Analysis of Gong et al.'s CCA2-Secure Homomorphic Encryption

Hyung Tae Lee, San Ling, and Huaxiong Wang

Division of Mathematical Sciences
School of Physical & Mathematical Sciences
Nanyang Technological University, Singapore
{hyungtaelee, lingsan, hxwang}@ntu.edu.sg

Abstract. It is a well-known result that homomorphic encryption is not secure against adaptive chosen ciphertext attacks (CCA2) because of its malleability property. Very recently, however, Gong *et al.* proposed a construction asserted to be a CCA2-secure additive homomorphic encryption (AHE) scheme; in their construction, the adversary is not able to obtain a correct answer when querying the decryption oracle on a ciphertext obtained by modifying the challenge ciphertext (Theoretical Computer Science, 2016). Because their construction is very similar to Paillier's AHE, it appeared to support an additive homomorphic property, though they did not specify an evaluation algorithm for the scheme in their paper.

In this paper, we present a simple CCA2 attack on their construction by re-randomizing the challenge ciphertext. Furthermore, we look into an additive homomorphic property of their construction. To do this, we first consider a typical candidate for an addition algorithm on ciphertexts, as provided for previous AHE constructions, and establish that it does not function correctly. Subsequently, we provide plausible evidence for the hardness of achieving an additive homomorphic property with their construction. According to our analysis, it seems hard to preserve additive homomorphic property of their construction without any modification.

In addition, as a minor contribution, we point out a flaw in the decryption algorithm of their construction and present a rectified algorithm for correct decryption.

Keywords: additive homomorphic encryption, adaptive chosen ciphertext attack, malleability

1 Introduction

Because homomorphic encryption allows computations on encrypted data, it has various applications, e.g., secure multiparty computation [6, 2], cloud computing [9], and electronic voting [4]. The security of such applications is directly affected by that of the homomorphic encryption employed, which has led to the question of what level of security homomorphic encryption can achieve. There have been several studies [7, 8, 1] demonstrating that encryption schemes with supporting homomorphic operations can be secure against non-adaptive chosen ciphertext attacks (CCA1), i.e., lunch time attack. On the other hand, Bellare *et al.* [3] have demonstrated that no homomorphic encryption scheme can be secure against adaptive chosen ciphertext attacks (CCA2) because of its malleability property.

Very recently, Gong *et al.* [5] presented quite a surprising result when they proposed a construction asserted to be an additive homomorphic encryption (AHE) scheme secure against CCA2. In seeking to achieve CCA2 security, they constructed an encryption scheme such that a message is located in the exponent to the base g^{ab}, where g is a generator of the underlying group, a is a fixed integer chosen by the key generation algorithm, and b is a random integer chosen by the encryption algorithm. They maintained that a polynomial-time adversary could not know the exact value of g^{ab} for the challenge ciphertext and therefore could not generate a suitable ciphertext that contributes to guessing the message corresponding to the challenge ciphertext. Hence, their construction seems secure against CCA2. Furthermore, their construction is very similar to Paillier's

AHE scheme [10], so it seems to allow additions on encrypted data, though they did not specify an evaluation algorithm in their paper.

In this paper, however, we present a simple CCA2 attack on their construction. Our attack is designed as follows: Assume that the challenge ciphertext $\mathcal{C} = (C_1, C_2, C)$ of a hidden message m is given. Then, C has the form $g^{ab(m+a)} \cdot A$ for a generator g of the underlying group $\mathbb{G}$, some element $A \in \mathbb{G}$, and integers a and b. The adversary chooses a random integer s ($\neq 0, 1$) and computes $\mathcal{C}^s = (C_1^s, C_2^s, C^s)$. Then, C^s can be transformed into the form $g^{abs(m+a)} \cdot A^s$, and $\mathcal{C}^s$ is still a valid ciphertext of the message m. Hence, when the adversary queries the decryption oracle on $\mathcal{C}^s$, it returns the message m.

Furthermore, we investigate an additive homomorphic property of their construction. To this end, we first present a typical candidate for an evaluation algorithm on ciphertexts, which is defined by component-wise group operations between ciphertexts. That is, for given ciphertexts $\hat{\mathcal{C}} = (\hat{C}_1, \hat{C}_2, \hat{C})$ and $\check{\mathcal{C}} = (\check{C}_1, \check{C}_2, \check{C})$, an evaluated ciphertext is defined as $\mathcal{C} = (\hat{C}_1 \cdot \check{C}_1, \hat{C}_2 \cdot \check{C}_2, \hat{C} \cdot \check{C})$. Then, we establish that this computation does not preserve the additive homomorphic property.

