A Hybrid Scheme of Public-Key Encryption and Somewhat Homomorphic Encryption

Jung Hee Cheon and Jinsu Kim

***Abstract*— We introduce a hybrid homomorphic encryption that combines public-key encryption (PKE) and somewhat homomorphic encryption (SHE) to reduce the storage requirements of most somewhat or fully homomorphic encryption (FHE) applications. In this model, messages are encrypted with a PKE and computations on encrypted data are carried out using SHE or FHE after homomorphic decryption. To obtain efficient homomorphic decryption, our hybrid scheme combines IND-CPA PKE without complicated message padding with SHE with a large integer message space. Furthermore, if the underlying PKE is multiplicative, the proposed scheme has the advantage that polynomials of arbitrary degree can be evaluated without bootstrapping. We construct this scheme by concatenating the ElGamal and Goldwasser–Micali schemes over a ring $\mathbb{Z}_N$ for a composite integer N whose message space is $\mathbb{Z}_N^{\times}$. To accelerate the homomorphic evaluation of the PKE decryption, we introduce a method to reduce the degree of the exponentiation circuit at the cost of additional public keys. Using the same technique, we present an efficient partial solution to an open problem which is to evaluate $\bmod q \bmod p$ arithmetic homomorphically for large p. As an independent interest, we also obtain a generic method for converting from private-key SHE to public-key SHE. Unlike the method described by Rothblum, we are free to choose the SHE message space.**

***Index Terms*— ElGamal, Goldwasser–Micali, Naccache-Stern, hybrid scheme, homomorphic encryption, fully homomorphic encryption, bootstrapping.**

I. Introduction

THE concept of computation on encrypted data without decryption was first introduced by Rivest, Adleman and Dertouzos in 1978 [19]. Thirty years later, Gentry proposed a fully homomorphic encryption (FHE) based on ideal lattices [7]. This scheme is far from being practical because of its large computational cost and large ciphertexts. Since then, considerable efforts have been made to devise more efficient schemes. However, most FHE schemes still have very large ciphertexts (millions of bits for a single ciphertext). This presents a considerable bottleneck in practical deployments.

We consider the following situation: several users upload data encrypted with a public-key FHE, a server carries out computations on the encrypted data and then sends them to an agency who has a decryption key for the FHE. This is common in typical FHE scenarios, such as medical and financial applications [16]. In this situation, one approach to reduce the storage requirement is to use AES encryption to encrypt data, and then perform homomorphic computations on ciphertexts after converting to FHE-ciphertexts. This method has a great advantage in storage and communication, because only small AES-ciphertexts are transmitted from user to server, and these are homomorphically decrypted only when their homomorphic computations are required. In an asymmetric setting, we can still use this approach by adding several public-key FHE ciphertexts of a session key. However this approach is not practical when the amount of messages transmitted simultaneously is small compared with the size of on FHE ciphertext. Moreover, the conversion of AES-ciphertexts into FHE-ciphertexts requires a leveled FHE with multiplicative depth of at least forty [2], [4], [9].[1]

In this paper, we explore an alternative method that encrypts messages with a public key encryption (PKE) and converts them into SHE-ciphertexts for homomorphic computations. In this approach, the ciphertext expansion ratio is only two or three regardless of the message size. Moreover, the decryption circuit is very shallow when the SHE allows large integers as messages. For example, the decryption circuit of ElGamal over $\mathbb{Z}_N$ has a multiplicative depth of nine under a SHE with the message space $\mathbb{Z}_N$ [8].[2] We can reduce the depth further by representing the secret exponent e as $\log_w e$ binary vectors of length w, which is an improvement over the Gentry-Halevi technique [8].

When using additive (resp. multiplicative) homomorphic encryption as the underlying PKEs, we obtain the additional advantage that additions (resp. multiplications) can be computed without converting to SHE. For multiplicative homomorphic encryptions (MHE) in particular, one can compute $\mathsf{SHE}(f(m_1, \ldots, m_k))$ from $\mathsf{PKE.Enc}(m_1), \ldots, \mathsf{PKE.Enc}(m_k)$ without (expensive) bootstrapping for any multivariate polynomial $f(x_1, \ldots, x_k)$ with polynomially many terms.

Manuscript received July 28, 2014; revised November 19, 2014; accepted January 14, 2015. Date of publication February 2, 2015; date of current version March 7, 2015. This work was supported by the ICT R&D program of MSIP/IITP [No.10047212, Development of homomorphic encryption supporting arithmetics on ciphertexts of size less than 1kB and its applications] and Samsung Electronics, Co., Ltd. [No. 045020130004]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shouhuai Xu.

The authors are with the Department of Mathematical Sciences, Seoul National University, Seoul 151-742, Korea (e-mail: jhcheon@snu.ac.kr; kjs2002@snu.ac.kr).

Digital Object Identifier 10.1109/TIFS.2015.2398359

[1] In [4], one chooses the bit length of secret key $\eta > \rho + L(\log \Theta + 9)$ to allow multiplicative depth L homomorphic evaluation. From the parameters in [4, Sec. 5], we can verify that the multiplicative level L of FHE is about forty, when we choose $\eta = 993, \rho = 86, \Theta = 28,125$ for homomorphic evaluation of AES decryption.

[2] The decryption of ElGamal encryption consists of exponentiation with secret 160-bit exponent e. Using the method to evaluate secret exponent e homomorphically with $2 \log e$ degree in Section III.B.1), one can evaluate the decryption circuit of ElGamal with SHE of multiplicative depth nine.

One problem when using MHE in the hybrid scheme is that the message space for MHE schemes is not usually closed under addition. For example, the (IND-CPA) ElGamal encryption over a ring R can only take messages with elements in a prime order subgroup, which covers only a small part of R. To resolve this, therefore, we construct a MHE whose message space is $\mathbb{Z}_N^\times$ for an RSA modulus $N = p_1 p_2$. The proposed scheme is constructed by combining ElGamal encryption over $\mathbb{Z}_N^\times$ and Goldwasser-Micali encryption over $\mathbb{Z}_N$, and is secure under the decisional Diffie–Hellman assumption and the quadratic residuosity assumption for common $N = pq$.

We remark that our technique solves the open problem of [13] when the FHE message space is $\mathbb{Z}_p$ for large p. We convert the double modulo reduction into a depth-3 circuit, and then, we apply the technique of [8]. Our improved technique plays an important role in this method, as the parameters depend heavily on the homomorphic capacity of FHE.

As an independent interest, we also present a generic method for converting from a private-key SHE to a public-key SHE using our hybrid scheme. In this case, the message space is $\mathbb{Z}_p$ for a large integer $p > 2$ and the public key is the encryption of the secret key of a PKE under private-key SHE, which is much smaller than that described in [20].

II. Preliminaries

In this section, we introduce some definitions and base problems needed to prove the security of our schemes.

Notation: For $m, n \in \mathbb{N}$, $[m, n]$ and $[m, n)$ denote the sets $\{m, m+1, \ldots, n-1, n\}$ and $\{m, m+1, \ldots, n-2, n-1\}$, respectively. We denote the element in $\mathbb{Z} \cap (-\frac{n}{2}, \frac{n}{2}]$ that is equivalent to a modulo n by $a \bmod n$ or $[a]_n$, and the unique integer in $(-\frac{\prod_i p_i}{2}, \frac{\prod_i p_i}{2}]$ that is congruent to m_i modulo p_i for all i by $CRT_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k)$. For an integer N, we use sets $\mathsf{J}_N := \{a \in \mathbb{Z}_N^\times | \left(\frac{a}{N}\right) = 1\}$ and $\mathsf{QR}_N := \{a \in \mathbb{Z}_N^\times | a = b^2$ for some $b \in \mathbb{Z}_N^\times\}$. We denote $a^{\phi(N)/p} \bmod N$ by $\left(\frac{a}{N}\right)_p$.

A. Hard Problems

Definition 1 (Decisional Diffie–Hellman Problem Over $\mathbb{G}$): Let $\mathbb{G}$ be a group with a generator g of order q. For a given tuple (g, g^a, g^b, g^c), the *decisional Diffie–Hellman (DDH) problem over* $\mathbb{G}$ is to determine whether $g^{ab} = g^c$.

Definition 2 (k-Quadratic Residuosity Problem Over $\mathbb{Z}_N$): Given an odd composite integer $N = pq$ such that $p \equiv q \equiv 1 \pmod{2^k}$ and $a \in \mathsf{J}_N$ where $\mathsf{J}_N := \{a \in \mathbb{Z}_N^\times | \left(\frac{a}{N}\right) = 1\}$, the *$k$-quadratic residuosity problem (k-QR) over* $\mathbb{Z}_N$ is to determine whether a is quadratic residue modulo N.

Definition 3 (Higher Residuosity Problem Over $\mathbb{Z}_N$): For a given odd composite integer $N = pq$, an integer d such that $d|(p-1)$, and an integer $a \in \mathbb{Z}_N$, the *higher residuosity problem (HR) over* $\mathbb{Z}_N$ is to determine whether a is the d-th residue modulo N.

We say that the DDH assumption over $\mathbb{G}_q$ holds if no polynomial time distinguisher can solve the DDH problem with non-negligible advantage with respect to the security parameter λ. The k-QR and HR assumptions over $\mathbb{Z}_N$ are defined similarly.

B. Homomorphic Encryption Schemes

Definition 4 (ElGamal Encryption Over a Ring): Let R be a ring. The *ElGamal encryption scheme* $\mathsf{EIG} = (\mathsf{EIG.KG}, \mathsf{EIG.Enc}, \mathsf{EIG.Dec})$ consists of the following algorithms:

- $\mathsf{EIG.KG}(\lambda)$: Choose a multiplicative cyclic subgroup $\mathbb{G}_q$ of prime order q in R such that the DDH assumption holds with respect to the security parameter λ. Choose a generator g of $\mathbb{G}_q$ and a random $e \in [0, q)$, and compute $y = g^e$. Output a public key $pk_{\mathsf{EIG}} = (R, \mathbb{G}_q, g, y)$ and a secret key $sk_{\mathsf{EIG}} = e$.
- $\mathsf{EIG.Enc}(pk_{\mathsf{EIG}}, m)$: Take as input the public key pk_{EIG} and a plaintext $m \in \mathbb{G}_q$. Choose a random $r \in [0, q)$ and compute g^{-r} and $m \cdot y^r$. Output $c = (g^{-r}, m \cdot y^r)$.
- $\mathsf{EIG.Dec}(sk_{\mathsf{EIG}}, c)$: Take as input the secret key sk_{EIG} and a ciphertext $c = (v, u) \in R^2$. Output $m = v^e u$.

