mathbb{G}$ if no $\tau$-time algorithm has advantage at least $\epsilon$ in solving the GDH problem in $\mathbb{G}$.*

Occasionally we drop the $\tau$ and $\epsilon$ and refer to the GDH in $\mathbb{G}$.

The *gap hashed Diffie-Hellman* (GHDH) problem [25] in $\mathbb{G}$ and function $h : \mathbb{G} \to \mathcal{D}$ is stated as follows. Let $\mathsf{A}$ be an algorithm, and we say that $\mathsf{A}$ has advantage $\epsilon$ in solving the GHDH problem in $\mathbb{G}$ and $h$ if

$$\frac{1}{2} \cdot |\Pr[\mathsf{A}^{\mathcal{O}}(g, g^{\alpha}, g^{\beta}, h(g^{\alpha\beta})) = 0] - \Pr[\mathsf{A}^{\mathcal{O}}(g, g^{\alpha}, g^{\beta}, T) = 0]| \geq \epsilon,$$

where the probability is over the random choice of generators $g$ in $\mathbb{G}$, the random choice of $\alpha$ and $\beta$ in $\mathbb{Z}_p$, the random choice of $T \in \mathcal{D}$, and the random bits consumed by $\mathsf{A}$. The oracle $\mathcal{O}$ is the DDH oracle.

**Definition 3.** *We say that the $(\tau, \epsilon)$-GHDH assumption holds in $\mathbb{G}$ and $h$ if no $\tau$-time algorithm has advantage at least $\epsilon$ in solving the GHDH problem in $\mathbb{G}$ and $h$.*

Occasionally we drop the $\tau$ and $\epsilon$ and refer to the GHDH in $\mathbb{G}$ and $h$.

**Hardcore Bits for the (Gap) Diffie-Hellman Key.** Roughly speaking, $h$ is called a *hardcore bit* function for the (gap) Diffie-Hellman key in $\mathbb{G}$ if the GHDH assumption in $\mathbb{G}$ and $h$ holds under only the GDH assumption in $\mathbb{G}$.

Let $\mathsf{A}$ be a $\tau$-time algorithm which has advantage $\epsilon$ in solving the GHDH problem in $\mathbb{G}$ and $h : \mathbb{G} \to \{0, 1\}$.

**Definition 4.** *We say that function $h : \mathbb{G} \to \{0, 1\}$ is a $(p_1, p_2)$ hardcore bit function in $\mathbb{G}$ if there exists a $p_1(\tau)$-time algorithm $\mathsf{B}$ which for any given $\mathsf{A}$, can solve the GDH problem with advantage $p_2(\epsilon)$ for some polynomials $p_1$ and $p_2$.*

See [19] for an example of hardcore bit functions for the (gap) Diffie-Hellman key.

## 3 The Proposed Scheme

In this section, we give the construction of our proposed scheme (its extensions with enhanced efficiency are given in the next section). As a basic scheme, we start with the standard ElGamal PKE scheme [16]. Since semantic security of the ElGamal scheme is based on only the DDH assumption (not the GDH assumption), with the use of hardcore bits we modify it to have semantic security based on the GDH assumption. See Appendix B for this semantically secure PKE scheme. Next, we further modify this scheme to have CCA-security by using the Dolev-Dwork-Naor paradigm [15]. Interestingly, in our proposed scheme, NIZK proofs which is required for [15] is not necessary due to the GDH assumption.[2]

[2] In general, for applying the technique of [15] we need NIZK proofs which make the resulting scheme complicated, however in our scheme it is not needed since the DDH oracle in the GDH assumption provides the necessary functionality for proving equivalence of plaintexts of different ciphertexts without neither any further computational assumption nor any additional ciphertext redundancy.

### 3.1 The Construction

Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $g \in \mathbb{G}$ be a generator. We assume that a group element of $\mathbb{G}$ is $k$-bit long where $k$ is the security parameter. Then, the construction of our proposed PKE scheme is as follows:

**Setup**$(1^k)$**:** Pick $dk = (x_0, \ldots, x_k, y_0, \ldots, y_k, z) \in \mathbb{Z}_p^{2k+3}$, and compute $X_i = g^{x_i}$, $Y_i = g^{y_i}$, and $Z = g^z$ for $i = 0, \ldots, k$. The decryption key is $dk$, and the public key is $PK = (\mathbb{G}, g, X_0, \ldots, X_k, Y_0, \ldots, Y_k, Z, h)$ where $h$ is a hardcore bit function in $\mathbb{G}$.