Subsequently, we provide plausible evidence that it is impossible to provide an addition algorithm for Gong *et al.*'s construction. To this end, we first simplify the problem of providing an addition algorithm for it with Paillier's AHE scheme $\mathsf{P.Enc}$. As a result, we obtain the following problem: Denote a ciphertext of Paillier's scheme of a message m by $\mathsf{P.Enc}(m)$. For any hidden integers $\hat{m}$, $\check{m}$, α, and β and a fixed hidden value a, when $\mathsf{P.Enc}(\alpha)$, $\mathsf{P.Enc}(\alpha(\hat{m} + a))$, $\mathsf{P.Enc}(\beta)$, and $\mathsf{P.Enc}(\beta(\check{m} + a))$ are given, generate $\mathsf{P.Enc}(\gamma(\hat{m} + \check{m} + a))$ and $\mathsf{P.Enc}(\gamma)$ for some nonzero scalar γ.

By using the additive homomorphic property of Paillier's scheme, we can generate ciphertexts of the form

$$\mathsf{P.Enc}(\alpha X_1(\hat{m} + a) + \alpha X_2 + \beta Y_1(\check{m} + a) + \beta Y_2 + Z) \tag{1}$$

for some scalars X_1, X_2, Y_1, Y_2, and Z. In order that the above ciphertext has the form $\mathsf{P.Enc}(\gamma(\hat{m} + \check{m} + a))$, an evaluation algorithm should find a solution of the following system of equations:

$$\begin{cases} \alpha X_1 = \beta Y_1 \neq 0 \\ a\alpha X_1 + \alpha X_2 + \beta Y_2 + Z = 0. \end{cases}$$

However, because a, α, and β are hidden values, it is impossible to find (X_1, X_2, Y_1, Y_2, Z) satisfying the above system of equations except with a negligible probability. Therefore, it seems hard to provide an addition algorithm for Gong *et al.*'s construction.

As a minor contribution, we also point out a flaw in the decryption algorithm of their original construction and provide a rectified algorithm for correct decryption.

Organization of the Paper. In Section 2, we provide Paillier's AHE scheme and Gong *et al.*'s recent construction. In Section 3, we demonstrate that the decryption algorithm of Gong *et al.*'s construction does not work correctly and provide a corrected version to accomplish the decryption. Section 4 presents our CCA2 attack on Gong *et al.*'s scheme. Finally, we discuss about an additive homomorphic property of their construction in Section 5.

2 Gong *et al.*'s Proposed CCA2-Secure Additive Homomorphic Encryption

In this section, we present some basic definitions related to public-key encryption. Then, we introduce Paillier's AHE scheme, which is the key foundation of Gong *et al.*'s construction and will

be utilized in our discussion about an additive homomorphic property of their construction in Section 5. We also provide the description of Gong *et al.*'s scheme.

2.1 Public Key Encryption and CCA2 Security

A public key encryption scheme consists of the following three algorithms:

- $\mathsf{KeyGen}(\kappa)$: This takes a security parameter κ as an input and outputs a public key pk and a secret key sk.
- $\mathsf{Enc}(\mathsf{pk}, m)$: This takes the public key pk and a message m as inputs and outputs a ciphertext C.
- $\mathsf{Dec}(\mathsf{sk}, C)$: This takes the secret key sk and a ciphertext C as inputs and outputs a message m'.

We say that a public key encryption scheme is correct if for all messages m and security parameters κ,

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m,$$

where pk and sk are outputs of $\mathsf{KeyGen}(\kappa)$.

The security of a public key encryption scheme is defined by the following game between a challenger and an adversary:

- **Setup:** The challenger obtains the public key pk and the secret key sk by running $\mathsf{KeyGen}(\kappa)$ for the security parameter κ and sends pk to the adversary.
- **Phase 1:** The adversary generates ciphertexts and sends them as queries to the decryption oracle, which outputs the plaintext message corresponding to the input ciphertext.
- **Challenge:** The adversary sends two messages m_0 and m_1 of equal length. The challenger randomly selects β from $\{0, 1\}$ and sends the adversary C_β obtained by running $\mathsf{Enc}(\mathsf{pk}, m_\beta)$.
- **Phase 2:** The adversary generates ciphertexts and sends them as queries to the decryption oracle. Note that he cannot send the challenge ciphertext C_β as a query.
- **Guess:** The adversary outputs $\beta' \in \{0, 1\}$.

The advantage of the adversary in the above game is defined to be $|\Pr[\beta = \beta'] - \frac{1}{2}|$. We say that a public key encryption scheme is CCA2-secure if there is no polynomial-time adversary whose advantage in the above game is non-negligible in the security parameter κ.

2.2 Paillier's Additive Homomorphic Encryption

In 1999, Paillier [10] proposed three public key encryption schemes based on a new assumption, called the *Decisional Composite Residuosity* (DCR) assumption. These schemes have been widely utilized in various applications because they are very efficient and allow additions on encrypted data.

Here we provide a description of the first scheme among them.

