A Generic Construction of CCA-Secure Cryptosystems without NIZKP for a Bounded Number of Decryption Queries

Goichiro Hanaoka and Hideki Imai*

Abstract

In this paper, we propose a generic construction of chosen-ciphertext secure cryptosystems against adversaries with a bounded number of decrytion queries from arbitrary semantically secure encryption in a black box manner. Our construction is not only an alternative to the previously known technique, i.e. the Naor-Yung paradigm [37, 19, 42], but also has some interesting properties. Especially, (1) it does not require *non-interactive zero-knowledge proof*, and (2) its component ciphertexts can be compressed into only *one* if the underlying encryption has a certain homomorphic property. Consequently, when applying our construction to the ElGamal encryption, ciphertext overhead of the resulting scheme will be only one group element which is considered *optimal* since it is the same as the original ElGamal. Disadvantages to previous schemes are that the upper bound of the number of decryption queries (e.g. 2^{30}) has to be known before set-up phase, and the size of public key is large.

1 Introduction

Background: Chosen-Ciphertext Secure Schemes. So far, construction methods for secure public key encryption have been intensively studied, and chosen-ciphertext security (CCA security, for short) [40, 19] is nowadays considered as a standard notion of security in practice. Furthermore, this security also implies universally composable security [12].

However, there exist only several CCA-secure schemes whose security is proven in the standard model. In particular, as generic constructions of CCA-secure schemes from arbitrary chosen-plaintext secure (CPA-secure) schemes, only the Naor-Yung paradigm [37] and its variants, e.g. [19, 42, 34], are known. Therefore, when (straightforwardly) designing a CCA-secure scheme from a given CPA-secure scheme (which may be newly proposed one), it would be significantly affected by the restriction which is due to the Naor-Yung paradigm. Namely, in the Naor-Yung paradigm, a sender has to encrypt a plaintext by using two distinct keys which results in two distinct ciphertexts, and add a non-interactive zero-knowledge proof (NIZKP) to guarantee the equivalence of the plaintexts of these two ciphertexts. Hence, its ciphertext length will be significantly larger than that of the underlying encryption scheme (even if the underlying scheme is homomorphic one such as ElGamal). Moreover, for constructing (general) NIZKP, existence of trap-door permutations is required to be assumed. It should be noticed that existence of CPA-secure schemes does not imply existence of trap-door permutations. Therefore, for flexibly designing various types of CCA-secure cryptosystems, more generic transforms are desired.

On the other hand, when we focus on the ElGamal cryptosystem and its specific properties, it is possible to have elegant and practical CCA-secure schemes, i.e. the Cramer-Shoup cryptosystem [17] and its variants. Especially, a variant which is proposed by Kurosawa and Desmedt [33] is most efficient among them, and its ciphertext overhead is two group elements and a message authentication tag (while the original Cramer-Shoup's is three group elements and more). This scheme is already practical in various

*National Institute of Advanced Industrial Science and Technology (AIST). {hanaoka-goichiro, h-imai}@aist.go.jp aspects. However, it is preferable to further reduce the ciphertext overhead since one group element and a message authentication tag are 1024-bit and 128-bit long, respectively (i.e. 2176 bits in total) if the underlying group is standard one over a finite field. For investigating this issue, it is important to know its lower bound in the first place. Especially, one may want to know if it is possible to construct a CCA-secure scheme with only one group element (i.e. 1024 bits) of ciphertext overhead which is the same as the plain ElGamal's.

Our Contribution. In this paper, we propose a new generic method for converting an arbitrary CPA-secure cryptosystem into another one with slightly weakened CCA security (which we call *CCA⁻ security*) in a black box manner. This can be considered as an alternative technique to the Naor-Yung paradigm for proving certain feasibility results on CCA-secure schemes. Especially, it is remarkable that our method does not require NIZKP nor any other computational assumptions if the upper bound of the number of decryption queries is known before set-up phase. Furthermore, in our construction, component ciphertexts can be compressed into only one if the underlying encryption scheme has a certain homomorphic property. These two properties enable us to design CCA⁻-secure schemes with short ciphertext length, and as an example, we show that it is possible to construct a CCA⁻-secure variant of the ElGamal whose ciphertext overhead is only one group element under reasonable assumptions. This implies that CCA security can be achieved without any ciphertext expansion from the plain ElGamal assuming that the upper bound of the number of decryption queries is known. However, we stress that this scheme is not very practical since its public key size is large.

Our approach is basically as follows. In the set-up phase, a user generates ℓ key pairs (dk_i, ek_i), $1 \leq i \leq \ell$ of the underlying CPA-secure encryption scheme, where dk_i and ek_i are a decryption key and its corresponding encryption key, respectively. The user publicizes $\mathcal{E} := \{ek_i\}_{1\leq i\leq \ell}$ while keeping $\mathcal{D} := \{dk_i\}_{1\leq i\leq \ell}$ as secret. When encrypting a plaintext M, a sender chooses $\mathcal{E}_s \subseteq \mathcal{E}$ by using *unduplicatable set selection* [42] which does not allow others to pick the same subset, and then, generates a ciphertext C by multiple encryption for M under all keys in $\mathcal{E}_s$. The user decrypts C by using the subset of $\mathcal{D}$ which corresponds to $\mathcal{E}_s$. We note that if the underlying encryption scheme is homomorphic, component ciphertexts for the multiple encryption can be unified into one ciphertext.

It is possible to provide unduplicatable set selection by using *strong one-time signature*, and this can be generically obtained from any one-way function. However, it may cause additional ciphertext overhead, and therefore, is not preferred for achieving short ciphertext size. In our ElGamal variant, we further remove the one-time signature by using a specific property of the ElGamal encryption.

Interestingly, *artificial abort*, which is a new proof technique proposed by Waters [45], plays an important role in the security proof of our proposed schemes. As far as we know, our schemes are the first ones whose security proof require the artificial abort, except for [45] and its variants.

Related Works. The notion of CCA security was introduced by Naor and Yung [37], and this was further extended by Rackoff and Simon [40] and Dolev, Dwork, and Naor [19]. Naor and Yung proposed a generic construction of non-adaptively CCA-secure cryptosystems from any semantically secure encryption [25] and NIZKP [7]. Dolev, Dwork, and Naor [19] and Sahai [42] later improved this idea and proposed adaptively CCA-secure constructions. Recently, Gertner, Malkin, and Myers [24] showed that for a large non-trivial class of constructions, it is impossible to construct a CCA-secure scheme from a CPA-secure scheme in a black box manner. We note that this does not contradict to our result since in our construction the upper bound of the number of decryption queries is assumed to be known, and this setting is out of their scope. For the same setting as ours, independently to our work, Pass, shelat, and Vaikuntanathan [39] showed a construction of non-malleable cryptosystems against chosen-ciphertext adversaries from any CPA-secure encryption in a non black box manner. Their construction still requires NIZKP and multiple component ciphertexts which may cause additional ciphertext overhead though their

NIZKP does not require any additional complexity assumption. As another independent work, Cramer, Hofheinz, and Kiltz [16] proposed the same generic construction as ours. However, they did not mention its application to ElGamal with short ciphertext.

Cramer and Shoup [17] proposed the first practical CCA-secure scheme by using a specific property of the ElGamal encryption. This was improved by Shoup [44] and Kurosawa and Desmedt [33]. Cramer and Shoup [18] further applied their methodology to [25] and [38].

Canetti, Halevi, and Katz [15] proposed a generic method for converting a (weakly secure) *identity-based encryption scheme* [43, 9] into a CCA-secure public key encryption scheme. Boneh and Katz [10] imporved its efficiency, and Kiltz [31] discussed a more relaxed condition for achieving CCA security. Boyen, Mei, and Waters [11] proposed practical CCA-secure schemes by using the basic idea of [15] and specific properties of [45] and [8]. Ciphertext overhead of their best scheme is two group elements. It should be noticed that these schemes are based on special types of groups called *bilinear groups*. The basic concept of our proposed method is similar to [15] though our method does not depend on bilinear groups.

Under the *random oracle* methodology [5], there exist many practical schemes, e.g. [6, 23]. However, this methodology is known to be problematic [13], and hence, in this paper we do not consider it.

2 Definitions

Throughout this paper, we use the following notations. Define $x \overset{R}{\leftarrow} \mathcal{X}$ as x being picked uniformly from a finite set $\mathcal{X}$ at random. If A is an algorithm, $x \leftarrow \mathsf{A}$ means that the output of A is x. When y is not a finite set nor an algorithm, $x \leftarrow y$ is an assignment operation. $|\cdot|$ is defined as the bit length if "$\cdot$" is an element of a finite set (respectively, the cardinality of the set if "$\cdot$" is a finite set). When we say that $\epsilon(k)$ is negligible, it means that for any constant c there exists $k_0 \in \mathbb{N}$, such that $\epsilon < (1/k)^c$ for any $k > k_0$.

Here, we review definitions for *public key encryption* (PKE), *strong one-time signature* (SOTS), *target collision resistant hash function* (TCRHF), *strong pseudorandom permutation* (SPRP), *cover-free family* (CFF), *key derivation function* (KDF), and the *decisional Diffie-Hellman* (DDH) *assumption*.

Public Key Encryption. A public key encryption (PKE) scheme Π consists of three probabilistic polynomial time (PPT) algorithms: $\Pi = (\mathsf{EGen}, \mathsf{Enc}, \mathsf{Dec})$. The key generation algorithm EGen takes as inputs 1^k, and generates decryption key dk and encryption key ek, where k is a security parameter. The encryption algorithm Enc takes as inputs ek and $M \in \mathcal{M}$, and outputs ciphertext $C \in \mathcal{C}$, where $\mathcal{M}$ and $\mathcal{C}$ are the plaintext and ciphertext spaces, respectively. The decryption algorithm Dec takes as inputs dk and C, and outputs M or $\perp$, where $\perp$ is a distinguished symbol. We require that for all $(dk, ek)(= \mathsf{EGen}(1^k))$, all M, and $C(= \mathsf{Enc}(ek, M))$, $\mathsf{Dec}(dk, C) = M$.