**Encrypt**$(PK, M)$**:** For a plaintext $M \in \{0,1\}$, pick a random $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and compute

$$\psi = (g^r, U_0^r, \ldots, U_k^r, h(Z^r) \oplus M),$$

where $U_i = X_i$ if $v_i = 0$, or $U_i = Y_i$ if $v_i = 1$, $v_i$ is $(i+1)$-th bit of $((g^r)_2 \| (h(Z^r) \oplus M))$, and $(W)_2$ denotes the binary representation of $W \in \mathbb{G}$. The ciphertext is $\psi$.

**Decrypt**$(dk, \psi, PK)$**:** For a ciphertext $\psi = (C_0, C_{1,0}, \ldots, C_{1,k}, C_2)$, check whether for all $i = 0, \ldots, k$, $C_0^{u_i} \stackrel{?}{=} C_{1,i}$ where $u_i = x_i$ if $v_i = 0$, or $u_i = y_i$ if $v_i = 1$ where $v_i$ is $(i+1)$-th bit of $((C_0)_2 \| C_2)$. If not, output $\perp$. Otherwise, output $M = C_2 \oplus h(C_0^z)$.

As earlier mentioned, the required operation for encryption in the proposed scheme is only exponentiations with a common exponent $r$ under fixed bases (which are contained in $PK$).

### 3.2 Security

Before going into a formal security proof of the proposed scheme, we consider the (in)security of its simplified scheme which would be helpful for understanding the essential part of the proposed scheme. The (insecure) simplified scheme is as follows: Suppose that the decryption key is reduced to be only $dk = (x_0, z)$ and the public key is $PK = (\mathbb{G}, g, X_0, Z, h)$. Let a ciphertext $\psi$ be $(g^r, X_0^r, h(Z^r) \oplus M)$ for a plaintext $M \in \{0,1\}$. Then, if the component "$X_0^r$" is valid (this can be checked by testing the consistency with $(g^r)^{x_0}$), the receiver would be convinced that the sender knows $r$ since without knowing $r$ (nor $x_0$), it seems hard to generate the Diffie-Hellman key $g^{r \cdot x_0}$. However, this is false. Namely, once an adversary sees a valid (challenge) ciphertext $(C_0, C_1, C_2)$, he can generate another valid ciphertext $(C_0^{r'}, C_1^{r'}, C_2)$ without knowing $\log_g C_0^{r'}$, where $r' \in \mathbb{Z}_p$ is a random. In other words, this scheme is *malleable*, and hence, insecure.

Our proposed scheme is considered as an enhanced version of the above scheme with non-malleability by using a similar technique to [15]. More specifically, in the proposed scheme, for each encryption the sender is enforced to choose a distinct subset from $\{X_0, \ldots, X_k, Y_0, \ldots, Y_k\}$ (instead of single $X_0$), and therefore, the above attack does not work any more. Consequently, the resulting scheme becomes CCA-secure.

The security of the above scheme is formally addressed as follows:

**Theorem 1.** *Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $h$ be a $(p_1, p_2)$ hardcore bit function in $\mathbb{G}$. Then, the above scheme is $(p_1^{-1}(\tau - o(\tau)), p_2^{-1}(\epsilon_{gdh}), q_D)$ CCA-secure assuming the $(\tau, \epsilon_{gdh})$ GDH assumption holds in $\mathbb{G}$.*

*Proof.* Assume we are given an adversary $\mathsf{A}$ which breaks CCA-security of the above scheme with running time $\tau$, advantage $\epsilon$, and $q_D$ decryption queries. We use $\mathsf{A}$ to construct another adversary $\mathsf{B}$ which, by using the DDH oracle, distinguishes hardcore bit $h$ of the (gap) Diffie-Hellman key in $\mathbb{G}$. This suffices for proving CCA-security of our scheme under the GDH assumption since existence of an algorithm which distinguishes a hardcore bit of the gap Diffie-Hellman key immediately implies existence of another algorithm which solves the GDH problem by the definition. Define adversary $\mathsf{B}$ as follows:

1. For a given GDH instance $(g, g^\alpha, g^\beta)$, $\mathsf{B}$ picks a random bit $\gamma$, and $(a_0, \ldots, a_k, b_0, \ldots, b_k) \in \mathbb{Z}_p^{2k+2}$. Let $v_i^\star$ be $(i+1)$-th bit of $((g^\beta)_2 \| \gamma)$ for $0 \leq i \leq k$.
2. $\mathsf{B}$ sets $Z = g^\alpha$, and $(X_i, Y_i) = (g^{a_i}, (g^\alpha)^{b_i})$ if $v_i^\star = 0$, or $(X_i, Y_i) = ((g^\alpha)^{a_i}, g^{b_i})$ if $v_i^\star = 1$, for $i = 0, \ldots, k$.
3. $\mathsf{B}$ inputs public key $PK = (\mathbb{G}, g, X_0, \ldots, X_k, Y_0, \ldots, Y_k, Z)$ and challenge ciphertext $\psi^\star = (g^\beta, (g^\beta)^{\mu_0}, \ldots, (g^\beta)^{\mu_k}, \gamma)$ to $\mathsf{A}$ where $\mu_i = a_i$ if $v_i^\star = 0$, or $\mu_i = b_i$ if $v_i^\star = 1$ for $i = 0, \ldots, k$.
   Notice that $\psi^\star$ is a valid ciphertext for plaintext $\gamma \oplus h(g^{\alpha\beta}) \in \{0, 1\}$. We also note that since the pair of plaintexts $(M_0, M_1)$ which are challenged is always $(0, 1)$, without loss of generality $\mathsf{B}$ may give $\psi^\star$ to $\mathsf{A}$ at this stage.
4. When $\mathsf{A}$ makes decryption query $\psi = (C_0, C_{1,0}, \ldots, C_{1,k}, C_2) \in \mathbb{G}^{k+2} \times \{0, 1\}$ (if a query is not in this form, then $\mathsf{B}$ simply rejects it), $\mathsf{B}$ proceeds as follows:
   (a) $\mathsf{B}$ determines a binary string $(v_i)_{0 \leq i \leq k} = ((C_0)_2 \| C_2)$, and checks whether for all $i = 0, \ldots, k$,
   $$\log_g U_i \stackrel{?}{=} \log_{C_0} C_{1,i}$$
   by using the DDH oracle (See Sec. 2.2), where $U_i = X_i$ if $v_i = 0$, or $U_i = Y_i$ if $v_i = 1$. If $\psi$ is in an invalid form, then $\mathsf{B}$ responds with "$\perp$".
   (b) If $\psi$ is in a valid form, then $\mathsf{B}$ picks (one of) $i$ such that $v_i \neq v_i^\star$. (We note that there always exists at least one such $i$ if $\psi \neq \psi^\star$.) $\mathsf{B}$ also picks $C_{1,i}$ and calculates $C_{1,i}^{1/a_i} = C_0^\alpha$ if $v_i = 0$, or $C_{1,i}^{1/b_i} = C_0^\alpha$ if $v_i = 1$. $\mathsf{B}$ responds with $M' = C_2 \oplus h(C_0^\alpha)$ to $\mathsf{A}$.
5. Finally, $\mathsf{A}$ outputs his guess $b'$ on $\gamma \oplus h(g^{\alpha\beta})$, and $\mathsf{B}$ outputs $b' \oplus \gamma$ as his guess on $h(g^{\alpha\beta})$.

Obviously, the above simulation is *always* perfect without even *any* negligible error probability. Therefore, $\mathsf{B}$'s advantage is completely the same as $\mathsf{A}$'s, i.e. $\epsilon$.

## 4 A Comparison on "Easiness-to-Understand"

In this section, we discuss "easiness-to-understand" of our proposed scheme by comparing it with the Kiltz scheme [25] and the Dolev-Dwork-Naor scheme [15] by focusing on their techniques for responding to decryption queries.