- $\mathsf{P.KeyGen}(\kappa)$: This takes a security parameter κ as an input and performs as follows:
  1. Select $\eta(\kappa)$-bit random primes p and q and set $n = pq$.

2. Compute $\lambda = \mathrm{lcm}(p-1, q-1)$.
3. Select an element g of order n in the multiplicative group $\mathbb{Z}_{n^2}^*$.
4. Output the public key $\mathsf{pk} = (n, g)$ and keep the secret key $\mathsf{sk} = \lambda$ private.

– $\mathsf{P.Enc}(\mathsf{pk}, m)$: Given the public key pk and a message $m \in \mathbb{Z}_n$, this performs as follows:
1. Select a random integer r from $\mathbb{Z}_n^*$.
2. Compute $C = g^m r^n \bmod n^2$ and output C.

– $\mathsf{P.Dec}(\mathsf{sk}, C)$: Given the secret key sk and a ciphertext C, this computes

$$m = \frac{L(C^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n,$$

where L is a function defined by $L(y) = \dfrac{y-1}{n}$ for $y < n^2$. Then, it outputs m.

2.3 Gong *et al.*'s Scheme

Based on Paillier's encryption scheme, Gong *et al.* [5] proposed a construction asserted to be a CCA2-secure AHE scheme. The description of their construction is as follows:

– $\mathsf{G.KeyGen}(\kappa)$: This takes a security parameter κ as an input and performs as follows:
1. Select $\eta(\kappa)$-bit random primes p and q and set $n = pq$.
2. Compute $\lambda = \mathrm{lcm}(p-1, q-1)$.
3. Compute a nontrivial factor t of λ and λ/t.
4. Select random numbers a, k, z_1, and z_2 from $\mathbb{Z}_n^*$.
5. Compute $g = 1 + kn$, $y = g^a \bmod n^2$, $y' = z_1^a g^{a^2} \bmod n^2$, and $y'' = z_1^a z_2^{tn} \bmod n^2$. We note that g has order n in the multiplicative group $\mathbb{Z}_{n^2}^*$.
6. Output the public key $\mathsf{pk} = (y, y', y'', z_1, n)$ and keep the secret key $\mathsf{sk} = (a, t, \lambda/t, \lambda)$ private.

– $\mathsf{G.Enc}(\mathsf{pk}, m)$: Given the public key pk and a message $m \in \mathbb{Z}_n$, this performs as follows:
1. Select random numbers r, r_1, and b from $\mathbb{Z}_n^*$.
2. Compute $B_x = y^b \bmod n^2$, $B'_x = (y')^b \bmod n^2$, $C_1 = z_1^{b(r+1)} \bmod n^2$, $C_2 = y^b r_1^n \bmod n^2$, and $C = B_x^m B'_x (y'')^{br} \bmod n^2$.
3. Output a ciphertext $\mathcal{C} = (C_1, C_2, C)$.

– $\mathsf{G.Dec}(\mathsf{sk}, \mathcal{C})$: Given the secret key sk and a ciphertext $\mathcal{C}$,
1. Parse $\mathcal{C}$ as (C_1, C_2, C).
2. Compute

$$m = \left(\frac{L(C \cdot C_1^{tn-a})^{\lambda/t} \bmod n^2)}{L(C_2^\lambda \bmod n^2)} \bmod n \right) - a \bmod n, \tag{2}$$

where L is a function defined by $L(y) = \dfrac{y-1}{n}$ for $y < n^2$.
3. Output m.

In an effort to achieve CCA2 security, Gong *et al.* attempted to prevent the adversary from launching CCA2 by not enabling him to obtain a correct answer when he queries the decryption oracle on a ciphertext obtained by modifying the challenge ciphertext. More precisely, in their construction, a message m is defined in the exponent to the base g^{ab}, where a is chosen by the key generation algorithm and b is chosen by the encryption algorithm. Here, each b is changed for each ciphertext, and they maintained that the adversary cannot succeed in CCA2 unless he finds g^{ab} corresponding to the challenge ciphertext, which is infeasible in polynomial time. However, this feature not only prevents their construction from supporting an additive homomorphic property, but is also insufficient for achieving CCA2 security. We will present our CCA2 attack on their construction in Section 4 and discuss its additive homomorphic property in Section 5.

