COMPRESS MULTIPLE CIPHERTEXTS USING ELGAMAL ENCRYPTION SCHEMES

MYUNGSUN KIM, JIHYE KIM, AND JUNG HEE CHEON

Abstract. In this work we deal with the problem of how to squeeze multiple ciphertexts without losing original message information. To do so, we formalize the notion of decomposability for public-key encryption and investigate why adding decomposability is challenging. We construct an ElGamal encryption scheme over extension fields, and show that it supports the efficient decomposition. We then analyze security of our scheme under the standard DDH assumption, and evaluate the performance of our construction.

1. Introduction

ElGamal encryption is one of fundamental public-key cryptosystems. One of its main advantages is that it is simple and efficient, but also that its chosen-plaintext security is clearly understood. Security overhead in terms of bandwidth, however, often becomes obstacles against its publicly wide use. ElGamal ciphertexts are typically at least as many bits as the prime modulus p. If the plaintext size is small comparatively to the size of p (as we can see in many practical scenarios), the relative size overhead becomes worse.

For example, assuming p is a 2048-bit prime, in a hybrid encryption scenario where a symmetric key size is 256-bit, the size overhead of ElGamal is roughly ten times the plaintext size. The situation becomes worse, as more computational power and more powerful mathematical analytic methods become available in the future, which results in a longer public key size. More specifically, let consider the case where a server should receive from multiple clients each shared secret-key encrypted under hybrid encryption. If the number of clients grows linearly in the above example, the size overhead also increases linearly. Thus, efficiency can be improved if a router is given an accessory to compress multiple ciphertexts. It might be possible to improve efficiency by optimizing a particular encryption algorithm. Instead, we focus on the way where anyone can have ability to optimize bandwidth overhead. As an immediate application of our proposal we can consider the case in which needs to aggregate ciphertexts from multiple sources, for example, a network of low-cost sensor nodes that send sensitive data over the internet to a recipient.

In this work, we study how to reduce unused spaces in ElGamal encryption. The unused space results from imbalance between the real plaintext size used during encryption and the prescribed plaintext size that a ciphertext can cover. We call this unutilized space *ciphertext overhead* – the size difference between a ciphertext and its embedded plaintext. When a single sender generates multiple ciphertexts, there are several ways to reduce the ciphertext overhead. (Refer to Related work for details.) On the other hand, ciphertext overhead considerably increases in a distributed setting where each ciphertext is generated by a different sender on a relatively small plaintext in the size. We focus on how to decrease the ciphertext overhead in the setting where *multiple ciphertexts* are generated from *distributed senders* to the same receiver.

2010 *Mathematics Subject Classification.* Primary: 94A60; Secondary: 11T71,14G50.
Key words and phrases. Ciphertext compression, Decomposability, Message rate.

1.1. **Our Contributions.** We show how to efficiently compress multiple ciphertexts and how to efficiently decompress each individual plaintexts. We define the notion of decomposability over semantically secure encryption to formalize decompression of a compressed ciphertext. We construct an ElGamal variant over extension fields to support efficient decomposability. We define message rate in order to measure the compression efficiency and analyze the message rate of our constructions.

1.2. **Related Work.** In a setting of a single arbitrary message, compact encryption [29, 1, 2] schemes can be a solution. Compact encryption allows to have an optimal ciphertext overhead in this setting. One of well-known compact encryption schemes is as follows: given a group $\mathbb{G}$ of prime order p with a generator g and a public/secret key pair $(y = g^x, x)$, a ciphertext for a plaintext m is $(g^r, m \oplus H(y^r))$ where a random r is chosen from $\mathbb{Z}_q^\times$ and a hash function $H : \mathbb{G} \to \{0,1\}^{|m|}$. The ciphertext overhead contains only one group element, regardless of the size of the plaintext. Moreover, when multiple ciphertexts are generated by a sender in sequence, the compact encryption scheme allows an optimal ciphertext overhead. When a new plaintext m_i is given, a sender computes $(\cdots((H(y^r) \oplus m_1) \oplus m_2 2^{|m_1|}) \oplus \cdots) \oplus m_i 2^{\sum_{j=1}^{i-1}|m_j|}$ by using $H : \mathbb{G} \to \{0,1\}^{\sum_{j=1}^{i}|m_j|}$. Then it is clear the compact encryption scheme has an optimal ciphertext overhead. Given multiple ciphertexts generated from distributed sources, however, the size is not optimal any more because the overhead increases linearly with the number of messages. Moreover, it is not clear how to compress ciphertexts because the group structure is broken by the hash function.

Gentry [13] proposed a scheme to compress Rabin ciphertexts and signatures (among other things) down to about $(2/3)\log N$ bits while ordinarily, RSA and Rabin ciphertexts are $\log N$ bits, where N is a composite modulus.

An encoding scheme proposed by Catagnos and Chevallier-Mames [11] may be used to compress multiple ciphertexts. However, an input plaintext size should be prohibitively small (e.g., at most up to 10 bits). Further, Johnson et al. [20] used as a message encoding a hash function from a bit string to a prime number for obtaining homomorphic signature.

1.3. **Organization.** The rest of this paper is organized as follows. In Section 2 we formally define the notion of decomposability. Section 3.1 describes an ordinary ElGamal encryption scheme is decomposable but inefficient. In Section 3.2, we give an ElGamal encryption scheme that satisfies both decomposability and efficiency. In Section 4, we further analyze that our decomposable encryption scheme runs efficiently.

2. Preliminaries

In this section, we remind the background regarding public-key homomorphic encryption, introduce a new notion called *decomposability*, and then describe the security model.

Notation. For $n \in \mathbb{N}$, $[n]$ denotes the set $\{1, \ldots, n\}$. If A is a probabilistic polynomial-time (PPT) machine, we use $a \leftarrow A$ to denote making A produce an output according to its internal randomness. In particular, if U is a set, then $r \xleftarrow{\$} U$ is used to denote sampling from the uniform distribution on U. For an integer a, $|a|$ denotes the bit length of a.