In our proposed scheme, for responding to a query the simulator picks a component from the query, and un-masks it by an ordinary exponentiation. Then, the simulator responds to the query with this result.

In the Kiltz scheme [25], there is only one redundant component in a ciphertext (which is practical and beautiful), and the simulator always picks it for extracting the answer of the query. However, since this construction utilizes the Boneh-Boyen-like technique [5], it requires more complicated and non-intuitive calculation than a single exponentiation. See, for example, Eq. (1) in the full version of [25] for the required calculation.

In the Dolev-Dwork-Naor scheme [15], similarly to our scheme, the simulator picks one of components of a decryption query, and simply decrypts this component ciphertext. This is due to that the NIZK proof guarantees that all decryption results of all component ciphertexts are identitcal. Therefore, this scheme is also considered easy-to-understand if *we do not mind using an NIZK proof as a black box*. However, actually NIZK proofs require the Karp reduction from the specific NP language (i.e. equality of plaintexts) to some NP complete language, and furthermore, the assumption of existence of enhanced trapdoor permutations is also necessary. Therefore, the full description of the scheme with a concrete construction of the NIZK proof becomes more complicated than our scheme.

## 5 Extensions

In this section, we give some ideas for enhancing efficiency of the proposed scheme. However, the structure of the resulting scheme becomes much less easy-to-understand.

### 5.1 Compressing Keys

It is possible to compress the size of keys by using target collision resistant hash functions (see Appendix C). Specifically, by replacing the vector $(v_i)_{0\le i\le k} = ((C_0)_2 \| C_2)$ with another vector $(v'_i)_{0\le i\le \ell-1} = \mathsf{TCR}(C_0, C_2)$ where $\mathsf{TCR} : \mathbb{G} \times \{0,1\} \to \{0,1\}^\ell$ is a target collision resistant hash function, sizes for both decryption and public keys are reduced by approximately $\ell/k \simeq 1/2$.[3]

### 5.2 Compressing Ciphertexts

Interestingly, our strategy of the security proof still works even if we compress the redundant components of a ciphertext by a product as

$$\psi = (g^r, \left(\prod_{v_i=0} X_i \cdot \prod_{v_i=1} Y_i\right)^r, h(Z^r) \oplus M).$$

The ciphertext overhead (i.e. ciphertext size minus plaintext size) of the resulting scheme becomes only two group elements, which is the same as the best known schemes, e.g. [8, 25].

[3] For $\ell$-bit security, the size of a group element is required to be at least $2\ell$-bit long, while the size of an output of $\mathsf{TCR}$ is required to be at least $\ell$-bit long.

### 5.3 Expanding the Plaintext Space

Similarly to [25], we can expand the size of plaintexts for arbitrary length with the GHDH assumption. Specifically, we replace the hardcore bit function $h : \mathbb{G} \to \{0,1\}$ with another hash function $h' : \mathbb{G} \to \{0,1\}^\nu$ where $\nu$ is sufficiently large, and assume the GHDH assumption holds in $\mathbb{G}$ and $h'$. Then, by encrypting a plaintext with CCA-secure DEM [20, 21, 22, 30] under the data encryption key $h'(Z^r)$ (instead of the simple one-time pad). See Appendix C for CCA-secure DEMs.

### 5.4 Applying the above Extensions Together

Here, we give a concrete construction of an enhanced version of our proposed scheme with all of the above extensions. Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $g \in \mathbb{G}$ be a generator. Let $\mathsf{TCR} : \mathbb{G} \to \{0,1\}^\ell$ be a target collision resistant hash function, and $h : \mathbb{G} \to \{0,1\}^\ell$ be a hash function (such that the GHDH assumption holds in $\mathbb{G}$ and $h$).[4] Let $(\mathsf{E}, \mathsf{D})$ be a CCA-secure DEM. Then, the construction of the enhanced scheme is as follows:

**Setup**$(1^k)$**:** Pick $dk = (x_0, \ldots, x_{\ell-1}, y_0, \ldots, y_{\ell-1}, z) \in \mathbb{Z}_p^{2\ell+1}$, and compute $X_i = g^{x_i}$, $Y_i = g^{y_i}$, and $Z = g^z$ for $i = 0, \ldots, \ell-1$. The decryption key is $dk$, and the public key is

$$PK = (\mathbb{G}, g, X_0, \ldots, X_{\ell-1}, Y_0, \ldots, Y_{\ell-1}, Z, h, \mathsf{TCR}).$$

