CRT-based Fully Homomorphic Encryption over the Integers*

Jinsu Kim[1], Moon Sung Lee[1], Aaram Yun[2] and Jung Hee Cheon[1]

[1] Seoul National University (SNU), Republic of Korea
[2] Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
kjs2002@snu.ac.kr, moolee@snu.ac.kr, aaramyun@unist.ac.kr, jhcheon@snu.ac.kr

Abstract. In 1978, Rivest, Adleman and Dertouzos introduced the basic concept of privacy homomorphism that allows computation on encrypted data without decryption. It was elegant work that precedes the recent development of fully homomorphic encryption schemes although there were found some security flaws, e.g., ring homomorphic schemes are broken by the known-plaintext attacks.

In this paper, we revisit one of their proposals, in particular the third scheme which is based on the Chinese Remainder Theorem and is ring homomorphic. The previous result is that only a single pair of known plaintext/ciphertext can break this scheme. However, by exploiting the standard technique to insert an error to a message before encryption, we can cope with this problem. We present a secure modification of their proposal by showing that the proposed scheme is fully homomorphic and secure against the chosen plaintext attacks under the decisional approximate GCD assumption and the sparse subset sum assumption when the message space is restricted to $\mathbb{Z}_2^k$.

Interestingly, the proposed scheme can be regarded as a generalization of the DGHV scheme with larger plaintext. Our scheme has $\tilde{O}(\lambda^5)$ overhead while the DGHV has $\tilde{O}(\lambda^8)$ for the security parameter λ. When restricted to the homomorphic encryption scheme with depth-$O(\log \lambda)$, the overhead is reduced to $\tilde{O}(\lambda)$. Our scheme can be used in applications requiring a large message space $\mathbb{Z}_Q$ for $\log Q = O(\lambda^4)$ or SIMD style operations on $\mathbb{Z}_Q^k$ for $\log Q = O(\lambda), k = O(\lambda^3)$, with $\tilde{O}(\lambda^5)$ ciphertext size as in the DGHV.

Keywords: privacy homomorphism, Chinese remainder theorem, homomorphic encryption, approximate gcd, DGHV

1 Introduction

The concept of computation on encrypted data without decryption was firstly introduced by Rivest, Adleman and Detourzos in 1978 [23]. They defined a *privacy homomorphism* to be an encryption $\mathbf{Enc} : \mathcal{P} \to \mathcal{C}$ which permits the computation $\mathbf{Enc}(m_1 * m_2)$ from $\mathbf{Enc}(m_1), \mathbf{Enc}(m_2)$ without revealing m_1 and m_2 for a certain binary operation $*$ on $\mathcal{P}$. They presented the five privacy homomorphisms [23], but no ring homomorphic scheme is secure under the known plaintext attacks [4].

One example of a privacy homomorphism given in [23] is as follows. Let p, q be large primes and $n = pq$. The plaintext space is $\mathbb{Z}_n$ and the ciphertext space is $\mathbb{Z}_p \times \mathbb{Z}_q$. An encryption of a message $m \in \mathbb{Z}_n$ is $(m \bmod p, m \bmod q)$ and a decryption is done using the Chinese Remainder Theorem (CRT). This cryptosystem is a privacy homomorphism under modular addition and modular multiplication. Unfortunately, it is shown that this privacy homomorphism is broken under the known plaintext attack [4]. In fact, $p | \gcd(m - c_1, n)$ and $q | \gcd(m - c_2, n)$ when $\mathbf{Enc}(m) = (c_1, c_2)$. Later, Domingo-Ferrer proposed its variants using additional secret key

* An extended abstract [7] will appear at Eurocrypt 2013, merged with some independent but overlapping work from Coron et al. [10]. A part of this paper was made public through [1].

$r_p \in \mathbb{Z}_p^*, r_q \in \mathbb{Z}_q^*$, but it is also broken under the known plaintext attacks [28, 8]. To avoid the known plaintext attacks, we may consider hiding a message by substituting a part of a message with a random error as in the recent fully homomorphic encryptions.

Basic Idea We denote by $a \bmod p$ the unique integer in $(-\frac{p}{2}, \frac{p}{2}]$ that is congruent to a modulo p, and by $\mathrm{CRT}_{(p_0,\ldots,p_k)}(m_0,\ldots,m_k)$ the unique integer in $(-\frac{\prod_i p_i}{2}, \frac{\prod_i p_i}{2}]$ which is congruent to m_i modulo p_i for all i. Our basic symmetric encryption scheme is as follows:

KeyGen(λ): Choose large pairwise coprime integers p_i $(i = 0,\ldots,k)$ and relatively small pairwise coprime integers Q_i $(i = 1,\ldots,k)$. Let $n = \prod_{i=0}^{k} p_i$. Output the secret key $sk = (p_0,\ldots,p_k)$ and the public parameter $pp = (n, Q_1,\ldots,Q_k)$. The message space is $\mathbb{Z}_Q$ for $Q = \prod_{i=1}^{k} Q_i$.

Enc(sk, m): Output $c = \mathrm{CRT}_{(p_0,\ldots,p_k)}(e, m_1+e_1Q_1,\ldots,m_k+e_kQ_k)$ where $m_i = m \bmod Q_i$ for all i, e is a random integer in $(-p_0/2, p_0/2]$ and $e_1,\cdots,e_k$ are ρ-bit random integers.

Dec(sk, c): Output $m = \mathrm{CRT}_{(Q_1,\ldots,Q_k)}(d_1,\ldots,d_k)$ where $d_i = (c \bmod p_i) \bmod Q_i$ for all i.

Since the CRT is a ring isomorphism from $\prod_i \mathbb{Z}_{p_i}$ to $\mathbb{Z}_n$ with respect to modular additions and multiplications, **Dec** function is also ring homomorphic. However, to decrypt a ciphertext correctly after operations of ciphertexts, the size of e_i and Q_i must be sufficiently smaller than that of p_i.

This scheme is a symmetric key encryption scheme which permits bounded number of modular additions and multiplications. We can extend this scheme to a somewhat homomorphic public key encryption scheme by publishing many encryptions of zero and k elementary elements $E_i = \mathrm{CRT}_{(p_0,\ldots,p_k)}(0,\ldots,0,1,0,\ldots,0)$.

We reduce the security of our Somewhat Homomorphic Encryption (SWHE) scheme to a decisional version of Approximate GCD problem (DACD). Approximate GCD (ACD) problem is to find p given many multiples of p with some errors (i.e. $x_i = pq_i + e_i$). We remark that the security of the DGHV scheme [13] is reduced to the ACD and that of its efficient variant [12] to a decisional version of the ACD problem that is slightly different from ours for modulus switching.

In fact, our scheme can be regarded as a generalization of the DGHV scheme, but with larger plaintext space. Our scheme can be extended to a Fully Homomorphic Encryption (FHE) through bootstrapping and squashing the decryption circuit as in [14, 13]. However, this could be done only when $Q_1 = \cdots = Q_k = 2$, and is not resolved for larger Q_i cases.

Let λ be the security parameter. The ciphertext size of our SWHE scheme is $\tilde{O}(\lambda^5)$ as in the DGHV scheme. While the plaintext size of the DGHV is $O(\lambda)$, that of ours is $O(\lambda^4)$ for $O(\lambda)$-bit $Q_1,\ldots,Q_k$ with $k = O(\lambda^3)$. Consequently, our scheme reduces the overheads (ratio of ciphertext computation and plaintext computation) from $\tilde{O}(\lambda^4)$ to $\tilde{O}(\lambda)$. For the fully homomorphic schemes, the overhead is reduced from $\tilde{O}(\lambda^8)$ to $\tilde{O}(\lambda^5)$ with message space $\mathbb{Z}_2^k$ for $k = O(\lambda^3)$.

Our scheme has an advantage in the applications requiring larger message space than [17]. When dealing with arithmetic on $\mathbb{Z}_Q$ for $\log Q = O(\lambda^4)$, our SWHE scheme can support $O(\lambda)$ multiplications with many additions. One of the important applications of homomorphic encryption schemes is to securely evaluate a multivariate function over the integers. Our scheme is the best choice when evaluating a function of degree $O(\lambda)$ with inputs $\Omega(\lambda^2)$. Also

our scheme can be used in the applications requiring SIMD style operations in k copies of $\mathbb{Z}_Q$ for $\log Q = \lambda, k = O(\lambda^3)$. Detailed comparison is given in Section 5.

Related works In 2009, Gentry [14, 15] introduced the first fully homomorphic encryption scheme based on ideal lattices which supports arbitrary many additions and multiplications on encrypted bit. His breakthrough paper drew an explosive interest and leads numerous researches in this area [13, 11, 12, 16, 26, 27, 24, 17, 3, 2]. Gentry's scheme and its variants [14, 15, 26, 27] are based on hard problems on ideal lattices. Another class of schemes [13, 11, 12] relies on the approximate GCD problem. The message space of the schemes is $\mathbb{Z}_2$ and so the overhead is rather high due to the large ciphertext expansion ratio. Our scheme improves their efficiency. Recent schemes based on the learning with error (LWE) or the ring-LWE are more efficient and accomplish polylogarithmic overhead for wide enough arithmetic circuits on $\mathbb{Z}_p$ for $p = poly(\lambda)$. For more related works, refer to [18].