We denote by λ a security parameter. A function $g : \mathbb{N} \to \mathbb{R}$ is called negligible if for every positive polynomial $\mu(\cdot)$ there is an integer N such that $g(n) < 1/\mu(n)$ for all $n > N$. We use standard asymptotic (O, o) notation to denote the growth of positive functions. We say that $f(n) = \tilde{O}(g(n))$ if $f(n) = O(g(n)\log^c n)$ for some fixed constant c.

2.1. **The Model.** This section gives a formal definition of decomposability in a public key setting. We start with the definition of public key encryption.

Public-Key Encryption. A public-key encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ consists of the following algorithms:

- KeyGen is a randomized algorithm that takes a security parameter λ as input, and outputs a secret key sk and a public key pk; pk defines a plaintext space $\mathbf{P}$ and a ciphertext space $\mathbf{C}$.
- Enc is a randomized algorithm that takes pk and a plaintext $m \in \mathbf{P}$ as input, and outputs a ciphertext $c \in \mathbf{C}$.
- Dec takes sk and $c \in \mathbf{C}$ as input, and outputs the plaintext m.

We say that an encryption scheme is correct if, for any key-pair $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and any $m \in \mathbf{P}$, it is the case that: $m \leftarrow \mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m))$.

Decomposability. Informally speaking, a public-key encryption $\mathcal{E}$ is decomposable if we can efficiently recover all original messages from a decrypted ciphertext which is obtained by compression of other multiple ciphertexts. Here compression should be efficient. A formal definition is as follows:

Definition 1 (Decomposability). *Let $k, \ell \in \mathbb{N}$. Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme as defined above. Let $\mathcal{T}_1$ be a set of polynomial-time computable functions from $\mathbf{C}^k$ to $\mathbf{C} \cup \{\perp\}$ and $\mathcal{T}_2$ a set of polynomial-time computable functions from $\mathbf{P}$ to $\mathbb{P}^\ell \cup \{\perp\}$ where $\mathbb{P} \subset \mathbf{P}$ and $\perp$ is a distinguished symbol indicating transformation failure. Then,* decomposable encryption *is given by a tuple of PPT algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{T}_1, \mathsf{T}_2)$ having the properties below.*

(1) Easy to compress: *For any vector of ciphertexts $\mathbf{c} = (c_1, \ldots, c_k)$ and for some $\mathsf{T}_1 \in \mathcal{T}_1$, $\mathsf{T}_1(\mathbf{c})$ outputs another ciphertext $C \in \mathbf{C}$ or $\perp$ where $c_i = \mathsf{Enc}_{pk}(m_i)$.*

(2) Easy to decompose: *For any plaintext $M = \mathsf{Dec}_{sk}(\mathsf{T}_1(\mathbf{c})) \in \mathbf{P}$ with some vector of ciphertexts $\mathbf{c}$ and for some $\mathsf{T}_2 \in \mathcal{T}_2$, $\mathsf{T}_2(M)$ outputs a set of messages $\mathbf{m} = \{m_1, \ldots, m_\ell\}$ or $\perp$ where $m_i \in \mathbb{P}$.*

(3) Correctness: *For any vector of plaintexts $\mathbf{m} = (m_1, \ldots, m_k)$ be a vector of input messages, and any vector of ciphertexts $\mathbf{c} = (c_1, \ldots, c_k)$ with $c_i = \mathsf{Enc}_{pk}(m_i)$, there exists a pair of $(\mathsf{T}_1, \mathsf{T}_2) \in (\mathcal{T}_1, \mathcal{T}_2)$ such that $(m_1, \ldots, m_k) = \mathsf{T}_2 \circ \mathsf{Dec} \circ \mathsf{T}_1(\mathbf{c})$.*

When getting an understanding of the meaning of the definition above, one should notice that since the output of decomposing loses the order of original messages, it should not be interpreted as a vector. Correctness of decomposability is ensured only as a set.

We can consider T_1 as a function to transform multiple ciphertexts to a single ciphertext such that corresponding plaintexts are obliviously combined into a single plaintext M. On the other hand, the transformation T_2 can be considered as a function to decode the single plaintext M into a set $\{m_1, \ldots, m_k\}$. When an encryption scheme has a decomposable property, we call it a *decomposable encryption* scheme.

We notice that decomposability is meaningful only if the output size of T_1 is "shorter" than its input size. As an example, an encryption scheme using T_1 as concatenation (denoted by $\|$) can be also decomposable: given an encryption scheme we define $\mathsf{T}_1 : \mathbf{C}^k \to \mathbf{C}$ as $(\mathsf{Enc}_{pk}(m_1), \ldots, \mathsf{Enc}_{pk}(m_k)) \mapsto \mathsf{Enc}_{pk}(m_1) \| \cdots \| \mathsf{Enc}_{pk}(m_k)$, and $\mathsf{T}_2 : \mathbf{P} \to \mathbf{P}^k$ as $m_1 \| \cdots \| m_k \mapsto (m_1, \ldots, m_k)$, respectively. However, since the output size of the first transformation T_1 is the same as the input size, decomposability does not help to compress the size of ciphertexts. From now on, we consider only non-trivial schemes to reduce the size overhead.

Security. The semantic security game for a decomposable encryption scheme is similar to the original semantic-security game for an encryption scheme [14], except that additional transformations $\mathsf{T}_1, \mathsf{T}_2$ are publicly given and the adversary sends to the challenger two challenging vectors of plaintexts of his choice. To distinguish from the semantic security of an encryption scheme denoted by IND-CPA, we denote the semantic security for decomposable encryption d-IND-CPA and define the d-IND-CPA game as follows:

Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{d-IND-CPA}}(1^\lambda)$:

(1) $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $(\mathsf{T}_1, \mathsf{T}_2) \xleftarrow{\$} (\mathcal{T}_1, \mathcal{T}_2)$, and then pk and $(\mathsf{T}_1, \mathsf{T}_2)$ are given to $\mathcal{A}$.

(2) $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}$ where $\mathbf{m}_0 = (m_{0,1}, \ldots, m_{0,k})$ and $\mathbf{m}_1 = (m_{1,1}, \ldots, m_{1,k})$ for each $m_{i,j} \in \mathbf{P}, i \in \{0,1\}$, and for all $j \in [k]$

(3) Choose $b \xleftarrow{\$} \{0,1\}$ and compute $\mathbf{c}_b = (\mathsf{Enc}_{pk}(m_{b,1}), \ldots, \mathsf{Enc}_{pk}(m_{b,k}))$.