Semantic security [25] for PKE, i.e. $\mathsf{IND\text{-}ATK}$ [40, 19, 2] where $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, is defined as follows. Let $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ and k be an adversary and the security parameter, respectively. For $\mathsf{atk} \in \{\mathsf{cpa}, \mathsf{cca}\}$, consider the following experiment:

$$\mathbf{Exp}^{\mathsf{ind\text{-}atk}}_{\mathsf{A},\Pi}(k) : [(dk, ek) \leftarrow \mathsf{EGen}(1^k); (M_0, M_1, s) \leftarrow \mathsf{A}_1^{\mathsf{D}}(ek); b \overset{R}{\leftarrow} \{0,1\};$$
$$C^\star \leftarrow \mathsf{Enc}(ek, M_b); b' \leftarrow \mathsf{A}_2^{\mathsf{D}}(C^\star, s); \text{return 1 if } b' = b, \text{ or 0 otherwise}],$$

where D is a decryption oracle which for given C, returns $M(\text{or } \perp)(= \mathsf{Dec}(dk, C))$ if $\mathsf{atk} = \mathsf{cca}$, or a random bit string if $\mathsf{atk} = \mathsf{cpa}$. The only restriction is that $C^\star$ is not allowed to submit to D. We define $\epsilon_{\textit{ind-atk},\mathsf{A}} = |\Pr[\mathbf{Exp}^{\mathsf{ind\text{-}atk}}_{\mathsf{A},\Pi}(k) = 1] - 1/2|$.

Definition 1 ($\mathsf{IND\text{-}ATK}$)**.** We say Π is (t, q, ϵ)-$\mathsf{IND\text{-}CCA}$ (resp. (t, ϵ)-$\mathsf{IND\text{-}CPA}$) *secure*, if for any A in time bound t with at most q queries to D, $\epsilon_{\textit{ind-cca},\mathsf{A}} \le \epsilon$ (resp. $\epsilon_{\textit{ind-cpa},\mathsf{A}} \le \epsilon$). We say that Π is *CCA-secure* (resp. *CPA-secure*) if ϵ is negligible. In particular, we say that Π is *CCA⁻-secure* if it is CCA-secure under assumption that q is known a priori.

Strong One-Time Signature. A signature scheme Σ consists of three PPT algorithms: $\Sigma = (\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver})$. The key generation algorithm SGen takes as inputs 1^k, and generates signing key sk and verification key vk, where k is a security parameter. The signing algorithm Sig takes as inputs sk, $m \in \{0,1\}^*$, and outputs (σ, m), where m is a message to be signed. The verification algorithm Ver takes as inputs vk, σ', and m', and outputs `accept` or `reject`. We require that for all $(sk, vk)(= \mathsf{SGen}(1^k))$, all m, all $(\sigma, m)(= \mathsf{Sig}(sk, m))$, we have $\mathsf{Ver}(vk, \sigma, m) = \texttt{accept}$.

Here, we define strong unforgeability for (one-time) signature [26, 1]. Let A and k be an adversary and the security parameter, respectively. Consider the following experiment:

$$\mathsf{Exp}^{\mathsf{sots}}_{\mathsf{A},\Sigma}(k) : [(sk, vk) \leftarrow \mathsf{SGen}(1^k); (\sigma^\star, m^\star) \leftarrow \mathsf{A}^{\mathsf{S}}(vk); \text{return } \mathsf{Ver}(vk, \sigma^\star, m^\star)],$$

where S is a signing oracle which for a given message m, returns (σ, m). A is allowed to submit a query to S for only *once*, and $(\sigma^\star, m^\star)$ is not allowed to be S's response. We define $\epsilon_{uf,\mathsf{A}} = \Pr[\mathsf{Exp}^{\mathsf{sots}}_{\mathsf{A},\Sigma}(k) = \texttt{accept}]$.

Definition 2 (SOTS). We say Σ is (t, ϵ)*-unforgeable* if for any A in time bound t, $\epsilon_{uf,\mathsf{A}} \le \epsilon$. We say that Σ is a *strong one-time signature scheme* (SOTS) if ϵ is negligible.

Here, "strong" means that it is hard to forge a signature even if its corresponding message has been asked to S. It is well known that SOTS can be derived from any one-way function [36, 41].

Cover Free Family. Let $\mathcal{L}$ be a finite set with $|\mathcal{L}| = u$ and $\mathcal{F} = \{\mathcal{F}_1, ..., \mathcal{F}_v\}$ be a family of subsets of $\mathcal{L}$.

Definition 3 (CFF). We say $(\mathcal{L}, \mathcal{F})$ is a (u, v, w)*-cover free family* (CFF) if $\mathcal{F}_i \not\subseteq \cup_{j \in \mathcal{S}_i} \mathcal{F}_j$ for all $i \in \{1, ..., v\}$ and for all $\mathcal{S}_i \subseteq \{1, ..., v\} \backslash \{i\}$ such that $|\mathcal{S}_i| \le w$.

There exist nontrivial constructions of CFF with $u = O(w^2 \log^2 v)$ and $|\mathcal{F}_i| = O(w \log v)$ for all $i \in \{1, ..., v\}$. This implies that there exists a (u, v, w)-CFF such that $u = O(\mathsf{poly}(k))$ and $v = \Omega(\mathsf{exp}(k))$ if $w = O(\mathsf{poly}(k))$ for a security parameter k. It should be noticed that for given $\mathcal{L}$ and index i, one can efficiently genarate $\mathcal{F}_i$. An example of concrete methods for generating CFF [21, 22] is as follows. Consider a code $\mathcal{C}$ of length N on an alphabet $\mathcal{Q}$ with $|\mathcal{Q}| = t$. Let a codeword $c \in \mathcal{C}$ be $(c_1, ..., c_N) \in \mathcal{Q}^N$, and $\mathcal{F}_c$ be $\{(i, c_i)\}_{1 \le i \le N}$. Let $\mathcal{F}$ denote $\{\mathcal{F}_c\}_{c \in \mathcal{C}}$, and $\mathcal{L}$ be $\{1, ..., N\} \times \mathcal{Q}$. When applying Reed-Solomon code as $\mathcal{C}$, $(\mathcal{L}, \mathcal{F})$ becomes a CFF.

Theorem 1. *For given N, t and w where t is a prime power and $N \le t+1$, there exists a $(tN, t^{\lceil (N+w-1)/w \rceil}, w)$-CFF.*

Target Collision Resistant Hash Function. Let $h : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^n$ be a family of functions, where k is a security parameter. Let $h_K : \{0,1\}^\ell \to \{0,1\}^n$ be an instance of h, which is indexed by $K \in \{0,1\}^k$, and A be an adversary. Then, consider the following experiment:

$$\mathsf{Exp}^{\mathsf{tcr}}_{\mathsf{A},h}(k) : [K \xleftarrow{R} \{0,1\}^k; x \xleftarrow{R} \{0,1\}^\ell; x' \leftarrow \mathsf{A}(K, x); \text{return 1 if } h_K(x') = h_K(x), \text{ or 0 otherwise}].$$

We define $\epsilon_{tcr,\mathsf{A}} = \Pr[\mathsf{Exp}^{\mathsf{tcr}}_{\mathsf{A},h}(k) = 1]$.

Definition 4 (TCRHF). We say h is (t, ϵ)*-target collision resistant* if for any A in time bound t, $\epsilon_{tcr,\mathsf{A}} \le \epsilon$. We say that h is a *target collision resistant hash function* (TCRHF) if ϵ is negligible.

TCRHF is a special case of *universal one-way hash function*, and can be constructed from any one-way function [36, 41].

Strong Pseudorandom Permutation. Let $\pi : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a family of permutations, and $\pi_K : \{0,1\}^\ell \to \{0,1\}^\ell$ be an instance of π, which is indexed by $K \in \{0,1\}^k$. Let $\mathcal{P}$ be the set of all permutations for ℓ-bit strings, and A be an adversary. Then, consider the following experiments:

$$\mathsf{Exp}^{\mathsf{sprp}}_{\mathsf{A},\pi}(k) : [K \xleftarrow{R} \{0,1\}^k; b \leftarrow \mathsf{A}^{\pi_K, \pi_K^{-1}}; \text{return } b],\ \mathsf{Exp}^{\mathsf{real}}_{\mathsf{A},\pi}(k) : [perm \xleftarrow{R} \mathcal{P}; b \leftarrow \mathsf{A}^{perm, perm^{-1}}; \text{return } b],$$

where permutations π_K, π_K^{-1}, $perm$, and $perm^{-1}$ are given to A as black boxes, and A can observe only their outputs which corresond to A's inputs. We define $\epsilon_{sprp,\mathsf{A}} = \frac{1}{2}|\Pr[\mathsf{Exp}^{\mathsf{sprp}}_{\mathsf{A},\pi}(k) = 1] - \Pr[\mathsf{Exp}^{\mathsf{real}}_{\mathsf{A},\pi}(k) = 1]|$.

Definition 5 (SPRP). We say π is (t, ϵ)-*strongly pseudorandom* if for any A in time bound t, $\epsilon_{sprp,\mathsf{A}} \leq \epsilon$. We say that π is a *strong pseudorandom permutation* (SPRP) if ϵ is negligible.

Luby and Rackoff [35] showed that an SPRP can be derived from any *pseudorandom function* (PRF) by using the four-round Feistel construction, while the standard psuedorandom permutation can be obtained from any PRF by the three-round Feistel construction.

Key Derivation Function. Let $\mathbb{G}$ be a cyclic group of prime order p. Let $f : \{0,1\}^k \times \mathbb{G}^2 \to \{0,1\}^n$ be a family of functions, and $f_K : \mathbb{G}^2 \to \{0,1\}^n$ be an instance of f, which is indexed by $K \in \{0,1\}^k$. Let A be an adversary. Then, consider the following experiments:

$$\mathsf{Exp}^{\mathsf{kd}}_{\mathsf{A},f}(k) : [K \xleftarrow{R} \{0,1\}^k; h_1, h_2 \xleftarrow{R} \mathbb{G}^2; b \leftarrow \mathsf{A}(h_1, f_K(h_1, h_2)); \text{return } b],$$
$$\mathsf{Exp}^{\mathsf{rnd}}_{\mathsf{A},f}(k) : [K \xleftarrow{R} \{0,1\}^k; h_1 \xleftarrow{R} \mathbb{G}; rnd \xleftarrow{R} \{0,1\}^n; b \leftarrow \mathsf{A}(h_1, rnd); \text{return } b].$$

We define $\epsilon_{kdf,\mathsf{A}} = \frac{1}{2}|\Pr[\mathsf{Exp}^{\mathsf{kd}}_{\mathsf{A},f}(k) = 1] - \Pr[\mathsf{Exp}^{\mathsf{rnd}}_{\mathsf{A},f}(k) = 1]|$.