Definition 5 (Goldwasser–Micali Encryption): The *Goldwasser-Micali encryption scheme* $\mathsf{GM} = (\mathsf{GM.KG}, \mathsf{GM.Enc}, \mathsf{GM.Dec})$ consists of the following algorithms:

- $\mathsf{GM.KG}(\lambda)$: Choose random primes p, q and compute $N = pq$ such that the 1-QR assumption holds with respect to the security parameter λ. Compute the quadratic non–residue y modulo N satisfying $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$. Output a public key $pk_{\mathsf{GM}} = (N, y)$ and a secret key $sk_{\mathsf{GM}} = (p)$.
- $\mathsf{GM.Enc}(pk_{\mathsf{GM}}, m)$: For a plaintext $m \in \{0, 1\}$, choose a random $x \in \mathbb{Z}_N^\times$ and output a ciphertext $c = y^m x^2 \mod N$.
- $\mathsf{GM.Dec}(sk_{\mathsf{GM}}, c)$: For a ciphertext $c \in \mathbb{Z}_N^\times$, if $\left(\frac{c}{p}\right)_2 = 1$ output 0. Otherwise output 1.

Definition 6 (Joye–Libert Encryption): The *Joye-Libert encryption scheme* $\mathsf{JL} = (\mathsf{JL.KG}, \mathsf{JL.Enc}, \mathsf{JL.Dec})$ consists of the following algorithms:

- $\mathsf{JL.KG}(\lambda)$: Choose an integer $k \geq 1$ and random primes p and q with $p \equiv q \equiv 1 \pmod{2^k}$, and set $N = pq$ such that the k-QR assumption holds with respect to the security parameter λ. Choose a random $y \in \mathsf{J}_N \backslash \mathsf{QR}_N$. Output a public key $pk_{\mathsf{JL}} = (N, y, k)$ and a secret key $sk_{\mathsf{JL}} = (p)$.
- $\mathsf{JL.Enc}(pk_{\mathsf{JL}}, m)$: For a plaintext $0 \leq m < 2^k$, choose a random $x \in \mathbb{Z}_N^\times$ and output a ciphertext $c = y^m x^{2^k} \bmod N$.
- $\mathsf{JL.Dec}(sk_{\mathsf{JL}}, c)$: For a ciphertext $c \in \mathbb{Z}_N^\times$, compute $z = \left(\frac{c}{p}\right)_{2^k}$ and then find and output $m \in \{0, 1\}^k$ such that the relation

$$\left[\left(\frac{y}{p}\right)_{2^k}\right]^m = z \pmod{p}$$

holds.

Remark 1: Note that the case $k = 1$ corresponds to the Goldwasser-Micali cryptosystem.

Definition 7 (Naccache-Stern Encryption): The *Naccache-Stern encryption scheme* $\mathsf{NS} = (\mathsf{NS.KG}, \mathsf{NS.Enc}, \mathsf{NS.Dec})$ consists of the following algorithms:

- $\mathsf{NS.KG}(\lambda)$: Choose random small primes $p_1, \ldots, p_k$, compute $u = \prod_{i=1}^{k/2} p_i$ and $v = \prod_{i=k/2+1}^{k} p_i$ and set $\sigma = uv$. Choose large primes a and b such that

$p = 2au+1$ and $q = 2bv+1$ are prime and set $N = pq$ such that the HR assumption holds with respect to the security parameter λ. Choose a random $g \bmod N$ of order $\phi(n)/4$. Output a public key $pk_{\mathsf{NS}} = (\sigma, N, g)$ and a secret key $sk_{\mathsf{NS}} = (p)$.

- $\mathsf{NS.Enc}(pk_{\mathsf{NS}}, m)$: For a plaintext $m \in \mathbb{Z}_\sigma$, choose $x \in \mathbb{Z}_N$ and output a ciphertext $c = x^\sigma g^m \bmod N$.
- $\mathsf{NS.Dec}(sk_{\mathsf{NS}}, c)$: For a ciphertext $c \in \mathbb{Z}_N^\times$, compute $z = c^{\phi(N)/p_i}$ and find m_i such that the relation

$$\left[\left(\frac{g}{N}\right)_{p_i}\right]^{m_i} = z \pmod N$$

holds for each i. Output

$$m = CRT_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k).$$

III. Encrypt With PKE and Compute With SHE

In this section, we describe the concept of a hybrid scheme that combines PKE and SHE. A message is encrypted using PKE, and converted to a ciphertext under SHE when homomorphic computations on the message are needed. The ciphertext is decrypted under SHE.

A. A Hybrid Scheme of PKE and SHE

Suppose that a client who has limited computation capability wants to compute $f(m_1, \ldots, m_k)$ for sensitive messages $\{m_1, \ldots, m_k\}$ and a multivariate polynomial f. The client could outsource the heavy computation to a server that has sufficient computing power. Currently, FHE is a good way of delegating computations, if we ignore the client's bandwidth and the server's storage. However, these are very significant measures in the construction of a cloud environment, as they are directly connected to the real cost.

The bandwidth and storage requirements can be reduced by combining PKE with a small ciphertext size and SHE that can evaluate the decryption circuit of the PKE, rather than its own decryption circuit. We propose a hybrid scheme to improve the efficiency of SHE when used in a cloud computing environment. Let $\mathsf{PKE} = (\mathsf{PKE.KG}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ be a PKE and $\mathsf{SHE} = (\mathsf{SHE.KG}, \mathsf{SHE.Enc}, \mathsf{SHE.Dec}, \mathsf{SHE.Eval})$ be a SHE. Suppose that there exists a circuit f_{Dec} such that $f_{\mathsf{Dec}}(sk, c) = m$ for all ciphertexts $c = \mathsf{PKE.Enc}(m)$ and secret key sk. We also assume that f_{Dec} can be evaluated homomorphically under SHE. The *Hybrid scheme* of PKE and SHE consists of the following five algorithms $\mathsf{Hyb} = (\mathsf{Hyb.KG}, \mathsf{Hyb.Enc}, \mathsf{Hyb.Conv}, \mathsf{Hyb.Eval}, \mathsf{Hyb.Dec})$:

- $\mathsf{Hyb.KG}(\lambda, \mathsf{PKE.KG}, \mathsf{SHE.KG})$: Run $\mathsf{PKE.KG}$ and $\mathsf{SHE.KG}$ to get $(pk_{\mathsf{PKE}}, sk_{\mathsf{PKE}}, pk_{\mathsf{SHE}}, sk_{\mathsf{SHE}})$. Output

$$pk_{\mathsf{Hyb}} = (pk_{\mathsf{PKE}}, pk_{\mathsf{SHE}}, \mathsf{SHE.Enc}(sk_{\mathsf{PKE}}))$$
$$sk_{\mathsf{Hyb}} = (sk_{\mathsf{SHE}}),$$

where $\mathsf{SHE.Enc}(sk_{\mathsf{PKE}})$ is an encryption of sk_{PKE} under SHE.

- $\mathsf{Hyb.Enc}(pk_{\mathsf{Hyb}}, m)$: For a plaintext $m \in \mathbf{M}_{pk_{\mathsf{PKE}}}$, output $c = \mathsf{PKE.Enc}(pk_{\mathsf{PKE}}, m)$.
- $\mathsf{Hyb.Conv}(pk_{\mathsf{Hyb}}, c)$: Evaluate the decryption circuit $\mathsf{PKE.Dec}$ with pk_{Hyb}. That is, compute and output a ciphertext C, where

$$C = f_{\mathsf{Dec}}\big(\mathsf{SHE.Enc}(sk_{\mathsf{PKE}}), \mathsf{SHE.Enc}(c)\big).$$

- $\mathsf{Hyb.Eval}(pk_{\mathsf{Hyb}}, f, C_1, \ldots, C_t)$: For a given circuit f and t ciphertexts $C_1, \ldots, C_t$ under SHE, output

$$C = \mathsf{SHE.Eval}(pk_{\mathsf{SHE}}, f, C_1, \ldots, C_t).$$

- $\mathsf{Hyb.Dec}(sk_{\mathsf{Hyb}}, c)$: For a ciphertext $C \in \mathbf{C}_{pk_{\mathsf{SHE}}}$, output $m = \mathsf{SHE.Dec}(sk_{\mathsf{SHE}}, C)$.

Remark 2: In the $\mathsf{Hyb.Conv}$ algorithm, note that

$$\begin{aligned} C &= f_{\mathsf{Dec}}(\mathsf{SHE.Enc}(sk_{\mathsf{PKE}}), \mathsf{SHE.Enc}(c)) \\ &= \mathsf{SHE.Enc}(f_{\mathsf{Dec}}(sk_{\mathsf{PKE}}, c)) = \mathsf{SHE.Enc}(m), \end{aligned}$$

since SHE can evaluate the decryption circuit of PKE f_{Dec}.

Remark 3: In our hybrid scheme, the homomorphic capacity of SHE at least exceeds the degree of the decryption circuit f_{Dec} of PKE.

Theorem 1 (Semantic Security of Hybrid Scheme): If both the PKE and the SHE are semantically secure, then so is the hybrid scheme.

Proof: Suppose a polynomial time adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ breaks the semantic security of the scheme with non-negligible advantage. We define the advantage of adversary $\mathcal{B}$ by

$$\begin{aligned} &\mathbf{Adv}_{\mathcal{B}}(\lambda) := \Big| \Pr\big[\mathcal{B}_2(x_0, x_1, pk_{\mathsf{Hyb}}, y) = b \big| \\ &(pk_{\mathsf{PKE}}, sk_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.KG}, (pk_{\mathsf{SHE}}, sk_{\mathsf{SHE}}) \leftarrow \mathsf{SHE.KG}, \\ &(x_0, x_1, pk_{\mathsf{Hyb}}) \leftarrow \mathcal{B}_1(pk_{\mathsf{PKE}}, pk_{\mathsf{SHE}}, \mathsf{SHE}(sk_{\mathsf{PKE}})), \\ &b \leftarrow \{0,1\}, y \leftarrow \mathsf{PKE.Enc}(x_b)\big] - 0.5\Big|, \end{aligned}$$

and denote this by $\mathbf{Adv}_{\mathcal{B}}$. We will show that we can use $\mathcal{B}$ to break either PKE or SHE. We now set up two games for PKE and SHE.