Organization In Section 2, we define some notations and basic problems. Our main scheme and the security proof are represented in Section 3 and Section 4, respectively. Applications and comparison with other schemes are given in Section 5. We explain how to extend our scheme to a fully homomorphic encryption in Section 6. Finally, further works are given in Section 7.

2 Preliminaries

Notation. We use $a \leftarrow A$ to denote choosing an element a from a set A randomly. When $\mathcal{D}$ is a distribution, we use $a \leftarrow \mathcal{D}$ to denote choosing an element a according to the distribution $\mathcal{D}$. We use $\mathbb{Z}_p := \mathbb{Z} \cap (\frac{-p}{2}, \frac{p}{2}]$ and $x \bmod p$ denotes a number in $\mathbb{Z} \cap (\frac{-p}{2}, \frac{p}{2}]$ which is equivalent to x modulo p. We use notation $(a_i)^k$ for a vector $(a_1, \ldots, a_k)$. So $\mathcal{D}_\rho(p_1, \ldots, p_k; Q_1, \ldots, Q_k; q_0)$ defined below can be denoted by $\mathcal{D}_\rho\big((p_i)^k; (Q_i)^k; q_0\big)$.

For pairwise coprime integers $p_1, \ldots, p_k$, we define $\mathrm{CRT}_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k)$ as a number in $\mathbb{Z} \cap (\frac{-x_0}{2}, \frac{x_0}{2}]$ which is equivalent to m_i modulo p_i for all $i \in \{1, \ldots, k\}$ where $x_0 = \prod_{i=1}^k p_i$. That is,

$$\mathrm{CRT}_{(p_1,\ldots,p_k)}(m_1, \ldots, m_k) \equiv \sum_{i=1}^{k} m_i \hat{p_i} \left(\hat{p_i}^{-1} \bmod p_i\right) \bmod x_0,$$

where $\hat{p_i} = \frac{x_0}{p_i} = \frac{\prod_{j=1}^k p_j}{p_i}$.

For η-bit primes $p_1, \ldots, p_k$ and ℓ_Q-bit integers $Q_1, \ldots, Q_k$, we define the following distributions:

$$\mathcal{D}_{\gamma,\rho}(p) := \{\text{choose } q \leftarrow \mathbb{Z} \cap [0, \frac{2^\gamma}{p}), e \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : \text{output } x = pq + e\},$$

$$\mathcal{D}_\rho(p_1, \ldots, p_k; q_0) := \Big\{\text{choose } e_0 \leftarrow \mathbb{Z} \cap [0, q_0), e_i \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) \text{ for } \forall i \in \{1, \ldots, k\}$$
$$: \text{output } x = \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)}(e_0, \ldots, e_k)\Big\},$$

$$\mathcal{D}_\rho(p_1, \ldots, p_k; Q_1, \ldots, Q_k; q_0) := \Big\{\text{choose } e_0 \leftarrow \mathbb{Z} \cap [0, q_0), e_i \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$$
$$\text{for } \forall i \in \{1, \ldots, k\} : \text{output } x = \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)}(e_0, e_1 Q_1, \ldots, e_k Q_k)\Big\}.$$

Remark 1. When $k = 1$, $\mathcal{D}_\rho(p_1; q_0)$ is identical to $\mathcal{D} := \{\text{choose } q \leftarrow \mathbb{Z} \cap [0, q_0), e \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho) : \text{output } x = p_1 q + e \bmod p_1 q_0\}$. For $x \leftarrow \mathcal{D}_\rho(p_1; q_0)$,

$$\begin{aligned} x &= \mathrm{CRT}_{(q_0,p_1)}(e_0, e_1) \\ &= e_0 p_1 (p_1^{-1} \bmod q_0) + e_1 q_0 (q_0^{-1} \bmod p_1) \bmod q_0 p_1 \\ &= e_0 p_1 \alpha + e_1 (p_1 \beta + 1) \bmod q_0 p_1 = (e_0 \alpha + e_1 \beta) p_1 + e_1 \bmod q_0 p_1 \end{aligned}$$

for some α and β. If e_0 is chosen from $\mathbb{Z} \cap [0, q_0)$ uniformly, $(e_0\alpha + e_1\beta) \bmod q_0$ is uniform in $\mathbb{Z} \cap [0, q_0)$ when $\gcd(\alpha, q_0) = 1$.

There are two versions of approximate GCD problem defined by Howgrave-Graham [19]. One is a general approximate GCD problem and the other is a partially approximate GCD problem.

General Approximate GCD problem. The (ρ, η, γ)-computational general approximate GCD problem is: for an η-bit prime p, given polynomially many samples from $\mathcal{D}_{\gamma,\rho}(p)$, find p.

Partially Approximate GCD problem. The (ρ, η, γ)-computational partially approximate GCD problem is: for an η-bit prime p, given a γ-bit integer $x_0 = pq_0$ and polynomially many samples from $\mathcal{D}_{\gamma,\rho}(p)$, find p.

In this paper, we use only partially approximate GCD problem, we omit the term 'partially' throughout the paper, and denote it by ACD. The ACD assumption is that ACD problem is hard for any polynomial time attacker.

3 Our Somewhat Homomorphic Encryption Scheme

We propose a homomorphic encryption supporting large integer arithmetic or SIMD operations. The message space is $\prod_{i=1}^{k} \mathbb{Z}_{Q_i}$. If $Q_1, \ldots, Q_k$ are pairwise coprime integers, the message space can be considered $\mathbb{Z}_Q$ where $Q = \prod_{i=1}^{k} Q_i$. On the other hand, our scheme can support SIMD operations when all Q_i's are the same.

3.1 Parameters

We give some descriptions about the parameters.

- λ : the security parameter
- ρ : the bit length of the error
- η : the bit length of the secret primes
- γ : the bit length of a ciphertext
- τ : the number of encryptions of zero in public key
- k : the number of distinct secret primes
- ℓ_Q : the bit length of Q_i for $i = 1, \ldots k$

Roughly speaking, k determines the size of the message space. The parameter ℓ_Q can be an integer from 2 to $\eta/8$ depending on the multiplicative depth of the scheme. The details of analysis are given in Section 3.3. The concrete parameters of our scheme are as follows:

- $\gamma = \eta^2\omega(\log\lambda)$, to resist Cohn and Heninger's attack [9] and the attack using Lagarias algorithm [20] on the approximate GCD problem (in Appendix C.3).
- $\eta = \tilde{\Omega}(\lambda^2 + \rho \cdot \lambda)$, to resist the factoring attack using the elliptic curve method [21] (in Section C.1) and to permit enough multiplicative depth (in Section 3.3).
- $\rho = \tilde{\mathcal{O}}(\lambda)$, to be secure against Chen-Nguyen's attack [6] and Howgrave-Graham's attack [19] (in Appendix C.2).
- $\tau = \gamma + \omega(\log\lambda)$, in order to use left-over hash lemma in the security proof which is given in Section 4.1.

We choose $\gamma = \tilde{\mathcal{O}}(\lambda^5), \eta = \tilde{\mathcal{O}}(\lambda^2), \rho = 2\lambda, \tau = \gamma + \lambda$ which is similar to the DGHV's convenient parameter setting [13].

3.2 The Construction

KeyGen$(\lambda, \rho, \eta, \gamma, \tau, \ell_Q, k)$: Choose η-bit distinct primes $p_1, \ldots, p_k$ and $q_0 \leftarrow \mathbb{Z} \cap [0, \frac{2^\gamma}{\prod_{i=1}^k p_i})$, and set $x_0 := q_0 \prod_{i=1}^k p_i$. Choose ℓ_Q-bit integers $Q_1, \ldots, Q_k$ with $\gcd(Q_i, x_0) = 1$ for $i = 1, \ldots, k$. Output the public key pk as follows:

$$pk = \Big(x_0 \,, \{Q_i\}_{i=1}^k, X := \big\{x_j = \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)}(e_{j0}, e_{j1}Q_1, \ldots, e_{jk}Q_k)\big\}_{j=1}^\tau,$$
$$Y := \big\{y_\ell = \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)}(e'_{\ell 0}, e'_{\ell 1}Q_1 + \delta_{\ell 1}, \ldots, e'_{\ell k}Q_k + \delta_{\ell k})\big\}_{\ell=1}^k \Big),$$

where $e_{j0}, e'_{\ell 0} \leftarrow \mathbb{Z} \cap [0, q_0), e_{ji} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho), e'_{\ell i} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$ for $i, \ell \in [1, k], j \in [1, \tau]$ and δ_{ij} is Kronecker delta. Output the secret key $sk = (p_1, \ldots, p_k)$.

Enc$(pk, \boldsymbol{m})$: For any $\boldsymbol{m} = (m_1, \ldots, m_k)$ with $m_i \in \mathbb{Z}_{Q_i}$, outputs $c = \sum_{i=1}^k m_i y_i + \sum_{j \in S} x_j \bmod x_0$ where S is a random subset of $\{1, \ldots \tau\}$.