Definition 6 (KDF). We say f is (t, ϵ)-*indistinguishable* in $\mathbb{G}$ if for any A in time bound t, $\epsilon_{kdf,\mathsf{A}} \leq \epsilon$. We say that f is a *key derivation function* (KDF) if ϵ is negligible.

It is possible to construct a KDF without any assumption by using the leftover hash lemma [29, 30] if the underlying function family is pair-wise independent. In practice, one can simply assume that a dedicated cryptographic hash function works as a KDF.

Decisional Diffie-Hellman Assumption. Let $\mathbb{G}$ be a multiplicative cyclic group of prime order p and g be a generator of $\mathbb{G}$. Let A and k be an adversary and the security parameter, respectively, where k denotes $|p|$. Then, consider the following experiments:

$$\mathsf{Exp}^{\mathsf{dh}}_{\mathsf{A},\mathbb{G}}(k) : [\alpha, \beta \xleftarrow{R} \mathbb{Z}_p^2; b \leftarrow \mathsf{A}(g, g^\alpha, g^\beta, g^{\alpha\beta}); \text{return } b],$$
$$\mathsf{Exp}^{\mathsf{rnd}}_{\mathsf{A},\mathbb{G}}(k) : [\alpha, \beta, \gamma \xleftarrow{R} \mathbb{Z}_p^3; b \leftarrow \mathsf{A}(g, g^\alpha, g^\beta, g^\gamma); \text{return } b].$$

We define $\epsilon_{ddh,\mathsf{A}} = \frac{1}{2}|\Pr[\mathsf{Exp}^{\mathsf{dh}}_{\mathsf{A},\mathbb{G}}(k) = 1] - \Pr[\mathsf{Exp}^{\mathsf{rnd}}_{\mathsf{A},\mathbb{G}}(k) = 1]|$.

Definition 7 (DDH). We say that the (t, ϵ)-*decisional Diffie-Hellman* (DDH) *assumption* holds in $\mathbb{G}$ if for any A in time bound t, $\epsilon_{ddh,\mathsf{A}} \leq \epsilon$.

3 Our Generic Constructions

In this section, we show our generic constructions of CCA$^-$-secure PKE schemes from arbitrary CPA-secure PKE schemes without using NIZKP.

3.1 Basic Idea

The basic idea of our construction is as follows. For a given CPA-secure PKE scheme Π, a user generates a (u, v, q)-CFF $(\mathcal{L}, \mathcal{F})$ and u pairs of keys $(dk_i, ek_i)_{1 \leq i \leq u}$ of Π. A sender picks a $\mathcal{F}_j$ from $\mathcal{F}$ in the unduplicatable set selection manner [42]. Then, he devided a plaintext M into $|\mathcal{F}_j|$ shares $(M_i)_{i \in \mathcal{F}_j}$, and encrypts M_i by using encryption key ek_i for all $i \in \mathcal{F}_j$. The user decrypts the ciphertext by using dk_i for $i \in \mathcal{F}_j$.

Due to the property of (u, v, q)-CFF and unduplicatable set selection, even if an adversary submits decryption queries for q times, it is guaranteed that there exists at least one dk_i which is not required for responding to these queries. Hence, a simulator can embed a given instance of Π, which he wants to break, to such dk_i. We note that there exists a (u, v, q)-CFF with $u = O(\mathsf{poly}(k))$ and $v = \Omega(\mathsf{exp}(k))$ if $q = O(\mathsf{poly}(k))$ for a security parameter k.

3.2 Construction

Let $\Pi = (\mathsf{EGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme and $\Sigma = (\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme. Let $h : \{0,1\}^k \times \{0,1\}^{|vk|} \to \{1, ..., v\}$ be a function family where the index space is $\{0,1\}^k$, and $|vk|$ denotes the size of verification key of Σ. In the security proof, Π, Σ, and h are viewed as a CPA-secure PKE, an SOTS, and a TCRHF (or, simply, a family of injective maps). Our generic construction Π' consists of the following algorithms:

Our Generic Construction

Key generation: For a given security parameter k,

1. generate a (u, v, q)-CFF $(\mathcal{L}, \mathcal{F})$ such that $\mathcal{L} = \{1, ..., u\}$,
2. $(dk_i, ek_i) \leftarrow \mathsf{EGen}(1^k)$ for $1 \leq i \leq u$, $K \xleftarrow{R} \{0,1\}^k$,
3. output $dk = ((dk_i)_{1 \leq i \leq u}, (\mathcal{L}, \mathcal{F}), K)$ and $ek = ((ek_i)_{1 \leq i \leq u}, (\mathcal{L}, \mathcal{F}), K)$.

Encryption: For given ek and a plaintext $M \in \mathcal{M}$,

1. $(sk, vk) \leftarrow \mathsf{SGen}(1^k)$, $j \leftarrow h_K(vk)$,
2. devide M into $(M_i)_{i \in \mathcal{F}_j}$ such that $\bigoplus_{i \in \mathcal{F}_j} M_i = M$,
3. $C_i \leftarrow \mathsf{Enc}(ek_i, M_i)$ for all $i \in \mathcal{F}_j$, $\sigma \leftarrow \mathsf{Sig}(sk, (C_i)_{i \in \mathcal{F}_j})$,
4. output $C = ((C_i)_{i \in \mathcal{F}_j}, \sigma, vk)$.

Decryption: For given dk and $C' = ((C'_i)_{i \in \mathcal{F}_{j'}}, \sigma', vk')$,

1. if $\mathsf{Ver}(vk', \sigma', (C'_i)_{i \in \mathcal{F}_{j'}}) = \texttt{reject}$ or $h_K(vk') \neq j'$, output "$\perp$",
2. else, $M'_i \leftarrow \mathsf{Dec}(dk_i, C_i)$ for all $i \in \mathcal{F}_{j'}$, $M' \leftarrow \bigoplus_{i \in \mathcal{F}_{j'}} M_i$,
3. output M'.

The security of the above scheme is addressed as follows.

Theorem 2. *The above scheme* Π' *is* (t, q, ϵ)-IND-CCA *if* Π *is* $(t + \tau_{cff} + (u-1)\tau_{egen} + \tau_{sgen} + O(u), \frac{1}{u}\epsilon - \frac{u+1}{2u}(\epsilon_{uf} + \epsilon_{tcr}))$-IND-CPA, Σ *is* (t, ϵ_{uf})*-unforgeable, and* h *is* (t, ϵ_{tcr})*-target collision resistant, where* τ_{cff}, τ_{egen}*, and* τ_{sgen} *are computational time for CFF generation,* EGen*, and* SGen*, respectively.*

3.3 Proof of Theorem 2

Now, we give a proof of Theorem 2. Interestingly, the *artificial abort* technique, which was recently proposed by Waters [45], plays an important role in our proof.

Construction of the Simulator. The goal of our simulator B is to break CPA security of Π by using a chosen-ciphertext adversary A against the above scheme Π'. For a given public key ek_0, B generates a (u,v,q)-CFF $(\mathcal{L},\mathcal{F})$ picks a random a from $\{1,...,u\}$, and sets $ek_a \leftarrow ek_0$. Then, B generates $(dk_i, ek_i) \leftarrow \mathsf{EGen}(1^k)$ for all $i \in \{1,...,u\}\backslash\{a\}$, and $(sk^\star, vk^\star) \leftarrow \mathsf{SGen}(1^k)$. B also picks a random K from $\{0,1\}^k$.

Next, B inputs $ek = ((ek_i)_{1\le i\le u}, (\mathcal{L},\mathcal{F}), K)$ as an encryption key of the proposed scheme. B responds to A's decryption query $C = ((C_i)_{i\in\mathcal{F}_j}, \sigma, vk)$ as follows. If $\mathsf{Ver}(vk,\sigma,(C_i)_{i\in\mathcal{F}_j}) = \mathtt{reject}$ or $h_K(vk) \neq j$, B returns "$\perp$". Else if $a \in \mathcal{F}_j$, B aborts the simulation and outputs a random $b' \in \{0,1\}$. Otherwise, B computes $M_i \leftarrow \mathsf{Dec}(dk_i, C_i)$ for all $i \in \mathcal{F}_j$.

When A submits M_0 and M_1, B picks a random $M' \in \mathcal{M}$ and submits M_0' and M_1' to its own encryption oracle, where $M_\beta' = M_\beta \oplus M'$ for $\beta \in \{0,1\}$. Let the returned challenge ciphertext be $C_a^\star$. Then, B sets $j^\star \leftarrow h_K(vk^\star)$, and generates $C_i^\star \leftarrow \mathsf{Enc}(ek_i, M_i)$ for $i \in \mathcal{F}_{j^\star}\backslash\{a\}$, where $\bigoplus_{i\in\mathcal{F}_{j^\star}\backslash\{a\}} M_i = M'$. B returns $C^\star = ((C_i^\star)_{i\in\mathcal{F}_{j^\star}}, \sigma^\star, vk^\star)$, where $\sigma^\star \leftarrow \mathsf{Sig}(sk^\star, (C_i^\star)_{i\in\mathcal{F}_{j^\star}})$.

Artificial Abort. Let $C^\ell = ((C_i^\ell)_{i\in\mathcal{F}_{j^\ell}}, \sigma^\ell, vk^\ell)$ $(1 \le \ell \le q)$ be A's decryption queries, and we say $C = ((C_i)_{i\in\mathcal{F}_j}, \sigma, vk)$ is *valid* if $\mathsf{Ver}(vk,\sigma,(C_i)_{i\in\mathcal{F}_j}) = \mathtt{accept}$ and $j = h_K(vk)$. Then, B calculates the following value t:

$$t = |\mathcal{F}_{j^\star} \backslash \cup_{j\in\mathcal{V}} \mathcal{F}_j|, \text{ where } \mathcal{V} = \{j^\ell \mid \ell \in \{\ell \mid C^\ell \text{ is valid}\}\}.$$

When A outputs b', then B outputs b' with probability $\frac{1}{t}$, or a random bit with probability $\frac{t-1}{t}$.