Before moving on to the next section, we remark on relationships between ciphertexts in Paillier's AHE scheme and those in Gong *et al.*'s construction to facilitate the reader's understanding. In fact, parts of a ciphertext in Gong *et al.*'s construction can be interpreted as ciphertexts in Paillier's encryption scheme with the public key $\mathsf{pk} = (n, g = 1 + kn)$. For a valid ciphertext $\mathcal{C} = (C_1, C_2, C)$ in Gong et al.'s encryption,

$$C_2 = y^b r_1^n = g^{ab} r_1^n \bmod n^2.$$

Hence, C_2 can be regarded as a ciphertext $\mathsf{P.Enc}(\mathsf{pk}, ab)$ when r_1 is a random element chosen in Paillier's encryption algorithm. Furthermore,

$$\begin{aligned} C &= B_x^m B_x'(y'')^{br} \bmod n^2 \\ &= (y^b)^m (y')^b (y'')^{br} \bmod n^2 \\ &= (g^{ab})^m (z_1^a g^{a^2})^b (z_1^a z_2^{tn})^{br} \bmod n^2 \\ &= \left(g^{ab(m+a)} (z_2^{brt})^n \right) \cdot z_1^{ab+abr} \bmod n^2, \end{aligned} \tag{3}$$

and hence C can be regarded as a multiplication of ciphertext $\mathsf{P.Enc}(\mathsf{pk}, ab(m+a))$ and z_1^{ab+abr}, where z_2^{brt} is a corresponding random element to generate a ciphertext $\mathsf{P.Enc}(\mathsf{pk}, ab(m+a))$. That is,

$$C = \mathsf{P.Enc}(\mathsf{pk}, ab(m+a)) \cdot z_1^{ab+abr} \bmod n^2$$

when z_2^{brt} is a randomly chosen element in the Paillier encryption algorithm. We will use these relationships in Section 5 to look into the impossibility of achieving an additive homomorphic property of Gong *et al.*'s construction.

3 Correction to the Decryption Algorithm of Gong *et al.*'s Scheme

In this section, we demonstrate that the decryption algorithm of Gong *et al.*'s construction does not function correctly and provide a rectified algorithm for correct decryption.

Incorrect Decryption of Gong *et al.*'s Scheme. Let $\mathcal{C} = (C_1, C_2, C)$ be a valid ciphertext. Then, we can represent the components as

$$C_1 = z_1^{b(r+1)} \bmod n^2, C_2 = y^b r_1^n \bmod n^2, \text{ and } C = B_x^m B_x'(y'')^{br} \bmod n^2,$$

where $B_x = y^b \bmod n^2$, $B'_x = (y')^b \bmod n^2$, and b, r, and r_1 are integers randomly chosen from $\mathbb{Z}_n^*$ in the encryption phase. Furthermore, the public keys satisfy $y = g^a \bmod n^2$, $y' = z_1^a g^{a^2} \bmod n^2$, and $y'' = z_1^a z_2^{tn} \bmod n^2$, where $g = 1 + kn$ for some $k \in \mathbb{Z}_n^*$ and a, z_1, and z_2 are randomly chosen integers from $\mathbb{Z}_n^*$. Let t be a nontrivial factor of λ in the secret key. Then,

$$\begin{aligned}
(C \cdot C_1^{(tn-a)})^{\lambda/t} &= \left(B_x^m B'_x (y'')^{br} (z_1^{b(r+1)})^{(tn-a)}\right)^{\lambda/t} \bmod n^2 \\
&= \left((y^b)^m (y')^b (y'')^{br} (z_1^{b(r+1)})^{(tn-a)}\right)^{\lambda/t} \bmod n^2 \\
&= \left((g^{ab})^m (z_1^a g^{a^2})^b (z_1^a z_2^{tn})^{br} (z_1^{b(r+1)})^{(tn-a)}\right)^{\lambda/t} \bmod n^2 \\
&= (g^{ab})^{(m+a)\lambda/t} (z_1^{b(r+1)tn} z_2^{tnbr})^{\lambda/t} \bmod n^2 \\
&= (g^{ab})^{(m+a)\lambda/t} (z_1^{b(r+1)} z_2^{br})^{n\lambda} \bmod n^2 \\
&= (g^{ab})^{(m+a)\lambda/t} \bmod n^2 \\
&= (1+kn)^{ab(m+a)\lambda/t} \\
&= 1 + kab(m+a)(\lambda/t)n \bmod n^2 \qquad (4)
\end{aligned}$$

and

$$\begin{aligned}
C_2^\lambda &= (y^b r_1^n)^\lambda = g^{ab\lambda} \bmod n^2 \\
&= (1+kn)^{ab\lambda} = 1 + kab\lambda n \bmod n^2. \qquad (5)
\end{aligned}$$

We note that the sixth equality in Equation (4) and the second equality in Equation (5) hold because the multiplicative order of each element in $\mathbb{Z}_{n^2}^*$ is a factor of $n\lambda$. Hence,

$$\begin{aligned}
&\left(\frac{L((C \cdot C_1^{tn-a})^{\lambda/t} \bmod n^2)}{L(C_2^\lambda \bmod n^2)} \bmod n\right) - a \bmod n \\
&= \left(\frac{L(1 + kab(m+a)(\lambda/t)n \bmod n^2)}{L(1 + kab\lambda n \bmod n^2)} \bmod n\right) - a \bmod n \\
&= \left(\frac{kab(m+a)(\lambda/t)}{kab\lambda} \bmod n\right) - a \bmod n \\
&= t^{-1}(m+a) \bmod n - a \bmod n.
\end{aligned}$$

Therefore, the decryption algorithm does not return the correct message corresponding to the ciphertext.

