A secure multi-party computation protocol without CRS supporting multi-bit encryption

Zong-Wu Zhu, Ru-Wei Huang*
School of Computer and Electronic Information, Guangxi University, Nanning, China
* ruweih@gxu.edu.cn

Abstract

To solve the problems in the existing fully homomorphic encryption (FHE)-based secure multi-party computation (SMC) protocols such as low efficiency, the FHE scheme that supports multi-bit encryption was modified during the generation of the public key so that the users could generate their public keys independently without the common random string (CRS) matrix. Further, a multi-bit Gentry-Sahai-Waters scheme (MGSW) scheme without CRS was constructed. The modified LinkAlgo algorithm was adopted to expand the single-key ciphertext into the multi-key ciphertext and simplify the way of generating the expanded ciphertext. In this way, a multi-key FHE (MFHE) scheme was achieved based on the MGSW scheme. Finally, a three-round SMC protocol without CRS was constructed using the MFHE scheme and the decisional learning with errors (DLWE) assumption, which was secure in the semi-malicious model. Compared to the existing protocols, the protocol proposed herein can support multi-bit encryption and is found with smaller ciphertext size and lower storage overhead and generate the expanded ciphertext in a simpler way. Overall performance is better than existing protocols.

1. Introduction

Secure Multi-party Computation (SMC), a method proposed by Yao [1], can securely compute a function without disclosing its data. Each party can get its results but has no access to others’ data. SMC now consists of several constructors, including verifiable secret sharing [2], oblivious transfer [3], mix and match [4] and homomorphic encryption [5–7]. Devised by Rivest et al. [8] in 1978, Fully Homomorphic Encryption (FHE) supports direct encryption of the plaintext, and the same computation can be done on the ciphertext to get decryption results, namely, \( f(Enc(m)) = Enc(f(m)) \). Due to this unique feature, FHE has displayed great potential among all constructors of SMC protocols, and thus attracted greater attention over time. Since the launch of a FHE scheme by Gentry [9] in 2009, a wealth of similar schemes emerged, including DGHV10 [10], BV11 [11], BGV12 [12], Bra12 [13], GSW13 [14], BV14 [15], CM15 [5], and NK15 [16]. Meanwhile, scholars in China and abroad also substantially investigated FHE-based SMC protocols.

In 2012, López-Alt et al. [17] put forward the concept of multi-key fully homomorphic encryption (MFHE) for the first time, and made use of NTRU [18] to construct the first
MFHE scheme. Therefore, an MFHE-based SMC protocol could be naturally created. Later, a lot of MFHE-based SMC protocols were constantly designed and improved. In 2016, Mukherjee et al. [6] established a two-round SMC protocol that achieved the best two-round interactions based on the GSW scheme, and proved that this scheme was secure in the malicious environment. The protocol needed to choose a common random string (CRS) matrix during the generation of a key, which undermined each user’s ability to generate their key independently. In the meantime, cascading and masking operations were added to the scheme, leading to a considerable volume of ciphertext matrix. Vijayakumar P et al. [19] proposed an efficient group key management technique that reduces the computational complexity without increasing the high storage complexity, thereby providing secure group communication in P2P networks. In the same year, a two-factor authentication scheme and a two-group key management scheme [20] were proposed to improve the security of vehicles communicating with a vehicle-mounted ad hoc network (VANET) environment. Audithan S et al. [21] proposed an anonymous authentication scheme to authenticate users, which is not easy to be maliciously accessed by attackers, and protects data transmitted in Internet business applications through mobile agents. In 2017, by increasing a new round of interaction and taking advantage of key homomorphism and threshold decryption, Wang Huiyong et al. [22] designed a simple three-round GSW-based SMC protocol with CRS, whose security was based on the Some-are-errorless LWE assumption, a variant of the LWE assumption. In this protocol, although a new round was added, the homomorphic computation depth and NAND gate complexity were reduced, and its overall efficiency was optimized compared to the MW16 scheme [6]. In 2018, Kim et al. [23] proposed the LinkAlgo algorithm to expand a single-key ciphertext into a multi-key one, and constructed for the first time a three-round SMC protocol without CRS that did not rely on a CRS matrix during the generation of a public key, allowing every user to generate their own public keys independently. This protocol met the multi-key CPA security requirements and could resist the attack of a semi-malicious adversary, but it was still slightly inferior to the protocol with CRS in terms of security and failed to prove that it was still secure in the malicious environment. In 2020, by employing the tool matrix and the encoding operations offered by Li Zengpeng [24] to improve the ciphertext expansion way of the KLP18 scheme [23], Tang Chunming et al. [25] built a three-round SMC protocol without CRS that outperformed the KLP18 scheme in efficiency, memory, and noise decryption, but it was only proven to be secure in the semi-malicious environment, as well. Dheerendra Mishra et al. [26] built a mutual authentication and key agreement scheme for mobile edge computing without the participation of trusted third parties, ensuring mutual authentication between users and edge servers and generating secure session keys. Vinoth R et al. [27] proposed a secure multi-factor authentication key agreement scheme for the Industrial Internet of Things (IIoT) to enable authorized users to remotely access sensing devices, effectively reducing communication during the authentication key agreement process and computational costs. In 2021, by referring to the multi-bit encryption scheme provided by Li Zengpeng, Tang Chunming et al. [28] pioneeringly developed a three-round SMC protocol with CRS that supports multi-bit encryption, whose security was based on the Ferr-LWE assumption and the Some-are-errorless LWE assumption, making it possible to resist the attack of an adversary in the malicious environment. In the same year, based on the multi-bit encryption scheme proposed by Li Zengpeng [24], Li Xixi et al. [29] modified his encryption algorithm and availed of the LinkAlgo algorithm to build a multi-bit multi-key FHE mechanism without CRS. Xia X et al. [30] proposed a cloud-assisted trustworthy assessment mechanism and an efficient anonymous authentication and key agreement scheme based on non-interactive zero-knowledge to ensure privacy protection and data security of IoT devices in smart cities.
It can be seen from the above work that current FHE-based SMC protocols have been equipped with or without CRS. An SMC protocol with CRS is found with higher security. It is also proven to be secure by using the non-commutative zero-knowledge proof in the malicious environment. Still, it needs to choose a CRS matrix during the parameter setting phase. All users are required to generate their keys with the help of this matrix, which dramatically limits the users’ ability to generate keys independently. In contrast, an SMC protocol without CRS supports all users to generate their keys independently by eliminating the need for a trusted organization to distribute the CRS matrix. Nevertheless, the existing SMC protocols without CRS are still generally plagued by overlarge ciphertext, colossal memory space and low efficiency.