(4) $b' \leftarrow \mathcal{A}(\mathbf{m}_0, \mathbf{m}_1, c_b)$ where $b' \in \{0,1\}$

(5) The output of the experiment is 1 if $b' = b$ and 0 otherwise.

We say that a decomposable encryption $\mathcal{E}$ scheme is *d-IND-CPA secure* if the advantage of an adversary $\mathcal{A}$ defined as $\left|\Pr[1 \leftarrow \mathbf{Exp}_{\mathcal{A}}^{\text{d-IND-CPA}}(1^\lambda)] - \frac{1}{2}\right|$ is negligible for all PPT machines $\mathcal{A}$. Note that the IND-CPA game for the underlying encryption is a special case of $k = 1$.

The d-IND-CPA security of decomposable encryption is implied by the IND-CPA security for the underlying encryption. Namely, an algorithm $\mathcal{A}$ that wins the d-IND-CPA game above with advantage ϵ can be used to construct an algorithm $\mathcal{B}$ that breaks the IND-CPA security of the underlying encryption with advantage $\frac{\epsilon}{2k}$. More specifically, when $\mathcal{A}$ makes a challenge query $(\mathbf{m}_0, \mathbf{m}_1)$, $\mathcal{B}$ chooses a random index j, makes the challenge query $(m_{0,j}, m_{1,j})$, and receives $c_{b,j}$ from the IND-CPA challenger. $\mathcal{B}$ picks a random bit b' and responds to $\mathcal{A}$ with a vector of ciphertexts $(c_{b',1}, \ldots, c_{b,j}, \ldots, c_{b',k})$ hoping that $b = b'$. If $\mathcal{A}$ outputs b^* and $b^* = b'$ then $\mathcal{B}$ outputs b'. Otherwise, $\mathcal{B}$ outputs a random bit. Since the probability that the j-th element is a correct target ($\mathcal{A}$ distinguishes with a non-negligible probability by the hybrid argument) and $b = b'$ is $\frac{1}{2k}$, the adversary $\mathcal{B}$ has the advantage $\frac{\epsilon}{2k}$.

2.2. **Cryptographic Assumption.** Let $\mathbb{G}_q$ be a cyclic group of prime order q, and let g be its generator. We assume that the DDH problem are hard in $\mathbb{G}_q$. For example, $\mathbb{G}_q$ could be a subgroup of order q in the group of modular residues $\mathbb{Z}_p^\times$ such that $q|p-1$, $|p| = 2048$, and $|q| = 256$, or it can be a group of points on an elliptic curve with order q for $|q| = 256$. For more examples of groups, where the DDH assumption is assumed to hold, see [7].

Definition 2. *The DDH problem is* (ϵ, t)-hard *in* $\mathbb{G}_q$*, if for every algorithm* $\mathcal{D}$ *running in time* t *we have:*

$$\left|\Pr\left[\alpha, \beta \xleftarrow{\$} \mathbb{Z}_q \middle| \mathcal{D}(g, g^\alpha, g^\beta, g^{\alpha\beta}) = 1\right] - \Pr\left[\alpha, \beta, \gamma \xleftarrow{\$} \mathbb{Z}_q \middle| \mathcal{D}(g, g^\alpha, g^\beta, g^\gamma) = 1\right]\right| \leq \epsilon.$$

3. Our Decomposable Encryption Scheme

In this section, we construct a decomposable encryption scheme from multiplicative homomorphic ElGamal encryption. We describe our construction in two steps: In Section 3.1, solely for presentation purposes, we explain how ElGamal encryption over a prime field is possibly converted into a decomposable encryption scheme. The resulting scheme helps to understand intuition of our construction and why message encoding/decoding algorithms are additionally needed. However, the first scheme is not efficient enough to be practical because of its inefficient transformation. In Section 3.2, we construct our proposed scheme based on another ElGamal variant, ElGamal encryption over an extension field. Similar techniques in Section 3.1 are used to have decomposability. Interestingly, unlike the first scheme, the resulting scheme is efficient and does not incur any inefficient transformation. However, security of ElGamal encryption over an extension field should be carefully examined because different types of attack can be applied.

3.1. **Basic but Inefficient Scheme.** An IND-CPA secure ElGamal is defined on a subgroup in which the DDH assumption holds. Let p, q be primes such that $p-1 = sq$ for some s. The size of p and q are determined by a security parameter. Given g, a generator of a subgroup $\mathbb{G}_q$, the public/secret key pair is $(y = g^x, x)$ for $x \overset{\$}{\leftarrow} [q-1]$, and $\mathbf{P} = \mathbb{G}_q$ and $\mathbf{C} = (\mathbb{G}_q)^2$. For any message $\tilde{m} \in \mathbb{G}_q$, the encryption algorithm is $\mathsf{Enc}_{pk}(\tilde{m}) = (g^r, \tilde{m}y^r) \pmod p = (u, v)$ with $r \overset{\$}{\leftarrow} [q-1]$. Given a ciphertext $c = (u, v) \in (\mathbb{G}_q)^2$, the decryption algorithm is $\mathsf{Dec}_{sk}(c) = vu^{-x} \pmod p \equiv \tilde{m}$.

To integrate decomposability into this ElGamal encryption scheme, we define two transformations T_1 and T_2. For T_1, we utilize multiplicative homomorphism embedded in ElGamal encryption as follows: $\mathsf{T}_1(c_1, \ldots, c_k) = \prod_{i=1}^k c_i$ where $c_i = (g^{r_i}, \tilde{m}_i y^{r_i})$ with random r_i. Let $C = \prod_{i=1}^k c_i = \left(g^{\sum_{i=1}^k r_i}, \left(\prod_{i=1}^k \tilde{m}_i\right) \cdot y^{\sum_{i=1}^k r_i}\right)$. Then given $\tilde{M} = \mathsf{Dec}_{sk}(C)$ it is natural to relate T_2 with a factoring algorithm of an integer $\tilde{M}$. However, unless all m_i's are primes, factorization cannot determine a unique set of plaintexts.