Modification for Correct Decryption. We can easily fix the decryption algorithm by multiplying the secret value t and the first term on the right side of Equation (2) as follows: For a ciphertext $\mathcal{C} = (C_1, C_2, C)$, define a decryption algorithm by

$$\mathsf{G.Dec}'(\mathsf{sk}, \mathcal{C}) = \left(t \cdot \frac{L((C \cdot C_1^{tn-a})^{\lambda/t} \bmod n^2)}{L(C_2^\lambda \bmod n^2)} \bmod n\right) - a \bmod n. \qquad (6)$$

Then, we do obtain the correct message corresponding to the ciphertext, because

$$\begin{aligned}
&\left(t \cdot \frac{L(C \cdot C_1^{tn-a})^{\lambda/t} \bmod n^2)}{L(C_2^{\lambda} \bmod n^2)} \bmod n\right) - a \bmod n \\
&= t(t^{-1}(m+a)) \bmod n - a \bmod n = ((m+a)-a) \bmod n \\
&= m \bmod n.
\end{aligned}$$

4 Adaptive Chosen Ciphertext Attack on Gong *et al.*'s Scheme

Now, we present our CCA2 attack on Gong *et al.*'s construction. Our attack is very straightforward and consists of a simple re-randomization by computing an exponentiation of the challenge ciphertext using an exponent that is a randomly chosen element in $(\mathbb{Z}_n \backslash \{0,1\})$. The result of this re-randomization is still a valid ciphertext of the same message as the challenge ciphertext. Therefore, the adversary can recover the exact message of the challenge ciphertext by querying the decryption oracle on the re-randomized ciphertext.

Let us explain our CCA2 attack more precisely. After the challenge phase of the CCA2 security game in Section 2.1, assume that the adversary receives the challenge ciphertext $\mathcal{C} = (C_1, C_2, C)$ from the challenger. Then, the challenge ciphertext of the message m_β for $\beta \in \{0,1\}$ can be represented as

$$C_1 = z_1^{b(r+1)} \bmod n^2, C_2 = y^b r_1^n \bmod n^2, \tag{7}$$

and

$$C = (y^b)^{m_\beta}(y')^b(y'')^{br} \bmod n^2, \tag{8}$$

where b, r, and r_1 are integers randomly chosen from $\mathbb{Z}_n^*$ by the encryption algorithm.

At this point, the adversary randomly selects an element s from $\mathbb{Z}_n \backslash \{0,1\}$ and computes $\mathcal{C}^s := (C_1^s, C_2^s, C^s)$. Then,

$$C_1^s = z_1^{(bs)(r+1)} \bmod n^2, C_2^s = y^{(bs)}(r_1^s)^n \bmod n^2,$$

and

$$C = (y^{(bs)})^{m_\beta}(y')^{(bs)}(y'')^{(bs)r} \bmod n^2.$$

Hence, we can see that $\mathcal{C}^s$ is obtained by substituting b and r_1 with bs and r_1^s in Equations (7) and (8), respectively. Thus, $\mathcal{C}^s$ is also a valid ciphertext of the message m_β, and the adversary can obtain the challenge message m_β by querying the decryption oracle on $\mathcal{C}^s$ in Phase 2 of the security game. Therefore, by our attack, Gong *et al.*'s construction is not CCA2-secure.

5 On the Additive Homomorphic Property of Gong *et al.*'s Scheme

In this section, we look into Gong *et al.*'s assertion of an additive homomorphic property for their scheme. In their original paper [5], the authors did not provide an addition algorithm on ciphertexts. Hence, we first present a typical candidate for an addition algorithm by considering

existing AHE schemes defined over multiplicative groups and show that it does not preserve an additive homomorphic property. Thereafter, we provide plausible evidence for the impossibility that their construction preserves an additive homomorphic property.