1.1 Objectives

To solve the problems of low efficiency of existing protocols, this paper converted the scheme of Chen Li et al. [31] into a multi-bit FHE scheme and constructed a multi-key homomorphic encryption scheme using the LinkAlgo algorithm. Finally, a three-round SMC protocol without CRS was designed to outperform all existing protocols in ciphertext size and storage overhead.

2. Knowledge base

2.1 Description of symbols

In this study, bold lowercase letters represent vectors, while bold capital letters refer to matrices. \( \mathbb{Z} \), \( \mathbb{R} \), and \( \mathbb{Z}_q \) refer to the set of integers, the set of real numbers, and the residue class ring of the integer modulo \( q \). The length of the \( n \)-dimensional vector \( \mathbf{a} \) is defined as its Euclidean norm \( \| \mathbf{a} \| = \sqrt{\sum_{i=1}^{n} a_i^2} \); the length of the vector set \( S \) is defined as \( \| S \| = \max_{\mathbf{a} \in S} \| \mathbf{a} \| \).

2.2 Definitions and theorems

Definition 1 [32] (LWE distribution): Define a secret vector \( \mathbf{s} \in \mathbb{Z}_q^n \), take uniform sampling \( \mathbf{a} \leftarrow \mathbb{Z}_q^n \), and choose \( e \leftarrow \mathcal{N} \) wherein \( \mathcal{N} \) is a discrete Gaussian distribution on \( \mathbb{Z} \). The sample distribution \( \mathcal{N}_{\mathbf{a},\mathbf{s}} \) is outputted in the form of \( (\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e \mod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \).

Definition 2 (Searching LWE, abbreviated as SLWE): Define \( m \) independent samples \( (\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \) from a given LWE distribution \( \mathcal{N}_{\mathbf{a},\mathbf{s}} \) to output \( \mathbf{s} \).

Definition 3 (Decisional LWE, abbreviated as DLWE): For the security parameter \( \lambda \), define \( n = n(\lambda) \) and \( q = q(\lambda) \geq 2 \), wherein \( n \) and \( q \) are integers. A distribution \( \chi = \chi(\lambda) \) on \( \mathbb{Z} \) is defined. The \( \text{LWE}_{n,q,\chi} \) problem is utilized to distinguish between the following two distributions: (1) Uniformly choose \( (\mathbf{a}_i, b_i) \) from \( \mathbb{Z}_q^{n+1} \); (2) Uniformly choose \( s \leftarrow \mathbb{Z}_q \) and then take \( e_i \leftarrow \mathcal{N} \) for the uniform sampling \( \mathbf{a}_i \leftarrow \mathbb{Z}_q^n \). Supposing that \( b_i = \langle \mathbf{a}_i, s \rangle + e_i \), \( (\mathbf{a}_i, b_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \) is outputted. According to the \( \text{LWE}_{n,q,\chi} \) assumption, the \( \text{LWE}_{n,q,\chi} \) problem is hard to solve.
Definition 4 The adversary models in the secure multi-party computation:

a. Semi-honest model: All participants will strictly follow the protocol and not change the protocol or its data. However, the intermediate computation results may be maintained and used for computing the private data of other participants.

b. Semi-malicious model: This model can be seen as an interactive Turing machine with reference and proof tapes. A semi-malicious adversary is obliged to record any data of a certain participant represented by it into the proof tape at any time. The adversary may decide whether to honestly execute the original protocol based on the inputs at random.

c. Malicious model: All computational participants in this model can arbitrarily alter and disclose the protocol and its data, and even interfere with its normal performance.

Definition 5 [31] a and b are determined as the vectors along $\mathbb{Z}_q^k$, k is a positive integer; q is a modulo; p is the power of 2; $t = \lceil \log_q p \rceil$; $N = kt$. The following function is defined:

$$mbDpt(a) = a' = (a_{1,1}, \ldots, a_{1,t}, \ldots, a_{k,1}, \ldots, a_{k,t}) \in \mathbb{Z}_p^N \quad \quad (1)$$

wherein $a'$ is a N-dimensional vector $a_i = \sum_{j=1}^{t} a_{ij} p^{j-1}$; $a_{ij} \in \mathbb{Z}_p$.

$$mbDpt^{(-1)}(a') = (\sum_{i=1}^{t} p' \cdot a_{1,i}, \ldots, \sum_{i=1}^{t} p' \cdot a_{k,i}) \quad \quad (2)$$

$$mbFlatten(a') = mbDpt(mbDpt^{(-1)}(a')) \quad \quad (3)$$

$$pofmb(b) = (b_1, pb_1, \ldots, p'^{-1}b_1, \ldots, b_k, pb_k, \ldots, p'^{-1}b_k) \quad \quad (4)$$

Theorem 1 Supposing that $e_i (i \in \{N\})$ are a series of independent random variables subject to a certain bounded distribution $B_p$, the random variable $e = \frac{1}{N} \sum_{i=1}^{N} e_i$ is also subject to $B_p$.