**Encrypt**$(PK, M)$**:** For a plaintext $M \in \mathcal{M}$, pick a random $r \xleftarrow{R} \mathbb{Z}_p$, and compute

$$\psi = \left(g^r, \left(\prod_{v_i=0} X_i \cdot \prod_{v_i=1} Y_i\right)^r, \mathsf{E}(h(Z^r), M)\right),$$

where $v_i$ is $(i+1)$-th bit of $\mathsf{TCR}(g^r)$. The ciphertext is $\psi$.

**Decrypt**$(dk, \psi, PK)$**:** For a ciphertext $\psi = (C_0, C_1, C_2)$, check whether

$$\prod_{v_i=0} (C_0^{x_i}) \cdot \prod_{v_i=1} (C_0^{y_i}) \stackrel{?}{=} C_1,$$

where $v_i$ is $(i+1)$-th bit of $\mathsf{TCR}(C_0)$. If not, output $\perp$. Otherwise, output $M = \mathsf{D}(h(C_0^z), C_2)$.

**Theorem 2.** *Let $\mathbb{G}$ be a multiplicative group with prime order $p$, $\mathsf{TCR}$ be a $(\tau, \epsilon_{tcr})$ target collision resistant hash function, and $(\mathsf{E}, \mathsf{D})$ be a $(\tau, \epsilon_{dem})$ CCA-secure DEM. Then, the above PKE scheme is $(\tau - o(\tau), \epsilon_{ghdh} + \epsilon_{tcr} + \epsilon_{dem}, q_D)$ CCA-secure assuming the $(\tau, \epsilon_{ghdh})$ GHDH assumption holds in $\mathbb{G}$ and $h$.*

The proof of the theorem is given in the full version of this paper.

The above scheme is much more efficient than the basic scheme in the previous section, and especially its ciphertext overhead is only two group elements which is the same as the best known schemes [8, 25]. However, this scheme is not very easy-to-understand any more, and furthermore, is still less efficient than [25] in all practical aspects.

[4] For $\ell$-bit security, length of outputs for both $\mathsf{TCR}$ and $h$ are determined as $\ell$-bit long.

## 6 Random Oracle Model vs. Standard Model

Here, we give a brief comparison of our proposed scheme with ECIES by focusing on their methods for simulating decryption oracles (under the GDH assumption). This comparison would clarify an essential difference between the random oracle model and the standard model. In both cases the main issue is that for given $g^r$ and $g^x$ which are contained in a decryption query and a public key, respectively, the simulator has to somehow produce $g^{r \cdot x}$ without knowing $x$ nor $r$.

In the security proof of ECIES, since the data encryption key is set as $K = H(g^{r \cdot x})$ where $H$ is a random oracle, one cannot properly encrypt a plaintext without submitting $g^{r \cdot x}$ to $H$. Hence, $g^{r \cdot x}$ can be extracted from $H$-queries. We notice that $g^{r \cdot x}$ is not embedded in the ciphertext, and consequently this trick does not require ciphertext redundancy which results in a very short ciphertext. See Appendix A for ECIES.

On the other hand, in the security proof of our proposed scheme (and the Kiltz scheme [25]), the simulator cannot observe the CCA adversary's inputs to $h$, and therefore, we have to enforce the CCA adversary to embed $g^{r \cdot x}$ (with a mask) into some redundant part (i.e. $(C_{1,0}, \ldots, C_{1,k})$ in $\psi$) of the ciphertext instead. The simulator extracts (masked) $g^{r \cdot x}$ from such a redundant part by using an incomplete decryption key. Hence, we see that it is difficult to remove redundancy of a ciphertext in the standard model.

Construction of redundancy free CCA-secure PKE schemes under reasonable assumptions in the standard model is a major open problem, and even in a relaxed notion of CCA-security, only the DDH-based scheme in [12] achieves it.

## Acknowledgement

The authors would like to thank Reynald Affeldt for his invaluable comments and discussions. The authors also would like to thank Takahiro Matsuda, SeongHan Shin and Rui Zhang for their helpful comments and suggestions.