In the game for PKE, the adversary $\mathcal{A}_{\mathsf{PKE}}$ only has access to pk_{PKE}. $\mathcal{A}_{\mathsf{PKE}}$ prepares the following game for $\mathcal{B}$.

1) $\mathcal{A}_{\mathsf{PKE}}$ runs $\mathsf{PKE.KG}$ and $\mathsf{SHE.KG}$ to obtain $(pk'_{\mathsf{PKE}}, sk'_{\mathsf{PKE}})$ and $(pk_{\mathsf{SHE}}, sk_{\mathsf{SHE}})$.
2) $\mathcal{A}_{\mathsf{PKE}}$ sends $(pk_{\mathsf{PKE}}, pk_{\mathsf{SHE}}, \mathsf{SHE.Enc}(sk'_{\mathsf{PKE}}))$ to $\mathcal{B}_1$ as a public key of Hyb.
3) $\mathcal{B}_1$ chooses x_0, x_1 and sends them to $\mathcal{A}_{\mathsf{PKE}}$.
4) $\mathcal{A}_{\mathsf{PKE}}$ selects $b \leftarrow \{0,1\}$ and sends $y \leftarrow \mathsf{PKE.Enc}(x_b)$ to $\mathcal{B}_2$.
5) $\mathcal{B}_2$ outputs $b' \in \{0,1\}$ to $\mathcal{A}_{\mathsf{PKE}}$ as an answer.
6) $\mathcal{A}_{\mathsf{PKE}}$ outputs $b = b'$.

Steps 2)−6) of this game are identical to the standard game for Hyb with an unmatched pair of pk_{Hyb} and pk_{PKE}. Therefore $\mathbf{Adv}_{\mathcal{A}_{\mathsf{PKE}}} \leq \mathbf{Adv}_{\mathcal{B}}$.

In the game for SHE, the adversary $\mathcal{A}_{\mathsf{SHE}}$ only has access to pk_{SHE}. $\mathcal{A}_{\mathsf{SHE}}$ prepares the following game for $\mathcal{B}$.

1) $\mathcal{A}_{\mathsf{SHE}}$ runs $\mathsf{SHE.KG}$ and $\mathsf{SHE.KG}$ to obtain $(pk_{\mathsf{PKE}}, sk_{\mathsf{PKE}})$ and $(pk'_{\mathsf{SHE}}, sk'_{\mathsf{SHE}})$.
2) $\mathcal{A}_{\mathsf{SHE}}$ runs $\mathsf{PKE.KG}$ again to get another secret key sk'_{PKE}.

3) $\mathcal{A}_{\mathsf{SHE}}$ sets $x_0 = sk_{\mathsf{PKE}}, x_1 = sk'_{\mathsf{PKE}}$ and is given $y \leftarrow \mathsf{SHE.Enc}(x_b)$ for some randomly chosen $b \in \{0,1\}$.
4) $\mathcal{A}_{\mathsf{SHE}}$ sends $(pk_{\mathsf{PKE}}, pk_{\mathsf{SHE}}, y)$ to $\mathcal{B}_1$ as a public key of Hyb.
5) $\mathcal{B}_1$ chooses x'_0, x'_1 and sends them to $\mathcal{A}_{\mathsf{SHE}}$.
6) $\mathcal{A}_{\mathsf{SHE}}$ selects $b' \leftarrow \{0,1\}$ and sends $y' \leftarrow \mathsf{PKE.Enc}(x'_{b'})$ to $\mathcal{B}_2$.
7) $\mathcal{B}_2$ outputs $b'' \in \{0,1\}$ to $\mathcal{A}_{\mathsf{SHE}}$ as an answer.
8) If $b'' = b'$, $\mathcal{A}_{\mathsf{SHE}}$ outputs $b = 0$, otherwise $b = 1$ is output.

In this game, if $b = 0$ then y is a valid encryption of sk_{PKE} that matches the public key of PKE contained in pk_{Hyb}. Steps 4) − 7) are identical to the standard game for Hyb. Then, $\mathcal{B}$ will have the advantage $\mathbf{Adv}_{\mathcal{B}}$. If $b = 1$, then y is an encryption of sk'_{PKE} that does not match the public key pk_{PKE}. In this case, the whole process is identical to the game we designed for PKE. Thus, the advantage of $\mathcal{A}_{\mathsf{SHE}}$ is

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}_{\mathsf{SHE}}} &= \Pr[\text{Correct guess}] - 0.5\\ &= \Pr[b=0|\text{Guess } 0] + \Pr[b=1|\text{Guess } 1] - 0.5\\ &= 0.5\cdot(0.5 + \mathbf{Adv}_{\mathcal{B}}) + 0.5\cdot(0.5 - \mathbf{Adv}_{\mathcal{A}_{\mathsf{PKE}}}) - 0.5\\ &= 0.5\mathbf{Adv}_{\mathcal{B}} - 0.5\mathbf{Adv}_{\mathcal{A}_{\mathsf{PKE}}}\end{aligned}$$

Therefore, if $\mathbf{Adv}_{\mathcal{B}}$ is non-negligible, either $\mathbf{Adv}_{\mathcal{A}_{\mathsf{PKE}}}$ or $\mathbf{Adv}_{\mathcal{A}_{\mathsf{SHE}}}$ is non-negligible. ∎

B. Additive Homomorphic Encryptions for PKE in the Hybrid Scheme

In constructing a hybrid scheme, candidate encryptions for an additive homomorphic IND-CPA PKE include Goldwasser–Micali [10], Paillier [18], Okamoto–Uchiyama [17], Naccache–Stern [15] and Joye–Libert [12] encryptions. The decryption circuit of each system requires additional circuits besides exponentiation, the Chinese remainder algorithm for the Goldwasser–Micali and Naccache–Stern encryptions, and integer division for the Paillier and Okamoto–Uchiyama encryptions. Because it is difficult to evaluate the integer division part efficiently, the latter encryptions are unsuitable for the construction of our hybrid scheme. Thus we only consider the Goldwasser–Micali, Joye–Libert, and Naccache–Stern encryptions for PKE in the hybrid scheme.

1) Goldwasser-Micali Encryption: The decryption circuit of Goldwasser-Micali encryption is as follows: for a given ciphertext $c \in \mathbb{Z}_N$, output 0 if $\left(\frac{c}{p}\right)_2 = 1$ and $\left(\frac{c}{q}\right)_2 = 1$ and output 1 otherwise. We can modify the decryption circuit as follows:

1. First, compute
$$m' = CRT_{(p,q)}(c^{(p-1)/2}, c^{(q-1)/2}).$$
2. Output $m = (1 - m')/2$.

Using the homomorphic evaluation of secret exponentiation and the Chinese remainder algorithm, it is possible to evaluate the decryption circuit of Goldwasser-Micali encryption.

We describe a homomorphic decryption method briefly: Suppose that the Goldwasser–Micali encryption scheme over $\mathbb{Z}_N$ and a SHE scheme with message space $\mathbb{Z}_N$ for $N = pq$ are given. Without loss of generality, we assume that $p > q$. To construct hybrid scheme, we use the following two facts:

1) For binary representation of $e = \sum_i e_i 2^i$,
$$v^e = v^{\sum_i e_i 2^i} = \prod_i v^{e_i 2^i} = \prod_i \left(e_i v^{2^i} + (1 - e_i)v^0\right)$$
2) $CRT_{(p,q)}(a,b) = (a\cdot q(q^{-1} \bmod p) + b\cdot p(p^{-1} \bmod q)) \bmod N$.

In Hyb.KG of hybrid scheme, we add encryptions of p_i, q_i for $i = 0, \ldots, \lceil \log p \rceil$ to the pk_{Hyb}, where $(p-1)/2 = \sum_i p_i 2^i$ and $(q-1)/2 = \sum_i q_i 2^i$. Since p and q are secret, the encryptions of $q(q^{-1} \bmod p)$ and $p(p^{-1} \bmod q)$ also need to be added to the public key of the hybrid scheme.

Now, we run Hyb.Conv algorithm to evaluate the decryption circuit of GM. Let us consider a GM ciphertext c. Below, we denote SHE.Enc by Enc and homomorphic multiplication and addition of SHE by $\cdot$ and $+$, respectively.

1) To evaluate exponentiation with secret exponent, compute
$$c_1 = \prod_i \left(\mathsf{Enc}(p_i)\cdot(c^{2^i} \bmod N) + (1 - \mathsf{Enc}(p_i))\right)$$
$$c_2 = \prod_i \left(\mathsf{Enc}(q_i)\cdot(c^{2^i} \bmod N) + (1 - \mathsf{Enc}(q_i))\right)$$
2) To evaluate the Chinese remaindering algorithm, compute
$$c_3 = c_1 \cdot \mathsf{Enc}\left(q(q^{-1} \bmod p)\right) + c_2 \cdot \mathsf{Enc}\left(p(p^{-1} \bmod q)\right)$$
3) Finally, we compute $c_4 = (1 - c_3)\cdot(2^{-1} \bmod N)$

In Step 1), for a given Goldwasser-Micali ciphertext c, we first compute $c^{2^i} \bmod N$ and then perform constant multiplication and addition under SHE, $d_i := \mathsf{Enc}(p_i)\cdot(c^{2^i} \bmod N) + (1 - \mathsf{Enc}(p_i))$. After a product of $\{d_0, \ldots, d_{\lceil \log p \rceil}\}$, we obtain SHE-ciphertext c_1 of degree $2\lceil \log p \rceil$. In Step 2) and 3), the degree of ciphertext is increased by 1,respectively, and so the degree of resulting ciphertext c_4 is $2\lceil \log p \rceil + 2$.

2) Naccache-Stern Encryption: The decryption of the Naccache–Stern encryption computes $z = (\frac{c}{N})_{p_i} = c^{\phi(N)/p_i}$ and find m_i by comparing it with $[(\frac{g}{N})_{p_i}]^j$ for all $j = 0, 1, \ldots, (p_i - 1)$. The message m is recovered by computing $m = CRT_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k)$.

The only difference from GM scheme is to fine m_i by searching z in the set $\{[(\frac{g}{N})_{p_i}]^j\}_{0\le j\le (p_i-1)}$. We use polynomial interpolation to obtain an encryption of m_i. That is, we construct a polynomial $f_i(x)$ of degree $p_i - 1$ such that
$$f_i\left(\left[\left(\frac{g}{N}\right)_{p_i}\right]^k\right) = k \pmod N,$$
for all $k \in \mathbb{Z}_{p_i}$. We use the same algorithms when evaluating $z = (\frac{c}{N})_{p_i} = c^{\phi(N)/p_i}$ and Chinese remaindering. The degree of the decryption circuit of Naccache–Stern is approximately $\log \phi(N) \cdot \max_i\{p_i - 1\}$.