Dec(sk, c): Output $(m_1, \ldots, m_k) = ((c \bmod p_1) \bmod Q_1, \ldots, (c \bmod p_k) \bmod Q_k)$.

Eval$(pk, \mathcal{C}, \boldsymbol{c} = (c_1, \ldots, c_t))$: Take as input public key pk, permitted circuit $\mathcal{C}$ with t inputs defined in Section 3.3 and a t-tuple of ciphertexts $\boldsymbol{c}$. Output $\mathcal{C}(c_1, \ldots, c_t)$ using **Add** and **Mul** given below.

Add(pk, c_1, c_2): Output $c_1 + c_2 \bmod x_0$.

Mul(pk, c_1, c_2): Output $c_1 \times c_2 \bmod x_0$.

Remark 2. $X = \{x_j\}_{j=1}^\tau$ is a set of encryptions of the zero vector, and y_ℓ is a encryption of the ℓ-th elementary vector E_ℓ in pk.

Remark 3. There are $(\tau + k)$ integers of γ-bit and k integers of ℓ_Q-bit in the public key. The public key size is $\tilde{\mathcal{O}}((\tau + k)\gamma + k\ell_Q) = \tilde{\mathcal{O}}(\lambda^{10})$ under the parameter setting in Section 3.1.

Remark 4. If $k = 1, Q_1 = 2$, then our scheme is the same as a noise-free variant of the DGHV [13].

A ciphertext $c \leftarrow \mathbf{Enc}(pk, \boldsymbol{m})$ can be written in the form,

$$\begin{aligned} c &= \sum_{\ell=1}^{k} m_\ell y_\ell + \sum_{j \in S} x_j \bmod x_0 \\ &= \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)} \left((\sum_{\ell=1}^{k} e'_{\ell 0} m_\ell), (\sum_{\ell=1}^{k} e'_{\ell 1} m_\ell) Q_1 + m_1, \ldots, (\sum_{\ell=1}^{k} e'_{\ell k} m_\ell) Q_k + m_k \right) \\ &\quad + \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)} \left((\sum_{j \in S} e_{j0}), (\sum_{j \in S} e_{j1}) Q_1, \ldots, (\sum_{j \in S} e_{jk}) Q_k \right) \\ &= \mathrm{CRT}_{(q_0,p_1,\ldots,p_k)} (e_0, e_1 Q_1 + m_1, \ldots, e_k Q_k + m_k) \end{aligned}$$

for some $e_0 \in \mathbb{Z} \cap [0, q_0), e_1, \ldots, e_k \in \mathbb{Z} \cap (-2^{\rho'}, 2^{\rho'})$, where $\rho' = \max\{\rho + \log k + \ell_Q, 2\rho + \log \tau\}$.

3.3 Correctness

We use the *integer circuits* with **Add** and **Mul** gates applied to integers rather than a bit. That is, boolean gates are replaced with integer operations. Now we show that the scheme is correct for permitted circuit. At first, we define a *permitted circuit* similar to Gentry [15].

Definition 1 (Permitted Circuit). *Let C be an integer circuit with t inputs. C is a* permitted circuit *if an output of C has absolute value at most $2^{\alpha(\eta-4)}$ whenever the absolute value of each t input is smaller than $2^{\alpha(\rho'+\ell_Q)}$ for any $\alpha \geq 1$.*

We denote $\mathcal{C}_{\mathcal{E}}$ as the set of permitted circuits. Now we will show that our scheme is correct for $\mathcal{C}_{\mathcal{E}}$, that is

$$\mathbf{Dec}(sk, C(c_1, \ldots, c_t)) = C(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_t)$$

where $C \in \mathcal{C}_{\mathcal{E}}$, $c_j \leftarrow \mathbf{Enc}(pk, \boldsymbol{m}_j)$ and $\boldsymbol{m}_j = (m_{j1}, \ldots, m_{jk})$ for $j = 1, \ldots, t$.

Theorem 1 (Correctness). *The scheme given in section 3.2 is correct for $\mathcal{C}_{\mathcal{E}}$.*

Proof. It is enough to show Lemma 6 and 7 that are in Appendix A for the proof of Theorem 1. □

Each noise of $c_1 + c_2$ is increased at most 1-bit. But the bit length of each noise for $c_1 \times c_2$ becomes about $2\rho' + 2\ell_Q$ which is two times larger than that of the original ciphertext. As you see, the noise expansion through multiplication is more significant than addition, so we focus on the multiplicative depth of permitted circuit.

Lemma 1. *Let C be an integer circuit and f be the multivariate polynomial computed by C. If $|\overrightarrow{f}| \cdot (2^{\rho'+\ell_Q})^d < 2^{\eta-4}$, then $C \in \mathcal{C}_{\mathcal{E}}$ where $|\overrightarrow{f}|$ is the ℓ_1 norm of the coefficient vector of f and $d = \deg f$.*

Proof. The proof is straightforward. □

From the above condition, we have

$$d < \frac{\eta - 4 - \log_2 |\overrightarrow{f}|}{\rho' + \ell_Q}$$

which is similar to the DGHV [13]. Since we want to support polynomial of degree λ, we choose $\eta \geq \rho' \cdot \Theta(\lambda)$ if we assume $\log_2 |\overrightarrow{f}|$ is relatively small to η, ρ'.

4 Security

In this section, we prove the security of our scheme. The security of the DGHV scheme is based on the ACD assumption defined in Section 2. On the contrary, the security of our scheme is based on the modified DACD (Decisional Approximate GCD) assumption which says that, for given a distribution $\mathcal{D} = \mathcal{D}_\rho(p; q_0)$ and some integer z, it is hard to determine whether z is chosen from $\mathcal{D}$ or not. In principle, DACD might be easier than ACD, but so far the only way to solve DACD is to first solve ACD. Therefore, we select the parameters of our scheme based on the attacks on the ACD problem [19, 9, 6]. The details are given in Appendix C.

4.1 Reduction to the Approximate GCD problem

To prove the semantic security of our scheme, we introduce another decisional version of approximate GCD problem.

Definition 2 (Decisional Approximate GCD Problem: DACD). *The (ρ, η, γ)-decisional approximate GCD problem is: for an η-bit prime p, given a γ-bit integer $x_0 = pq_0$ and polynomially many samples from $\mathcal{D}_\rho(p; q_0)$, determine $b \in \{0, 1\}$ from $z = x + r \cdot b \bmod x_0$ where $x \leftarrow \mathcal{D}_\rho(p; q_0)$ and $r \leftarrow \mathbb{Z} \cap [0, x_0)$.*

We assume that DACD problem is hard for any polynomial time distinguisher. In the following, we introduce new problems that have a role bridging the gap between DACD problem and our scheme. Overall, our scheme is semantically secure based on the DACD assumption.

Definition 3 (Decisional Approximate GCD_Q Problem: DACD_Q).
The $(\rho, \eta, \gamma, l_Q)$-decisional approximate GCD_Q problem is: for an η-bit prime p and a ℓ_Q-bit integer Q, given a γ-bit integer $x_0 = pq_0$ with $\gcd(x_0, Q) = 1$, and polynomially many samples from $\mathcal{D}_\rho(p; Q; q_0)$, determine $b \in \{0, 1\}$ from $z = x + r \cdot b \bmod x_0$ where $x \leftarrow \mathcal{D}_\rho(p; Q; q_0)$ and $r \leftarrow \mathbb{Z} \cap [0, x_0)$.

Definition 4 (k-Decisional Approximate GCD_Q Problem: k-DACD_Q).
The $(\rho, \eta, \gamma, l_Q)$-$k$-decisional approximate GCD_Q problem is : for η-bit distinct primes $p_1, \ldots, p_k$ and ℓ_Q-bit integers $Q_1, \ldots, Q_k$, given a γ-bit integer $x_0 := q_0 p_1 \cdots p_k$, with $\gcd(x_0, Q_i) = 1$ for $i = 1, \ldots, k$, and polynomially many samples from $\mathcal{D} := \mathcal{D}_\rho\big((p_i)^k; (Q_i)^k; q_0\big)$ and a set $Y := \{y_\ell = \mathrm{CRT}_{(q_0, p_1, \ldots, p_k)}(e_{\ell 0}, e_{\ell 1} Q_1 + \delta_{\ell 1}, \ldots, e_{\ell k} Q_k + \delta_{\ell k}) \big| e_{\ell 0} \leftarrow \mathbb{Z}_{q_0}, e_{\ell i} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$ for $\ell, i \in \{1, \ldots, k\}\}$, determine $b \in \{0, 1\}$ from $z = x + r \cdot b \bmod x_0$ where $x \leftarrow \mathcal{D}$ and $r \leftarrow \mathbb{Z} \cap [0, x_0)$.

We say that the DACD assumption holds if no polynomial time distinguisher can solve the DACD problem with non-negligible advantage. The k-DACD_Q assumption is defined similarly. Now we show that our somewhat homomorphic encryption scheme is semantically secure under the DACD assumption. This is done by three steps. In the following, arrows indicate polynomial time reductions.