To cope with this problem, we consider a map from a given message to a *prime number.*[1] We define a bijective map $\Psi : \mathbf{P} \to \mathbb{P}_\omega$ which converts a message $\tilde{m} \in \mathbf{P}$ into a prime number $m \in \mathbb{P}_\omega \subset \mathbf{P}$, where $\mathbb{P}_\omega$ is a set of primes equal to or less than ω bits for some $\omega \in \mathbb{N}$. The resulting encryption algorithm is $\mathsf{Enc}_{pk}(\tilde{m}) = (g^r, my^r)$ where $m = \Psi(\tilde{m})$. The map Ψ is discussed in more detail later in this section.

Correctness in decomposability is provided because a prime factorization determines a unique set of plaintexts, $\{m_1, \ldots, m_k\}$ for some k and the set of original messages $\{\tilde{m}_1, \ldots, \tilde{m}_k\}$ are extracted through Ψ^{-1} to each m_i. It is obvious that this construction is a decomposable encryption scheme for a set of primes $\mathbb{P}_\omega$ and some $k < \frac{\log p}{\omega}$. The decomposable encryption scheme is correct with $k < \frac{\log p}{\omega}$ since $m_1 \cdots m_k \le 2^{k\omega} < p$, and so $m_1 \cdots m_k = m_1 \cdots m_k \pmod p$.

In the following we describe our techniques to construct a bijective map Ψ.

Message Encoding and Decoding. We assign a message to a prime number by using a small-sized random padding and checking whether the padded message is a prime number. Namely, we append a padding γ to the message $\tilde{m}$, and then check whether $m = \tilde{m} \parallel \gamma$ is a prime number. When we define $\tilde{m} \parallel \gamma = \tilde{m}2^{|\gamma|} + \gamma$, the size of γ is determined by the distribution of primes. Let $\pi(x)$ be the number of primes equal to or less than x. Huxley [19] proved that

$$\pi(x + \Delta(x)) - \pi(x) \sim \frac{\Delta(x)}{\log x} \tag{3.1}$$

is true for almost all x if $\Delta(x) = x^{1/6+\varepsilon}$ ($\varepsilon > 0$ fixed). (See [28] for a survey on this topic.) This result implies that there exists at least a prime number if $|\gamma| = \lceil \frac{\omega}{6} \rceil$. (e.g., if $\omega = 32$ and $|\gamma| = 6$ then $\frac{\Delta(x)}{\log x} = 2$; when we increase the $|\gamma|$ by 8, we can expect to have eight primes at least.) The follow lemma shows that our encoding algorithm runs well with high probability.

Lemma 1. *Let x be the maximum value of messages and γ be a padded message as defined in above. Let $\rho > 0$ and define $s = \lceil |\gamma| \log(1/\rho) \rceil$. Then the procedure outputs a prime number in $\mathbb{G}_q$ with probability at least $1 - \rho^{\lceil \log x \rceil}$.*

Proof. Assume that the procedure runs s times. The probability that all s trials do not give any prime number is at most $\left(1 - \frac{\log x}{\Delta(x)}\right)^s < \left(e^{-\log x/\Delta(x)}\right)^{\lceil |\gamma| \log(1/\rho) \rceil} = e^{\lceil \log x \log \rho \rceil} = \rho^{\lceil \log x \rceil}$

[1] A technique mapping messages to primes has been used by Kim et al. [26] for private set intersection which runs on additive homomorphism by using Paillier encryption.

by $\Delta(x) = |\gamma|$ and Equation (3.1). Hence the procedure outputs at least a prime number with probability $1 - \rho^{\lceil \log x \rceil}$ as required. $\square$

The inverse map of Ψ is clear by simply removing the random padding.

Although we construct a decomposable encryption scheme as above, it is not practical due to the inefficiency of T_2 based on factorization. More specifically, trial division by primes demands a complexity of $\sqrt{p}$ for extracting a prime factor m_i. The expected time for Pollard's rho algorithm to find a factor m_i of M is $O(\sqrt{m_i})$ [9]. If for any prime factor m_i of M, $m_i - 1$ is smooth with respect to some relatively small bound B, we can use Pollard's p-1 factoring algorithm whose running time for finding the factor m_i is $O(\omega B/\log B)$ [30]. When one fails to factor a given message using Pollard's p-1 algorithm, we can apply the elliptic curve factoring algorithm whose expected running time is $L_{m_i}[\frac{1}{2}, \sqrt{2}]$ [27]. For example, recovering 128-bit messages from its compression would take 2^{64}.

Thus, this construction could be practical only for small-sized messages. In the next section, we construct a decomposable encryption scheme which efficiently covers medium or large messages.

3.2. **Our Efficient Construction.** In the multiplicative homomorphic encryption, decomposability can be added by utilizing multiplicative homomorphism as T_1 and a factoring algorithm as T_2, and defining an encoding/decoding scheme for correct message recovery. The challenge is how to combine all these algorithms efficiently, while preserving the security of an encryption scheme. To have efficient T_1 and T_2 transformations we consider the following facts:

- The ring of polynomials over a finite field is *unique factorization domains* (UFD) in which every non-zero element is uniquely written as a product of irreducible elements.
- Factoring a given polynomial over a finite field into irreducibles is carried out efficiently.

Thus, we examine an ElGamal encryption scheme over extension fields which satisfy the above properties and construct efficient transformations on it. Interestingly, ElGamal encryption over extension fields is also used for a different application of privacy-preserving set union. However, we notice that the security analysis in [18] is not rigorous and misses even attacks executable in extension fields.

In the following we overview the ElGamal encryption scheme over extension fields, demonstrate efficient transformations and encoding/decoding schemes, respectively, and discuss about the security of ElGamal encryption over extension fields focusing on the attack overlooked in [18].

Overview of ElGamal over Extension Fields. The description of the ElGamal encryption scheme over extension fields consists of the following algorithms.