3) Joye–Libert Encryption: The homomorphic decryption of Joye–Libert encryption is the same as the first part of Naccache–Stern decryption if we replace the parameters N, p_i, and g by p, 2^k, and y, respectively. In this case, we need to use SHE with a $\mathbb{Z}_Q$ message space where $Q = 2^k N$. The degree of the decryption circuit is approximately $\log p \cdot (2^k - 1)$.

Remark 4: Although the elliptic curve ElGamal cryptosystem [14] has a small ciphertext, we do not consider it in this paper, since it is difficult to evaluate the inverse map of the message encoding and the addition of points on the elliptic curve.

C. Multiplicative Homomorphic Encryptions for PKE in the Hybrid Scheme

We may consider ElGamal encryption [5] as a candidate for a multiplicative PKE in constructing a hybrid scheme. An ordinary ElGamal encryption over a ring R has a message space $\mathbb{G}_q \subset R$ of prime order q. In other words, elements not in the prime subgroup cannot be securely encrypted under ElGamal encryption. It is possible to take all nonzero element in R as a message only when $R = \mathbb{F}_{2^n}$ for n such that $2^n - 1$ is prime, but we cannot use the ElGamal encryption over $\mathbb{F}_{2^n}$ for PKE. This is because the DLP in an extension field with a small characteristic is no longer considered a hard problem [1], [11]. Unlike the extension field case, it is impossible to construct an ElGamal encryption whose message space is a full domain over an integer ring.

We propose a new multiplicative homomorphic encryption whose message space is $\mathbb{Z}_N^\times$, which covers almost all nonzero elements of $\mathbb{Z}_N$. Our scheme is a combination of the ElGamal scheme over $\mathbb{Z}_N$ [5] and the Goldwasser-Micali encryption scheme over $\mathbb{Z}_N$ [10] for a common $N = p_1 p_2$, and the ciphertext consists of three elements in $\mathbb{Z}_N^\times$. We call this the *EGM* encryption scheme.

First, we need the following Lemma to construct EGM encryption.

Lemma 1: Let $N = p_1 p_2$, where $p_1 = 2q_1 + 1$ and $p_2 = 4q_2 + 1$ for distinct primes p_1, p_2, q_1, q_2 and $q_1, q_2 \neq 2$. Let $\mathsf{J}_N := \{a \in \mathbb{Z}_N^\times | \left(\frac{a}{N}\right) = 1\}$. Then, J_N is a cyclic subgroup in $\mathbb{Z}_N^\times$ of order $4q_1q_2$.

Proof: The order of $\mathbb{Z}_N^\times$ is $\phi(N) = 8q_1q_2$, and the order of J_N is $\phi(N)/2 = 4q_1q_2$. A subgroup of order $4q_1q_2$ in $\mathbb{Z}_N^\times$ is isomorphic to $\mathbb{Z}_2 \oplus \mathbb{Z}_{2q_1q_2}$ or $\mathbb{Z}_{4q_1q_2}$, since the group $\mathbb{Z}_N^\times$ is isomorphic to $\mathbb{Z}_{p_1}^\times \oplus \mathbb{Z}_{p_2}^\times \cong \mathbb{Z}_{2q_1} \oplus \mathbb{Z}_{4q_2}$. If there is an element $\alpha \in \mathbb{Z}_N^\times$ of order 4, then $\mathsf{J}_N \cong \mathbb{Z}_{4q_1q_2}$ and so J_N is a cyclic group of order $4q_1q_2$.

Let us consider generators g_1 of order $2q_1$ in $\mathbb{Z}_{p_1}^\times$ and g_2 of order $4q_2$ in $\mathbb{Z}_{p_2}^\times$ and the map

$$CRT_{(p_1,p_2)} : \mathbb{Z}_{p_1}^\times \times \mathbb{Z}_{p_2}^\times \to \mathbb{Z}_N^\times.$$

Let $\alpha := CRT_{(p_1,p_2)}(g_1^{q_1}, g_2^{q_2})$. Then we can easily verify that the order of α is 4. In addition,

$$\left(\frac{\alpha}{N}\right) = \left(\frac{\alpha}{p_1}\right)\left(\frac{\alpha}{p_2}\right) = \left(\frac{g_1^{q_1}}{p_1}\right)\left(\frac{g_2^{q_2}}{p_2}\right) = (-1)\cdot(-1) = 1.$$

Therefore $\mathsf{J}_N \cong \mathbb{Z}_{4q_1q_2}$ if $q_1, q_2 \neq 2$. ∎

We use the parameters N, p_1, p_1, q_1, q_2 and J_N as defined in Lemma 1. We remark that the Jacobi symbol of -1 is $\left(\frac{-1}{N}\right) = \left(\frac{-1}{p_1}\right)\left(\frac{-1}{p_2}\right) = -1$. Take an element $\sigma \in \mathbb{Z}_N^\times$ with $\left(\frac{\sigma}{p_1}\right) = \left(\frac{\sigma}{p_2}\right) = -1$. We define a bijective map $\iota : \mathbb{Z}_N^\times \to \mathsf{J}_N \times \{0,1\}$ by

$$m \mapsto (\hat{m}, \check{m}) = \left(m \cdot \left(\frac{m}{N}\right), \left(1 - \left(\frac{m}{N}\right)\right)/2\right).$$

The $\mathsf{EGM} = (\mathsf{EGM.KG}, \mathsf{EGM.Enc}, \mathsf{EGM.Dec})$ encryption is as follows:

- $\mathsf{EGM.KG}(\lambda)$: Choose a generator g of J_N with order $\phi(N)/2$ in $\mathbb{Z}_N^\times$ and a random $e \in [0, 4q_1q_2)$, and compute $y \equiv g^e \bmod N$. Output a public key $pk_{\mathsf{EGM}} = (N, g, y, \sigma)$ and a secret key $sk_{\mathsf{EGM}} = (e, p_1, p_2)$.
- $\mathsf{EGM.Enc}(pk_{\mathsf{EGM}}, m)$: For a plaintext $m \in \mathbb{Z}_N^\times$, compute $\iota(m) = (\hat{m}, \check{m})$ and choose a random $r \in [0, N^2)$[3] and a random $h \in \mathbb{Z}_N$. Output a ciphertext $c = (c_1, c_2, c_3) = (g^{-r}, \hat{m} y^r, \sigma^{\check{m}} h^2)$.
- $\mathsf{EGM.Dec}(sk_{\mathsf{EGM}}, c)$: Take as input the secret key sk_{EGM} and a ciphertext $c = (c_1, c_2, c_3) \in (\mathbb{Z}_N^\times)^3$. Compute and output a message $m \equiv c_1^e c_2 \cdot CRT_{(p_1,p_2)}(c_3^{q_1}, c_3^{2q_2}) \bmod N$.

The EGM encryption is multiplicatively homomorphic over $\mathbb{Z}_N^\times$, which covers almost all nonzero element of $\mathbb{Z}_N$. Note that $CRT_{(p_1,p_2)}(c_3^{q_1}, c_3^{2q_2}) = 1$ if $\check{m}$ is even, and $CRT_{(p_1,p_2)}(c_3^{q_1}, c_3^{2q_2}) = -1$ otherwise. From this, the EGM encryption is correct for an unlimited number of multiplications on encrypted data. Additionally, the EGM encryption is semantically secure under the DDH assumption over J_N and the QR assumption over $\mathbb{Z}_N$ for common $N = p_1 p_2$.

Theorem 2 (Multiplicative Homomorphism): For any positive integer k, suppose that $c_i = \mathsf{EGM.Enc}(pk_{\mathsf{EGM}}, m_i)$ for all $i \in [1, k]$. Then,

$$\mathsf{EGM.Dec}(sk_{\mathsf{EGM}}, \prod_{i=1}^{k} c_i) = \prod_{i=1}^{k} m_i,$$

where the multiplication of two ciphertexts is defined as the componentwise product. That is, EGM encryption is multiplicatively homomorphic.

Proof: Suppose that $c_i := (c_{i1}, c_{i2}, c_{i3}) = (g^{r_i}, \hat{m}_i y^{r_i}, \sigma^{\check{m}_i} h_i^2)$ for $i \in [1, k]$. Then

$$C = (C_1, C_2, C_3) := \prod_{i=1}^{k} c_i = (g^{-\sum_{i=1}^{k} r_i}, \prod_{i=1}^{k} \hat{m}_i y^{\sum_{i=1}^{k} r_i}, \sigma^{\sum_{i=1}^{k} \check{m}_i} \prod_{i=1}^{k} h_i^2)$$

We remark that if $\sum_{i=1}^{k} \check{m}_i$ is even, then $CRT_{(p_1,p_2)}(C_3^{q_1}, C_3^{2q_2}) = 1$ and $\prod_{i=1}^{k}(\frac{m_i}{N}) = 1$. Otherwise, $CRT_{(p_1,p_2)}(C_3^{q_1}, C_3^{2q_2}) = -1$ and $\prod_{i=1}^{k}(\frac{m_i}{N}) = -1$.

[3]The statistical distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is at most $1/N$, where $\mathcal{D}_1 := \{\text{choose } r \leftarrow [0, N^2) : \text{output } r \bmod \phi(N)\}$ and $\mathcal{D}_2 := \{\text{choose } r \leftarrow [0, \phi(N)) : \text{output r}\}$. In fact, we can choose $N^{1+\epsilon}$ for some $\epsilon > 0$ instead of N^2.

Thus, we obtain

$$\begin{aligned} & C_1^e C_2 \cdot CRT_{(p_1,p_2)}(C_3^{q_1}, C_3^{2q_2}) \\ &= \Big(\prod_{i=1}^{k} \hat{m}_i\Big) \cdot CRT_{(p_1,p_2)}(C_3^{q_1}, C_3^{2q_2}) \\ &= \Big(\prod_{i=1}^{k} m_i \left(\frac{m_i}{N}\right)\Big) \cdot CRT_{(p_1,p_2)}(C_3^{q_1}, C_3^{2q_2}) \\ &= \prod_{i=1}^{k} m_i \pmod N. \end{aligned}$$

■

Theorem 3 (Semantic Security): The EGM scheme is semantically secure under the DDH assumption over J_N and the QR assumption over $\mathbb{Z}_N$.