Theorem 2 [33] For any $m \geq n[\log q]$, there is a matrix $G \in \mathbb{Z}_q^{n \times m}$ and its corresponding "short primary image" matrix function $G^{-1}()$ to achieve $G^{-1}(M) \in \{0,1\}^{m \times m'}$ and $GG^{-1}(M) = M$ for any matrix $M \in \mathbb{Z}_q^{m \times m'}$, wherein $m'$ can be any number.

Theorem 3 [31] In terms of the MGSW scheme proposed herein, $L$ is the maximum NAND gate depth of the circuit to be computed; in the absence of homomorphic computation, if $C$ is the ciphertext obtained by encrypting 0, when $|\langle C_{m-1}, s' \rangle| < q / (4p(pN + 1)^L)$, the scheme is correct.

Proof. By analyzing the correctness of homomorphic addition and multiplication, the noise is not greater than $pN + 1$ times that of the original ciphertext after each homomorphic computation. As a result, when $|\langle C_{m-1}, s' \rangle| < q / (4p(pN + 1)^L)$, $|\langle C_{m-1}, s' \rangle| < q / (4p)$ is achieved after no more than $L$ homomorphic computations. According to the decryption algorithm, if $|\langle C_{m-1}, s' \rangle| < q / (4p)$, $(s', C_{m-1}) / (q / 2p) < 1 / 2$ is realized. Consequently, if the encrypted information is 0, $\langle C_{m-1}, s' \rangle$ is closer to 0 than $q / (2p)$, and

$$\mu = \langle s', C_{m-1} \rangle / (q / 2p) + 1 / 2 \mod 2 < [1 / 2 + 1 / 2] \mod 2 = 0;$$

when the opposite happens, the correctness of the scheme can be guaranteed.
3. MGSW scheme that supports multi-bit encryption

The MGSW scheme constructed in this section is a modified FHE scheme that supports multi-bit encryption based on Literature [31], whose security is based on the DLWE assumption. Each user in Literature [31] relies on the CRS matrix $A$ to generate their public keys during the key generation process, undermining their ability to generate a public key independently. In contrast, our scheme is more advanced because the participants do not need any CRS matrix to generate public keys; instead, each of them can generate his/her public keys independently by randomly choosing a matrix $A$ from $\mathbb{Z}_q^{n \times n}$. The scheme is detailed as follows:

For the given modulo $q$ and the dimension $N$, the ciphertext $C$ is an $N \times N$ dimension matrix defined on $\mathbb{Z}_p$, and each matrix component is far less than $q$. The secret key $sk$ of $C$ is defined as an $N$-dimension vector along $\mathbb{Z}_q$. Supposing that the plaintext $\mu$ is a small integer, when $Csk = \mu sk+e$, $C$ is defined as the ciphertext of $\mu$ wherein $e$ is a small error vector. During the decryption process, first take the $i$th row $C_i$ of $C$, compute $x = (C_i sk) = \mu sk_i + e_i$, and output $\mu = \lfloor x/sk_i \rfloor$, wherein $sk_i$ is the $i$th element of $sk$, $e_i$ is the $i$th element of $e$, and $i \in [0,N-1]$. The information $\mu$ can be considered an eigenvalue of the ciphertext matrix $C$, while the secret key $sk$ is the approximate eigenvector of $C$ corresponding to the eigenvalue $\mu$.

**MGSW.Setup($\lambda$, $L$):** $\lambda$ is the security parameter; $L$ is the maximum NAND gate depth of the circuit; the lattice dimension $n = n(\lambda, L)$; for the modulo $q$, the error distribution $\chi = \chi(\lambda, L)$; $m = m(\lambda, L) = O(n\log q)$. By choosing appropriate parameters, the LWE assumption holds, thus outputting params $= (n, q, \chi, m)$.

**MGSW.Keygen(n, q):** For the positive integer $n$, take the depth of homomorphic computation as $l$, randomly and uniformly choose $A \in \mathbb{Z}_q^{n \times n}$ from $\mathbb{Z}_q^{n \times n}$, and take the sample $s$ from the discrete Gaussian distribution $\chi(\lambda)$ on $\mathbb{Z}^{n \times d}$. The public key $pk = (A, b = A \cdot s + e) \in \mathbb{Z}_q^{n \times m}$ is computed, and the secret key is $sk = (\frac{-s}{1}) \in \mathbb{Z}_q^{n+1}$.

**MGSW.Encrypt(pk, $\mu$):** For the plaintext $\mu \in \mathbb{Z}_p$, randomly and uniformly choose $r_i e_i \sim \chi(n)$, $e_{i+1} \sim \chi$ and $i = 1, \ldots, (n+1)\cdot t$, compute $C_{i,1} = A^T \cdot r_i + e_i \in \mathbb{Z}_q^n$ and $C_{i,2} = b^T \cdot r_i + e_{i+1} \in \mathbb{Z}_q$. Supposing that $e$ is a matrix composed of $m = (n+1)\cdot t$ ciphertexts arranged as column vectors, whose dimension is $(n+1)\times m$, thus outputting the ciphertext $C = \text{mbFlatten}(\mu \cdot I_N + \text{mbDpt}(C)) \in \mathbb{Z}_p^{m \times m}$.

**MGSW.Decrypt(sk, $C$):** For the ciphertext $C \in \mathbb{Z}_p^{m \times m}$ and the secret key $sk = (\frac{-s}{1}) \in \mathbb{Z}_q^{n+1}$, supposing that $s' = \text{pofmb}(sk)$, compute and output the plaintext $\mu = \lfloor s', C_{m,1} \rfloor / (q/2p) + \frac{1}{2} \bmod p$.