- $\mathsf{KeyGen}(1^\lambda)$: The key generation algorithm chooses a large prime p and n such that $p^n - 1 = sq$ for some prime q and an integer s. Then select an irreducible polynomial $f(X) \in \mathbb{F}_p[X]$ of degree n and a generator $g(X)$ from $\mathbb{G}_q$ where $g(X) = h(X)^{\frac{p^n-1}{q}}$, where $h(X)$ is a generator of $(\mathbb{F}_p[X]/\langle f(X)\rangle)^\times$. It computes $y(X) \equiv g(X)^x \bmod f(X)$ where a secret key x is randomly chosen from $[p^n - 2]$. publishes a public key $pk = \langle \mathbb{G}_q, g(X), y(X), f(X)\rangle$.
- $\mathsf{Enc}_{pk}(M(X))$: Encryption with the public key pk and message $M(X) = (X - m) \in \mathbb{G}_q$ proceeds as follows. First, a random value $r \in [q-1]$ is chosen. The ciphertext is then published as:
$$C(X) = (u(X), v(X)) := (g(X)^r \bmod f(X), M(X) \cdot y(X)^r \bmod f(X)).$$
- $\mathsf{Dec}_{sk}(C(X))$: Suppose that a ciphertext $C(X)$ is encrypted with a public key pk and we have a secret key. Then, the ciphertext can be decrypted as:
$$M(X) \equiv v(X)u(X)^{-x} \pmod{f(X)}.$$

To make ElGamal encryption semantically secure, a subgroup where the DDH assumption holds should be used: if a generator of $(\mathbb{F}_p[X]/\langle f(X)\rangle)^\times$ is used, this ElGamal encryption scheme is not IND-CPA secure since it is easy to test elements of the multiplicative group. In fact, in this setting if messages are small, the ElGamal encryption scheme is subject to the Boneh-Joux-Nguyen attack [8]. Thus, we use the subgroup $\mathbb{G}_q$, as above, generated by $g(X) = h(X)^s$.

We note that there exists a sufficiently small number s such that $p^n - 1 = sq$. Furthermore, when n is prime, we have that $p^n - 1 = (p-1)\Phi_n(p)$ where $\Phi_n(p)$ is the n-th cyclotomic polynomial. Assuming the Bateman-Horn conjecture [4, 5], the number of primes of the form $(p^n - 1)/(p-1) = \Phi_n(p)$ not exceeding x, denoted by $H(x)$, is given by

$$H(x) \sim 0.76 \cdots \int_2^{x^{1/2}} (\log u)^{-2} du.$$

Thus we know that the probability that $\Phi_n(p)$ is prime for an integer $p \ll x$ is not small.

Message Encoding and Decoding. For a message $\tilde{m} \in \mathbb{F}_p$ we take the smallest positive integer γ such that $(X - \tilde{m} \parallel \gamma)^q \equiv 1 \bmod f(X)$, where $(\tilde{m} \parallel \gamma)$ denotes $\tilde{m}2^{|\gamma|} + \gamma \in \mathbb{F}_p$. Assuming the linear polynomials of order q are uniformly distributed over all linear polynomials in $\mathbb{F}_p[X]/\langle f(X)\rangle$, the expected number of trials becomes s. Thus we know that the encoding process can be performed efficiently and its output is an irreducible element in $\mathbb{G}_q$. Alternatively, we can directly use $(X - \tilde{m})^s \in \mathbb{G}_q$ when s is sufficiently smaller than n. However, this method is less efficient than the random-padding technique due to the condition $k < \frac{n}{s\omega}$ in terms of message rate. For the discussion about message rate refer to the next section.

Decoding is straightforward by using a constant term in each linear polynomial after removing the random padding of fixed size.

Transformations. We take T_1 as multiplication on $\mathbf{C}$ since the ElGamal encryption scheme is multiplicatively homomorphic. When the transformation T_2 is given by *factoring polynomials* over a finite field $\mathbb{F}_p$, we have polynomial-time algorithms for factoring in $\mathbb{F}_p[X]$, which is a key difference from the ElGamal encryption given in Section 3.1.

Lemma 2. *Let $\mathbb{P}_\omega$ be a set of irreducible polynomials of degree less than or equal to ω in $\mathbb{F}_p[X]$. The ElGamal encryption scheme given in Section 3.2 is decomposable on $\mathbb{P}_\omega$ for $k < \frac{n}{\omega}$.*

Proof. The proof is straightforward from the the fact that $\mathbb{F}_p[X]$, the ring of polynomials over a field $\mathbb{F}_p$, is a UFD, and there exist efficient transformations T_1 and T_2. □

3.3. **Security Analysis.** In this section we show that the ElGamal encryption on extension fields is IND-CPA secure, so that our decomposable construction is d-IND-CPA secure.

As mentioned above, the ElGamal encryption is secure against the Boneh-Joux-Nguyen attack [8]. Let $v(X) = M(X) \cdot y(X)^r$ be the second component of a ciphertext from the encryption algorithm. The Boneh-Joux-Nguyen attack works only if an adversary can make $y(X)^{rq} \in \mathbb{G}_s$ by raising $v(X)$ to the power of q and manage to compute discrete logarithms in $\mathbb{G}_s$. However, as $y(X)^{rq} \in \mathbb{G}_q$ and $\mathbb{G}_q$ has a large prime order, the adversary cannot efficiently find the random exponent r.

Next we should check our encryption scheme would be subject to index calculus attacks, since we moved from the multiplicative group of a prime field to the multiplicative group of an extension field. Recall that the encryption scheme works on such a subgroup $\mathbb{G}_q$ of the multiplicative group $(\mathbb{F}_p[X]/\langle f(X)\rangle)^\times \cong \mathbb{F}_{p^n}^\times$. During parameter selection, p is a prime such that $p^n - 1 = sq$ for a small even number s and a large prime number q. Note that p is not a prime power and the extension degree n is a prime number. Then we can see that an attacker has complexity of $\sqrt{q}$ to compute discrete logarithms in $\mathbb{G}_q$ using a square-root algorithm directly in $\mathbb{G}_q$, such as Pollard's rho algorithm. We have two efficient methods of calculating the index calculus: the number field sieve method [15] in a prime field and the function field sieve method [3] in a finite field with a large extension degree. However, the function field sieve by Adleman can efficiently extract discrete logarithms over finite fields of *small* characteristic. In the case of a medium-size characteristic, we should consider variations of the number field sieve method and the function field sieve method [21, 22]. In particular, they are efficient when $\log p < O(\sqrt{n}\log n)$ holds. Using medium-sized values of prime p with large-size subgroup prevents from being affected by these algorithms. Moreover, Gower [16, §1.3] pointed out that when p is a prime, the attacker will not be able to mount the Granger-Vercauteren attack [17].