Step 1: (ρ, η, γ)-DACD $\longrightarrow$ $(\rho, \eta, \gamma, l_Q)$-$\mathrm{DACD}_Q$ (Lemma 2)

Step 2: $(\rho, \eta, \gamma, l_Q)$-$\mathrm{DACD}_Q$ $\longrightarrow$ $(\rho, \eta, \gamma + (k-1)\eta, l_Q)$-$k$-$\mathrm{DACD}_Q$ (Lemma 3)

Step 3: $(\rho, \eta, \gamma + (k-1)\eta, l_Q)$-$k$-$\mathrm{DACD}_Q$ $\longrightarrow$ our scheme (Theorem 2)

The first step is rather easily done by multiplying Q thanks to the knowledge of the exact multiple of p. In the second step, DACD_Q problem with $x_0 = q_0 p_1$ is converted to k-DACD_Q problem by choosing additional $k-1$ primes $p_2, \ldots, p_k$ and computing necessary terms including $x'_0 = q_0 \prod_{i=1}^{k} p_i$. In the proof, we use a hybrid argument and lose a factor of k in the success probability. Finally, the last step is done by interpreting the input of k-DACD_Q problem as a public key of the scheme.

Lemma 2. *The (ρ, η, γ)-DACD problem is reducible to the $(\rho, \eta, \gamma, l_Q)$-$\mathrm{DACD}_Q$ problem.*

Proof. Suppose a polynomial time distinguisher $\mathcal{B}$ solves the $(\rho, \eta, \gamma, l_Q)$-$\mathrm{DACD}_Q$ problem with an advantage ϵ. We construct a polynomial time distinguisher $\mathcal{A}$ that solves the (ρ, η, γ)-DACD problem with the same advantage. Suppose $\mathcal{A}$ is given γ-bit integer $x_0 = pq_0, z = x + r \cdot b$, and polynomially many samples $X = \{x_i \mid x_i \leftarrow \mathcal{D}_\rho(p; q_0) \text{ for } i = 1, \ldots, \tau\}$. $\mathcal{A}$ works as follows:

1. Choose a ℓ_Q-bit integer Q such that $\gcd(x_0, Q) = 1$.
2. Construct samples $X' := \{x \cdot Q \bmod x_0 \mid x \in X\}$ and $z' := z \cdot Q \bmod x_0$.
3. Give (x_0, Q, X', z') to $\mathcal{B}$.
4. Output b' where b' is $\mathcal{B}$'s answer.

We verify that the statistical distance of $\mathcal{D}' = \{x \leftarrow \mathcal{D}_\rho(p; q_0) : \text{Output } y = x \cdot Q \bmod x_0\}$ and $\mathcal{D}_\rho(p; Q; q_0)$ is negligible when $\gcd(x_0, Q) = 1$. Consider a map $\phi_Q : \mathbb{Z}_{q_0} \to \mathbb{Z}_{q_0}$ defined by $x \mapsto x \cdot Q$. Since $\gcd(x_0, Q) = 1$, ϕ_Q is a ring isomorphism and so $\Delta(\mathcal{D}', \mathcal{D}_\rho(p; Q; q_0)) = 0$. It is easy to see that z' is uniform in $\mathbb{Z} \cap [0, x_0)$ when z is randomly chosen in $\mathbb{Z} \cap [0, x_0)$. Hence in this case, $\Pr[\mathcal{A}(\mathcal{D}_\rho(p; q_0), z) = 1] = \Pr[\mathcal{B}(\mathcal{D}', z') = 1]$. On the other hand, if z is randomly chosen in $\mathcal{D}_\rho(p; q_0)$, then z' is uniform in $\mathcal{D}'$ and so $\Pr[\mathcal{A}(\mathcal{D}_\rho(p; q_0), z) = 1] = \Pr[\mathcal{B}(\mathcal{D}', z') = 1]$. Thus

$$\mathbf{Adv}(\mathcal{A}) = |\Pr[\mathcal{A}(\mathcal{D}_\rho(p; q_0), z_1) = 1] - \Pr[\mathcal{A}(\mathcal{D}_\rho(p; q_0), z_2) = 1]| = \epsilon$$

by the definition of algorithm $\mathcal{B}$ and the fact $\Delta(\mathcal{D}', \mathcal{D}_\rho(p; Q; q_0)) = 0$ where $z_1 \leftarrow \mathcal{D}_\rho(p; q_0)$ and $z_2 \leftarrow \mathbb{Z} \cap [0, x_0)$. □

Lemma 3. *Let $p_1, \ldots, p_k$ be distinct η-bit primes, $x_0 = q_0 p_1$ be γ_1-bit integer, and $x'_0 = q_0 \prod_{i=1}^{k} p_i$ be γ-bit integer. Then, the $(\rho, \eta, \gamma_1, l_Q)$-DACD$_Q$ problem is reducible to the $(\rho, \eta, \gamma, l_Q)$-$k$-DACD$_Q$ problem with the advantage of the latter k times that of the former on average.*

Proof. Suppose a polynomial time distinguisher $\mathcal{B}$ solves the $(\rho, \eta, \gamma, l_Q)$-$k$-$\mathrm{DACD}_Q$ problem. We construct a polynomial time distinguisher $\mathcal{A}$ that solves the $(\rho, \eta, \gamma_1, l_Q)$-$\mathrm{DACD}_Q$ problem. We are assuming $\mathcal{B}$'s advantage depends only on the parameter, not the specific value such as p_i.
For $x_0 = q_0 p_1$ and $x'_0 = q_0 \prod_{i=1}^{k} p_i$, we define $\mathcal{D}_i := \mathcal{D}_\rho(p_1, \ldots, p_i; Q_1, \ldots, Q_i; q_0 \prod_{j=i+1}^{k} p_j)$ and $\mathcal{D}_0 := \mathbb{Z} \cap [0, x'_0)$. Note that the support[1] of $\mathcal{D}_i$ is included in the support of D_{i-1} for $i \in \{1, \ldots, k\}$. Suppose $\mathcal{B}$ can distinguish z between $\mathcal{D}_0$ and $\mathcal{D}_k$ with advantage ϵ. Then by the standard hybrid argument, $\mathcal{B}$ should distinguish z between $\mathcal{D}_i$ and $\mathcal{D}_{i-1}$ for some $i \in \{1, \ldots, k\}$ with advantage at least ϵ/k. Let us denote this index as i_0. Since $\mathcal{B}$'s advantage only depends on the parameters, this means $\mathcal{B}$ can distinguish between $\mathcal{D}_{i_0}$ and $\mathcal{D}_{i_0-1}$ for any η-bit primes $p_1, \ldots, p_k$.
For the time being, let us assume that i_0 is known. Let the input of the distinguisher $\mathcal{A}$ be

[1] The support of a distribution is a set of elements having non-zero probability in the distribution.

an integer $x_0 = q_0 p_1, Q_1$, polynomially many samples x_i from $\mathcal{D}$ and $z = x + r \cdot b$ where $\mathcal{D} := \mathcal{D}_\rho(p_1; Q_1; q_0), x \leftarrow \mathcal{D}, r \leftarrow \mathbb{Z} \cap [0, x_0)$ and $b \in \{0, 1\}$. We define a set $\mathcal{I}_1 := (x_0, Q_1, \{x_i\}_{i=1}^\tau)$. Using input $(\mathcal{I}_1, z)$, $\mathcal{A}$ constructs an input $(\mathcal{I}_2, z')$ which will be given to the distinguisher $\mathcal{B}$ as follows:

1. Choose ℓ_Q-bit integers $Q_2, \ldots, Q_k$, η-bit distinct primes $p_2, \ldots, p_k$ such that $\gcd(Q_i, x_0) = \gcd(p_i, x_0) = 1$ for $i \in \{2, \ldots, k\}$.
2. Let $x'_0 = x_0 \cdot \prod_{i=2}^k p_i = q_0 \prod_{i=1}^k p_i$.
3. For each sample x_i from $\mathcal{D}$, choose $e_{ij} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$ for $j \in \{2, \ldots, k\}$, and construct a sample from the distribution $\mathcal{D}' := \mathcal{D}_\rho(p_1, \ldots, p_k; Q_1, \ldots, Q_k; q_0)$ by $x'_i = \mathrm{CRT}_{(x_0, p_2, \ldots, p_k)}(x_i, e_{i2}Q_2, \ldots, e_{ik}Q_k)$.
4. To make a set Y, choose $e'_{\ell j} \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$, $s_\ell \leftarrow \mathcal{D}$ for $\ell \in [1, k], j \in [2, k]$ and construct $y'_\ell = \mathrm{CRT}_{(x_0, p_2, \ldots, p_k)}(s_\ell + \delta_{\ell 1}, e'_{\ell 2}Q_2 + \delta_{\ell 2}, \ldots, e'_{\ell k}Q_k + \delta_{\ell k})$.
5. For $z = x + r \cdot b$, let $z' = \mathrm{CRT}_{(x_0, p_2, \ldots, p_k)}(z, e'_2 Q_2, \ldots, e'_{i_0} Q_{i_0}, e'_{i_0+1}, \ldots, e'_k)$ where $e'_i \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$ for $i \in \{2, \ldots, i_0\}$ and $e'_i \leftarrow \mathbb{Z} \cap [0, p_i)$ for $i \in \{i_0 + 1, \ldots, k\}$.