Proof of Theorem 3: Under the attacker scenario, the attacker first receives a public key of the encryption scheme, and outputs a message $m_0, m_1 \in \mathbb{Z}_N$. The challenger returns an encryption of m_b for a randomly chosen bit b. Finally, the attacker outputs a guess b' and succeeds if $b = b'$. We use hybrid argument to prove semantic security. We use a sequence of games and denote S_i the event that the attacker succeeds in Game$_i$.

Game$_0$: This is the original attack scenario. That is, we simulate the challenger by running EGM.KG to obtain a public key $pk_0 = (N, g, y, \sigma)$ and a secret key $sk_0 = (e, p_1, p_2)$.

Game$_1$: This game is the same as Game$_0$, except for the following modification to the key generation. Instead of choosing $\sigma \in \mathbb{Z}_N^{\times}$ with $(\frac{\sigma}{p_1}) = (\frac{\sigma}{p_2}) = -1$, we choose it as $\sigma' = t^2$ for a randomly chosen t from $\mathbb{Z}_N$. That is, $pk_1 = (N, g, y, \sigma')$.

It is clear that any significant difference between $\Pr[S_0]$ and $\Pr[S_1]$ leads immediately to an effective statistical test for solving the QR problem over $\mathbb{Z}_N$. Thus we obtain

$$|\Pr[S_0] - \Pr[S_1]| \le \mathbf{Adv}_{\mathsf{QR}},$$

where $\mathbf{Adv}_{\mathsf{QR}}$ denotes the advantage in solving QR problem over $\mathbb{Z}_N$.

Game$_2$: This game is the same as Game$_1$, except for the following modification to the encryption of m_b. Instead of encrypting a m_b as $c = (g^{-r}, \hat{m}_b y^r, \sigma'^{\check{m}_b} h^2)$, we compute $c = (g^{-r}, u, \sigma'^{\check{m}_b} h^2)$ for randomly chosen $r \in [0, N^2)$, $h \in \mathbb{Z}_N$ and $u \in \mathsf{J}_N$ and send c to the attacker as a challenge ciphertext.

It is also clear that any significant difference between $\Pr[S_1]$ and $\Pr[S_2]$ leads immediately to an effective statistical test for solving the DDH problem over J_N. Thus

$$|\Pr[S_1] - \Pr[S_2]| \le \mathbf{Adv}_{\mathsf{DDH}},$$

where $\mathbf{Adv}_{\mathsf{DDH}}$ denotes the advantage in solving DDH problem over J_N.

Since the challenge ciphertext c in Game$_2$ is independent from message m_b, $\Pr[S_2]$ is 1/2. Thus we obtain that

$$\begin{aligned} \left|\Pr[S_0] - \frac{1}{2}\right| &= |\Pr[S_0] - \Pr[S_2]| \\ &\le |\Pr[S_0] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_2]| \\ &\le \mathbf{Adv}_{\mathsf{QR}} + \mathbf{Adv}_{\mathsf{DDH}}. \end{aligned}$$

Thus the advantage of the attacker in Game$_0$ is negligible under DDH assumption over J_N and QR assumption over $\mathbb{Z}_N$. ■

1) Encryption of Zero: Since the EGM scheme has the message space $\mathbb{Z}_N^{\times}$, we cannot encrypt zero or any multiples of p_1 or p_2 in $\mathbb{Z}_N$. As the probability of an encryptor choosing the multiples of p_1 or p_2 is negligible, we are only concerned with zero message.

Borrowing an idea in [21], we can modify the scheme by appending λ Goldwasser–Micali encryptions of an encoding defined by $0 \mapsto \vec{r} \in_R \{0,1\}^{\lambda}$ and $1 \mapsto 0^{\lambda} \in \{0,1\}^{\lambda}$ for 2^{λ} security. The ciphertext of the modified EGM scheme is of the form $(g^{-r}, \hat{m} y^r, h^2 \sigma^{\check{m}}, \mathsf{GM.Enc}(r_1), \ldots, \mathsf{GM.Enc}(r_\lambda))$, where $(r_1, \ldots, r_\lambda) = (0, \ldots, 0)$ for a nonzero message $m \in \mathbb{Z}_N^{\times}$ and a random λ-bit element for the zero element. Note that $\hat{m}$ and $\check{m}$ can be taken arbitrary from $\mathbb{Z}_N^{\times}$ when $m = 0$. The random $(r_1, \ldots, r_\lambda)$ in the appended ciphertext is preserved under multiplications with $1 - 2^{-\lambda}$ probability. The decryption algorithm is similar to the original EGM using a polynomial $f(r_1, \ldots, r_\lambda) = \frac{(-1)^{\lambda}}{\lambda!} \prod_{i=1}^{\lambda} (r_1 + \cdots + r_\lambda - i)$.

2) EGM Encryption: The decryption circuit of EGM encryption consists of three secret exponentiations, two multiplications and one Chinese remaindering algorithm. Similar to the Goldwasser–Micali encryption, we can evaluate the decryption circuit of degree $\log e + \log p_1$ ($\approx \lambda^2$). The message space of EGM is $\mathbb{Z}_N^{\times}$ and the ciphertext space is $(\mathbb{Z}_N^{\times})^3$. Thus we choose the message space of SHE to be $\mathbb{Z}_N$ to construct the hybrid scheme.

IV. Homomorphic Evaluation of Exponentiation

To enhance the performance of the hybrid scheme, we must evaluate a modular exponentiation by a secret exponent efficiently; this is related to the decryption circuit of Goldwasser–Micali, Naccache–Stern, Joye–Libert and EGM encryptions. Actually, this problem has been dealt with by Gentry and Halevi [8] in evaluating the depth-3 decryption circuit of the form $\Sigma\Pi\Sigma$. Their idea is to express the secret key e of the ElGamal encryption as a binary representation, and then convert the exponentiation into a multivariate polynomial. In their approach, the ElGamal decryption circuit is represented by a 4λ–degree polynomial that is too large to evaluate efficiently, where λ is the security parameter. We present an improved algorithm to evaluate an exponentiation with a small degree multivariate polynomial.

A. Improved Exponentiation Using Vector Decomposition

Gentry and Halevi [8] first proposed a method to homomorphically evaluate an exponentiation using a secret exponent. They expand the secret key e of ElGamal encryption as a binary representation $e = \sum_i e_i 2^i$, with $e_i \in \{0,1\}$, and compute v^e as follows:

$$v^e = v^{\sum_i e_i 2^i} = \prod_i v^{e_i 2^i}.$$

They use the Lagrange interpolation to compute $v^{e_i 2^i}$, that is, $v^{e_i 2^i} = e_i v^{2^i} + (1 - e_i) v^0$. The degree of their exponentiation circuit is approximately $2 \log e$ ($\approx 4\lambda$), where λ is the

security parameter. Reducing the degree of exponentiation of secret exponent is a meaningful approach, since the degree of the decryption circuit is directly related to the selection of parameters for SHE. We consider a w-ary ($w > 2$) representation of the secret e to reduce the degree of the exponentiation circuits. Using this w-ary representation, there are $\log_w e$ individual terms $v^{e_i w^i}$, which is fewer than $\log_2 e$. However, a$(w-1)$-degree polynomial is required to express each $v^{e_i w^i}$ for $e_i \in [0, w-1]$ when using Lagrange interpolation. Indeed, this increases the degree of exponentiation by e from $2\log e$ to $w \log_w e$.

Instead of Lagrange interpolation, we use a vector representation of e_i to reduce the degree in computing $v^{e_i w^i}$. First, we expand the secret e in a w-ary representation, $e = \sum_{\ell=0}^{\lfloor \log_w e \rfloor} e_\ell w^\ell$. Then, v^e can be written in the form $v^e = v^{\sum_{\ell=0}^{n} e_\ell w^\ell} = \prod_{\ell=0}^{n} v^{e_\ell w^\ell}$, where $e_\ell \in [0, w-1]$ and $n = \lfloor \log_w e \rfloor$. We define an embedding map π as:

$$\pi : W \longrightarrow \mathbb{Z}^w$$
$$a \longmapsto \vec{f}_{a+1}$$

where $W := \{0, 1, \ldots, w-1\}$ and $F := \{\vec{f}_1, \ldots, \vec{f}_w\}$ is the standard basis in $\mathbb{Z}^w$. We denote $\pi(e_i) = (e_{i0}, \ldots, e_{i(w-1)}) \in \mathbb{Z}^w$, where $e_{ik} \in \{0, 1\}$ for all $i \in [0, n]$ and $k \in [0, w-1]$. We also define a vector $\vec{v}_i := (1, v^{w^i}, v^{2w^i}, \cdots, v^{(w-1)w^i}) \in \mathbb{G}_q^w$. Then it can easily be verified that $v^{e_i w^i} = \langle \vec{v}_i, \pi(e_i) \rangle$, where $\langle \cdot, \cdot \rangle$ is the ordinary inner product in $\mathbb{Z}^w$. To operate this procedure publicly, we add encryptions of $e_{\ell k}$ for all $\ell \in [0, n]$, $k \in [0, w-1]$ under SHE to the public key. In fact, we can omit an encryption of e_{i0}, since e_{i0} is equal to $1 - \sum_{k=1}^{w-1} e_{ik}$ which can be computed homomorphically.

$\mathsf{Eval.Exp.Setup}(pk_{\mathsf{SHE}}, e, w)$: Take as input a public key pk_{SHE} of SHE, secret exponent e, and expansion parameter w.

1) Expand $e = \sum_{i=0}^{n} e_i w^i$ and compute $\pi(e_i) = (e_{i0}, \ldots, e_{i(w-1)})$ for all $i \in [0, n]$, where $n = \lfloor \log_w e \rfloor$.
2) Output

$$\bar{E}_e := \Big\{ \mathsf{SHE.Enc}(e_{ik}) : i \in [0, n], k \in [0, w-1] \Big\}.$$

$\mathsf{Eval.Exp}(pk_{\mathsf{SHE}}, \bar{E}_e, w, v)$: Take as input the public key pk_{FHE} of FHE, set $\bar{E}_e$ output by $\mathsf{Eval.Exp.Setup}$ and v.

1) Encrypt v^{kw^i} for all $i \in [0, n]$, $k \in [0, w-1]$ under SHE.
2) Output

$$c := \prod_{i=0}^{n} \Big(\sum_{k=0}^{w-1} \mathsf{SHE.Enc}(e_{ik}) \cdot \mathsf{SHE.Enc}(v^{kw^i}) \Big).$$

Remark 5: In Step 1, we assume v, w, n are public so that v^{kw^i} can be evaluated publicly.