Estimating Probabilities. Now, we estimate $\Pr[\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{B},\Pi}(k) = 1]$. Let $\mathsf{win.cpa}$ denote an event that $\mathbf{Exp}^{\mathsf{ind\text{-}cpa}}_{\mathsf{B},\Pi}(k) = 1$, $\mathsf{win.cca}$ denote an event that $\mathbf{Exp}^{\mathsf{ind\text{-}cca}}_{\mathsf{A},\Pi'}(k) = 1$, $\mathsf{succeed}$ denote an event that B does not abort the simulation and $a \in \mathcal{F}_{j^\star}$, forge denote an event that A submits a valid decryption query which forms $C' = ((C_i')_{i\in\mathcal{F}_{j^\star}}, \sigma', vk^\star)$ such that $(C_i')_{i\in\mathcal{F}_{j^\star}} \neq (C_i^\star)_{i\in\mathcal{F}_{j^\star}}$ or $\sigma' \neq \sigma^\star$, find denote an event that A submits a valid decryption query which forms $C' = ((C_i')_{i\in\mathcal{F}_{j^\star}}, \sigma', vk')$ such that $vk' \neq vk^\star$, and $\mathsf{a.abort}$ denote an event that B outputs a random bit in the artificial abort phase. We assume that $\Pr[\mathsf{win.cca}] = 1/2 + \epsilon_{\mathsf{cca}}$. For simplicity, let ideal be an event that $\overline{\mathsf{forge}} \wedge \overline{\mathsf{find}}$. Then, we have that

$$\begin{aligned} \Pr[\mathsf{win.cpa}] &\ge \Pr[\mathsf{win.cpa}|\mathsf{ideal}]\Pr[\mathsf{ideal}] \\ &\ge \Pr[b' = b|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{ideal}]\Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{ideal}]\Pr[\mathsf{ideal}] \\ &\quad + \frac{1}{2}\Pr[\overline{\mathsf{succeed}} \vee \mathsf{a.abort}|\mathsf{ideal}]\Pr[\mathsf{ideal}]. \end{aligned} \tag{1}$$

Let *coin* be A's random coin, and $\mathcal{R}$ be the set of all possible values of *coin*. Then, we have the following lemmas:

Lemma 1. *For all $R \in \mathcal{R}$, we have that*

$$\Pr[b' = b|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{ideal} \wedge coin = R] = \Pr[\mathsf{win.cca}|\mathsf{ideal} \wedge coin = R].$$

Proof. This is obvious since B's simulation is perfect if $\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}$ is true. □

Lemma 2. *For all $R \in \mathcal{R}$, we have that*

$$\Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} | \mathsf{ideal} \wedge coin = R] = \frac{1}{u}.$$

Proof. Let $Suc(R)$ denote $\Pr[\mathsf{succeed}|\mathsf{ideal} \wedge coin = R]$, and $t(R)$ be the value of t (see **Artificial Abort.**) according to the decryption queries by A with coin R. If ideal is true, there always exists non-empty subset $\mathcal{D}$ of $\{1, ..., u\}$ such that $\mathcal{D} \subseteq \mathcal{F}_{j^\star}$ and $|\mathcal{D} \cap \cup_{j \in \mathcal{V}} \mathcal{F}_j| = 0$ due to the property of the (u, v, q)-CFF. This implies that $\mathsf{succeed}$ occurs if $a \in \mathcal{D}$, and therefore, $Suc(R) = t(R)/u$. On the other hand, $\Pr[\overline{\mathsf{a.abort}}|\mathsf{succeed} \wedge \mathsf{ideal} \wedge coin = R] = 1/t(R)$ by definition. Then we have that

$$\begin{aligned} \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{ideal} \wedge coin = R] &= \Pr[\overline{\mathsf{a.abort}}|\mathsf{succeed} \wedge \mathsf{ideal} \wedge coin = R] \cdot Suc(R) \\ &= \frac{1}{t(R)} \cdot \frac{t(R)}{u} = \frac{1}{u}, \end{aligned}$$

which proves the lemma. □

From Lemmas 1 and 2, we have that

$$\begin{aligned} &\Pr[b' = b|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{ideal}] \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{ideal}] \Pr[\mathsf{ideal}] \\ &\quad = \sum_{R \in \mathcal{R}} (\Pr[b' = b|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{ideal} \wedge coin = R] \\ &\qquad\qquad \cdot \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{ideal} \wedge coin = R] \Pr[coin = R|\mathsf{ideal}] \Pr[\mathsf{ideal}]) \\ &\quad = \sum_{R \in \mathcal{R}} (\Pr[\mathsf{win.cca}|\mathsf{ideal} \wedge coin = R] \cdot \frac{1}{u} \cdot \Pr[coin = R|\mathsf{ideal}] \Pr[\mathsf{ideal}]) \\ &\quad = \frac{1}{u} \cdot \Pr[\mathsf{win.cca}|\mathsf{ideal}] \Pr[\mathsf{ideal}]. \end{aligned} \tag{2}$$

Lemma 3. *We have that* $\Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{ideal}] = 1/u$.

Proof. From Lemma 2, we have $\sum_{R \in \mathcal{R}} \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{ideal} \wedge coin = R] \Pr[coin = R|\mathsf{ideal}] = 1/u \cdot \sum_{R \in \mathcal{R}} \Pr[coin = R|\mathsf{ideal}] = 1/u$. □

From Eqs. 1, 2 and Lemma 3, we have that

$$\Pr[\mathsf{win.cpa}] \geq \frac{1}{u} \cdot \Pr[\mathsf{win.cca}|\mathsf{ideal}] \Pr[\mathsf{ideal}] + \frac{1}{2} \cdot \frac{u-1}{u} \cdot \Pr[\mathsf{ideal}]. \tag{3}$$

Lemma 4. *We have that* $\Pr[\mathsf{ideal}] \geq 1 - \epsilon_{uf} - \epsilon_{tcr}$.

Proof. Event forge immediately implies forgery against SOTS Σ, and event find immediately implies collision in TCRHF h (namley, for given K and $vk^\star$, vk' such that $h_K(vk') = h_K(vk^\star)$ is found). Hence, $\Pr[\overline{\mathsf{ideal}}]$ is at most $\epsilon_{uf} + \epsilon_{tcr}$. □

From Eq. 3 and Lemma 4, we have that

$$\begin{aligned} \Pr[\mathsf{win.cpa}] &\geq \frac{1}{u} \cdot (\Pr[\mathsf{win.cca}] - \epsilon_{uf} - \epsilon_{tcr}) + \frac{1}{2} \cdot \frac{u-1}{u} \cdot (1 - \epsilon_{uf} - \epsilon_{tcr}) \\ &= \frac{1}{2} + \frac{1}{u}\epsilon_{cca} - \frac{u+1}{2u}(\epsilon_{uf} + \epsilon_{tcr}). \end{aligned}$$

From the above discussions, it can be easily seen that the claimed bound of the running time of B holds (notice that additional computational time of $O(u)$ is required for calculating t). This completes the proof of the theorem. □

3.4 A Variant for Key-Homomorphic Encryption

If the underlying PKE scheme Π has a certain homomorphic property, which is formally defined below, we can compress the component ciphertexts into one.

Definition 8 (Key Homomorphism). We say a PKE scheme Π is *key-homomorphic* for operations $\langle +, \cdot \rangle$ if for any two key pairs (dk_1, ek_1) and (dk_2, ek_2) of Π, $(dk_1 + dk_2, ek_1 \cdot ek_2)$ is also a key pair of Π.

Then, we can modify the encryption algorithm of the above scheme as follows: $(sk, vk) \leftarrow \mathsf{SGen}(1^k)$; $j \leftarrow h_K(vk)$; $C^{-} \leftarrow \mathsf{Enc}(\prod_{i \in \mathcal{F}_j} ek_i, M)$; $\sigma \leftarrow \mathsf{Sig}(sk, C^{-})$; output $C = (C^{-}, \sigma, vk)$. M can be recovered by using $\sum_{i \in \mathcal{F}_j} dk_i$. By this modification, the size of ciphertext can be significantly reduced. ElGamal is an example of key-homomorphic encryption schemes, and in Sec. 4, we further improve efficiency of the ElGamal-based construction by removing σ and vk.

3.5 Remarks

Here, we give some remarks which are worth discussing.

Necessity of Artificial Abort. As mentioned before, the artificial abort technique plays an important role in the security proof. This is primarily due to that we cannot calculate $\Pr[b' = b | \mathsf{succeed} \wedge \mathsf{ideal}]$ but $\Pr[b' = b | \mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{ideal}]$. See Eq. 2 for details.

Injective Maps as TCRHF. It is obvious that there is no collision in injective maps, and therefore a family of injective maps is considered a TCRHF without any assumption. More specifically, if $h : \{0,1\}^k \times \{0,1\}^{|vk|} \rightarrow \{1, ..., v\}$ is a family of such maps, it is $(t, 0)$-target collision resistant for any t. However, it should be noted that $v \geq 2^{|vk|}$ is a necessary condition for existence of injective maps.

Unsuitability of Trivial CFFs. In the above schemes, we notice that the underlying (u, v, q)-CFF is necessary to satisfy $u = O(\mathsf{poly}(k))$ due to its implementability. Recall that u is the number of key pairs which are used in the scheme. On the other hand, one might think that v is not required to be $\Omega(\mathsf{exp}(k))$, and that a trivial CFF (i.e. $(q+1, q+1, q)$-CFF) can be utilized. However, this is false. Namely, in our schemes $h : \{0,1\}^k \times \{0,1\}^{|vk|} \rightarrow \{1, ..., v\}$ has to be a TCRHF, and one can easily break its security if $v = O(\mathsf{poly}(k))$. This implies that trivial CFFs are not suitable.