In Step 3 and 4, since

$$\begin{aligned}
x'_i &= \mathrm{CRT}_{(x_0, p_2, \ldots, p_k)}(x_i, e_{i2}Q_2, \ldots, e_{ik}Q_k) \\
&= \mathrm{CRT}_{(q_0, p_1, p_2, \ldots, p_k)}(e_{i0}, e_{i1}Q_1, e_{i2}Q_2, \ldots, e_{ik}Q_k) \\
y'_\ell &= \mathrm{CRT}_{(x_0, p_2, \ldots, p_k)}(s_\ell + \delta_{\ell 1}, e'_{\ell 2}Q_2 + \delta_{\ell 2}, \ldots, e'_{\ell k}Q_k + \delta_{\ell k}) \\
&= \mathrm{CRT}_{(q_0, p_1, p_2, \ldots, p_k)}(e'_{\ell 0}, e'_{\ell 1}Q_1 + \delta_{\ell 1}, e'_{\ell 2}Q_2 + \delta_{\ell 2}, \ldots, e'_{\ell k}Q_k + \delta_{\ell k})
\end{aligned}$$

for some $e_{i0}, e'_{\ell 0} \in \mathbb{Z} \cap [0, q_0), e_{i1}, e'_{\ell 1} \in \mathbb{Z} \cap (-2^\rho, 2^\rho)$ for $i \in [1, \tau], \ell \in [1, k]$, the set Y given to $\mathcal{B}$ is suitable. The distinguisher $\mathcal{A}$ gives these input $\mathcal{I}_2 = (x'_0, \{Q_i\}_{i=1}^k, \{x'_i\}_{i=1}^\tau, \{y'_\ell\}_{\ell=1}^k)$ and z' to $\mathcal{B}$, and use $\mathcal{B}$'s answer to its answer. Interchanging p_1 and p_{i_0}, we know that z is sampled from D_{i_0} or $D_{i_0 - 1}$. This can be distinguished by $\mathcal{B}$ with advantage ϵ/k, and thus $\mathcal{A}$'s advantage is at least ϵ/k. Since we do not know i_0, we randomly choose i_0 and get the average advantage of ϵ/k.

□

To complete the proof of the semantic security of our scheme, we need two more lemmas. Lemma 4 shows that the distribution of fake public key is indistinguishable from that of the correct public key. Lemma 5 implies that an encryption from $\mathcal{A}$ is correct form for the scheme.

Lemma 4. *For the parameters $(\lambda, \rho, \eta, \gamma, \tau, l_Q, k)$, let $pk = (x_0, \{Q_i\}_{i=1}^k, \{x_j\}_{j=1}^\tau, \{y_l\}_{l=1}^k)$ and $sk = (p_1, \ldots, p_k)$ be chosen as in the* **KeyGen** *of our scheme. And let us choose x'_j uniformly from $\mathbb{Z}_{x_0} = \mathbb{Z} \cap \left(-\frac{x_0}{2}, \frac{x_0}{2}\right]$ for $j = 1, \ldots, \tau$. Then, pk and pk' are computationally indistinguishable if we define pk' as $(x_0, \{Q_i\}_{i=1}^k, \{x'_j\}_{j=1}^\tau, \{y_l\}_{l=1}^k)$, under the $(\rho, \eta, \gamma, l_Q)$-k-DACD$_Q$ assumption.*

Lemma 5. *For the parameters $(\lambda, \rho, \eta, \gamma, \tau, l_Q, k)$, let $pk = (x_0, \{Q_i\}_{i=1}^k, \{x_j\}_{j=1}^\tau, \{y_l\}_{l=1}^k)$ and $sk = (p_1, \ldots, p_k)$ be chosen as in the* **KeyGen** *of our scheme. Let $\boldsymbol{m} = (m_1, \ldots, m_k)$ where $m_i \in \mathbb{Z}_{Q_i}$. For every $z \in \mathcal{D}_\rho(p_1, \ldots, p_k; Q_1, \ldots, Q_k; q_0)$, the following distribution*

$$\mathcal{C}_{pk,z}(\boldsymbol{m}) = \left\{ S \subset_R \{1, \ldots, \tau\} : \text{Output } c' \leftarrow \sum_{i=1}^k m_i y_i + \sum_{j \in S} x_j + z \bmod x_0 \right\}$$

is computationally close to the distribution $\mathbf{Enc}(pk, \boldsymbol{m})$*, under the* $(\rho, \eta, \gamma, l_Q)$*-k-DACD*$_Q$ *assumption.*

Now we prove the semantic security of our scheme.

Theorem 2. *The cryptosystem given in section 3 is semantically secure, if the* $(\rho, \eta, \gamma, l_Q)$*-k-DACD*$_Q$ *assumption holds.*

Proof. Suppose a polynomial time algorithm $\mathcal{B}$ breaks the semantic security of the scheme with non-negligible advantage. We construct a polynomial time algorithm $\mathcal{A}$ that solves the $(\rho, \eta, \gamma, l_Q)$-$k$-DACD$_Q$ problem with non-negligible advantage. For η-bit distinct primes $p_1, \ldots, p_k$ and ℓ_Q-bit integers $Q_1, \ldots, Q_k$, the input of $\mathcal{A}$ is $(x_0, (Q_i)^k, \mathcal{D}_\rho((p_i)^k; (Q_i)^k; q_0), Y, z)$ where $x_0 = q_0 \prod_{i=1}^k p_i$ is a γ-bit integer. The algorithm $\mathcal{A}$ works as follows:

1. $\mathcal{A}$ gives tuples $(x_0, (Q_i)^k, X := \{x_j \leftarrow \mathcal{D}_\rho((p_i)^k; (Q_i)^k; q_0)\}_{j=1}^\tau$, $Y := \{y_1, \ldots, y_k\})$ to $\mathcal{B}$ as the public key.
2. $\mathcal{B}$ chooses $\{\boldsymbol{m}_0 = (m_{01}, \ldots, m_{0k}), \boldsymbol{m}_1 = (m_{11}, \ldots, m_{1k})\}$ and sends it to $\mathcal{A}$.
3. $\mathcal{A}$ computes $c' = \sum_{\ell=1}^k m_{bi} y_\ell + \sum_{j \in J} x_j + z \bmod x_0$ for randomly chosen $b \in \{0, 1\}$ where $J \subset \{1, \ldots, \tau\}$ is a random subset, and give c' to $\mathcal{B}$.
4. $\mathcal{B}$ outputs $b' \in \{0, 1\}$.
5. If $b = b'$, then $\mathcal{A}$ outputs 0. Otherwise, outputs 1.

We see that the public key given to $\mathcal{B}$ is correctly formed and distributed. It is easy to see that c' is uniform in $\mathbb{Z}_{x_0}$ when z is randomly chosen in $\mathbb{Z}_{x_0}$. Hence in this case, the advantage of $\mathcal{A}$ is zero since c' does not reveal any information of $\boldsymbol{m}_b$ and $\mathcal{B}$'s probability of correct guessing is exactly $1/2$. On the other hand, if z is randomly chosen in $\mathcal{D}_\rho((p_i)^k; (Q_i)^k; q_0)$, then c' is computationally indistinguishable from the correct encryption of $\boldsymbol{m}_b$ by Lemma 5 when we choose τ larger than $\gamma + \omega(\log \lambda)$. Thus, in this case, the probability of correct answer for $\mathcal{B}$ is at most negligibly different from that of $\mathcal{B}$. This shows that the advantage of $\mathcal{A}$ is non-negligible, violating the $(\rho, \eta, \gamma, l_Q)$-$k$-DACD$_Q$ assumption. Therefore, the cryptosystem given is section 3 is semantically secure. □

5 Applications

In this section, we compare our somewhat homomorphic scheme with other homomorphic schemes when used for integer arithmetic and vector arithmetic (SIMD operations).

5.1 Secure Large Integer Arithmetic

Secure integer arithmetic is one of the most important applications of homomorphic encryption schemes. It includes frequently used statistical functions such as mean, standard deviation, logistical regression, and secure evaluation of a multivariate function over the integers. Some applications may require very large integer inputs in the computation of these functions. For the homomorphic computation of these functions, one may use FHEs. However, the large ciphertext expansion and rather high cost of bootstrapping make this cumbersome and inefficient. Also SWHE on small message space can not be efficient since it requires a large depth for integer operations. In fact, even an addition of two λ-bit integers using bit operations needs computing degree-$O(\lambda)$ polynomial over $\mathbb{Z}_2$ due to carry computation. For

Message space (bit length)	BV [3]	Ours	
		$\gamma = \tilde{O}(\lambda^4)$	$\gamma = \tilde{O}(\lambda^5)$
λ	λ^3	λ^3	λ^4
λ^2	λ^3	λ^2	λ^3
λ^3	λ^3	λ	λ^2
λ^4	λ^3	-	λ

Fig. 1. The comparison of overheads when allowing $\log \lambda$ multiplicative depth ($\tilde{\mathcal{O}}$ is omitted.)

this reason, it is very important to construct an efficient somewhat homomorphic scheme supporting large integer arithmetic on encrypted data. In this subsection, we compare our SWHE with other schemes.