Finally we check whether the ElGamal variant on extension fields is IND-CPA secure and the decomposable construction relying on this variant is secure.

Theorem 1. *The ElGamal scheme on extension fields given in Section 3.2 is IND-CPA secure assuming the DDH assumption holds in a cyclic subgroup $\mathbb{G}_q$ of $(\mathbb{F}_p[X]/\langle f(X)\rangle)^\times$.*

Proof. We prove security by defining two hybrid experiments Game_0 and Game_1 where Game_0 is the real IND-CPA game.

Game$_0$. Fix an efficient adversary $\mathcal{A}$. To make things more precise and more concrete, we give an algorithmical description of the attack game as follows:

$\alpha \overset{\$}{\leftarrow} \mathbb{Z}_q, y(X) = g(X)^\alpha$
$M_0(X), M_1(X) \leftarrow \mathcal{A}(\tilde{r}, y(X))$ where $\tilde{r}$ is sampled uniformly at random from some set
$b \overset{\$}{\leftarrow} \{0,1\}, \beta \overset{\$}{\leftarrow} \mathbb{Z}_q, u(X) = g(X)^\beta, t(X) = y(X)^\beta, v(X) = M_b(X)\cdot t(X)$
$b' \leftarrow \mathcal{A}(\tilde{r}, y(X), u(X), v(X))$

It is clear that this algorithm faithfully represents the IND-CPA game. If we define E_0 to be the event that $b = b'$, then the adversarys advantage is $|\Pr[\mathsf{E}_0] - 1/2|$.

Game$_1$. We now make one small change to the above game. Namely, instead of computing $t(X)$ as $y(X)^\beta$, we compute it as $g(X)^\gamma$ for randomly chosen $\gamma \in \mathbb{Z}_q$. We can describe the resulting game algorithmically as follows:

$\alpha \overset{\$}{\leftarrow} \mathbb{Z}_q, y(X) = g(X)^\alpha$
$M_0(X), M_1(X) \leftarrow \mathcal{A}(\tilde{r}, y(X))$ where $\tilde{r}$ is sampled uniformly at random from some set
$b \overset{\$}{\leftarrow} \{0,1\}, \beta \overset{\$}{\leftarrow} \mathbb{Z}_q, u(X) = g(X)^\beta, \gamma \overset{\$}{\leftarrow} \mathbb{Z}_q, t(X) = y(X)^\gamma, v(X) = M_b(X)\cdot t(X)$
$b' \leftarrow \mathcal{A}(\tilde{r}, y(X), u(X), v(X))$

Let E_1 be the event that $b = b'$ in Game_1. We first show that $\Pr[\mathsf{E}_1] = 1/2$.

Claim 1. $\Pr[\mathsf{E}_1] = 1/2$

In this claim we would like to prove that the adversarys output b' is independent of the challenger's bit b. It is enough to show that $b, \tilde{r}, y(X), u(X), v(X)$ are mutually independent, which implies that b and $b' \leftarrow \mathcal{A}(\tilde{r}, y(X), u(X), v(X))$ are independent. If $b, \tilde{r}, y(X), u(X)$ are fixed, then so are $M_0(X), M_1(X)$, since they are determined by $\tilde{r}, y(X)$. Moreover, the conditional distribution of $t(X)$ is the uniform distribution on $\mathbb{G}_q$, and hence from this, we see that the conditional distribution of $v(X) = M_b(X)\cdot t(X)$ is the uniform distribution on $\mathbb{G}_q$.

Claim 2. *Let ϵ be an advantage of an efficient algorithm which distinguishes between a DDH tuple and a random tuple. Then, $|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| = \epsilon$.*

We know that ϵ is negligible under the DDH assumption. The proof of this is essentially the observation that in Game_0, the tuple $\langle y(X), u(X), t(X)\rangle$ is of the form $\langle g(X)^\alpha, g(X)^\beta, g(X)^{\alpha\beta}\rangle$, while in Game_1, it is of the form $\langle g(X)^\alpha, g(X)^\beta, g(X)^\gamma\rangle$, and so the adversary should not tell the difference, under the DDH assumption. More precisely, our distinguishing algorithm $\mathcal{D}$ works as follows:

```
Distinguishing Algorithm $\mathcal{D}(y(X), u(X), t(X))$
$M_0(X), M_1(X) \leftarrow \mathcal{A}(\tilde{r}, y(X))$
$b \xleftarrow{\$} \{0,1\}, v(X) = M_b(X) \cdot t(X)$
$b' \leftarrow \mathcal{A}(\tilde{r}, y(X), u(X), v(X))$
if $b = b'$ then output 1; otherwise output 0
```

If the input to $\mathcal{D}$ is of the form $\langle g(X)^\alpha, g(X)^\beta, g(X)^{\alpha\beta}\rangle$, then computation proceeds just as in Game$_0$, and therefore

$$\Pr\left[\alpha, \beta \xleftarrow{\$} \mathbb{Z}_q \middle| \mathcal{D}\left(g(X)^\alpha, g(X)^\beta, g(X)^{\alpha\beta}\right) = 1\right] = \Pr\left[\mathsf{E}_0\right].$$

If the input to $\mathcal{D}$ is of the form $\langle g(X)^\alpha, g(X)^\beta, g(X)^{\gamma}\rangle$, then computation proceeds just as in Game$_1$, and therefore

$$\Pr\left[\alpha, \beta, \gamma \xleftarrow{\$} \mathbb{Z}_q \middle| \mathcal{D}\left(g(X)^\alpha, g(X)^\beta, g(X)^{\gamma}\right) = 1\right] = \Pr\left[\mathsf{E}_1\right].$$

From this, it follows that the advantage of $\mathcal{D}$ is equal to $|\Pr[E_0] - \Pr[\mathsf{E}_1]|$.