## 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: Proc. of CCS 1993, pp. 62–73 (1993)
3. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
4. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications. In: Proc. of STOC 1988, pp. 103–112 (1988)
5. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)

6. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
7. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
8. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: Proc. of CCS 2005, pp. 320–329 (2005)
9. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: Proc. of FOCS 2001, pp. 136–145 (2001)
10. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: Proc. of STOC 1998, pp. 209–218 (1998)
11. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
12. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., shelat, A., Vaikuntanathan, V.: Bounded CCA2-secure encryption. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 502–518. Springer, Heidelberg (2007)
13. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
14. 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)
15. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: Proc. of STOC 1991, pp. 542–552 (1991)
16. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. on Inform. Theory 31(4), 469–472 (1985)
17. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 53–68. Springer, Heidelberg (1999)
18. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
19. Goldreich, O., Levin, L.A.: A hard-core predicate for all one-way functions. In: Proc. of STOC 1989, pp. 25–32 (1989)
20. Halevi, S.: EME*: extending EME to handle arbitrary-length messages with associated data. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 315–327. Springer, Heidelberg (2004)
21. Halevi, S., Rogaway, P.: A tweakable enciphering mode. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 482–499. Springer, Heidelberg (2003)
22. Halevi, S., Rogaway, P.: A parallelizable enciphering mode. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 292–304. Springer, Heidelberg (2004)
23. 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)
24. 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)

25. 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), http://eprint.iacr.org/2007/036
26. 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)
27. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proc. of STOC 1990, pp. 427–437 (1990)
28. Okamoto, T., Pointcheval, D.: The gap-problems: a new class of problems for the security of cryptographic schemes. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
29. Okamoto, T., Pointcheval, D.: REACT: rapid enhanced-security asymmetric cryptosystem transform. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 159–175. Springer, Heidelberg (2001)
30. Phan, D.H., Pointcheval, D.: About the security of ciphers (semantic security and pseudo-random permutations). In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 182–197. Springer, Heidelberg (2004)
31. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
32. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
33. Shoup, V.: Using hash functions as a hedge against chosen ciphertext attack. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 275–288. Springer, Heidelberg (2000)
34. Shoup, V.: A proposal for an ISO standard for public key encryption (version 2.1) (manuscript, 2001)
35. Waters, B.: Efficient identity based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)

## A A Brief Review of ECIES

Here, we give a brief review of ECIES [1]. Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $g \in \mathbb{G}$ be a generator, and $H : \mathbb{G} \to \{0,1\}^\ell$ be a hash function, which will be viewed as a random oracle in the security analysis. Let $(\mathsf{E}, \mathsf{D})$ is a CCA-secure DEM (see Appendix C). Then, the construction of ECIES is as follows:

**Setup**$(1^k)$**:** Pick $z \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and compute $Z = g^z$. The decryption key is $z$, and the public key is $PK = (\mathbb{G}, g, Z, H)$.

**Encrypt**$(PK, M)$**:** For a plaintext $M \in \mathcal{M}$, pick a random $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and compute $\psi = (g^r, \mathsf{E}(H(Z^r), M))$. The ciphertext is $\psi$.

**Decrypt**$(dk, \psi, PK)$**:** For a ciphertext $\psi = (C_0, C_1)$, output $M = \mathsf{D}(H(C_0^z), C_1)$.

**Proposition 1 ([34]).** *Let $\mathbb{G}$ be a multiplicative group with prime order $p$, $H$ be a random oracle, and $(\mathsf{E}, \mathsf{D})$ be a $(\tau, \epsilon_{dem})$ CCA-secure DEM. Then, the above scheme is $(\tau - o(\tau), \epsilon_{gdh} + \epsilon_{dem}, q_D)$ CCA-secure assuming the $(\tau, \epsilon_{gdh})$ GDH assumption holds in $\mathbb{G}$.*

An intuitive explanation of the security is as follows: Due to the standard KEM/DEM composition theorem [33], it is sufficient to prove that the KEM part of the above scheme is CCA-secure (since the DEM part is already CCA-secure). Therefore, for proving security, by using a CCA adversary $\mathsf{A}$ against the KEM, we will construct another algorithm $\mathsf{B}$ which solves the GDH problem.