There are several homomorphic encryptions based on various hard problems. Gentry's scheme and its optimized variants are still less efficient. FHEs based on the ACD are conceptually simpler, but support only bit operations. More recent schemes based on (R)LWE appear to be most efficient in various parameters. We compare our scheme with the BV scheme in this category [3]. Large integer arithmetic with this scheme can be found in [22].

In the BV scheme [3], $q > t > (\ell+1)^\lambda (2^{2\lambda})^\lambda$ and n must be larger than $\lambda\ell$ to permit depth-λ on ℓ-bit integers where the ciphertext space is $\mathbb{Z}_q[x]/(x^n+1)$ and the message space is $\mathbb{Z}_t[x]/(x^n+1)$. These constraints can be a disadvantage to encrypt large integers, since the overhead is about $\tilde{O}(n \log q/\ell) \approx \tilde{O}(\lambda^3)$. To make encryption scheme supporting degree λ on polynomial on large integers in our scheme, it is sufficient to choose $O(\lambda)$-bit pairwise coprime integers $Q_1, \ldots, Q_k$. The advantage of our scheme in the overhead stands out, as the plaintext space gets larger.

We give a comparison between [3] and ours with respect to the bit length of the message space in Fig. 1. If one accepts the lattice rule of thumb conjecture [27], one may need $\gamma = \tilde{O}(\lambda^5)$ in our scheme. However, considering estimation of lattice reduction time in known attacks, practical parameter for γ in [12] approximately matches to $\tilde{O}(\lambda^4)$ and this seems to be sufficient for the security.

5.2 Secure Supporting SIMD Operations

For the parallel computations on encrypted data, homomorphic encryptions supporting SIMD operations can be useful. Especially, when it supports integer arithmetic, it can be used to securely evaluate a multivariate function simultaneously on many input vectors. There are several homomorphic encryptions supporting SIMD operations including [25, 17] that use CRT in certain polynomial ring $\mathbb{A}_p := \mathbb{Z}_p[x]/(F(x))$ where $F(x)$ is irreducible over $\mathbb{Z}$ but factors into smaller degree polynomials over $\mathbb{Z}_p$. In contrast, we use CRT in $\mathbb{Z}_n$ for a certain composite n.

Smart and Vercauteren's FHE in [25] supports SIMD operations on $\mathbb{F}_{2^d}$. To evaluate depth-$\log(\lambda)$ arithmetic circuit of average width $\Omega(\lambda)$ over $\mathbb{F}_2$, the overhead is $\tilde{\mathcal{O}}(N\sqrt{N}/\lambda) \approx \tilde{\mathcal{O}}(\lambda^5)$ where $\lambda = 2^7$, since one must choose $N \approx 2^{27}$.

Gentry et al. [17] show that any t-gate, depth-L arithmetic circuit of average width $\Omega(\lambda)$ over $\mathbb{F}_p$ can be evaluated homomorphically in time $t \cdot \tilde{\mathcal{O}}(L) \cdot \mathit{polylog}(\lambda)$ (Theorem 3 in [17]) with $p = \mathit{poly}(\lambda)$. Thus, it cannot support large message space such as $\mathbb{Z}_Q$ where $Q \approx 2^\lambda$.

In our scheme, the overhead is $\tilde{O}(\lambda^2)$ with SIMD operations on k copies of $\mathbb{Z}_2$ for $k = O(\lambda^3)$. If we choose $Q_1 = \cdots = Q_k = Q \approx 2^\lambda$, the overhead is reduced to $\tilde{O}(\lambda)$ with SIMD operations on k copies of $\mathbb{Z}_Q$. Thus, our scheme gets more attractive in dealing with SIMD operations with larger integers.

6 Fully Homomorphic Encryption

Our homomorphic encryption can be converted to a fully homomorphic encryption via Gentry's squashing technique as is done in DGHV [13] when all Q_i's are equal to two, based on the sparse subset sum assumption. Note that message space is $\mathbb{Z}_2^k$. Recall that the decryption of message component m_i is done by $m_i \leftarrow (c \bmod p_i) \bmod Q_i = (c \bmod p_i) \bmod 2$. Our sparse subset consists of y_j's such that

$$\frac{1}{p_i} \approx \sum_{j=1}^{\Theta} s_{ij} \cdot y_j,$$

and the secret key s_{ij}'s are included in the public key as an encrypted form $\mathbf{Enc}(pk, \boldsymbol{s}_j)$ where $\boldsymbol{s}_j = (s_{1j}, s_{2j}, \ldots, s_{kj})$ for $j = 1, \ldots, \Theta$. Using the same subset $\{y_j\}_{j=1}^{\Theta}$ for every secret prime p_i, parallel computation is possible. By this way, we can lower the multiplicative depth of the decryption to be computed homomorphically.

The remained question is whether this can be done when Q_i is larger than two. Since computing Q_i-ary addition seems to need more complex carry computations than a binary addition, the proof is not straightforward.

7 Further Works

In this paper, we proposed a secure ring homomorphic encryption scheme by modifying a privacy homomorphism suggested by Rivest, Adleman and Dertouzos [23]. Our scheme can be extended to a fully homormorphic encryption scheme under the decisional Approximate GCD assumption and the sparse subset sum assumption. One strong point of our scheme is the small overhead when using as a somewhat homomorphic encryption scheme of bounded depth over the integers. Especially, when we are to securely evaluate a multivariate function of degree $O(\lambda)$ with input size $\Omega(\lambda^2)$, our scheme has the smallest overhead.

Our scheme is analogous to the DGHV schemes [13] and the public key size is similar to that of the DGHV. It would be interesting how to apply the public key compression technique of [12] to our homomorphic scheme. Also, it is an open problem to design an efficient squashing step for our scheme with $Q_i > 2$, as well as the DGHV.

Acknowledgments We would like to thank Taekyoung Kwon and Hyung Tae Lee for valuable comments. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2012-0001243).

References

1. Memoirs of the 6th Cryptology Paper Contest, arranged by a Korean government organization (2012)
2. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer Berlin / Heidelberg, 2012.

3. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer Berlin / Heidelberg, 2011.
4. E. Brickell and Y. Yacobi. On privacy homomorphisms (extended abstract). In D. Chaum and W. Price, editors, *Advances in Cryptology EUROCRYPT 87*, volume 304 of *Lecture Notes in Computer Science*, pages 117–125. Springer Berlin Heidelberg, 1988.
5. J. Buhler, J. Lenstra, H.W., and C. Pomerance. Factoring integers with the number field sieve. In A. Lenstra and J. Lenstra, HendrikW., editors, *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 50–94. Springer Berlin Heidelberg, 1993.
6. Y. Chen and P. Nguyen. Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 502–519. Springer Berlin Heidelberg, 2012.
7. J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. *To Appear at Eurocrypt 2013.*
8. J. H. Cheon, W.-H. Kim, and H. S. Nam. Known-plaintext cryptanalysis of the domingo-ferrer algebraic privacy homomorphism scheme. *Inf. Process. Lett.*, 97(3):118–123, 2006.
9. H. Cohn and N. Heninger. Approximate common divisors via lattices. *IACR Cryptology ePrint Archive*, 2011:437, 2011.
10. J.-S. Coron, T. Lepoint, and M. Tibouchi. Batch fully homomorphic encryption over the integers. Cryptology ePrint Archive, Report 2013/036, 2013.
11. J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In P. Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer Berlin / Heidelberg, 2011.
12. J.-S. Coron, D. Naccache, and M. Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer Berlin / Heidelberg, 2012.
13. M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In H. Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer Berlin / Heidelberg, 2010.
14. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.
15. C. Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 116–137. Springer Berlin / Heidelberg, 2010.
16. C. Gentry and S. Halevi. Implementing gentry's fully-homomorphic encryption scheme. In K. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer Berlin / Heidelberg, 2011.
17. C. Gentry, S. Halevi, and N. Smart. Fully homomorphic encryption with polylog overhead. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer Berlin / Heidelberg, 2012.
18. C. Gentry, S. Halevi, and N. Smart. Homomorphic evaluation of the aes circuit. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer Berlin / Heidelberg, 2012.
19. N. Howgrave-Graham. Approximate integer common divisors. In *CaLC*, pages 51–66, 2001.
20. J. C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM J. Comput.*, 14(1):196–209, 1985.
21. J. Lenstra, H. W. Factoring integers with elliptic curves. *The Annals of Mathematics*, 126(3):pp. 649–673, 1987.
22. M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *CCSW*, pages 113–124, 2011.
23. R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphism. *Foundations of Secure Computation*, pages 168–177, 1978.
24. P. Scholl and N. Smart. Improved key generation for gentry's fully homomorphic encryption scheme. In L. Chen, editor, *Cryptography and Coding*, volume 7089 of *Lecture Notes in Computer Science*, pages 10–22. Springer Berlin / Heidelberg, 2011.

25. N. Smart and F. Vercauteren. Fully homomorphic simd operations. *To appear in Designs, Codes and Cryptography.*
26. N. Smart and F. Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Nguyen and D. Pointcheval, editors, *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer Berlin / Heidelberg, 2010.
27. D. Stehlé and R. Steinfeld. Faster fully homomorphic encryption. In M. Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 377–394. Springer Berlin / Heidelberg, 2010.
28. D. Wagner. Cryptanalysis of an algebraic privacy homomorphism. In *ISC*, pages 234–239, 2003.