**MGSW.Add($C_1, C_2$):** Input the ciphertexts $C_1$ and $C_2$, and output the new ciphertext $C = \text{mbFlatten}(C_1 + C_2)$ after the homomorphic addition.

**MGSW.Mult($C_1, C_2$):** Input the ciphertexts $C_1$ and $C_2$, and output the new ciphertext $C = \text{mbFlatten}(C_1 \cdot C_2)$ after the homomorphic multiplication.

3.1 Correctness

During multi-bit encryption and decryption, in the absence of homomorphic computation, if the plaintext is $\mu'$, we can obtain $\mu = \lfloor (\mu', C_{m,1}) / (q/2p) + \frac{1}{2} \bmod p \rfloor = \lfloor \mu' + e/(q/2p) + \frac{1}{2} \rfloor \bmod p$ after the decryption according to the encryption and decryption processes; when $|e/(q/2p)| < 1/2$, $\mu = \mu'$, so it is correctly decrypted. For homomorphic addition, $C = \text{mbFlatten}((\mu_1 + \mu_2)I_N + \text{mbDmp}(C_1 + C_2))$, and MGSW.Dec($sk, C$) = $(\mu_1 + \mu_2) \bmod p$. For homomorphic multiplication, $C sk = \mu_1 \mu_2 sk + \mu_1 e_1 + C_1 e_2$, and MGSW.Dec($sk, C$) = $\mu_1 \mu_2$. Therefore, correct decryption is achieved.
After homomorphic multiplication, since the coefficient of both $\mu_3$ and $C_1$ are limited to $\mathbb{Z}_p$, the noise is not greater than $pN+p$ times that of the original ciphertext. Consequently, during the multi-bit encryption, the limitation exerted by Theorem 3 on the noise becomes $\langle C_{m-1}, s' \rangle < q/[4p(pN + p)^2]$. Considering $pN = pkt \gg p$, the influence of this change on the modulo $q$ can be ignored.

### 3.2 Security

**Theorem 4** Supposing that the parameters $n = \text{poly}(\lambda)$ and $q = \text{poly}(\lambda)$ constitute a polynomial with the security parameter $\lambda$, and that the Attacker can distinguish the ciphertext of the MGSW scheme and the uniform distribution on $\mathbb{Z}_p^{m \times n}$ with a non-ignorable advantage, the DLWE problem is also solved. Therefore, if it is supposed that this problem is hard to solve, the MGSW scheme can meet the plaintext’s (IND-CPA) security criteria.

**Proof.** The Theorem is proven by defining the following game sequences:

a) Game0.

 Initialization: The Challenger runs MGSW.Keygen($n,q$) to generate the public-secret key pair (pk, sk), and gives the public key to the Attacker $\hat{A}$.

 Step 1: The Attacker may encrypt the information $\mu \in \{0,1\}$ independently or through the Challenger. If the latter method is adopted, the Challenger needs to return the ciphertext accurately.

 Challenge: At a given time, the Attacker challenges the Challenger and sends the challenge plaintext $\mu_1, \mu_2 \in \{0,1\}$. By randomly choosing $b \in \{0,1\}$, the Challenger runs MGSW.Encrypt($pk, \mu$), computes and sends the challenge ciphertext $C$ to the Challenger.

 Step 2: Just as done in Step 1, the Attacker may encrypt the information $\mu \in \{0,1\}$ independently or through the Challenger.

 Guessing: The Attacker guesses the challenge plaintext chosen from the challenge step, and outputs $b' \in \{0,1\}$. Thus, Game0 is a classic IND-CPA attacking game where the Attacker’s advantage is marked as:

$$\text{Adv}_{\text{CPA}}(\hat{A}) = \left| \Pr[\hat{A}(pk, \text{MGSW.Encrypt}(pk, \mu_0)) = 1] - \Pr[\hat{A}(pk, \text{MGSW.Encrypt}(pk, \mu_1)) = 1] \right|$$  \hspace{1cm} (5)

b) Game1.

 In Game1, except for the initialization period, the Challenger follows all other steps as with Game0.

 Initialization: The Challenger randomly and uniformly chooses the public key $A \leftarrow \mathbb{Z}_q^{(n+1) \times m}$ and gives it to the Attacker $\hat{A}$.

 The Attacker’s advantage in Game1 is marked as $\text{Adv}_{\text{Game1}}(\hat{A})$. In Game1, the public key is no longer generated through the secret key. As the public key in Game0 can be seen as $n \times 1 \text{LWE}_{q,n,k}$ instances, the public key in Game1 is randomly chosen from the uniform distribution. Therefore, if the $\text{DLWE}_{q,n,k}$ problem can be solved via the non-ignorable advantage, Game0 and Game 1 can also be distinguished based on the same edge. If the assumption $\text{DLWE}_{q,n,k}$ holds, the advantage of the Attacker to distinguish between Game0 and Game1 is negligible, thus getting

$$\text{Adv}_{\text{CPA}}(\hat{A}) = \text{Adv}_{\text{Game1}}(\hat{A}) + \text{negl}(\lambda)$$  \hspace{1cm} (6)

c) In Game2, except for the challenge period, the Challenger follows all other steps as with Game1. At a given time, the Attacker challenges the Challenger and sends the challenge plaintext $\mu_1, \mu_2 \in \{0,1\}$. The Challenger randomly chooses $b \in \{0,1\}$ and $C \leftarrow \mathbb{Z}_q^{n+1}$, and sends the challenge ciphertext $C' = c + (a, \mu'$ $| q/2]$ to the Attacker. The advantage of the Attacker in Game2 is marked as $\text{Adv}_{\text{Game2}}(\hat{A})$. Likewise, $\text{Adv}_{\text{Game1}}(\hat{A}) \leq \text{Adv}_{\text{Game2}}(\hat{A}) + \text{negl}(\lambda)$. 