Non-adaptively CCA-secure Construction. If only non-adaptive CCA security is required, the proposed scheme can be modified into significantly simpler one. The encryption algorithm of the modified scheme is as follows: $j \stackrel{R}{\leftarrow} \{1, ..., v\}$; devide M into $(M_i)_{i \in \mathcal{F}_j}$ such that $\bigoplus_{i \in \mathcal{F}_j} M_i = M$; $C_i \leftarrow \mathsf{Enc}(ek_i, M_i)$ for all $i \in \mathcal{F}_j$; output $C = (j, (C_i)_{i \in \mathcal{F}_j})$. Furthermore, in this scheme a trivial CFF (i.e. $(q+1, q+1, q)$-CFF) is available since security of h is not required to be taken into account. We should note that artificial abort is still necessary for proving the security of this modified scheme.

Public Verifiablity. In the proposed scheme, validity of a ciphertext is publicly verifiable, and therefore it is not difficult to extend this scheme to be a threshold cryptosystem. We note that Naor-Yung and its variants are also publicly verifiable.

4 Our ElGamal Variant with Optimal Ciphertext Length

In this section, we further reduce the ciphertext length of our proposed scheme by assuming more specific properties of the underlying CPA-secure PKE schemes. This results in CCA˜-secure PKE schemes with *optimal* ciphertext length, which means that its ciphertext size is the same as that for the underlying PKE schemes. Especially, we demonstrate a concrete example of this method by using ElGamal as the underlying PKE scheme, and show that the ciphertext overhead of the resulting scheme is only one group element, which is the same as the plain ElGamal.

4.1 Basic Idea

Here, we address a general idea for reducing ciphertext overhead of our construction in the previous section. As shown in Sec. 3.4, component ciphertexts can be compressed to be one if the underlying PKE scheme is key-homomorphic. Hence, optimal ciphertext length can be achieved if σ and vk are further removed. We notice that σ and vk are used to provide unduplicatable set selection, and therefore, they may be removed if there exists another way for providing this functionality rather than SOTS. In our proposed scheme, unduplicatable set selection without SOTS can be achieved by assuming additional property of the underlying PKE scheme Π such that a ciphertext of Π consists of two parts $C = (c_1, c_2)$, and c_1 and c_2 are computed as functions of only randomness *coin*, and of *ek*, M and *coin*, respectively. If Π has this property as well as key-homomorphism, then unduplicatable set selection can be provided by using c_1 part. ElGamal is an example of such an encryption scheme, and we next show a concrete construction.

4.2 Our ElGamal Variant

In this subsection, we propose a variant of ElGamal encryption which is CCA˜-secure under the DDH assumption in the standard model. Ciphertext overhead of the proposed scheme is only one group element which is considered optimal.

Let $\mathbb{G}$ be a multiplicative cyclic group of prime order p and g be a generator of $\mathbb{G}$. Let $h : \{0,1\}^k \times \mathbb{G} \to \{1, ..., v\}$ be a function family where the index space is $\{0,1\}^k$, $\pi : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a permutation family where the index space is $\{0,1\}^k$, and $f : \{0,1\}^k \times \mathbb{G}^2 \to \{0,1\}^k$ be a function family where the index space is $\{0,1\}^k$. In the security proof, h, π and f are viewed as a TCRHF (or, simply, a family of injective maps), an SPRP, and a KDF, respectively. Our ElGamal variant Π' consists of the following algorithms:

Our ElGamal Variant with Optimal Ciphertext Length

Key generation: For a given security parameter k,

1. generate a (u, v, q)-CFF $(\mathcal{L}, \mathcal{F})$ such that $\mathcal{L} = \{1, ..., u\}$,
2. generate $\mathbb{G}$, p and g,
3. $x_i \leftarrow \mathbb{Z}_p$ and $y_i \leftarrow g^{x_i}$ for $1 \le i \le u$, $K, K' \xleftarrow{R} \{0,1\}^k$,
4. output $dk = ((x_i)_{1\le i\le u}, \mathbb{G}, p, g, (\mathcal{L}, \mathcal{F}), K, K')$ and $ek = ((y_i)_{1\le i\le u}, \mathbb{G}, p, g, (\mathcal{L}, \mathcal{F}), K, K')$.

Encryption: For given ek and a plaintext $M \in \{0,1\}^\ell$,

1. $r \xleftarrow{R} \mathbb{Z}_p$, $c_1 \leftarrow g^r$, $j \leftarrow h_K(c_1)$, $y \leftarrow \prod_{i\in\mathcal{F}_j} y_i$,
2. $\overline{K} \leftarrow f_{K'}(c_1, y^r)$, $c_2 \leftarrow \pi_{\overline{K}}(M)$,

3. output $C = (c_1, c_2)$.

Decryption: For given dk and a ciphertext $C' = (c'_1, c'_2)$,

1. $j' \leftarrow h_K(c'_1)$, $x \leftarrow \sum_{i \in \mathcal{F}_{j'}} x_i$,
2. $\overline{K}' \leftarrow f_{K'}(c'_1, {c'_1}^x)$, $M' \leftarrow \pi_{\overline{K}'}(c'_2)$,
3. output M'.

The security of the above scheme is addressed as follows.

Theorem 3. *The above scheme Π' is $(t - u \cdot \tau_{exp}, q, 4(u \cdot \epsilon_{ddh} + \epsilon_{kdf} + \frac{u+1}{2}\epsilon_{tcr} + \epsilon_{sprp}))$-*IND-CCA *if $(t + \tau_{cff} + (u-1)\tau_{exp} + O(u), \epsilon_{ddh})$-DDH assumption holds, h is (t, ϵ_{tcr})-target collision resistant, f is (t, ϵ_{kdf})-indistinguishable, and π is (t, ϵ_{sprp})-strongly pseudorandom, where τ_{cff} and τ_{exp} are computational time for CFF generation and exponentiation over $\mathbb{G}$, respectively.*

4.3 Proof of Theorem 3

Now, we give a proof of Theorem 3. The proof idea is basically similar to Theorem 2, but the details are totally different. Actually, the proof consists of the following two steps: (1) first, we prove that the above scheme Π' is secure if $\overline{K}$ is totally unknown to any adversary and π is an SPRP, and (2) then, we prove that any adversary cannot obtain any partial knowledge of $\overline{K}$.

Security of Session Key. Now, we formally address adversary A's inability of extracting partial knowledge of $\overline{K}$ as follows. Let $\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k)$ denote the following experiment. For a randomly generated (dk, ek) of the above scheme Π', ek is given to an adversary A. For $c_1 \in \mathbb{G}$, let $\overline{K}(c_1)$ be $f_{K'}(c_1, c_1^x)$ where $x = \sum_{i \in \mathcal{F}_j} x_i$ and $j = h_K(c_1)$. For a random $b \in \{0, 1\}$, $(c_1^\star, \overline{K}^\star)$ is given to A where $\overline{K}^\star = \overline{K}(c_1^\star)$ if $b = 1$, or $\overline{K}^\star$ is a random k-bit string otherwise. The adversary is allowed to access a decryption oracle D which for given c_1, returns $\overline{K}(c_1)$. The only restriction is that $c_1^\star$ is not allowed to submit to D. Finally, A outputs his guess b'. Define $\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1$ if and only if $b' = b$. We assume that for any A in time bound t with at most q queries to D, $\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1$ with probability at most $\frac{1}{2} + \epsilon_{kem}$. We notice that this is based on the standard security definition of *key encapsulation mechanism* (KEM) [44].

Then, Theorem 3 can be immediately proved from the following lemmas.

Lemma 5. *The above scheme Π' is $(t - u \cdot \tau_{exp}, q, 4\epsilon_{kem} + 4\epsilon_{sprp})$-*IND-CCA *if π is (t, ϵ_{sprp})-strongly pseudorandom.*

Lemma 6. *In the above scheme Π', $\epsilon_{kem} \leq u \cdot \epsilon_{ddh} + \epsilon_{kdf} + \frac{u+1}{2}\epsilon_{tcr}$ if $(t + \tau_{cff} + (u-1)\tau_{exp} + O(u), \epsilon_{ddh})$-DDH assumption holds, h is (t, ϵ_{tcr})-target collision resistant, and f is (t, ϵ_{kdf})-indistinguishable.*

4.4 Proof of Lemma 5

Construction of the Simulator. The goal of our simulator B is to distinguish real session key $\overline{K}$ from a random key, or to distinguish an instance $\pi_{\overline{K}^\star}$ of permutation family π from a random permutation *perm*. B interacts with a chosen-ciphertext adversary A against the above scheme Π'. Before the simulation, B flips a random coin $\mathcal{COIN} \in \{0, 1\}$. If $\mathcal{COIN} = 0$, B tries to break security of session key, or else, B tries to break security of π.

If $\mathcal{COIN} = 0$, B works as follows. For a given public key ek of Π', B passes it to A. Challenge $(c_1^\star, \overline{K}^\star)$ is also given to B. For A's decryption query $C = (c_1(\neq c_1^\star), c_2)$, B asks $\overline{K} = \overline{K}(c_1)$ to B's own decryption oracle, and returns M such that $M = \pi_{\overline{K}}^{-1}(c_2)$. Also, for A's decryption query $C = (c_1^\star, c_2)$, B returns M such that $M = \pi_{\overline{K}^\star}^{-1}(c_2)$. When A submits M_0 and M_1, B picks a random $\beta \in \{0,1\}$, computes $c_2^\star = \pi_{\overline{K}^\star}(M_\beta)$, and returns $C^\star = (c_1^\star, c_2^\star)$ as a challenge ciphertext. B outputs 1 if A's output is identical to β, or 0 otherwise.

If $\mathcal{COIN} = 1$, B works as follows. B generates key pair (dk, ek) of Π', B passes ek to A. B also picks a random $c_1^\star$ from $\mathbb{G}$. B responds to A's decryption query $C = (c_1(\neq c_1^\star), c_2)$ by decrypting it with dk. Also, for A's decryption query $C = (c_1^\star, c_2)$, B submits c_2 to B's own permutation oracle, which is $\pi_{\overline{K}^\star}^{-1}$ or $perm^{-1}$. B returns the oracle's answer to A. When A submits M_0 and M_1, B picks a random $\beta \in \{0,1\}$, and sends M_β to B's own permutation oracle, which is $\pi_{\overline{K}^\star}$ or $perm$. Let the oracle's answer be $c_2^\star$. B returns $C^\star = (c_1^\star, c_2^\star)$ as a challenge ciphertext. B outputs 1 if A's output is identical to β, or 0 otherwise.