For a given GDH instance $(g, g^\alpha, g^\beta)$, $\mathsf{B}$ sets $PK = (\mathbb{G}, g, g^\alpha, H)$, where random oracle $H$ is controlled by $\mathsf{B}$, and $C_0^\star = g^\beta$ as a challenge ciphertext. $PK$ and $(C_0^\star, K^\star)$ is given to $\mathsf{A}$, where $K^\star \stackrel{R}{\leftarrow} \{0,1\}^\ell$ and $\mathsf{A}$'s goal is to correctly guess if $K^\star = H(g^{\alpha\beta})$ or not. For a decryption query $C_0$, by using the DDH oracle $\mathcal{O}$, $\mathsf{B}$ searches an $H$ query $w$ which has been submitted by $\mathsf{A}$ such that $(g, g^\alpha, C_0, w)$ forms a Diffie-Hellman tuple. $\mathsf{B}$ returns $H(w)$ if there is such an $H$ query, or a random $K_{C_0} \in \{0,1\}^\ell$ otherwise. Similarly, for an $H$ query $w$, by using $\mathcal{O}$, $\mathsf{B}$ searches a decryption query $C_0$ which has been submitted by $\mathsf{A}$ such that $(g, g^\alpha, C_0, w)$ forms a Diffie-Hellman tuple. $\mathsf{B}$ responds as $H(w) = K_{C_0}$ if there is such a decryption query, or $H(w) \stackrel{R}{\leftarrow} \{0,1\}^\ell$ otherwise. $\mathsf{B}$ keeps the above answers in his memory, and returns the same answer for the same query. Since it is information-theoretically impossible to distinguish $H(g^{\alpha\beta})$ from a random $\ell$-bit string without submitting $g^{\alpha\beta}$ to $H$, $\mathsf{A}$ submits it to random oracle $H$ at some point. Therefore, $\mathsf{B}$ can also output $g^{\alpha\beta}$ with non-negligible probability by picking it from $\mathsf{A}$'s $H$ queries (and $\mathsf{B}$ can correctly select it with the help of $\mathcal{O}$).

## B The ElGamal PKE Scheme with the GDH Assumption

Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $g \in \mathbb{G}$ be a generator. Then, the construction of the GDH-based semantically secure PKE scheme is as follows:

**Setup**$(1^k)$**:** Pick $z \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and compute $Z = g^z$. The decryption key is $dk = z$, and the public key is $PK = (\mathbb{G}, g, Z, h)$ where $h$ is a hardcore bit function in $\mathbb{G}$.

**Encrypt**$(PK, M)$**:** For a plaintext $M \in \{0,1\}$, pick a random $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$, and compute $\psi = (g^r, h(Z^r) \oplus M)$. The ciphertext is $\psi$.

**Decrypt**$(dk, \psi, PK)$**:** For a ciphertext $\psi = (C_0, C_1)$, output $M = C_1 \oplus h(C_0^z)$.

**Theorem 3.** *Let $\mathbb{G}$ be a multiplicative group with prime order $p$, and $h$ be a $(p_1, p_2)$ hardcore bit function in $\mathbb{G}$. Then, the above scheme is $(p_1^{-1}(\tau - o(\tau)), p_2^{-1}(\epsilon_{gdh}), 0)$ CCA-secure (i.e. semantically secure) assuming the $(\tau, \epsilon_{gdh})$ GDH assumption holds in $\mathbb{G}$.*

*Proof.* Assume we are given an adversary $\mathsf{A}$ which breaks semantic security of the above scheme with running time $\tau$ and advantage $\epsilon$. We use $\mathsf{A}$ to construct another adversary $\mathsf{B}$ which distinguishes hardcore bit $h$ of the Diffie-Hellman key in $\mathbb{G}$. This suffices for proving semantic security of the above scheme under the GDH assumption. Define adversary $\mathsf{B}$ as follows: For a given GDH instance $(g, g^\alpha, g^\beta)$, $\mathsf{B}$ sets $PK = (\mathbb{G}, g, g^\alpha, h)$ and $\psi^\star = (g^\beta, \gamma)$ where $\gamma$ is a random bit. $PK$ and $\psi^\star$ are given to $\mathsf{A}$. Finally, $\mathsf{A}$ outputs his guess $b'$ on $\gamma \oplus h(g^{\alpha\beta})$, and $\mathsf{B}$ outputs $b' \oplus \gamma$ as his guess on $h(g^{\alpha\beta})$.