d) In Game3, except for the challenge period, the Challenger follows all other steps as with Game2. The Challenger sends the challenge ciphertext to the Attacker. The advantage of the Attacker in Game3 is marked as \( \text{Adv}_{\text{Game3}}(\hat{A}) \). In Game3, both the public key and the challenge ciphertext are taken from uniform distributions, and do not contain any information of the plaintext, so \( \text{Adv}_{\text{Game1}}(\hat{A}) = 0 \). Since \( C \) in both Game2 and Game3 is taken from the uniform distribution on \( \mathbb{Z}_q^{n+1} \), \( C' \) in Game2 and \( C \) in Game3 are statistically indistinguishable, namely, \( \text{Adv}_{\text{Game2}}(\hat{A}) \leq \text{Adv}_{\text{Game3}}(\hat{A}) + \negl(\lambda) \).

As \( \text{Adv}_{\text{Game3}}(\hat{A}) = 0 \), if the assumption \( \text{DLWE}_{q,n,2n+1} \) holds, \( \text{Adv}_{\text{CPA}}(\hat{A}) = \negl(\lambda) \), and the MGSW scheme meets the IND-CPA security criteria.

### 4. Modified LinkAlgo algorithm

It is often the case that the construction of a multi-key FHE scheme relies on the homomorphic computation of the ciphertexts under different keys, but the MGSW scheme proposed herein can just generate the multi-bit single-key ciphertext. Consequently, in our scheme, the LinkAlgo algorithm [23] is adopted for expanding the multi-bit single-key ciphertext into the multi-bit multi-key ciphertext. As mentioned in the KLP18 scheme [23], the expansion of multi-key ciphertext involves complicated steps, leading to low efficiency, high memory space and loud decryption noise. Therefore, the complicated ciphertext expansion way is optimized herein to get simpler expanded ciphertext, further increasing the efficiency. The modified LinkAlgo algorithm is detailed as follows:

For the matrix \( R \in \{0,1\}^{m\times n} \), \( V^{(s,t)} \) is the \( \beta \) noise ciphertext of \( R^{(s,t)} \) encrypted with the GSW encryption algorithm under \( (pk, sk) = (K, \hat{t}) \). Supposing that \( (pk', sk') = (K', \hat{t}') \) is another pair of keys, by inputting \( pk' \) and all \( R^{(s,t)} \) into the modified LinkAlgo algorithm, \( Y \) is outputted to achieve \( tY = tK^eR + e \) and \( \|e\|_\infty \leq m^2\beta \) (\( e \) is the noise), thus outputting the optimized expanded ciphertext \( \hat{C} \).

**Modified LinkAlgo Algorithm**

**Input:** \( pk' \) and \( \{R^{(s,t)}\}_{s, t \in [m]} \)

**Output:** \( \{V^{(s,t)}\}_{s, t \in [m]} = \{\text{GSW.Enc}(R^{(s,t)}, pk)\}_{s, t \in [m]} \)

1. Define \( L_{st}[a, b] = \begin{cases} K'[a, s]; & t = b \\ 0; & \text{other} \end{cases} \)

2. Output \( Y = \sum_{s=1}^{m} \sum_{t=1}^{m} V^{(s,t)}G^{-1}(L_{st}) \in \mathbb{Z}_q^{m\times m} \)

Output:

\[
\hat{C}_i = \begin{bmatrix}
C_i - Y_i^1 & 0 & \cdots & 0 & 0 \\
0 & C_i - Y_i^2 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
Y_i^1 & \cdots & C_i & \cdots & Y_i^\ell \\
0 & \cdots & \cdots & \cdots & 0
\end{bmatrix}
\]
Now, by defining \( tY = tKR + e \), the following detailed processes are presented to prove that \( \|e\|_\infty \leq m^3 \beta \) holds:

\[
\begin{align*}
tY &= \sum_{s,t} tV^{(s,t)}G^{-1}(L_{s,t}) \\
&= \sum_{s,t} (tR^{(s,t)}G + e_{s,t})G^{-1}(L_{s,t}) \\
&= \sum_{s,t} (tR^{(s,t)}L_{s,t} + e'_{s,t}) \\
&= t \sum_{s,t} R^{(s,t)}L_{s,t} + \sum_{s,t} e'_{s,t} \\
\end{align*}
\]

wherein \( e'_s = e_sG^{-1}(L_{s}) \) and \( \|e'_s\| \leq m\beta \).

Now, it is time to prove \( \sum_{s=1}^{m} \sum_{t=1}^{m} R^{(s,t)}L_{s,t} = KR \)

\[
\sum_{s=1}^{m} \sum_{t=1}^{m} R^{(s,t)}L_{s,t} = \sum_{s=1}^{m} \sum_{t=1}^{m} \begin{pmatrix} 0 & \cdots & R^{(s,t)}K^{(1,s)} & \cdots & 0 \\ \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \\ 0 & \cdots & \sum_{s=1}^{m} R^{(s,t)}K^{(n,s)} & \cdots & 0 \end{pmatrix}
\]

\[
= \sum_{t=1}^{m} \begin{pmatrix} 0 & \cdots & \sum_{s=1}^{m} R^{(s,t)}K^{(1,s)} & \cdots & 0 \\ \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \\ 0 & \cdots & \sum_{s=1}^{m} R^{(s,t)}K^{(n,s)} & \cdots & 0 \end{pmatrix}
\]

\[
= \sum_{t=1}^{m} \begin{pmatrix} 0 & \cdots & \langle K_{1}^{row}, R_{1}^{col} \rangle & \cdots & 0 \\ \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \\ 0 & \cdots & \langle K_{m}^{row}, R_{m}^{col} \rangle & \cdots & 0 \end{pmatrix} = KR
\]