Throughout this section, we assume that two valid ciphertexts $\hat{\mathcal{C}} = (\hat{C}_1, \hat{C}_2, \hat{C})$ and $\check{\mathcal{C}} = (\check{C}_1, \check{C}_2, \check{C})$ under the same public key are given and that they satisfy the following relationships: Let pk be (y, y', y'', z_1, n), where $g = 1 + kn$, $y = g^a \bmod n^2$, $y' = z_1^a g^{a^2} \bmod n^2$, and $y'' = z_1^a z_2^{tn} \bmod n^2$ for integers $a, k, z_1, z_2 \in \mathbb{Z}_n^*$. Then, there exist $\hat{b}$, $\check{b}$, $\hat{r}$, $\check{r}$, $\hat{r}_1$, and $\check{r}_1$ in $\mathbb{Z}_n^*$ such that

$$\hat{C}_1 = z_1^{\hat{b}(\hat{r}+1)} \bmod n^2, \hat{C}_2 = y^{\hat{b}} \hat{r}_1^n \bmod n^2, \hat{C} = (y^{\hat{b}})^{\hat{m}} (y')^{\hat{b}} (y'')^{\hat{b}\hat{r}} \bmod n^2$$

and

$$\check{C}_1 = z_1^{\check{b}(\check{r}+1)} \bmod n^2, \check{C}_2 = y^{\check{b}} \check{r}_1^n \bmod n^2, \check{C} = (y^{\check{b}})^{\check{m}} (y')^{\check{b}} (y'')^{\check{b}\check{r}} \bmod n^2.$$

Typical Candidate for an Addition Algorithm. A typical candidate for an addition algorithm on ciphertexts of AHE defined over multiplicative groups is to multiply ciphertexts component-wise. Let us define $\mathcal{C} = (C_1, C_2, C)$ as a component-wise multiplication between two ciphertexts $\hat{\mathcal{C}}$ and $\check{\mathcal{C}}$. That is,

$$C_1 = \hat{C}_1 \cdot \check{C}_1, C_2 = \hat{C}_2 \cdot \check{C}_2, \text{ and } C = \hat{C} \cdot \check{C}.$$

Then,

$$\begin{aligned}
C \cdot C_1^{tn-a} &= (\hat{C} \cdot \check{C}) \cdot (\hat{C}_1 \cdot \check{C}_1)^{tn-a} \bmod n^2 \\
&= \left((y^{\hat{b}\hat{m}+\check{b}\check{m}}) (y')^{\hat{b}+\check{b}} (y'')^{\hat{b}\hat{r}+\check{b}\check{r}} \right) \cdot \left(z_1^{\hat{b}(\hat{r}+1)+\check{b}(\check{r}+1)} \right)^{tn-a} \bmod n^2 \\
&= \left((g^{a(\hat{b}\hat{m}+\check{b}\check{m})}) (z_1^a g^{a^2})^{\hat{b}+\check{b}} (z_1^a z_2^{tn})^{\hat{b}\hat{r}+\check{b}\check{r}} \right) \cdot \left(z_1^{\hat{b}(\hat{r}+1)+\check{b}(\check{r}+1)} \right)^{tn-a} \bmod n^2 \\
&= \left((g^{a(\hat{b}\hat{m}+\check{b}\check{m})}) \cdot g^{a^2(\hat{b}+\check{b})} \right) \cdot \left(z_1^{\hat{b}(\hat{r}+1)+\check{b}(\check{r}+1)} z_2^{\hat{b}\hat{r}+\check{b}\check{r}} \right)^{tn} \bmod n^2 \\
&= \left(g^{a(\hat{b}\hat{m}+\check{b}\check{m})+a^2(\hat{b}+\check{b})} \right) \cdot \left(z_1^{\hat{b}(\hat{r}+1)+\check{b}(\check{r}+1)} z_2^{\hat{b}\hat{r}+\check{b}\check{r}} \right)^{tn} \bmod n^2
\end{aligned}$$

and hence

$$\begin{aligned}
L((C \cdot C_1^{tn-a})^{\lambda/t} \bmod n^2) &= L(((1+kn)^{a(\hat{b}\hat{m}+\check{b}\check{m})+a^2(\hat{b}+\check{b})})^{\lambda/t} \bmod n^2) \\
&= \frac{1 + k(a(\hat{b}\hat{m} + \check{b}\check{m}) + a^2(\hat{b} + \check{b}))(\lambda/t)n - 1}{n} \bmod n \\
&= k \left(a(\hat{b}\hat{m} + \check{b}\check{m}) + a^2(\hat{b} + \check{b}) \right) (\lambda/t) \bmod n.
\end{aligned}$$

The first equality in the above equation holds because the multiplicative order of each element in $\mathbb{Z}_{n^2}^*$ is a factor of $n\lambda$. Moreover, the following holds:

$$\begin{aligned}
L(C_2^\lambda \bmod n^2) &= L(((y^{\hat{b}})\hat{r}_1^n(y^{\check{b}})\check{r}_1^n)^\lambda \bmod n^2) \\
&= L(((y)^{\hat{b}+\check{b}}(\hat{r}_1\check{r}_1)^n)^\lambda \bmod n^2) \\
&= L(((g)^{a(\hat{b}+\check{b})}(\hat{r}_1\check{r}_1)^n)^\lambda \bmod n^2) \\
&= L((1+kn)^{(a(\hat{b}+\check{b}))\lambda} \bmod n^2) \\
&= \frac{1+ka(\hat{b}+\check{b})\lambda n - 1}{n} \bmod n \\
&= ka(\hat{b}+\check{b})\lambda \bmod n.
\end{aligned}$$