Combining Claim 1 and Claim 2, we see that $|\Pr[E_0] - \Pr[\mathsf{E}_1]| = \epsilon$, and this is negligible. That completes the proof of security of the ElGamal encryption on extension fields. $\square$

The following theorem states that the decomposable encryption scheme satisfies the d-IND-CPA security.

Theorem 2. *Assuming the DDH problem is intractable, the decomposable encryption based on ElGamal variant is d-IND-CPA secure.*

Proof. As we discussed in Section 2.1, the security of our decomposable encryption scheme is naturally implied by the security of the above ElGamal encryption. $\square$

4. Performance Analysis

In this section, we analyze efficiency of our decomposable encryption schemes in terms of message rate and computation efficiency. To estimate compression efficiency, we define message rate and give an analysis on message rate. Then we analyze the efficiency of transformations used in Section 3.2. We also present the whole computational complexity of the decomposable encryption scheme given in Section 3.2.

4.1. Compression Efficiency.

We analyze how our scheme effectively compresses given multiple ciphertexts. For this purpose, we define *message rate.* Roughly speaking, the message rate is the percentage of plaintexts that occupies an output ciphertext of T_1 when a decomposable encryption scheme works correctly.

Definition 3 (Message Rate). *The message rate, denoted by MR, over the ciphertext with respect to T_1 is the total bits of plaintexts contained in ciphertexts T_1 takes as input divided by the total number of bits it produces as output:*

$$MR = \frac{\log \mathsf{V}_{\mathbf{P}}}{\log \mathsf{V}_{\mathbf{C}}}$$

where $\mathsf{V}_{\mathbf{P}}$ is the total number of bits of messages in $(c_1, \ldots, c_k)$ given to T_1 as input and $\mathsf{V}_{\mathbf{C}}$ is the total number of bits T_1 gives as output in $\mathbf{C}$.

We first compute the message rate of the ElGamal-based decomposable construction given in Section 3.1. We demand that $k < \frac{\log p}{\omega + |\gamma|}$ for correctly recovering the original messages. We can easily see that the message rate of this construction becomes $\frac{k\omega}{2\log p}$. For example, let p be a 1024-bit prime, ω a 32-bit message, and $|\gamma|$ a 10-bit random padding. Since $k \leq 24$, in this setting $MR = \frac{768}{2048} = 0.375$.

When a decomposable encryption is used by the ElGamal encryption in Section 3.2, we require that $k < n$ because $\omega = 1$. Its message rate becomes $\frac{k \log \tilde{m}}{2n \log p}$. For a fair comparison, let $\tilde{m}$ be a 32-bit message, p be a 42-bit prime, and n be a 27-bit prime. Then we have $k \leq 26$. Hence, this decomposable encryption scheme has $MR = \frac{832}{2268} \approx 0.367$.

Now let us consider the transmission of k ElGamal ciphertexts without use of a decomposable encryption scheme. For simplicity, suppose that all plaintexts are of the same size ω. We then see that the message rate MR in this setting is $\frac{\omega}{2 \log p}$. If the same values above, i.e., $\omega = 32$ and $\log p = 1024$, are used, we have $MR = \frac{32}{2048} \approx 0.016$. Hence a decomposable ElGamal encryption scheme allows us to have the message rate 23 times higher than an ordinary ElGamal encryption scheme. This means that we can utilize bandwidth more efficiently with a decomposable encryption scheme.

Remark 1. *If k is larger than n, the messages can not be recovered from the compressed ciphertext, since $\prod_{i=1}^{k}(X - m_i) \neq \prod_{i=1}^{k}(X - m_i) \bmod f(X)$. When there are more than k ciphertexts, one way we can use is to compress after partitioning them into groups of k or less elements. We conjecture that a decomposable encryption scheme using an IND-CPA secure encryption has the message rate less than $\frac{1}{2}$ due to the random part of its underlying encryption.*

4.2. **Computation Efficiency.** The major bottleneck of our decomposable encryption scheme described in 3.2 is the transformation T_2, which factors a given polynomial of degree n into irreducibles. Milestones in the development of polynomial-time algorithms for factoring in $\mathbb{F}_p[X]$ are the algorithms of Berlekamp [6], Cantor & Zassenhaus [10], von zur Gathen & Shoup [34] and Kaltofen & Shoup [24]. See the surveys [33, 23, 32]. A straightforward implementation of Berlekamp's algorithm [6] uses $O(n^3 + n^{1+o(1)} \log p)$ operations in $\mathbb{F}_p$. Presently, there are practical algorithms that factor degree n polynomials over $\mathbb{F}_p$ in $\tilde{O}(n^2 + n \log p)$ operations, and sub-quadratic algorithms that rely on fast matrix multiplication [24]. When the Cantor-Zassenhaus algorithm [10] is used, it requires an expected number of $O(n^{2+o(1)} \log p)$ operations in $\mathbb{F}_p$. One of the asymptotically fastest algorithms for factoring polynomials, due to von zur Gathen and Shoup [34], requires an expected number of $O(n^{2+o(1)} + n^{1+o(1)} \log p)$ operations in $\mathbb{F}_p$. Further, Umans [31] proposed randomized algorithms for factoring degree n univariate polynomials over $\mathbb{F}_p$ that use $O(n^{1.5+o(1)} + n^{1+o(1)} \log p)$ field operations, when the characteristic is small.

In our decomposable encryption scheme, the encryption algorithm requires two exponentiations in $\mathbb{F}_{p^n}$. Since two exponentiations involve a constant number of multiplications over $\mathbb{F}_{p^n}$, it takes $O(n^{\log_2 3})$ using Karatsuba method [25] or $\tilde{O}(n)$ by fast Fourier transform [12]. Therefore, the total computational complexity is bounded by $O(n^{2+o(1)} \log p)$.

5. Conclusion and Further Work

In this work we gave an answer to the problem of how to squeeze multiple ciphertexts without losing original message information. We present the notion of decomposability for public-key homomorphic encryption and construct an efficient ElGamal encryption over extension fields to support decomposability. Our scheme on a subgroup of order q in $\mathbb{F}_{p^n}$ is efficient when $\frac{p^n-1}{q}$ is small.

An interesting question is to find an encoding algorithm of a message into an order q group even when the cofactor $\frac{p^n-1}{q}$ is large. Another open problem is to find an upper bound of the message rate and further design the "optimal" encryption scheme that achieves this rate.