Therefore, \( tY = tKR + e \), wherein \( e = \sum_{s=1}^{m} \sum_{t=1}^{m} e'_{s,t} \) and \( \|e\|_\infty \leq m^3 \beta \).
5. Multi-key Fully Homomorphic Encryption (MFHE)

In this section, based on the above MGSW scheme that supports multi-bit encryption, the modified LinkAlgo algorithm is adopted to expand the multi-bit single-key ciphertext to the multi-bit multi-key ciphertext and construct an MFHE scheme. To be specific, \( G \) and \( G^{-1}(\cdot) \) are the same as those described in Theorem 2; \( G \) is expanded into \( \hat{G}_r = \text{diag}(G, \cdots G) \in \mathbb{Z}_{q^{mN}}^{nN\times mN} \), and \( \hat{G}_r^{-1}(\cdot) \) is its corresponding function. This scheme is composed by the polynomial algorithm \( \text{MFHE} = (\text{Setup}, \text{Keygen}, \text{Enc}, \text{Expand}, \text{Eval}, \text{Dec}) \) for a series of probability events, as detailed below:

- **MFHE.Setup(\( 1^\lambda, 1^\ell \))**
  1. Run MGSW.Setup(\( 1^\lambda, 1^\ell \))
  2. Output \( \text{params} = (n, q, \chi, m) \)

- **MFHE.Keygen(\( \text{params} \))**
  1. Run MGSW.Keygen(\( n, q \))
  2. Output \( (pk, sk) = (K, \tilde{t}) \)

- **MFHE.Enc(\( pk, \mu \))**
  1. Run MGSW.Encrypt(\( pk, \mu \))
  2. Output \( C = \text{mbFlatten}(\mu I_{m} + \text{mbDpt}(C)) \)

- **MFHE.Expand((\( pk_1, pk_2, \cdots, pk_t \), i, C) \rightarrow (\hat{C}_i) \)**
  After the modified LinkAlgo algorithm is adopted to input the key and new ciphertext, the expanded ciphertext is outputted as:

  \[
  \hat{C}_i = \begin{bmatrix}
  C_i - Y_i^1 & 0 & \cdots & 0 & 0 \\
  0 & C_i - Y_i^2 & \cdots & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  Y_i^1 & \cdots & C_i & \cdots & Y_i^t \\
  \vdots & \vdots & \vdots & \vdots & \vdots \\
  0 & 0 & \cdots & 0 & C_i - Y_i^t
  \end{bmatrix}
  \]

- **MFHE.Eval(\( \text{params}, f, \hat{C}_1, \cdots, \hat{C}_l \)) \rightarrow (\hat{C}^*) \**
  1. Present \( l \) expanded ciphertexts \( \hat{C}_i (i = 1, \cdots, l) \) and the matrix functions \( \hat{G}_r^{-1}(\cdot) \) corresponding to \( \hat{G}_r = \text{diag}(\hat{G}, \cdots, \hat{G}) \)
  2. Output \( \hat{C}^* \)

- **MFHE.Dec(\( \text{params}, (\tilde{t}_1, \cdots, \tilde{t}_{N_r}), \hat{C}_n \)) \rightarrow (\mu) \**
  1. Present the key \( \tilde{t} = (\tilde{t}_1, \cdots, \tilde{t}_{N_r}) \) and an expanded ciphertext \( \hat{C}_n \), and define \( \tilde{w} = \underbrace{0, \cdots, 0}_{m\ell}, \underbrace{[q/2], \cdots, [q/2]}_{\ell} \in \mathbb{Z}_q^{(m+1)\ell} \)
  2. Run MGSW.Decrypt(\( sk, C \))
  3. Output \( \mu \)
Threshold decryption can be implemented on the above-expanded ciphertext $C_i$, as detailed below:

1. Present an expanded ciphertext $\hat{C}$ and the $i$th key $i_j \in \mathbb{Z}_{q_i}^{(m+1) \times r}$, and divide the ciphertext $\hat{C}$ into $N$-row submatrices $\hat{C}_i (\hat{C} = (\hat{C}_1^T, \cdots, \hat{C}_N^T))$ wherein $\hat{C}_i^T \in \mathbb{Z}_{q_i}^{(m+1) \times N}$. Define $\hat{w} = \left[ \begin{array}{c} 0, \ldots, 0, \frac{q}{2}, \ldots, \frac{q}{2} \end{array} \right] \in \mathbb{Z}_{q_i}^{(m+1) \times r}$.
2. Compute $\gamma_i = \hat{i} \hat{C} G^{-1} (\hat{w}^T)$
3. Output $p_i = \gamma_i + e_i$, wherein $e_i$ is a small noise.