A Correctness

Lemma 6. *If $c \leftarrow \boldsymbol{Enc}(pk, \boldsymbol{m})$ for $\boldsymbol{m} \in \prod_{i=1}^{k} \mathbb{Z}_{Q_i}$, then $c = p_i a_i + b_i Q_i + m_i$ for some a_i, b_i with $|b_i Q_i + m_i| < 2^{(\rho' + \ell_Q)}$ for all $i = 1, \ldots, k$.*

Proof. The proof is straightforward. If $c \leftarrow \mathbf{Enc}(pk, \boldsymbol{m})$, then

$$c = \mathrm{CRT}_{(q_0, p_1, \ldots, p_k)}(e_0, e_1 Q_1 + m_1, \ldots, e_k Q_k + m_k) = p_i a_i + e_i Q_i + m_i$$

for some a_i and $|e_i Q_i + m_i| < 2^{\rho' + \ell_Q}$ for all $i = 1, \ldots, k$. □

Lemma 7. *Let $C \in \mathcal{C}_{\mathcal{E}}$ and $c_j \leftarrow \boldsymbol{Enc}(pk, \boldsymbol{m}_j)$, where $\boldsymbol{m}_j = (m_{j1}, \ldots, m_{jk})$ for $j = 1, \ldots, t$. Let $m'_i \leftarrow C(m_{1i}, \ldots, m_{ti})$ and $c \leftarrow \boldsymbol{Eval}(pk, C, c_1, \ldots, c_t)$. Then $c = p_i a_i + b_i Q_i + m'_i$ for some a_i, b_i with $|b_i Q_i + m'_i| < p_i/8$ for all $i = 1, \ldots, k$.*

Proof. Let f be the multivariate polynomial computed by C. Then

$$\begin{aligned} c \bmod p_i &= f\big(c_1, \ldots, c_t\big) \bmod p_i \\ &= f\big(c_1 \bmod p_i, \ldots, c_t \bmod p_i\big) \bmod p_i. \end{aligned}$$

Since $C \in \mathcal{C}_{\mathcal{E}}$ and $|c_j \bmod p_i| < 2^{\rho' + \ell_Q}$ for all $j = 1, \ldots, t$ by Lemma 6,

$$\Big| f\big(c_1 \bmod p_i, \ldots, c_t \bmod p_i\big) \Big| < 2^{\eta - 4} < p_i/8$$

for all $i = 1, \ldots, k$. Thus $c \bmod p_i = f\big(c_1 \bmod p_i, \ldots, c_t \bmod p_i\big)$. Also,

$$\begin{aligned} (c \bmod p_i) \bmod Q_i &= f\big(c_1 \bmod p_i, \ldots, c_t \bmod p_i\big) \bmod Q_i \\ &= f\Big(\big(c_1 \bmod p_i\big) \bmod Q_i, \ldots, \big(c_t \bmod p_i\big) \bmod Q_i\Big) \bmod Q_i \\ &= f(m_{1i}, \ldots, m_{ti}) \bmod Q_i \\ &= m'_i \bmod Q_i \end{aligned}$$

for all $i = 1, \ldots, k$. □

B Proofs

Proof of Lemma 4. Suppose a polynomial time distinguisher $\mathcal{B}$ may distinguish pk from pk' with advantage ϵ. Using $\mathcal{B}$, we construct a polynomial time distinguisher $\mathcal{A}$ that solves the $(\rho, \eta, \gamma, l_Q)$-$k$-DACD$_Q$ problem. Note that the distinguisher $\mathcal{A}$ has access to the oracle

$\mathcal{D} = \mathcal{D}_\rho((p_i)^k; (Q_i)^k; q_0)$. For $r = 0, \ldots, \tau$, define $pk^{(r)} = (x_0, \{Q_i\}_{i=1}^k, \{x_j^{(r)}\}_{j=1}^\tau, \{y_l\}_{l=1}^k)$ by setting

$$\begin{aligned} x_1^{(r)} &\leftarrow \mathbb{Z}_{x_0}, \\ &\vdots \\ x_r^{(r)} &\leftarrow \mathbb{Z}_{x_0}, \\ x_{r+1}^{(r)} &\leftarrow \mathcal{D}, \\ &\vdots \\ x_\tau^{(r)} &\leftarrow \mathcal{D}. \end{aligned}$$

We see that $pk^{(0)}$ has the same distribution as pk, and $pk^{(\tau)}$ has the same distribution as pk'. For $r = 1, \ldots, \tau$, we define

$$pr_r := \Pr[\mathcal{B}(pk^{(r-1)}) = 1] - \Pr[\mathcal{B}(pk^{(r)}) = 1].$$

(Note that in the above formula we omitted other information $\mathcal{B}$ has: $\lambda, \rho, \eta, \gamma, \tau, l_Q, k$.)

Now we are ready to fully define the algorithm $\mathcal{A}$. It has given a number z, which either is from $\mathcal{D}$, or is uniformly random on $\mathbb{Z}_{x_0}$.

The algorithm $\mathcal{A}$ first picks r randomly from $\{1, \ldots, \tau\}$, and selects x_j^* $(j = 1, \ldots, \tau)$ as follows

$$\begin{aligned} x_1^* &\leftarrow \mathbb{Z}_{x_0}, \\ &\vdots \\ x_{r-1}^* &\leftarrow \mathbb{Z}_{x_0}, \\ x^* &:= z, \\ x_{r+1}^* &\leftarrow \mathcal{D}, \\ &\vdots \\ x_\tau^* &\leftarrow \mathcal{D}. \end{aligned}$$

Then $\mathcal{A}$ runs $\mathcal{B}$ with input $pk^* := (x_0, \{Q_i\}_{i=1}^k, \{x_j^*\}_{j=1}^\tau, \{y_l\}_{l=1}^k)$, and echoes the output of $\mathcal{B}$ as its own answer. Clearly, if z is chosen from $\mathcal{D}$, then pk^* has the same distribution as $pk^{(r-1)}$, and if z is chosen uniformly from $\mathbb{Z}_{x_0}$, then pk^* has the same distribution as $pk^{(r)}$. Now, if z is from $\mathcal{D}$, we have

$$\Pr[\mathcal{A}(z) = 1] = \frac{1}{\tau} \sum_{r=1}^{\tau} \Pr[\mathcal{B}(pk^{(r-1)}) = 1],$$

and if z is from $\mathbb{Z}_{x_0}$, then

$$\Pr[\mathcal{A}(z) = 1] = \frac{1}{\tau} \sum_{r=1}^{\tau} \Pr[\mathcal{B}(pk^{(r)}) = 1].$$

(Again we omit from the notation other information $\mathcal{A}$ has other than z.)

The difference between the two probabilities is equal to

$$\begin{aligned}\frac{1}{\tau}\sum_{r=1}^{\tau}\Pr[\mathcal{B}(pk^{(r-1)})=1]-\frac{1}{\tau}\sum_{r=1}^{\tau}\Pr[\mathcal{B}(pk^{(r)})=1]&\\ =\frac{1}{\tau}\left(\Pr[\mathcal{B}(pk^{(0)})=1]-\Pr[\mathcal{B}(pk^{(\tau)})=1]\right)&\\ =\frac{1}{\tau}\left(\Pr[\mathcal{B}(pk)=1]-\Pr[\mathcal{B}(pk')=1]\right)&\\ =\epsilon/\tau.&\end{aligned}$$

Therefore, in this case $\mathcal{A}$ is a distinguisher which solves the (ρ,η,γ,l_Q)-k-DACD$_Q$ problem with advantage ϵ/τ. Under the (ρ,η,γ,l_Q)-k-DACD$_Q$ assumption, we conclude that the distinguisher $\mathcal{B}$ cannot exist. □

Proof of Lemma 5. Since we are making (ρ,η,γ,l_Q)-k-DACD$_Q$ assumption, according to Lemma 4, instead of normally chosen public key $pk=(x_0,\{Q_i\}_{i=1}^k,\{x_j\}_{j=1}^{\tau},\{y_l\}_{l=1}^k)$, we may use $pk'=(x_0,\{Q_i\}_{i=1}^k,\{x'_j\}_{j=1}^{\tau},\{y_l\}_{l=1}^k)$ with x'_j chosen uniform randomly from $\mathbb{Z}_{x_0}$, since both are computationally indistinguishable from each other.