Remark 6: In Step 2, we use the trivial encryption of v^{kw^i} for all i and k, since they contain no secret information.

Theorem 4 (Correctness): If

$$c \leftarrow \mathsf{Eval.Exp}(pk_{\mathsf{SHE}}, \bar{E}_e, w, v),$$

then $v^e \leftarrow \mathsf{SHE.Dec}_{sk_{\mathsf{SHE}}}(c)$.

The proof of Theorem 4 is straightforward. It has been verified that the degree of exponentiation is approximately $2\log_w e$, which is $\log w$ times smaller than the original method. Using our method, the exponentiation of a large secret exponent can be homomorphically evaluated with a small degree polynomial at a cost of $\tilde{O}(w)$ additional public keys for an arbitrary integer w.

B. Improve the Bootstrapping Without Squashing

Gentry and Halevi [8] proposed a new method to construct FHE without squashing, called *chimeric* FHE. The chimeric FHE uses a multiplicative homomorphic encryption (MHE) to bootstrap a SHE without squashing, thereby removing the assumption on the hardness of the sparse subset sum problem. Gentry and Halevi's technique in [8] expresses the decryption of the SHE scheme as a depth-3 ($\sum \prod \sum$) arithmetic circuit. They temporarily switch to a ciphertext under MHE, such as ElGamal, to compute $\prod$ part. Then they homomorphically evaluate the decryption circuit of MHE to obtain a ciphertext under SHE. Using their method, SHE only needs to evaluate the MHE decryption circuit of fixed degree $2\log e$, rather than its own decryption circuit. Using our efficient evaluation of exponentiation given in Section IV-A, we can reduce the degree from $2\log e$ to $2\log_w e$. Moreover, our method can handle a more general class of SHE whose decryption circuit is a composition of restricted depth-3 circuit $\sum \prod \sum$ and several low depth circuits. By applying our technique, we show that any SHE with a decryption circuit of type $[\cdot]_q \bmod p$ can be bootstrapped if it can evaluate degree $2\log_w e$ circuits.

1) Evaluate Double Modulo Reduction: The bottleneck of the bootstrapping process is the homomorphic computation of $[\cdot]_q \bmod p$. We use the terminology "double modulo reduction" to denote $[\cdot]_q \bmod p$. In general, when the plaintext is $\mathbb{Z}_2$, the bootstrapping proceeds via bit operations on binary representations of integers. However, it is not easy to bootstrap a ciphertext when the message space is $\mathbb{Z}_p$, where $p > 2$. We propose a method to evaluate $[\cdot]_q \bmod p$ homomorphically for large p using Gentry and Halevi's idea [8]. As an application, we present a bootstrapping method of [2] with sufficiently large message. Since the modulus switching technique cannot be used to handle errors of ciphertexts in batch FHE [2], the selection of parameters for FHE is heavily dependent on the homomorphic capacity of the scheme. Thus our improved technique plays an important role in bootstrapping ciphertexts.

We assume that q is equivalent to 1 modulo p. Then $[c]_q \bmod p$ can be written as in the DGHV scheme [22]:

$$[c]_q \bmod p = c - \lfloor c/q \rceil \cdot q \bmod p = c - \lfloor c/q \rceil \bmod p.$$

In comparison with the DGHV scheme, it is hard to express division by p as a low degree polynomial over $\mathbb{Z}_p$ when p is greater than two. To apply the technique in [8], we first modify the division part using Gentry's squashing technique [7]. Let us consider parameters κ, Θ, θ such that $\kappa = \gamma\eta/\rho'$, $\Theta = \omega(\kappa \cdot \log \lambda)$, and $\theta = \lambda$, where γ is a bit length of c, η is a bit length of q which is larger than $\rho' = 2\lambda$.

Set $x_q = \lfloor 2^\kappa/q \rceil$ and choose a Θ-bit random vector $\vec{s} = (s_1, \ldots, s_\Theta)$ with Hamming weight θ. Choose a random integer $u_i \in \mathbb{Z} \cap [0, p \cdot 2^\kappa)$ for $i = 1, \ldots, \Theta$ such that $\sum_i s_i u_i = x_q \pmod{p \cdot 2^\kappa}$. Set $y_i = u_i/2^\kappa$, which is smaller than p with κ precision after the binary point. In addition, $[\sum_i s_i y_i]_p = (1/q) - \Delta_q$ for some $|\Delta_q| < 2^{-\kappa}$. To bootstrap a ciphertext c output by a permuted circuit, we first compute $z_i \leftarrow [c \cdot y_i]_p$, keeping only $n = \lceil \log_2 \theta \rceil + 3$ precision after the binary point for $i = 1, \ldots, \Theta$. That is, $[c \cdot y_i] = z_i - \Delta_i$ for some Δ_i with $|\Delta_i| \le 1/16\theta$. We have

$$\begin{aligned}
&\left[(c/q) - \sum_{i=1}^{\Theta} s_i z_i\right]_p \\
&\quad = \left[(c/q) - \sum_{i=1}^{\Theta} s_i [c \cdot y_i]_p - \sum_{i=1}^{\Theta} s_i \Delta_i\right]_p \\
&\quad = \left[(c/q) - c\left[\sum_{i=1}^{\Theta} s_i \cdot y_i\right]_p - \sum_{i=1}^{\Theta} s_i \Delta_i\right]_p \\
&\quad = \left[(c/q) - c\left[(1/q) - \Delta_q\right]_p - \sum_{i=1}^{\Theta} s_i \Delta_i\right]_p \\
&\quad = \left[c \cdot \Delta_q - \sum_{i=1}^{\Theta} s_i \Delta_i\right]_p .
\end{aligned}$$

Since the bit length of c is at most $2^{\gamma(\eta-4)/(\rho'+2)} < 2^{\kappa-4}$, we have $c \cdot \Delta_q \le 1/16$. Additionally, it can be observed that $|\sum s_i \Delta_i| \le \theta \cdot 1/16\theta = 1/16$. Thus, we have

$$[c]_q \bmod p = c - \lfloor c/q \rceil \bmod p = c - \left\lfloor \sum s_i z_i \right\rceil \bmod p. \quad (1)$$

To apply the chimeric technique [8], we convert the above subset sum into a $\sum\prod\sum$ form, as defined as follows:

Definition 8: Let $\mathcal{L} = \{L_j(x_1, \ldots, x_n)\}$ be a set of polynomials, all in the same n variables. An arithmetic circuit C is an $\mathcal{L}$-restricted depth-3 circuit over $\vec{x} := (x_1, \ldots, x_n)$ if there exists multi sets $S_1, \ldots, S_t \subset \mathcal{L}$ and constants $\lambda_0, \lambda_1, \ldots, \lambda_t$ such that

$$C(\vec{x}) = \lambda_0 + \sum_{i=1}^{t} \lambda_i \cdot \prod_{L_j \in S_i} L_j(x_1, \ldots, x_n).$$

The degree of C with respect to $\mathcal{L}$ is $d = \max_i |S_i|$.

Equation (1) can be converted as follows:

$$\begin{aligned}
[c]_q \bmod p &\equiv c - \lfloor c/q \rfloor \bmod p \\
&\equiv c - \left\lfloor c \cdot \sum_{i=1}^{\Theta} s_i z_i \right\rfloor \bmod p \\
&\equiv \underbrace{c - \sum_{i=1}^{\Theta} s_i z_i'}_{\text{simple part}} - \underbrace{\left\lfloor 2^{-n} \sum_{i=1}^{\Theta} s_i z_i'' \right\rceil}_{\text{complicated part}} \bmod p,
\end{aligned}$$

where $z_i = z_i' + z_i'' \cdot 2^{-n}$ for integers $z_i' \in [0, p)$ and $z_i'' \le 2^n \le 8\Theta$. As well as the simple part, the "complicated part" can be expressed as a $\mathcal{L}_A$-restricted depth-3 circuit C, provided we choose p such that $p > 8\Theta^2$ by the following lemmas.

Lemma 2 [8]: Let Q be a prime with $Q > 2\Theta^2$. Regarding the "complicated part" of the decryption circuit, there is an univariate polynomial $f(x)$ of degree $\le 2\Theta^2$ such that $f(\sum_{i=1}^{\Theta} s_i z_i'') = \lfloor Q^{-n} \sum_{i=1}^{\Theta} s_i z_i'' \rceil \bmod Q$.

Lemma 3 [8]: Let T, Θ be positive integers, and $f(x)$ a univariate polynomial over $\mathbb{Z}_Q$ (for Q prime, $Q \ge T\Theta + 1$). Then there is a multilinear symmetric polynomial M_f on $T\Theta$ variables such that

$$\begin{aligned}
&f\Big(\sum_{i=1}^{\Theta} s_i z_i''\Big) \\
&= M_f(\underbrace{s_1, \ldots, s_1}_{z_1''}, \underbrace{0, \ldots, 0}_{T - z_1''}, \ldots, \underbrace{s_\Theta, \ldots, s_\Theta}_{z_\Theta''}, \ldots, \underbrace{0, \ldots, 0}_{T - z_\Theta''}),
\end{aligned}$$

for all $\vec{s} = (s_1, \ldots, s_\Theta) \in \{0, 1\}^\Theta$ and $z_1'', \ldots, z_\Theta'' \in [0, T]$.

Lemma 4 [8]: Let $Q \ge \Theta + 1$ be a prime, let $A \subset \mathbb{Z}_Q$ have cardinality $\Theta + 1$, let $\vec{x} = (x_1, \ldots, x_\Theta)$ be variables, denote $\mathcal{L}_A = \{(a + x_i : a \in A, 1 \le i \le \Theta\}$. For every multilinear symmetric polynomial $M(\vec{x})$ over $\mathbb{Z}_Q$, there is a circuit $C(\vec{x})$ such that:

- C is a $\mathcal{L}_A$-restricted depth-3 circuit over $\mathbb{Z}_Q$ such that $C(\vec{x}) \equiv M(\vec{x}) \bmod Q$.
- C has $\Theta + 1$ product gates of $\mathcal{L}_A$-degree Θ, one gate for each value $a_j \in A$, with the j-th gate computing the value $\lambda_j \prod_i (a_j + x_i)$ for some constant λ_j.
- A description of C can be computed efficiently given the values $M(\vec{x})$ at all $\vec{x} = 1^i 0^{\Theta - i}$.