MFHE.FinDec($p_1, \cdots, p_t, \mu$)

1. Input $p_1, \cdots, p_t$, and compute $p = \sum_{i=1}^{t} p_i$
2. Output $\mu = ||\frac{p}{q_i^2}||$.

5.1 Correctness of expanded ciphertext

$\hat{C}_i \leftarrow$ MFHE.Expand $((p\hat{k}_1, p\hat{k}_2, \cdots, p\hat{k}_t), (i, C))$ can be obtained by expanding the ciphertext of the $i$th user, wherein $C$ is the multi-key ciphertext of the plaintext $\mu$ after being encrypted by the MGSW scheme under $(K, \hat{i})$. By defining the multi-key $\hat{i} = (i_1, \cdots, i_N)$ and the public matrix $\hat{G}$, if $\hat{C}$ satisfies $i \hat{C}_i = \mu \hat{t} \hat{G}$, wherein $M_i = \text{diag}(M_1, \cdots, M_t)$, it is natural for us to promote the MGSW scheme. Next, the correctness of the expanded ciphertext will be proven:

\[ i \hat{C}_i = [t_1 (C - Y_1^t) + t_2 X_1, t_2 (C - Y_2^t) + t_2 Y_1, \cdots, t_N (C - Y_N^t) + t_N Y_1] \]

For all $j \neq i$, $t_j C - t_j Y_1^t + t_j X_1 = t_j K M_j = \mu t_j G - t_j K M_j - e_j + t_j K M_j + e_j = \mu t_j G + \tilde{e}_j$.

Therefore, $i \hat{C}_i = \mu \hat{t} \hat{G}_N + \hat{e}$, wherein $\hat{e} = [\tilde{e}_1, \cdots, \tilde{e}_2, \cdots, \tilde{e}_N]$.

To be specific, $\tilde{e}_j = -e_j + e_j + t_j K M_j \leq B_x + mB_x + B_x = (m + 2)B_x$, and $e_j = t_j C - t_j G$, and $\|e\|_{\infty} = mB_x$.

Therefore, $\|e\|_{\infty} \leq (m+2)B_x$. To get it correctly decrypted, the condition $(m+2)B_x \leq q/(4mN)$ shall be met. An appropriate parameter $q$ can be chosen to achieve the result that $(m+2)B_x \leq q/(4mN)$ holds.

5.2 Security of the proposed scheme

a) First of all, the scheme encryption resembles that of the above MGSW scheme constructed. It can be known from the security of the MGSW scheme that this encryption process meets the IND-CPA security criteria.

b) Security of the expanded ciphertext. The expanded ciphertext is obtained from the Lin-Kalgo algorithm where $Y$ is generated by $V^{(t)}$ and $G^{-1}(L_{x,t})$, as $G^{-1}(L_{x,t})$ means the bit decomposition of $L_{x,t}$ and $V^{(t)}$ refers to the encryption of $K^{(t)}$. It can be seen from Literature [13] that $V^{(t)} G^{-1}(L_{x,t})$ and the matrix uniformly chosen from $\mathbb{Z}_q^{(m+1) \times N}$ are indistinguishable in terms of computation, so this expanded ciphertext is secure.

Therefore, the proposed scheme herein is proven to be secure.
6. Three-round Secure Multi-party Computation (SMC) protocol

This section utilizes the above MFHE scheme to construct a three-round SMC protocol without CRS that supports multi-bit encryption. Although the best two-round interactions have been realized in the protocol with CRS in Literature [6], an SMC protocol without CRS requires at least three rounds to complete the entire protocol: As no CRS matrix is used in the protocol, each participant generates his/her key pair independently, and distributes the key to compute and release the partially decrypted result. As a result, an SMC protocol with CRS requires at least three rounds.

Next, we will prove that the SMC protocol constructed herein is secure in the semi-malicious environment; that is to say, the protocol is secure when facing a semi-malicious adversary, who is weaker than the malicious adversary but more robust than the semi-honest adversary.

6.1. Security of the protocol

Next, we will prove that the SMC protocol constructed herein is secure in the semi-malicious and semi-malicious models, as detailed below:

**Preprocessing:** Set up the parameters (params) ← MFHE.Setup(1^λ, 1^λ) to make sure that all participants share the parameter settings.

**Input:** For i∈[N], each participant P_i inputs the private data x_i∈{0,1} to compute the function f((0,1)^N→(0,1)), wherein L is the circuit depth of f.

**Round 1.** For each participant P_i, the following steps are carried out.
- Generate (pk_i, sk_i) ← MFHE.Keygen(params).
- Release the public key {pk_i}_{i∈[N]}.

**Round 2.** Each participant P_i receives the public key {pk_i}_{i∈[N]} \{i\} from others and follows the steps below.
- Use the public key pk_i to encrypt the plaintext m' and get the ciphertext C ← MFHE.Enc(pk_i, m').
- Run the expanded algorithm to get the expanded ciphertext C_i ← MFHE.Expand((pk_i, pk_2, ..., pk_n), i, C), and then release the expanded ciphertext C_i.

**Round 3.** Each participant P_i receives the ciphertext C_i from others and follows the steps below.
- Output the assessed ciphertext C_i ← MFHE.Eval(params, f, C_1, ..., C_n) after homomorphic computation.
- Threshold encryption is implemented. Get the partially decrypted result p_i ← MFHE.PartDec(C_i, i) and then release p_i.

**Output:** Each participant P_i receives the partially decrypted result {p_j}_{j≠i} from others, and computes the final decryption result μ ← MFHE.FinDec(p_1, ..., p_n).

6.1. Security of the protocol

Next, we will prove that the SMC protocol constructed herein is secure in the semi-malicious environment; that is to say, the protocol is secure when facing a semi-malicious adversary, who is weaker than the malicious adversary but more robust than the semi-honest adversary.

a) First, a PPT simulator S is designed for a semi-malicious adversary who has captured N−1 users. The semi-malicious adversary in the static state is marked as A. Assuming that P_h is the only surviving honest participant. On behalf of P_h, Simulator S follows the steps below.

In the second round, Simulator S replaces the real input of the honest participant P_h with 0 for encryption. Subsequently, Simulator S obtains the inputs and secret keys of N−1 captured participants from "the proof tape." S sends these inputs into an ideal machine to get the output y and the ciphertext C after the homomorphic computation. Then, S computes and simulates the partially decrypted p_h ← S(y, C, h, {sk_i}_{i∈[N]} \{h\}) for P_h and discloses the partially decrypted result simulated in the third round to replace the real decryption.