Hence, we need only to compare $\mathcal{C}_{pk,z}(\boldsymbol{m})$ and $\mathbf{Enc}(pk,\boldsymbol{m})$ under the 'false' public key pk'. The output of $\mathcal{C}_{pk,z}(\boldsymbol{m})$ is $c'=\sum_{i=1}^k m_i y_i+\sum_{j\in S}x'_j+z \bmod x_0$, and the output of $\mathbf{Enc}(pk,\boldsymbol{m})$ is $c=\sum_{i=1}^k m_i y_i+\sum_{j\in S}x'_j \bmod x_0$. But since x'_j are uniformly chosen modulo x_0, we may use the Leftover Hash Lemma, more specifically Lemma 1 from [13], to conclude that the distribution of $\sum_{j\in S}x'_j$ is statistically indistinguishable from uniform random distribution on $\mathbb{Z}_{x_0}$, hence both distributions, c' and c, are uniform random on $\mathbb{Z}_{x_0}$. Switched to the correct public key, this implies that two distributions are computationally indistinguishable. □

C Known Attacks on the ACD Problem

In [13], many attacks on the ACD problem are described. Including those attacks, we analyze additional attacks on ACD problem. We present the attacks depending on the number of approximate multiple of p. Suppose that we are given approximate multiples of p which are $x_0,x_1,\ldots,x_m$ where only x_0 is an exact multiple. If $m=0$, we can only use factoring algorithm such as elliptic curve method [21] or number field sieve [5]. If $m=1$, Howgrave-Graham [19] and Chen-Nguyen [6] can be applied. In the case of $m\geq 2$, the Lagarias algorithm for simultaneous Diophantine approximation [20] or Cohn and Heninger's attack [9] can be considered.

C.1 The Factoring Attacks

Given $x_0=pq_0$, a exact multiple of p, one can recover p by using factoring algorithm such as elliptic curve method or number field sieve. Lenstra's elliptic curve method for factoring algorithm [21] runs in time $\exp(\mathcal{O}(\eta^{1/2}))$ where $\log p=\eta$. To avoid this attack, we have to choose $\eta\geq\mathcal{O}(\lambda^2)$. The time needed by the general number field sieve [5] to factor γ-bit integer x_0 is $\exp(\mathcal{O}(\gamma^{1/3}))$. Thus we choose $\gamma\geq\mathcal{O}(\lambda^3)$.

C.2 The Approximate GCD Problem of Two Numbers

Chen-Nguyen [6] Only one sample from $\mathcal{D}_{\gamma,\rho}(p)$ is used for solving ACD problem in Chen-Nguyen's paper [6] which considers e is in $[0, 2^\rho)$ rather than $(-2^\rho, 2^\rho)$. Given $x_0 = pq_0$ and $x_1 = pq_1 + e_1$, they start with the following observation:

$$p = \gcd\Big(x_0, \prod_{i=0}^{2^\rho - 1}(x_1 - i) (\mathrm{mod}\ x_0)\Big),$$

which holds with overwhelming probability. And then construct polynomial $f_j(x)$ defined as

$$f_j(x) = \prod_{i=0}^{j-1}(x_1 - (x+i)) (\mathrm{mod}\ x_0).$$

They notice that

$$\prod_{i=0}^{2^\rho - 1}(x_1 - i) (\mathrm{mod}\ x_0) = \prod_{k=0}^{2^{\rho' + (\rho \mathrm{mod} 2)} - 1} f_{2^{\rho'}}(2^{\rho'} k) (\mathrm{mod}\ x_0)$$

where $\rho' = \lfloor \rho/2 \rfloor$. Using FFT and multi-point evaluation, it can recover p in $\tilde{\mathcal{O}}(2^{\rho/2})$ complexity rather than $\tilde{\mathcal{O}}(2^\rho)$ of the brute force attack. So we choose parameter ρ such that $\rho > \mathcal{O}(\lambda)$.

Howgrave-Graham [19] It suggests two approaches for solving ACD problem, one is the continued fraction method and the other is Coppersmith's method. We describe Coppersmith's method, since it has better results than the continued fraction with respect to a bit-length of noise e. Given $x_0 = pq_0, x_1 = pq_1 + e_1$, it defines $q_0(x) := x_0, q_1(x) := x_1 + x$ and constructs lattice L with the coefficient vector of $p_i(x) = q_0(x)^{u-i} q_1(x)^i$ for some fixed integer u and $i = 0, \ldots, u$. When e is the correct noise of x_1, $p_i(-e) = 0 \mod p^u$ for $i = 0, \ldots, u$. If the length of the b_1 which is the first vector of LLL output is smaller than p^u, then we find all roots of b_1 over the integers not in $\mathbb{Z}_{p^u}$ and one of the roots is the correct noise e. The result of this algorithm is that it outputs all integer $p \geq x_0^\beta$ such that there exists an e with $e \leq x_0^{\beta^2}$, and p divides both x_0 and $x_1 - e$ by using LLL algorithm. Thus we choose the parameter $\gamma > \eta^2/\rho$ to resist this attack.

C.3 The Approximate GCD Problem of Many Numbers

Cohn and Heninger [9] They generalize Howgrave-Graham's approach to the case one is given many samples from $\mathcal{D}_{\gamma,\rho}(p)$. Given $x_0 = pq_0$ and $x_1, \ldots, x_m \leftarrow \mathcal{D}_{\gamma,\rho}(p)$, their approach is as follows:

1. Construct m-variate polynomial $Q_i(y_1, \ldots, y_m)$ with small coefficients satisfying

$$Q_i(e_1, \ldots, e_m) \equiv 0 \mod p^k$$

whenever $x_i \equiv e_i \mod p$ for $i = 1, \ldots, m$ and for some $k > 0$.

2. Construct a lattice L generated by the coefficient vector of Q_i's.
3. Apply LLL algorithm and get m shortest vector $v_1, \ldots, v_m$ satisfying

$$|v_1| \leq \cdots \leq |v_m| \leq 2^{(\dim L)/4} (\det L)^{1/(\dim L + 1 - m)}.$$

4. If $2^{(\dim L)/4} (\det L)^{1/(\dim L+1-m)} < p^k$, then $V_i(e_1, \ldots, e_m) = 0$ over the integers where V_i is corresponding polynomial of the vector v_i for $i = 1, \ldots, m$.
5. By solving multivariate equation $V_1, \ldots, V_m$, it can recover $e_1, \ldots, e_m$.

In step 5, it needs heuristic assumption such that equation $V_1, \ldots, V_m$ is algebraically independent. The result is as follows: By using multivariate version of Coppersmith's attack, the size of noise can be extended to $x_0^{\beta \frac{m+1}{m}}$ from $x_0^{\beta^2}$ where m is the number of samples from $\mathcal{D}_{\gamma,\rho}(p)$. If $m = 1$, the result is the same as that of Howgrave-Graham. However Cohn and Heninger's approach has one constraint such that $\beta^2 \log x_0 \gg 1$. We do not know that how much $\beta^2 \log x_0$ is lager than 1, since the analysis in [9] is given only asymptotically. So we have to analyze more concretely to get practical parameters.

Simultaneous Diophantine Approximation (SDA) [20] This attacker is considered in [13]. Given $x_0 = pq_0$ and $x_i = pq_i + e_i \leftarrow \mathcal{D}_{\gamma,\rho}(p)$ for $i = 1, \ldots, m$, note that the rational numbers $y_i = x_i/x_0$ are the instances of the simultaneous diophantine approximation problem. Consider

$$y_i = \frac{x_i}{x_0} = \frac{pq_i + e_i}{pq_0} = \frac{q_i}{q_0} + \frac{s_i}{q_0} \tag{1}$$

where $s_i = e_i/p \approx 2^{\rho-\eta}$. We now try to use Lagarias's algorithm to solve this problem. Consider the lattice spanned by row vectors of following matrix:

$$L = \begin{pmatrix} 2^\rho & x_1 & x_2 & \cdots & x_m \\ 0 & -x_0 & 0 & \cdots & 0 \\ 0 & 0 & -x_0 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \cdots & -x_0 \end{pmatrix}.$$

Suppose that $\boldsymbol{v}$ be the target vector of lattice L, that is $\boldsymbol{v} = (q_0, \ldots, q_t) \cdot L$. Then $\|\boldsymbol{v}\| \approx 2^{\gamma-\eta+\rho}\sqrt{m+1}$, since

$$\boldsymbol{v} = (q_0, \ldots, q_m) \cdot L = (q_0 \cdot 2^\rho, q_0 x_1 - x_0 q_1, \ldots, q_0 x_m - x_0 q_m).$$

with $|q_0 2^\rho| \approx 2^{\gamma-\eta+\rho}$ and $|q_0 x_i - x_0 q_i| = |x_0 q_0 (\frac{x_i}{x_0} - \frac{q_i}{q_0})| = |x_0 s_i| \approx 2^{\gamma-\eta+\rho}$. Since Minkowski's bound is $(\det L)^{1/\dim L}\sqrt{m+1} \approx 2^{\frac{\gamma m+\rho}{m+1}}\sqrt{m+1} = 2^{\gamma - \frac{\gamma-\rho}{m+1}}\sqrt{m+1}$, only if $m + 1 > \frac{\gamma-\rho}{\eta-\rho}$, the target vector $\boldsymbol{v}$ can be the shortest vector in L. Actually, there are exponentially many vector of length smaller than $2^\gamma\sqrt{m+1}$, which is longer than $2^{\eta-\rho}$. As a rule of thumb, it requires time roughly $2^{m/k}$ to output a 2^k approximation of the shortest vector. So it can be found $2^{\eta-\rho}$ approximation of the shortest vector in $2^{m/(\eta-\rho)} > 2^{(\gamma-\rho)/(\eta-\rho)^2}$. Thus one must choose parameters γ, η such that $\gamma/\eta^2 = \omega(\log \lambda)$ to avoid the attack on the ACD problem using Lagarias algorithm.