References

[1] M. Abe, E. Kiltz, and T. Okamoto. Chosen ciphertext security with optimal ciphertext overhead. In J. Pieprzyk, editor, *Advances in Cryptology-AsiaCrypt*, LNCS 5350, pages 355–371, 2008. 2

[2] M. Abe, E. Kiltz, and T. Okamoto. Compact CCA-secure encryption for messages of arbitrary length. In S. Jarecki and G. Tsudik, editors, *PKC*, LNCS 5443, pages 377–392, 2009. 2

[3] L. Adleman. The function field sieve. In L. Adleman and M.-D. Huang, editors, *ANTS*, LNCS 877, pages 108–121, 1994. 8

[4] P. Bateman and R. Horn. A heuristic asymptotic formula concerning the distribution of prime numbers. *Mathematics of Computation*, 16:363–367, 1962. 7

[5] P. Bateman and R. Stemmler. Waring's problem for algebraic number fields and primes of the form $(p^r - 1)/(p^d - 1)$. *Illinois J. Math.*, 6(1):142–156, 1962. 7

[6] E. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):713–735, 1970. 10

[7] D. Boneh. The decision Diffie-Hellman problem. In J. Buhler, editor, *ANTS*, LNCS 1423, pages 48–63, 1998. 4

[8] D. Boneh, A. Joux, and P. Nguyen. Why textbook ElGamal and RSA encryption are insecure. In T. Okamoto, editor, *Advances in Cryptology-AsiaCrypt*, LNCS 1976, pages 30–43, 2000. 7

[9] R. Brent. An improved Monte Carlo factorization algorithm. *BIT Numerical Mathematics*, 20:176–184, 1980. 6

[10] D. Cantor and H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36:587–592, 1981. 10

[11] G. Castagnos and B. Chevallier-Mames. Towards a DL-based additively homomorphic encryption scheme. In J. Garay, A. Lenstra, M. Mambo, and R. Peralta, editors, *ISC*, LNCS 4779, pages 362–375, 2007. 2

[12] J. Cooley and J. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. 10

[13] C. Gentry. How to compress rabin ciphertexts and signatures (and more). In M. K. Franklin, editor, *Advances in Cryptology-Crypto*, LNCS 3152, pages 179–200, 2004. 2

[14] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. 3

[15] D. M. Gordon. Discrete logarithms in $GF(p)$ using the number field sieve. *SIAM J. Discrete Math.*, 6(1):124–138, 1993. 8

[16] J. Gower. Prime order primitive subgroups in torus-based cryptography. IACR Cryptology ePrint Archive 2006: 466, 2006. 8

[17] R. Granger and F. Vercauteren. On the discrete logarithm problem on algebraic tori. In V. Shoup, editor, *Advances in Cryptology-Crypto*, LNCS 3621, pages 66–85, 2005. 8

[18] J. Hong, J. W. Kim, J. Kim, K. Park, and J. H. Cheon. Constant-round privacy preserving multiset union. In *Cryptology ePrint Archive 2011:138*, 2011. 6

[19] M. Huxley. On the difference between consecutive primes. *Inventiones Math.*, 15:164–170, 1972. 5

[20] R. Johnson, D. Molnar, D. Song, and D. Wagner. Homomorphic signature schemes. In B. Preneel, editor, *CT-RSA*, LNCS 2271, pages 244–262, 2002. 2

[21] A. Joux and R. Lercier. The function field sieve in the medium prime case. In S. Vaudenay, editor, *Advances in Cryptology-EuroCrypt*, LNCS 4004, pages 254–270, 2006. 8

[22] A. Joux, R. Lercier, N. Smart, and F. Vercauteren. The number field sieve in the medium prime case. In C. Dwork, editor, *Advances in Cryptology-Crypto*, LNCS 4117, pages 326–344, 2006. 8

[23] E. Kaltofen. Polynomial factorization: a success story. In J. Rafael Sendra, editor, *ISSAC*, pages 3–4, 2003. 10

[24] E. Kaltofen and V. Shoup. Subquadratic-time factoring of polynomials over finite fields. *Mathematics of Computation*, 67(223):1179–1197, 1998. 10

[25] A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics Doklady*, 7:595–596, 1963. 10

[26] M. Kim, H. T. Lee, and J. H. Cheon. Mutual private set intersection with linear complexity. In *WISA*, 2011. 5

[27] H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126(3):649–673, 1987. 6

[28] H. Maier. Primes in short intervals. *Michigan Mathematical Journal*, 32(2):221–225, 1985. 5

[29] D. H. Phan and D. Pointcheval. Chosen-ciphertext security without redundancy. In C.-S. Laih, editor, *Advances in Cryptology-AsiaCrypt*, LNCS 2894, pages 1–18, 2003. 2

[30] J. Pollard. Theorems on factorization and primality testing. *Proceedings of the Cambridge Philosophical Society*, 76:521–528, 1974. 6

[31] C. Umans. Fast polynomial factorization and modular composition in small characteristic. In *STOC*, pages 481–490, 2008. 10
[32] J. von zur Gathen. Who was who in polynomial factorization. In B. Trager, editor, *ISSAC*, page 2, 2006. 10
[33] J. von zur Gathen and D. Panario. Factoring polynomials over finite fields: A survey. *J. Symb. Comput.*, 31(1/2):3–17, 2001. 10
[34] J. von zur Gathen and V. Shoup. Computing frobenius maps and factoring polynomials. *Computational Complexity*, 2:187–224, 1992. 10

CHRI & Dept. of Mathematical Sciences, Seoul National University, 1 Gwanak-ro, Gwangak-gu, Seoul 151-747, Republic of Korea
E-mail address: `msunkim@snu.ac.kr`

School of Electrical Engineering, Kookmin University, 77 Jeongneung-no, Seongbuk-gu, Seoul 136-702, Republic of Korea
E-mail address: `jihyek@kookmin.ac.kr`

CHRI & Dept. of Mathematical Sciences, Seoul National University, 1 Gwanak-ro, Gwangak-gu, Seoul 151-747, Republic of Korea
E-mail address: `jhcheon@snu.ac.kr`