Estimating Probabilities. In the above simulation, B's views for [$\overline{K}^\star$ is random $\wedge\, \mathcal{COIN} = 0$] and [given permutation is $\pi_{\overline{K}^\star} \wedge \mathcal{COIN} = 1$] are identical. Let $\Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{B},\Pi'}(k) = 0 | \overline{K}^\star$ is random $\wedge\, \mathcal{COIN} = 0] = \Pr[\mathsf{Exp}^{\mathsf{sprp}}_{\mathsf{B},\pi}(k) = 1 | \mathcal{COIN} = 1] = \frac{1}{2} + \lambda$. Then, the probability of $\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{B},\Pi'}(k) = 1$ is estimated as follows.

$$\begin{aligned}
\Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{B},\Pi'}(k) = 1] &= \frac{1}{2} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{B},\Pi'}(k) = 1 | \mathcal{COIN} = 0] + \frac{1}{2} \cdot \frac{1}{2} \\
&= \frac{1}{4} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{B},\Pi'}(k) = 1 | \overline{K}^\star = \overline{K}(c_1^\star) \wedge \mathcal{COIN} = 0] \\
&\quad + \frac{1}{4} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{B},\Pi'}(k) = 1 | \overline{K}^\star \text{ is random} \wedge \mathcal{COIN} = 0] + \frac{1}{4} \\
&\leq \frac{1}{4} \cdot (\frac{1}{2} + \epsilon_{cca}) + \frac{1}{4} \cdot (\frac{1}{2} - \lambda) + \frac{1}{4} \\
&= \frac{1}{2} + \frac{1}{4}(\epsilon_{cca} - \lambda).
\end{aligned}$$

Similarly, the probability of $\mathbf{Exp}^{\mathsf{sprp}}_{\mathsf{B},\pi}(k) = 1$ is estimated as follows.

$$\begin{aligned}
\Pr[\mathbf{Exp}^{\mathsf{sprp}}_{\mathsf{B},\pi}(k) = 1] &= \frac{1}{2} \cdot \Pr[\mathbf{Exp}^{\mathsf{sprp}}_{\mathsf{B},\pi}(k) = 1 | \mathcal{COIN} = 1] + \frac{1}{2} \cdot \frac{1}{2} \\
&= \frac{1}{2} \cdot (\frac{1}{2} + \lambda) + \frac{1}{4} = \frac{1}{2} + \frac{1}{2}\lambda.
\end{aligned}$$

Since $\Pr[\mathbf{Exp}^{\mathsf{rnd}}_{\mathsf{B},\pi}(k) = 1] = \frac{1}{2}$, we have that

$$\begin{aligned}
\epsilon_{sprp,\mathsf{B}} &= \frac{1}{2} | \Pr[\mathbf{Exp}^{\mathsf{sprp}}_{\mathsf{B},\pi}(k) = 1] - \Pr[\mathbf{Exp}^{\mathsf{rnd}}_{\mathsf{B},\pi}(k) = 1] | \\
&= \frac{1}{2} | \frac{1}{2} + \frac{1}{2}\lambda - \frac{1}{2} | = \frac{1}{4}|\lambda|.
\end{aligned}$$

From the assumptions, we have that

$$\epsilon_{kem} \geq \frac{1}{4}(\epsilon_{cca} - \lambda), \quad \epsilon_{sprp} \geq \frac{1}{4}|\lambda|.$$

Consequently, we have that

$$\epsilon_{cca} \leq 4\epsilon_{kem} + \lambda \leq 4\epsilon_{kem} + 4\epsilon_{sprp}.$$

From the above discussions, it can be easily seen that the claimed bound of the running time of B holds. This completes the proof of the lemma. □

4.5 Proof of Lemma 6

Construction of the Simulator. The goal of our simulator B is to distinguish a Diffie-Hellman tuple from random one by using a chosen-ciphertext adversary A which can distinguish a valid session key $\overline{K}$ of Π' from a random one. For a given DDH instance $(g, g^{\alpha}, g^{\beta}, g^{\gamma}) \in \mathbb{G}^4$, B generates a (u, v, q)-CFF $(\mathcal{L}, \mathcal{F})$ picks a random a from $\{1, ..., u\}$, and sets $y_a \leftarrow g^{\alpha}$. Then, B generates (x_i, y_i) for all $i \in \{1, ..., u\} \backslash \{a\}$ such that $y_i = g^{x_i}$. B also picks randoms K, K' from $\{0,1\}^k$.

Next, B passes $ek = ((y_i)_{1 \le i \le u}, \mathbb{G}, p, g, (\mathcal{L}, \mathcal{F}), K, K')$ to A as an encryption key of Π'. B also gives challenge $(c_1^{\star}, \overline{K}^{\star})$ to A, where

$$c_1^{\star} = g^{\beta}, \quad \overline{K}^{\star} = f_{K'}(g^{\beta}, g^{\gamma} \cdot \prod_{i \in \mathcal{F}_{j^{\star}} \backslash \{a\}} (g^{\beta})^{x_i}),$$

and $j^{\star} = h_K(c_1^{\star})$. B responds to A's decryption query c_1 as follows. If $a \in \mathcal{F}_j$ such that $j = h_K(c_1)$, B aborts the simulation and outputs a random $b' \in \{0,1\}$. For $c_1 \in \mathbb{G}$, let $\overline{K}(c_1)$ be $f_{K'}(c_1, c_1^{x})$ where $x = \sum_{i \in \mathcal{F}_j} x_i$ and $j = h_K(c_1)$. If $a \notin \mathcal{F}_j$, B computes $\overline{K}(c_1)$ and returns it to A.

Artificial Abort. Let c_1^{ℓ} $(1 \le \ell \le q)$ be A's decryption queries, and j^{ℓ} be $h_K(c_1^{\ell})$. Then, B calculates the following value t:

$$t = |\mathcal{F}_{j^{\star}} \backslash \cup_{\ell \in \{1,...,q\}} \mathcal{F}_{j^{\ell}}|.$$

When A outputs b', then B outputs b' with probability $\frac{1}{t}$, or a random bit with probability $\frac{t-1}{t}$.

Estimating Probabilities. Now, we estimate $\epsilon_{ddh,\mathsf{B}} = \frac{1}{2}|\Pr[\mathbf{Exp}^{\mathsf{dh}}_{\mathsf{B},\mathbb{G}}(k) = 1] - \Pr[\mathbf{Exp}^{\mathsf{rnd}}_{\mathsf{B},\mathbb{G}}(k) = 1]|$. Let real denote an event that real session key $\overline{K}$ is given to A, random denote an event that a random k-bit string is given to A, fake denote an event that $f_K(g^{\beta}, h_1)$ is given to A (instead of a random k-bit string) where $h_1 \overset{R}{\leftarrow} \mathbb{G}$, $\mathsf{succeed}$ denote an event that B does not abort the simulation and $a \in \mathcal{F}_{j^{\star}}$, find denote an event that A submits a decryption query $c_1 (\neq c_1^{\star})$ such that $h_K(c_1) = h_K(c_1^{\star})$, and $\mathsf{a.abort}$ denote an event that B outputs a random bit in the artificial abort phase. Then, we have that

$$\begin{aligned} \Pr[\mathbf{Exp}^{\mathsf{dh}}_{\mathsf{B},\mathbb{G}}(k) = 1] &\ge \Pr[\mathbf{Exp}^{\mathsf{dh}}_{\mathsf{B},\mathbb{G}}(k) = 1|\overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}] \\ &\ge \Pr[b' = 1|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{real} \wedge \overline{\mathsf{find}}] \\ &\quad \cdot \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}] \\ &\quad + \frac{1}{2} \Pr[\overline{\mathsf{succeed}} \vee \mathsf{a.abort}|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}]. \end{aligned} \tag{4}$$

Let *coin* be A's random coin, and $\mathcal{R}$ be the set of all possible values of *coin*. Then, we have the following lemmas:

Lemma 7. *For all $R \in \mathcal{R}$, we have that*

$$\Pr[b' = 1|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] = \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R].$$

Proof. This is obvious since B's simulation is perfect if $\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}$ is true. □

Lemma 8. *For all $R \in \mathcal{R}$, we have that*

$$\Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] = \frac{1}{u}.$$

Proof. Let $Suc(R)$ denote $\Pr[\mathsf{succeed}|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R]$, and $t(R)$ be the value of t (see **Artificial Abort.**) according to the decryption queries by A with coin R. If $\overline{\mathsf{find}}$ is true, there always exists non-empty subset $\mathcal{D}$ of $\{1, ..., u\}$ such that $\mathcal{D} \subseteq \mathcal{F}_{j^\star}$ and $|\mathcal{D} \cap \cup_{\ell \in \{1,...,q\}} \mathcal{F}_{j^\ell}| = 0$ due to the property of the (u, v, q)-CFF. This implies that $\mathsf{succeed}$ occurs if $a \in \mathcal{D}$, and therefore, $Suc(R) = t(R)/u$. On the other hand, $\Pr[\overline{\mathsf{a.abort}}|\mathsf{succeed} \wedge \mathsf{ideal} \wedge coin = R] = 1/t(R)$ by definition. Then we have that

$$\Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] = \frac{1}{t(R)} \cdot \frac{t(R)}{u} = \frac{1}{u},$$

which proves the lemma. □

From Lemmas 7 and 8, we have that

$$\begin{aligned}
&\Pr[b' = 1|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}] \\
&= \sum_{R \in \mathcal{R}} (\Pr[b' = 1|\mathsf{succeed} \wedge \overline{\mathsf{a.abort}} \wedge \mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] \\
&\qquad \cdot \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] \Pr[coin = R|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}]) \\
&= \sum_{R \in \mathcal{R}} (\Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] \cdot \frac{1}{u} \cdot \Pr[coin = R|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}]) \\
&= \frac{1}{u} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}]. \qquad (5)
\end{aligned}$$

Lemma 9. *We have that* $\Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}}] = 1/u$.