Therefore, our modified decryption algorithm in Equation (6) outputs $\frac{\hat{b}\hat{m}+\check{b}\check{m}}{\hat{b}+\check{b}} \mod n$, not $\hat{m}+\check{m} \mod n$, by the following computation:

$$\begin{aligned}
\mathsf{G.Dec}'(\mathsf{sk}, \mathcal{C}) &= \left(t\frac{L((C \cdot C_1^{tn-a})^{\lambda/t} \bmod n^2)}{L(C_2^\lambda \bmod n^2)} \bmod n \right) - a \bmod n \\
&= t\frac{k\left(a(\hat{b}\hat{m}+\check{b}\check{m}) + a^2(\hat{b}+\check{b})\right)(\lambda/t)}{ka(\hat{b}+\check{b})\lambda} \bmod n - a \bmod n \\
&= \frac{\hat{b}\hat{m}+\check{b}\check{m}}{\hat{b}+\check{b}} \bmod n.
\end{aligned}$$

Furthermore, because $\hat{b}$ and $\check{b}$ are randomly chosen by the encryption algorithm, the receiver who decrypts ciphertexts cannot know $\hat{b}$ and $\check{b}$ and hence cannot recover $\hat{m}+\check{m} \mod n$ from $\frac{\hat{b}\hat{m}+\check{b}\check{m}}{\hat{b}+\check{b}} \mod n$. Therefore, an additive homomorphic property of Gong *et al.*'s construction cannot be preserved by using this typical candidate.

Discussion on the Impossibility of Preserving Additive Homomorphic Property in Gong *et al.*'s Scheme. Now, we provide plausible evidence for the impossibility of providing an addition algorithm for Gong *et al.*'s construction. To do this, we will first simplify the problem of providing an addition algorithm for their construction by replacing their ciphertexts with Paillier's and then examine the hardness of this simplified problem.

As seen in Section 2.3, parts of the ciphertexts in Gong *et al.*'s scheme can be regarded as ciphertexts in the Paillier encryption. Hence, we can replace $\hat{\mathcal{C}} = (\hat{C}_1, \hat{C}_2, \hat{C})$ and $\check{\mathcal{C}} = (\check{C}_1, \check{C}_2, \check{C})$ by

$$(\hat{C}_1, \mathsf{P.Enc}(\mathsf{pk}, a\hat{b}), \mathsf{P.Enc}(\mathsf{pk}, a\hat{b}(\hat{m}+a)) \cdot z_1^{a\hat{b}+a\hat{b}\hat{r}} \bmod n^2) \tag{9}$$

and

$$(\check{C}_1, \mathsf{P.Enc}(\mathsf{pk}, a\check{b}), \mathsf{P.Enc}(\mathsf{pk}, a\check{b}(\check{m}+a)) \cdot z_1^{a\check{b}+a\check{b}\check{r}} \bmod n^2), \tag{10}$$

respectively, where pk is $(n, g = 1+kn)$. Here, the role of C_1 in a ciphertext $\mathcal{C} = (C_1, C_2, C)$, where $C = \left(g^{ab(m+a)}(z_2^{brt})^n\right) \cdot z_1^{ab+abr} \bmod n^2$ in Equation (3), is to remove z_1^{ab+abr} from the value of C

for correct decryption. Hence, we can regard C_1 as having no effect on the message in the case of a valid ciphertext, and we may ignore the C_1 component and z_1^{ab+abr} in the C component for constructing an evaluation algorithm.

Since an evaluation algorithm should take only public parameters and ciphertexts as inputs, by replacing $a\hat{b}$ and $a\check{b}$ with α and β in Equations (9) and (10), respectively, we can define the problem of providing an addition algorithm for Gong *et al.*'s construction as follows: Let $\mathsf{P.Enc}$ be Paillier's encryption algorithm. For any hidden $\hat{m}, \check{m}, \alpha, \beta \in \mathbb{Z}_n$ and a fixed value $a \in \mathbb{Z}_n^*$, when $\mathsf{P.Enc}(\mathsf{pk}, \alpha)$, $\mathsf{P.Enc}(\mathsf{pk}, \beta)$, $\mathsf{P.Enc}(\mathsf{pk}, \alpha(\hat{m}+a))$, and $\mathsf{P.Enc}(\mathsf{pk}, \beta(\check{m}+a))$ are given, generate $\mathsf{P.Enc}(\mathsf{pk}, \gamma)$ and $\mathsf{P.Enc}(\mathsf{pk}, \gamma(\hat{m}+\check{m}+a))$ for some nonzero integer γ.