Thus we obtain the following theorem:

Theorem 5: Let q, p be primes such that $q > p > 8\Theta^2$. For any $A \subset \mathbb{Z}_p$ of cardinality at least $8\Theta^2 + 1$, the double modulo reduction $[\cdot]_q \bmod p$ can be expressed as $\mathcal{L}_A$-restricted depth-3 circuit C of $\mathcal{L}_A$-degree at most $8\Theta^2$ having at most $8\Theta^2 + \Theta + 1$ product gates. Thus, the double modulo reduction circuit can be evaluated using the chimeric technique.

2) Bootstrapping in [2]: Cheon *et al.* [2] proposed a FHE based on the Chinese remainder theorem. The decryption of the scheme consists of $[\cdot]_{p_i} \bmod Q_i$ with $i \in [1, k]$. Their scheme only achieves "bootstrapping" when all Q_i's are two. They raised the problem of evaluating the double modulo reduction $[\cdot]_{p_i} \bmod Q_i$ homomorphically when some of Q_i are greater than 2. We can partially solve this problem with the above technique, and so complete the bootstrapping stage for sufficiently large Q_i. This gives a FHE that deals with large integers. Note that if the Q_i's are relatively prime, we can map a plaintext on $\mathbb{Z}_M$ with $M = \prod_i Q_i$ into $\prod_i \mathbb{Z}_{Q_i}$, thus enabling large integer arithmetic on $\mathbb{Z}_M$. In this bootstrapping procedure, our improved technique for the evaluation of exponentiation plays an important role, since the parameters of the FHE are heavily dependent on the homomorphic capacity of the scheme. Using our method, the homomorphic capacity can be reduced from $2 \log e$ to $2 \log_w e$ at the cost of public key size $\tilde{O}(w)$.

V. Discussions

We now discuss some typical applications of FHE in database and cloud computing environments and

analyze the advantages of our hybrid scheme under various scenarios.

A. Application Model

1) Database Encryption: Let us consider the situation in which a government agency collects medical records of patients from a hospital, and extracts some statistical information from the records. When lots of data are stored, it becomes more important to protect the data from misuse by insiders or hacking by outsiders. To reduce the risk, the data may be encrypted prior to storage. Under this scenario, we give an efficient storage solution using a hybrid scheme.

First, the agency generates $(pk_{\mathsf{Hyb}}, sk_{\mathsf{Hyb}})$ using the hybrid scheme. The secret key sk_{Hyb} is stored in a secure area, the public key pk_{PKE} is made public to hospitals, and pk_{Hyb} is made public to a database. Each hospital uploads its medical records to the database after encryption under pk_{PKE}. The agency requests the database to perform some computations on the patient data to extract information. The database performs homomorphic computations on the encrypted data using pk_{Hyb}. After evaluating the requested computations, the resulting ciphertexts are sent to the agency. The decryption is carried out in the agency's secure area.

2) Outsourcing of Computations in Cloud Environment: Suppose that a client who has limited computing power wants to perform heavy computation on private data. Our hybrid scheme can be used to protect his privacy, i.e., the computations of PKE-encrypted data are outsourced along with pk_{Hyb} to a cloud that has huge computing power and storage. The cloud performs the outsourced computations, and returns the resulting ciphertext encrypted under SHE. Although the fact that the client must send the large-size pk_{Hyb} appears to be a weakness of our hybrid scheme, this is only a minor concern, since pk_{Hyb} is sent to the cloud only once in the outsourcing procedure.

Remark 7: Since the client may not trust the cloud, it is desirable that the cloud can prove that the computations on the encrypted data were done correctly. Proof of the correctness when using only FHE was given in [3]. Further study on this problem is needed when the hybrid scheme is used to delegate the computations.

B. Advantages

The advantages of using our scheme in the above scenarios include small bandwidth, reduced storage requirements, and computational efficiency. In this section, we consider the scale invariant fully homomorphic encryption based on RLWE proposed in [6].

1) Small Bandwidth and Storage Saving: In a cloud environment, each client encrypts their messages using limited computing power and storage and a server manages the encrypted data with its large computing power and storage. However, the current FHE's may not be suited to this environment, because their large ciphertext size entails a large communication cost. For example, the ciphertext size of scale-invariant FHE based on RLWE allowing multiplicative depth ten in [6] is about 600KB for 80-bit security: the dimension n is 10067 and the modulus q is a 230-bit integer.

In the above scenario, encryptors (each hospital or client) can encrypt data using an efficient AES or PKE scheme to reduce the bandwidth, instead of an inefficient SHE only. Ciphertexts of only a few thousand bits are then sent to the database. This could reduce the encryptors' bandwidth dramatically. When transmitting μ-bit data using the hybrid scheme of AES and SHE, encryptors need to send encryptions of their own AES secret key to convert AES ciphertexts into SHE ciphertexts as well as data encryptions. (say, *conversion key*) Since the size of SHE ciphertext which supports multiplicative depth fifty (forty for decryption of AES and ten for homomorphic evaluation) is 16MB, the encryptors send an additional 128×16MB conversion key.[4]

On the other hand, when using the hybrid scheme of EGM and SHE, encryptors only send EGM encryption of size $\lceil \frac{\mu}{1024} \rceil \cdot 3072$bit to send μ-bit without any additional conversion key. The conversion key of the EGM scheme is sent to the server by decryptor (the government agency).

After receiving the ciphertext from each encryptor, the server stores and computes on the ciphertexts which contain secret information of encryptors. The server can reduce the storage requirement by storing only small AES or PKE ciphertexts, instead of large SHE ciphertexts. The server converts these to SHE ciphertexts and performs the necessary operations only when required. In the case of hybrid scheme of AES and SHE, the server has to store the conversion key of each clients as well as the public key (containing evaluation key) to evaluate functions on encrypted data. In the hybrid scheme of AES and SHE, the public key size is $4n \log q = 32$MB.[5]

Let us consider the hybrid scheme of EGM and SHE. The message space of the EGM scheme is $\mathbb{Z}_N^\times$ for a 1024-bit integer $N = pq$, and the ciphertext is of the form $(C_1, C_2, C_3) \in (\mathbb{Z}_N^\times)^3$. To evaluate the decryption of EGM, we choose the message space of SHE to be $\mathbb{Z}_N$. In scale invariant SHE based on RLWE [6], we choose the dimension and modulus as follows:

$$n = \frac{(L(\log t + \log n + 23) - 8.5)(\kappa + 110)}{7.2}$$
$$\log q = L(\log t + \log n + \log\log q),$$

for multiplicative depth L, message space $\mathbb{Z}_t$ and security parameter κ. Thus, we obtain the ciphertext size of 3GB by substituting $L = 20$ and $\log t = 1024$. In this case, we have to add encryptions of EGM secret key in the public key to convert EGM ciphertexts into SHE ciphertexts. Since q_1 and q_2 are 512 bit integers, we add 1024 additional ciphertexts to the public key when using binary expansion of

[4]We refer parameter analysis of [9]. In state-wise bit-slicing variant of AES evaluation, they encrypt each bit of AES secret key separately. When the SHE allows homomorphic multiplications unto depth fifty, the degree n of the base ring is 51234, and the modulus q is a 1250-bit integer. We can obtain $2n \log q (= 16$MB) size SHE ciphertext.

[5]Here, if we use a new key switching technique proposed in [9], the evaluation key size is $2n \log q$.

TABLE I
COMPARISON BETWEEN SHE SCHEMES, THE HYBRID SCHEME OF AES WITH SHE, AND THE HYBRID SCHEME OF EGM WITH SHE IN REGARD TO THE TRANSMITTED CIPHERTEXT SIZE, CIPHERTEXT EXPANSION RATIO, SHE CIPHERTEXT SIZE, PUBLIC KEY SIZE, AND MULTIPLICATIVE DEPTH OF SHE

Schemes	Transmitted Ciphertext Size (μ-bit encryption)	Ciphertext Expansion Ratio (Asymptotic)	SHE Ciphertext Size	Public-Key Size	Multiplicative Depth of SHE (allowing homo. eval. of depth 10)
SHE Only	$\left\lceil \frac{\mu}{2\cdot 10^3} \right\rceil \cdot$600KB	2400	600KB	1.2MB	10
AES with SHE	128$\cdot$ 16MB+ $\left\lceil \frac{\mu}{128} \right\rceil \cdot$128bit	1	16MB	32MB	50
EGM with SHE	$\left\lceil \frac{\mu}{1024} \right\rceil \cdot$3072bit	3	3GB	3TB	20

q_1 and q_2. Therefore, the size of public key is 3TB. If we can evaluate the modulo N arithmetic under SHE with a smaller message space, we expect to reduce the size of ciphertext and public key of SHE. In the hybrid scheme of EGM and SHE, the public key size is much larger. However, differently from the hybrid scheme of AES and SHE, it has an advantage that the server do not need to store each encryptor's conversion key.

Our hybrid scheme of EGM and SHE has more efficient bandwidth and storage than the hybrid scheme of AES and SHE scheme when more than 1500 encryptors participate in the above scenario and each encryptor sends data whose size is smaller than 1GB.[6]

We summarize the theoretical comparison of the size of the transmitted ciphertext, SHE-ciphertext and public key in Table I when each cryptosystem allows homomorphic evaluation of depth ten multivariate functions.

2) Efficient Computing: In our hybrid scheme, we can choose an additive or multiplicative homomorphic PKE depending on the property of the circuit we are to evaluate. Suppose that the server is to evaluate multivariate polynomials f and g, where f has polynomially many monomials on inputs and g has polynomially many linear factors. We will use a multiplicative homomorphic PKE in the first case, and an additive homomorphic PKE in the second case.