Proof. From Lemma 8, we have $\sum_{R \in \mathcal{R}} \Pr[\mathsf{succeed} \wedge \overline{\mathsf{a.abort}}|\mathsf{real} \wedge \overline{\mathsf{find}} \wedge coin = R] \Pr[coin = R|\mathsf{real} \wedge \overline{\mathsf{find}}] = 1/u \cdot \sum_{R \in \mathcal{R}} \Pr[coin = R|\mathsf{real} \wedge \overline{\mathsf{find}}] = 1/u$. □

From Eqs. 4, 5 and Lemma 9, we have that

$$\begin{aligned}
\Pr[\mathbf{Exp}^{\mathsf{dh}}_{\mathsf{B},\mathbb{G}}(k) = 1] &\geq \frac{1}{u} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real} \wedge \overline{\mathsf{find}}] \Pr[\overline{\mathsf{find}}] + \frac{1}{2} \cdot \frac{u-1}{u} \cdot \Pr[\overline{\mathsf{find}}] \\
&\geq \frac{1}{u} \cdot (\Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real}] - \epsilon_{tcr}) + \frac{1}{2} \cdot \frac{u-1}{u} \cdot (1 - \epsilon_{tcr}) \\
&= \frac{1}{u} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real}] + \frac{u-1}{2u} - \frac{u+1}{2u}\epsilon_{tcr}. \qquad (6)
\end{aligned}$$

By a similar discussion as the above, we also have that

$$\Pr[\mathbf{Exp}^{\mathsf{rnd}}_{\mathsf{B},\mathbb{G}}(k) = 0] \geq \frac{1}{u} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 0|\mathsf{fake}] + \frac{u-1}{2u} - \frac{u+1}{2u}\epsilon_{tcr}. \qquad (7)$$

Lemma 10. *We have that* $\frac{1}{2}|\Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 0|\mathsf{random}] - \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 0|\mathsf{fake}]| \leq \epsilon_{kdf}$.

Proof. It is easy to construct an algorithm which can distinguish an output of f_K from a random string by using A with advantage more than ϵ_{kdf} if the above statement is not true. □

From Eqs. 6, 7 and Lemma 10, we have that

$$\begin{aligned}
&\frac{1}{2}|\Pr[\mathbf{Exp}^{\mathsf{dh}}_{\mathsf{B},\mathbb{G}}(k) = 1] - \Pr[\mathbf{Exp}^{\mathsf{rnd}}_{\mathsf{B},\mathbb{G}}(k) = 1]| \\
&\qquad \geq \frac{1}{2}|\frac{1}{u} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 1|\mathsf{real}] + \frac{u-1}{2u} - \frac{u+1}{2u}\epsilon_{tcr} \\
&\qquad\qquad -(1 - \frac{1}{u} \cdot \Pr[\mathbf{Exp}^{\mathsf{kem}}_{\mathsf{A},\Pi'}(k) = 0|\mathsf{fake}] - \frac{u-1}{2u} + \frac{u+1}{2u}\epsilon_{tcr})| \\
&\qquad \geq \frac{1}{u}\epsilon_{kem} - \frac{1}{u}\epsilon_{kdf} - \frac{u+1}{2u}\epsilon_{tcr}.
\end{aligned}$$

Consequently, we have that $\epsilon_{kem} \leq u \cdot \epsilon_{ddh} + \epsilon_{kdf} + \frac{u+1}{2}\epsilon_{tcr}$.

From the above discussions, it can be easily seen that the claimed bound of the running time of B holds (notice that additional computational time of $O(u)$ is required for calculating t). This completes the proof of the theorem. □

4.6 Remarks

Here, we give some remarks on our ElGamal variant.

Reasonability of Assumptions. As discussed above, our ElGamal variant is secure under the DDH assumption, and existence of a TCRHF, an SPRP, and a KDF. The DDH assumption is widely believed to be reasonable, and many other schemes, e.g. [20, 17, 33], are based on this assumption. A TCRHF can be derived from any one-way function. An SPRP can be derived from any PRF by using the four-round Feistel construction. A KDF can be constructed without any computational assumption by using leftover hash lemma. (In practice, one can simply assume that a dedicated cryptographic hash function works as a KDF.) Hence, assumptions that are required in our scheme are considered reasonable.

Compressing Keys. Though the proposed scheme is optimal in terms of ciphertext overhead, the sizes of both decryption and encryption keys are large. However, the size of a decryption key can be significantly reduced as follows. Let $f : \{0,1\}^k \times \{1,...,u\} \to \mathbb{Z}_p$ be a PRF, $mst \overset{R}{\leftarrow} \{0,1\}^k$ be a "master" decryption key, and f_{mst} be an instance of f indexed by mst. Let $x_i \leftarrow f_{mst}(i)$. Then, $(x_i)_{1\leq i\leq u}$ can be used as (a part of) the decryption key of the proposed scheme. By this modification, a user has to keep only mst as his decryption key. On the other hand, it is difficult to compress an encryption key, and this is still an open problem. In Sec. 6, we show a partial solution for it in which a sender does not need to store the receiver's public key but a short *state* instead [3].

Extending Message Space. For a plaintext M with $|M| = n \cdot \ell$ for arbitrary $n \geq 2$, it is possible to extend the message space by using the CMC mode of operation [27]. This method does not require any additional assumption.

5 Comparison

Here, we discuss comparison among ElGamal variants in terms of ciphertext size, and it is summarized in Table 1. So far, under the DDH assumption the Kurosawa-Desmedt scheme [33] is considered as the most efficient scheme, and its ciphertext overhead is two group elements and an authentication tag. We note that it is difficult to remove the authentication tag from the Kurosawa-Desmedt scheme even if an SPRP is used for the data encapsulation part (like our proposed scheme). This is primarily due to that the key encapsulation part of the Kurosawa-Desmedt scheme is not CCA-secure [28]. Under stronger assumptions, e.g. the DBDH assumption [14] or the square-DBDH assumption [32], the authentication tag can be removed [11, 32]. However, even in these schemes there are still two group elements as ciphertext overhead. Furthremore, these schemes require bilinear groups. In our ElGamal variant, its ciphertext overhead is only one group element which can be considered as optimal since this is the same as the original ElGamal scheme. Moreover, its underlying assumption is the DDH assumption, and it does not require bilinear groups. This result implies that it is possible to construct a CCA-secure scheme from an underlying CPA-secure scheme in the standard model without redundancy if the upper bound of the number of decryption queries is known, e.g. 2^{30}.

Table 1: Efficiency comparison for ElGamal variants in terms of ciphertext size, where $|g|$, $|mac|$, and $|M|$ are sizes for a group element, an authentication tag, and a plaintext, respectively. For concreteness, one can think of $|mac| = 128$. DBDH and square-DBDH means the *decisional bilinear Diffie-Hellman assumption* [14] and the *decisional square bilinear Diffie-Hellman assumption* [32], respectively.

	ciphertext size	assumption	security	bilinear group
ElGamal [20]	$2\|g\|$ $(\|M\| \leq \|g\|)$	DDH	CPA-secure	–
Cramer-Shoup [17]	$4\|g\|$ $(\|M\| \leq \|g\|)$	DDH	CCA-secure	–
Shoup [44]	$3\|g\| + \|mac\| + \|M\|$	DDH	CCA-secure	–
Kurosawa-Desmedt [33]	$2\|g\| + \|mac\| + \|M\|$	DDH	CCA-secure	–
Boyen-Mei-Waters [11]	$2\|g\| + \|M\|$	DBDH	CCA-secure	necessary
Kiltz [32]	$2\|g\| + \|M\|$	square-DBDH	CCA-secure	necessary
Proposed (Sec. 4)	$\|g\| + \|M\|$	DDH	CCA$^-$-secure	–

6 Compressing Public Key: a Partial Solution

In our ElGamal variant, the size of a public key is large, and hence it is not very practical. However, this problem can be partially solved by the *stateful public key encryption* technique [3]. Notice that the main motivation of [3] is to reduce computational costs for encryption, but in our scheme we apply this technique for compressing a public key. As a result, a sender only needs to generate and store a short private *state* which is significantly compressed from a receiver's public key.

6.1 Basic Idea

In our ElGamal variant, a sender picks a random r and generates $\overline{K}$ which is computed as a function of r and a receiver's public key. Our basic idea is to store g^r and $\overline{K}$ as the sender's private state, and re-use them for multiple times. Then, the sender does not need to store nor read the huge public key any more by securely keeping his state. However, we should note that the resulting scheme is totally insecure if the above idea is straightforwardly applied, and therefore a careful discussion is required. We deal with this issue by replacing an SPRP with a *randomized symmetric key encryption which is secure against both chosen-plaintext and chosen-ciphertext attacks* [4] (like [3]).

6.2 Definitions

$\mathsf{IND\text{-}CCA}$ for Stateful Setting. Due to the existance of the private state, the security definition is required to be slightly modified. Namely, in addition to the original CCA environment, an adversary is also given an encryption oracle which for a given plaintext M, returns its corresponding ciphertext C under a fixed state. The challenge ciphertext $C^\star$ is also generated under the same state. For details of the definition, see [3]. We note that the authors of [3] also consider multi-sender setting, but for our scheme it is sufficient to investigate a simpler setting where we can assume there exists only one sender.

CPA&CCA-Secure Symmetric Key Encryption. A symmetric key encryption (SKE) scheme Π consists of two PPT algorithms: $\Pi = (\mathsf{SEnc}, \mathsf{SDec})$. The encryption algorithm SEnc takes as inputs $K \in \{0,1\}^k$ and $M \in \mathcal{M}$, and outputs ciphertext $C \in \mathcal{C}$, where k is a security parameter, and $\mathcal{M}$ and $\mathcal{C}$ are the plaintext and ciphertext spaces, respectively. The decryption algorithm SDec takes as inputs K and C, and outputs M or $\perp$, where $\perp$ is a distinguished symbol. We require that for all $K \in \{0,1\}^k$, all M, and $C(= \mathsf{SEnc}(K, M))$, $\mathsf{SDec}(K, C) = M$.

CPA&CCA-security of SKE [4] is defined as follows. Let $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ and k be an adversary and the security parameter, respectively. Consider the following experiment:

$$\begin{aligned}\mathbf{Exp}^{\mathsf{ske}}_{\mathsf{A},\Pi}(k) : [K \overset{R}{\leftarrow} \{0,1\}^k; (M_0, M_1, s) \leftarrow \mathsf{A}_1^{\mathsf{D},\mathsf{E}}; b \overset{R}{\leftarrow} \{0,1\}; C^\star \leftarrow \mathsf{SEnc}(K, M_b);\\ b' \leftarrow \mathsf{A}_2^{\mathsf{D},\mathsf{E}}(C^\star, s); \text{return } 1 \text{ if } b' = b, \text{ or } 0 \text{ otherwise}],\end{aligned}$$

where D is a decryption oracle which for given C, returns $M(\text{or } \perp)(= \mathsf{SDec}(K, C))$, and E is an encryption oracle which for given M, returns $C(= \mathsf{SEnc}(K, M))$. The only restriction is that $C^\star$ is not allowed to submit to D. We define $\epsilon_{ske,\mathsf{A}} = |\Pr[\mathbf{Exp}^{\mathsf{ske}}_{\mathsf{A},\Pi}(k) = 1] - 1/2|$.