Because Paillier's encryption supports only an additive homomorphic property, a solver of the above problem allows only scalar multiplications and additions on ciphertexts. Hence, he can only obtain ciphertexts of the form

$$\begin{aligned}&\mathsf{P.Enc}(\mathsf{pk}, \alpha X_1(\hat{m}+a) + \alpha X_2 + \beta Y_1(\check{m}+a) + \beta Y_2 + Z)\\ &= \mathsf{P.Enc}(\mathsf{pk}, \alpha X_1(\hat{m}+a) + \beta Y_1(\check{m}+a) + \alpha X_2 + \beta Y_2 + Z),\end{aligned} \tag{11}$$

by computing

$$\mathsf{P.Enc}(\mathsf{pk}, \alpha(\hat{m}+a))^{X_1} \times \mathsf{P.Enc}(\mathsf{pk}, \alpha)^{X_2} \times \mathsf{P.Enc}(\mathsf{pk}, \beta(\check{m}+a))^{Y_1} \times \mathsf{P.Enc}(\mathsf{pk}, \beta)^{Y_2} \times \mathsf{P.Enc}(\mathsf{pk}, Z)$$

for some scalars X_1, X_2, Y_1, Y_2, and Z.

To generate a ciphertext of the form $\mathsf{P.Enc}(\mathsf{pk}, \gamma(\hat{m}+\check{m}+a))$ from ciphertexts of the form (11) for any $\hat{m}, \check{m}$, a tuple of scalars (X_1, X_2, Y_1, Y_2, Z) should be a solution of the following system of equations:

$$\begin{cases}\alpha X_1 = \beta Y_1 = \gamma \neq 0\\ a\alpha X_1 + \alpha X_2 + \beta Y_2 + Z = 0\end{cases} \tag{12}$$

because if the above satisfies, then the following holds for any $\hat{m}, \check{m}$:

$$\begin{aligned}(11) &= \mathsf{P.Enc}(\mathsf{pk}, \alpha X_1(\hat{m}+a) + \beta Y_1(\check{m}+a) + \alpha X_2 + \beta Y_2 + Z)\\ &= \mathsf{P.Enc}(\alpha X_1(\hat{m}+\check{m}+a) + a\alpha X_1 + \alpha X_2 + \beta Y_2 + Z)\\ &= \mathsf{P.Enc}(\gamma(\hat{m}+\check{m}+a)).\end{aligned}$$

However, it is infeasible to solve the above system of equations because a, α, and β are hidden. Therefore, it seems hard to provide an addition algorithm for Gong *et al.*'s construction without any modification.

6 Conclusion

Very recently, Gong *et al.* proposed a construction asserted to be a CCA2-secure AHE scheme [5]. In this paper, we first identified that their decryption algorithm does not function correctly and provided the rectified algorithm for correct decryption. Subsequently, we provided a simple CCA2 attack on their construction by re-randomizing the challenge ciphertext with a randomly chosen exponent in $\mathbb{Z}_n \backslash \{0, 1\}$. We also pointed out that their construction seems hard to support an additive homomorphic property by considering a typical candidate for an addition algorithm and providing plausible evidence for achieving an additive homomorphic property with their construction. As a result, we conclude that their construction is in fact not a CCA2-secure homomorphic encryption scheme.

Acknowledgements

This work was supported by Research Grant TL-9014101684-01 and the Singapore Ministry of Education under Research Grant MOE2013-T2-1-041.

References

1. F. Armknecht, S. Katzenbeisser, and A. Peter. Group homomorphic encryption: characterizations, impossibility results, and applications. *Des. Codes Cryptography*, 67(2):209–232, 2013.
2. G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 483–501. Springer, 2012.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *LNCS*, pages 26–45. Springer, 1998.
4. I. Damgård, M. Jurik, and J. B. Nielsen. A generalization of Paillier's public-key system with applications to electronic voting. *Int. J. Inf. Sec.*, 9(6):371–385, 2010.
5. L. Gong, S. Li, Q. Mao, D. Wang, and J. Dou. A homomorphic encryption scheme with adaptive chosen ciphertext security but without random oracle. *Theoretical Computer Science*, 609(1):253–261, 2016.
6. L. Kissner and D. X. Song. Privacy-preserving set operations. In V. Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *LNCS*, pages 241–257. Springer, 2005.
7. H. Lipmaa. On the CCA1-security of Elgamal and Damgård's Elgamal. In X. Lai, M. Yung, and D. Lin, editors, *Inscrypt 2010*, volume 6584 of *LNCS*, pages 18–35. Springer, 2011.
8. J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On CCA-secure somewhat homomorphic encryption. In A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography (SAC) 2011*, volume 7118 of *LNCS*, pages 55–72. Springer, 2012.
9. A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In H. J. Karloff and T. Pitassi, editors, *Symposium on Theory of Computing Conference (STOC) 2012*, pages 1219–1234. ACM, 2012.
10. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.