First, let us consider

$$f(x_1, \ldots, x_n) = \sum_{i \in I} M_i(x_1, \ldots, x_n),$$

which is an n-variable polynomial over a ring R. Each M_i is a monomial of f. If the degree of f is large, several decryption or modulus switching procedures are required when using ordinary FHE. These are slow, or increase the ciphertext size. However, using our hybrid scheme, we can evaluate f without bootstrapping, regardless of its degree. Suppose the ciphertexts are encrypted under a multiplicative encryption E with key (pk, sk) and SHE can evaluate the decryption circuit $\mathsf{D}(sk, \cdot)$ of E_{pk}. Given $c_1 = \mathsf{E}_{pk}(m_1), \ldots, c_n = \mathsf{E}_{pk}(m_n)$, we can compute $\mathsf{SHE.Enc}(f(m_1, \ldots, m_n))$ with $\mathsf{SHE.Enc}(sk)$. Below, we denote $\mathsf{SHE.Enc}$ by SHE and $\mathsf{SHE.Dec}(a) = \mathsf{SHE.Dec}(b)$ by $a \equiv b$.

$$\begin{aligned}
&\mathsf{SHE}(f(m_1, \ldots, m_n)) \\
&\quad \equiv \sum_i \mathsf{SHE}(M_i(m_1, \ldots, m_n)) \\
&\qquad (\because \mathsf{SHE} \text{ is additive homomorphic.}) \\
&\quad \equiv \sum_i \mathsf{SHE}\{\mathsf{D}(sk, \mathsf{E}_{pk}(M_i(m_1, \ldots, m_n)))\} \\
&\qquad (\because \mathsf{D} \circ \mathsf{E} \text{ is an identity}) \\
&\quad \equiv \sum_i \mathsf{SHE}\{\mathsf{D}(sk, M_i(\mathsf{E}_{pk}(m_1), \ldots, \mathsf{E}_{pk}(m_n)))\} \\
&\qquad (\because \mathsf{E} \text{ is multiplicative homomorphic.}) \\
&\quad \equiv \sum_i \bar{\mathsf{D}}(\mathsf{SHE}(sk), \mathsf{SHE}(M_i(\mathsf{E}_{pk}(m_1), \ldots, \mathsf{E}_{pk}(m_n)))),
\end{aligned}$$

where $\bar{\mathsf{D}}$ is the decryption circuit encrypted with SHE and the last equality holds because SHE can evaluate D with $\mathsf{SHE}(sk)$. More specifically, we follow the steps:

1) Given $c_1 = \mathsf{E}_{pk}(m_1), \cdots, c_n = \mathsf{E}_{pk}(m_n)$, compute $M_i(c_1, \cdots, c_n) = \mathsf{E}_{pk}(M_i(m_1, \cdots, m_n))$ for each i.
2) Encrypt $\mathsf{E}_{pk}(M_i(m_1, \cdots, m_n))$ with SHE and then evaluate the decryption circuit $\bar{\mathsf{D}}$ with the encrypted secret key $\mathsf{SHE}(sk)$ of E_{pk}. (Observe: one may use trivial encryption on $\mathsf{E}_{pk}(M_i)$.)
3) Add them to obtain $\mathsf{SHE}(f(m_1, \ldots, m_n))$.

Now let us consider that the server is to compute a polynomial $g(x_1, \ldots, x_n) = \prod_{i \in I} L_i(x_1, \ldots, x_n)$ where each L_i is a linear multivariate factor of g. Suppose that the ciphertexts are encrypted under an additive homomorphic encryption E. In this case, we follow the steps:

1) Given $c_1 = \mathsf{E}_{pk}(m_1), \cdots, c_n = \mathsf{E}_{pk}(m_n)$, compute $L_i(c_1, \cdots, c_n) = \mathsf{E}_{pk}(L_i(m_1, \cdots, m_n))$ for each i.
2) Encrypt $\mathsf{E}_{pk}(L_i(m_1, \cdots, m_n))$ with SHE and then evaluate the decryption circuit $\bar{\mathsf{D}}$ with the encrypted secret key $\mathsf{SHE}(sk)$ of E_{pk}. (Observe: one may use trivial encryption on $\mathsf{E}_{pk}(L_i)$.)
3) Multiply them to obtain $\mathsf{SHE}(f(m_1, \ldots, m_n))$.

Remark 8: After converting the ciphertexts, we could compute them under SHE rather than PKE. Therefore, better use is made of the hybrid scheme when evaluating fixed multivariate polynomials.

[6]Suppose that m encryptors participate in and each encryptor sends k-bit data to the server. Then our approach is more advantageous when 3TB $<$ $(2m)$GB and $3k$-bit $<$ (2GB + k-bit).

VI. Generic Conversion of SHE From Private-Key to Public-Key

Rothblum [20] showed the way how to transform any additively homomorphic private-key encryption scheme into a public-key homomorphic encryption scheme when the message is $\mathbb{Z}_2$. To apply this method, the private-key SHE needs to be compact which means that the length of a homomorphically generated encryption is independent of the number of ciphertexts from which it was created. An additive homomorphic encryption is converted from private-key to public key by adding a number of encryptions of zero and one to the public key.

We can consider our hybrid scheme to be a generic conversion of SHE from private-key to public key whose message space is $\mathbb{Z}_p$ for some large prime p. We need only add encryptions of the secret key of a PKE under the private SHE to the public key instead of $\{\mathsf{SHE.Enc}_i(0)\}_i$ and $\{\mathsf{SHE.Enc}_i(1)\}_i$. The encryption algorithm of "public-key" SHE is made up of Hyb.Enc and Hyb.Conv. Given a hybrid scheme $\mathsf{Hyb} = (\mathsf{Hyb.KG}, \mathsf{Hyb.Enc}, \mathsf{Hyb.Conv}, \mathsf{Hyb.Dec}, \mathsf{Hyb.Eval})$ of PKE and a private key SHE scheme PrivSHE, we could construct a public key SHE PubSHE as follows:

PubSHE.KG(λ): Run PKE.KG to obtain pk_{PKE} and sk_{PKE}. Output a public key $pk_{\mathsf{PubSHE}} = (pk_{\mathsf{Hyb}})$ and a secret key $sk_{\mathsf{PubSHE}} = (sk_{\mathsf{Hyb}})$.

PubSHE.Enc(pk_{PubSHE}, m): For a plaintext $m \in \mathbb{Z}_p$, encrypt m under Hyb.Enc and then output a ciphertext $C \leftarrow \mathsf{Hyb.Conv}(pk_{\mathsf{PubSHE}}, c)$ where $c \leftarrow \mathsf{Hyb.Enc}(m)$.

PubSHE.Dec(sk_{PubSHE}, C): For a ciphertext C, output a message $m \leftarrow \mathsf{Hyb.Dec}(sk_{\mathsf{PubSHE}}, C)$.

The semantic security of this conversion follows that of the hybrid scheme.

VII. Conclusion

We proposed a hybrid scheme that combines public key encryption and somewhat homomorphic encryption. The proposed scheme is suitable for cloud computing environments since it has small bandwidth, low storage requirement, and supports efficient computing on encrypted data.

Our solution provides a trade-off between the size of the transmitted ciphertexts and the conversion costs. While the ciphertext expansion of PKE is larger than that of AES, it can be homomorphically evaluated with a SHE of much smaller multiplicative depth.

The parameters of our hybrid scheme are very large when the message space of the underlying FHE is $\mathbb{Z}_N$. For an efficient implementation, we need a method to evaluate mod N arithmetic using an FHE whose message space is $\mathbb{Z}_M$ for small $M > 2$.

Acknowledgement

The authors would like to thank Hyung Tae Lee and the anonymous reviewers for their valuable and helpful comments.

References

[1] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé. (2013). "A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic." [Online]. Available: http://eprint.iacr.org/2013/400

[2] J. H. Cheon *et al.*, "Batch fully homomorphic encryption over the integers," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7881, T. Johansson and P. Nguyen, Eds. Berlin, Germany: Springer-Verlag, 2013, pp. 315–335.

[3] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6223, T. Rabin, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 483–501.

[4] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Scale-invariant fully homomorphic encryption over the integers," in *Public-Key Cryptography* (Lecture Notes in Computer Science), H. Krawczyk, Ed. Berlin, Germany: Springer-Verlag, 2014, pp. 311–328.

[5] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 196, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer-Verlag, 1984, pp. 10–18.

[6] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," in *Proc. IACR Cryptol.*, 2012, p. 144.

[7] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput. (STOC)*, 2009, pp. 169–178.

[8] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," in *Proc. IEEE 52nd Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2011, pp. 107–109.

[9] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 850–867.

[10] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[11] A. Joux, "A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic," in *Selected Areas in Cryptography* (Lecture Notes in Computer Science). New York, NY, USA: Springer, 2013, pp. 355–379.

[12] M. Joye and B. Libert, "Efficient cryptosystems from 2^k-th power residue symbols," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2013.

[13] J. Kim, M. S. Lee, A. Yun, and J. H. Cheon, "CRT-based fully homomorphic encryption over the integers," in *Proc. IACR Cryptol.*, 2013, paper 2013/057.

[14] A. J. Menezes and S. A. Vanstone, "Elliptic curve cryptosystems and their implementation," *J. Cryptol.*, vol. 6, no. 4, pp. 209–224, 1993.

[15] D. Naccache and J. Stern, "A new public key cryptosystem based on higher residues," in *Proc. 5th ACM Conf. Comput. Commun. Security*, 1998, pp. 59–66.

[16] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proc. 3rd ACM Cloud Comput. Security Workshop (CCSW)*, 2011, pp. 113–124.

[17] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1403, K. Nyberg, Ed. Berlin, Germany: Springer-Verlag, 1998, pp. 308–318.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1592, J. Stern, Ed. Berlin, Germany: Springer-Verlag, 1999, pp. 223–238.

[19] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Found. Secure Comput.*, vol. 4, no. 11, pp. 169–180, 1978.

[20] R. Rothblum, "Homomorphic encryption: From private-key to public-key," in *Theory of Cryptography* (Lecture Notes in Computer Science), vol. 6597, Y. Ishai, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 219–234.

[21] T. Sander, A. Young, and M. Yung, "Non-interactive cryptocomputing for NC^1," in *Proc. 40th Annu. Symp. Found. Comput. Sci. (FOCS)*, New York, NY, USA, Oct. 1999, pp. 554–567.

[22] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6110, H. Gilbert, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 24–43.

Jung Hee Cheon is currently a Professor with the Department of Mathematical Sciences, Seoul National University, Seoul, Korea, and the Director of the Cryptographic Hard Problems Research Initiatives. He received the B.S. and Ph.D. degrees in mathematics from the Korea Advanced Institute of Science and Technology, Daejeon, in 1991 and 1997, respectively. After working as a Senior Researcher with the Electronics and Telecommunications Research Institute, Daejeon, and a Visiting Scientist with Brown University, Providence, RI, USA, he was an Assistant Professor with Information and Communications University, Daejeon, for two years. His research interests include computational number theory, cryptography, and information security. He was the Cochair of ICISC, ANTS, and Asiacrypt, and served as a Program Committee Member of many IACR conferences. He was a recipient of the best paper award at Asiacrypt in 2008.

Jinsu Kim is currently pursuing the Ph.D. degree with the Department of Mathematical Sciences, Seoul National University (SNU), Seoul, Korea. He received the B.S. degree in mathematical education from SNU, in 2009. His research interests include computational number theory, cryptography, and information security.