Definition 9 (IND-CPA&CCA)**.** We say Π is (t, q_d, q_e, ϵ)-IND-CPA&CCA *secure*, if for any A in time bound t with at most q_d and q_e queries to D and E, respectively, $\epsilon_{ske,\mathsf{A}} \leq \epsilon$. We say that Π is *CPA&CCA-secure* if ϵ is negligible.

It is easy to construct a CPA&CCA-secure SKE by a generic encrypt-then-mac composition [4] with, for example, AES and HMAC. In this case, its ciphertext overhead will be 80 bits (for IV of AES) + 128 bits (for MAC).

6.3 Construction

Construction of the stateful scheme is the same as our ElGamal variant except for that

- SPRP π is replaced with CPA&CCA-secure SKE (SEnc,SDec). Namely, **Encryption** algorithm is modified as $c_2 \leftarrow \mathsf{SEnc}(\overline{K}, M)$, and **Decryption** algorithm is modified as $M' \leftarrow \mathsf{SDec}(\overline{K'}, c_2')$.
- In **Encryption** algorithm, r is picked from $\mathbb{Z}_p$ uniformly at random for the first time. Then, the sender keeps only $(g^r, \overline{K})$ as his private state, and erases ek from his storage. From the second time, for encrypting a message M', he only computes $c_2 \leftarrow \mathsf{SEnc}(\overline{K}, M')$ and outputs $C = (c_1, c_2)$ where $c_1 = g^r$.

Security of the above scheme can be easily proven by a straightforward combination of proofs of Theorem 3 of this paper and Theorem 4.1 of [3]. More specifically, for proving IND-CCA for stateful setting, it is necessary to simulate an encryption oracle under a fixed state (see Sec. 6.2), and this can not be straightforwardly done since a given DDH instance is embedded to the state and therefore the simulator does not know it. However, we can cope with this problem by using properties of a CPA&CCA-secure SKE with the same proof technique as [3].

6.4 Performance

In the above scheme, a huge public key ek is significantly compressed into a short private state $(g^r, \overline{K})$ by each sender. Notice that a sender does not need to keep "g^r" part as secret, and can store it even in a public storage. Due to the use of a CPA&CCA-secure SKE, ciphertext overhead increases for 208 bits from the original (stateless) scheme. However, the total size of a ciphertext is still shorter than that of Kurosawa-Desmedt.

Acknowledgement

The authors would like to thank Nuttapong Attrapadung, Yang Cui, Jun Furukawa, Eike Kiltz, Rui Zhang for their comments and suggestions. The authors also would like to thank Steven Myers for clarifying consistency between [24] and our work, and Rafael Pass for sending their draft of [39] to us.

References

[1] J.H. An, Y. Dodis, and T. Rabin, “On the security of joint signature and encryption,” Proc. of Eurocrypt’02, pp.83-107, 2002.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, “Relations among notions of security for public-key encryption schemes,” Proc. of CRYPTO’98, pp.26-45, 1998.

[3] M. Bellare, T. Kohno, and V. Shoup, “Stateful public-key cryptosystems: how to encrypt with one 160-bit exponentiation,” to appear in Proc. of CCS’06.

[4] M. Bellare and C. Namprempre, “Authenticated encryption: relations among notions and analysis of the generic composition paradigm,” Proc. of Asiacrypt’00, pp.531-545, 2000.

[5] M. Bellare and P. Rogaway, “Random oracles are practical: a paradigm for designing efficient protocols,” Proc. of CCS’93, pp.62-73, 1993.

[6] M. Bellare and P. Rogaway, “Optimal asymmetric encryption,” Proc. of Eurocrypt’94, pp.92-111, 1994.

[7] M. Blum, P. Feldman, and S. Micali, “Non-interactive zero-knowledge and its applications,” Proc. of STOC’88, pp.103-112, 1988.

[8] D. Boneh and X. Boyen, “Efficient selective-ID secure identity-based encryption without random oracles,” Proc. of Eurocrypt’04, pp.223-238, 2004.

[9] D. Boneh and M.K. Franklin, “Identity-based encryption from the Weil pairing,” Proc. of CRYPTO’01, pp.213-229, 2001.

[10] D. Boneh and J. Katz, “Improved efficiency for CCA-secure cryptosystems built using identity-based encryption,” Proc. of CT-RSA’05, pp.87-103, 2005.

[11] X. Boyen, Q. Mei, and B. Waters, “Direct chosen ciphertext security from identity-based techniques,” Proc. of CCS’05, pp.320-329, 2005.

[12] R. Canetti, “Universally composable security: a new paradigm for cryptographic protocols,” Proc. of FOCS’01, pp.136-145, 2001.

[13] R. Canetti, O. Goldreich, and S. Halevi, “The random oracle methodology, revisited,” Proc. of STOC’98, pp.209-218, 1998.

[14] R. Canetti, S. Halevi, and J. Katz, “A forward-secure public-key encryption scheme,” Proc. of Eurocrypt’03, pp.255-271, 2003.

[15] R. Canetti, S. Halevi, and J. Katz, “Chosen-ciphertext security from identity-based encryption,” Proc. of Eurocrypt’04, pp.207-222, 2004.

[16] R. Cramer, D. Hofheinz, and E. Kiltz, “A note on bounded chosen ciphertext security from black-box semantical security,” manuscript.

[17] R. Cramer and V. Shoup, “A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack,” Proc. of CRYPTO’98, pp.13-25, 1998.

[18] R. Cramer and V. Shoup, “Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption,” Proc. of Eurocrypt’02, pp.45-64, 2002.

[19] D. Dolev, C. Dwork, and M. Naor, “Non-malleable cryptography,” Proc. of STOC’91, pp. 542-552, 1991.

[20] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” IEEE Trans. on Inform. Theory, 31(4), pp.469-472, 1985.

[21] P. Erdös, P. Frankl and Z. Furedi, “Families of finite sets in which no sets is covered by the union of two others,” J. of Combin. Theory Ser. A 33, pp.158-166, 1982.

[22] P. Erdös, P. Frankl and Z. Furedi, “Families of finite sets in which no sets is covered by the union of r others,” Israel Journal of Math., 51, pp.79-89, 1985.

[23] E. Fujisaki and T. Okamoto, “Secure integration of asymmetric and symmetric encryption schemes,” Proc. of CRYPTO’99, pp.537-554, 1999.

[24] Y. Gertner, T. Malkin, and S. Myers, “Towards a separation of semantic and CCA security for public key encryption,” to appear in Proc. of TCC’07.

[25] S. Goldwasser and S. Micali, “Probabilistic encryption,” J. Comput. Syst. Sci., 28(2), pp.270-299, 1984.

[26] S. Goldwasser, S. Micali, and R.L. Rivest, “A digital signature scheme secure against adaptive chosen-message attacks,” SIAM J. Comput., 17(2), pp.281-308, 1988.

[27] S. Halevi and P. Rogaway, “A tweakable enciphering mode,” Proc. of CRYPTO’03, pp.482-499, 2003.

[28] J. Herranz, D. Hofheinz, and E. Kiltz, “The Kurosawa-Desmedt key encapsulation is not chosen-ciphertext secure,” Cryptology ePrint Archive, 2006/207, 2006.

[29] R. Impagliazzo, L.A. Levin, and M. Luby, “Pseudo-random generation from one-way functions,” Proc. of STOC’89, pp.12-24, 1989.

[30] R. Impagliazzo and D. Zuckerman, “How to recycle random bits,” Proc. of FOCS’89, pp.248-253, 1989.

[31] E. Kiltz, “Chosen-ciphertext security from tag-based encryption,” Proc. of TCC’06, pp.581-600, 2006.

[32] E. Kiltz, “On the limitations of the spread of an IBE-to-PKE transformation,” Proc. of PKC’06, pp.274-289, 2006.

[33] K. Kurosawa and Y. Desmedt, “A new paradigm of hybrid encryption scheme,” Proc. of CRYPTO’04, pp.426-442, 2004.

[34] Y. Lindell, “A simpler construction of CCA2-secure public-key encryption under general assumptions,” Proc. of Eurocrypt’03, pp.241-254, 2003.

[35] M. Luby and C. Rackoff, “How to construct pseudorandom permutations from pseudorandom functions,” SIAM J. Comput., 17(2), pp.373-386, 1988.

[36] M. Naor and M. Yung, “Universal one-way hash functions and their cryptographic applications,” Proc. of STOC’89, pp.33-43, 1989.

[37] M. Naor and M. Yung, “Public-key cryptosystems provably secure against chosen ciphertext attacks,” Proc. of STOC’90, pp.427-437, 1990.

[38] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” Proc. of Eurocrypt’99, pp.223-238, 1999.

[39] R. Pass, a. shelat, and V. Vaikuntanathan, “Bounded-CCA secure non-malleable encryption,” manuscript.

[40] C. Rackoff and D.R. Simon, “Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack,” Proc. of CRYPTO’91, pp.433-444, 1991.

[41] J. Rompel, “One-way functions are necessary and sufficient for secure signatures,” Proc. of STOC’90, pp.387-394, 1990.

[42] A. Sahai, “Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security,” Proc. of FOCS’99, pp.543-553, 1999.

[43] A. Shamir, “Identity-based cryptosystems and signature schemes,” Proc. of Crypto’84, LNCS 196, Springer-Verlag, pp.47-53, 1985.

[44] V. Shoup, “Using hash functions as a hedge against chosen ciphertext attack,” Eurocrypt’00, pp.275-288, 2000.

[45] B. Waters, “Efficient identity based encryption without random oracles,” Proc. of Eurocrypt’05, LNCS 3494, Springer-Verlag, pp.114-127, 2005.