Obviously, we can also prove semantic security of the above scheme under the computatinal Diffie-Hellman assumption since in the above proof the DDH oracle is never used. However, for enhancing it to have CCA-security we need the DDH oracle.

# C Cryptographic Tools

## C.1 Target Collision Resistant Hash Functions

Let $\mathsf{TCR} : \mathcal{X} \to \mathcal{Y}$ be a hash function, $\mathsf{A}$ be an algorithm, and $\mathsf{A}$'s advantage $\mathsf{AdvTCR}_{\mathsf{A}}$ be

$$\mathsf{AdvTCR}_{\mathsf{A}} = \Pr[\mathsf{TCR}(x') = \mathsf{TCR}(x) \in \mathcal{Y} \wedge x' \neq x |\ x \stackrel{R}{\leftarrow} \mathcal{X};\ x' \stackrel{R}{\leftarrow} \mathsf{A}(x)].$$

**Definition 5.** We say that $\mathsf{TCR}$ is a $(\tau, \epsilon)$ *target collision resistant hash function* if for all $\tau$-time algorithm $\mathsf{A}$, we have that $\mathsf{AdvTCR}_{\mathsf{A}} < \epsilon$.

It is obvious that any injective mapping can be used as a perfectly secure target collision resistant hash function.

## C.2 Data Encapsulation Mechanism

**The Model.** A data encapsulation mechanism (DEM) scheme consists of the following two algorithms:

$\mathsf{E}(K, M)$ Takes as input a data encryption key $K \in \mathcal{K}$ and a plaintext $M \in \mathcal{M}$, and outputs a ciphertext $\psi$.

$\mathsf{D}(K, \psi)$ Takes as input a data encryption key $K \in \mathcal{K}$ and a ciphertext $\psi$, and outputs the plaintext $M \in \mathcal{M}$.

We require that if $\psi \leftarrow \mathsf{E}(K, M)$ then $\mathsf{D}(K, \psi) = M$.

**Chosen-Ciphertext Security.** CCA-security of a DEM is defined using the following game between an attack algorithm $\mathsf{A}$ and a challenger. Both the challenger and $\mathsf{A}$ are given $1^k$ as input.

**Setup.** The challenger chooses a data encryption key $K \in \{0,1\}^k$.

**Query I.** Algorithm $\mathsf{A}$ adaptively issues decryption queries $\psi_1, \ldots, \psi_m$. For query $\psi_i$, the challenger responds with $\mathsf{D}(K, \psi_i)$.

**Challenge.** At some point, $\mathsf{A}$ submits a pair of plaintexts $(M_0, M_1) \in \mathcal{M}^2$. Then, the challenger picks a random $b \in \{0,1\}$, runs algorithm $\mathsf{E}$ to obtain the challenge ciphertext $\psi^\star \leftarrow \mathsf{E}(K, M_b)$, and give $\psi^\star$ to $\mathsf{A}$.

**Query II.** Algorithm $\mathsf{A}$ continues to adaptively issue decryption queries $\psi_{m+1}$, $\ldots, \psi_{q_D}$. For query $\psi_i (\neq \psi^\star)$, the challenger responds as **Query I**.

**Guess.** Algorithm $\mathsf{A}$ outputs its guess $b' \in \{0,1\}$ for $b$ and wins the game if $b = b'$.

Let $\mathsf{AdvDEM}_{\mathsf{A}}$ denote the probability that $\mathsf{A}$ wins the game.

**Definition 6.** We say that a DEM is $(\tau, \epsilon, q_D)$ *CCA-secure* if for all $\tau$-time algorithms $\mathsf{A}$ who make a total of $q_D$ decryption queries, we have that $|\mathsf{AdvDEM}_{\mathsf{A}} - 1/2| < \epsilon$.