By using a series of hybrid attacking games REAL_{π, A, Z}, HYB_{π, A, Z}, and IDEAL_{F, S, Z}, it is proven that the real and simulation results are indistinguishable. The definitions and proving
means of these games resemble those in Literature [6], so it is concluded that
REAL_{r,A,Z}^statstatHYB_{r,A,Z} and HYB_{r,A,Z}compcompIDEAL_{r,A,Z}. More proving details can be
found in Literature [6]. It is finally proven that the real and simulation computations are indistinguishable, namely, IDEAL_{r,A,Z}compcompREAL_{r,A,Z}.

b) Assuming that the adversary has corrupted multiple honest participants, just like Literature [6], a pseudorandom function can be adopted to prove that the protocol proposed herein is secure.

Therefore, the protocol is secure in a semi-malicious environment. The proving process ends.

7. Comparisons and analyses of protocol performance

Compared to the KLP18 scheme [23] that supports single-bit encryption only, the proposed protocol can support multi-bit encryption. Assuming that the information to be encrypted is in B-bit, the KLP18 scheme needs to be encrypted for B times, but only one encryption is required by adopting our protocol.

In comparison with Literature [29], the scheme in this study needs to be improved from two aspects:

a) Ciphertext size: Both the basic MGSW scheme proposed herein and the scheme offered in Literature [29] support multi-bit encryption. However, the former changes how the GSW scheme is implemented, requiring smaller ciphertext size and achieving \((n+1)^2[\log q]^2\); the latter demands a ciphertext size of \((n+1)^2[\log q]^2\). For B-bit encryption and decryption, the ciphertext sizes in Literature [10] and this study are \((n+B)^2[\log q]^2\) and \((n+B)^2[\log q]^2\), respectively.

b) Storage overhead: Although the GSW scheme is adopted as the basic scheme for both this study and Literature [29], the former’s ciphertext size is much smaller than the latter, suggesting that our protocol dramatically reduces the ciphertext size. In this way, the protocol proposed in this study occupies a smaller memory space but offers higher overall efficiency than in Literature [25].

The comparisons of protocol performance are detailed in Table 1, where “basic” refers to the basic scheme adopted in the protocol; “multi-bit” suggests whether the scheme supports multi-bit encryption; “Storage” is the Storage overhead; n is the lattice dimension; q is the modulo; p means the power of 2; B is the bit quantity inputted.

8. Conclusion

In this study, the key generation algorithm in the FHE scheme offered by Chen Li et al. was modified to construct the MGSW scheme so that the participants do not need to rely on the CRS matrix to generate their keys. Further, the MGSW scheme and the LinkAlgo algorithm were adopted for achieving the MFHE scheme. Finally, a three-round interactive SMC protocol without CRS that supports multi-bit encryption was designed using the MFHE scheme. Its security was based on the DLWE assumption, and it was proven secure in the semi-malicious environment.
model. The protocol proposed herein outperforms the existing ones in terms of the support for multi-bit encryption, ciphertext size and storage overhead, and functions more efficiently as a whole.

Yet, all existing FHE-based SMC protocols without CRS were proven to be secure in the semi-malicious environment only, but cannot resist the attack of a malicious adversary [34]. How to construct an SMC protocol without CRS in a malicious environment remains to be solved, indicating our future research direction.

Author Contributions
Writing – original draft: Zong-Wu Zhu.
Writing – review & editing: Ru-Wei Huang.

References
1. Yao A C. Protocols for secure computations[C]//23rd annual symposium on foundations of computer science (sfcs 1982). IEEE, 1982: 160–164.
2. Gennaro R, Rabin M O, Rabin T. Simplified VSS and fast-track multi-party computations with applications to threshold cryptography[C]//Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing, 1998: 101–111.
3. Goldreich O. Secure multi-party computation[J]. Manuscript. Preliminary version, 1998, 78.
4. Jakobsson M, Juels A. Mix and match: Secure function evaluation via ciphertexts[C]//International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2000: 162–177.
5. Clear M, McGoldrick C. Multi-identity and multi-key leveled FHE from learning with errors[C]//Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2015: 630–656.
6. Mukherjee P, Wichs D. Two round multi-party computations via multi-key FHE[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2016: 735–763.
7. Brakerski Z, Halevi S, Polychroniadou A. Four-round secure computation without setup[C]//Theory of Cryptography Conference. Springer, Cham, 2017: 645–677.
8. Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms[J]. Foundations of secure computation, 1978, 4(11): 169–180.
9. Gentry C. Fully homomorphic encryption using ideal lattices[C]//Proceedings of the forty-first annual ACM symposium on computing theory. 2009: 169–178.
10. Van Dijk M, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers[C]//Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 2010: 24–43.
11. Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE[J]. SIAM Journal on Computing, 2014, 43(2): 831–871.
12. Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping[J]. ACM Transactions on Computation Theory (TOCT), 2014, 6(3): 1–36.
13. Brakerski Z. Fully homomorphic encryption without modulus switching from classical GapSVP[C]//Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2012: 868–886.
14. Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[C]//Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2013: 75–92.
15. Brakerski Z, Vaikuntanathan V. Lattice-based FHE as secure as PKE[C]//Proceedings of the 5th conference on Innovations in theoretical computer science. 2014: 1–12.
16. Nuida K, Kurosawa K. (Batch) fully homomorphic encryption over integers for non-binary message spaces[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2015: 537–555. https://doi.org/10.1186/1471-2105-16-S18-S6
17. López-Alt A, Tromer E, Vaikuntanathan V. On-the-fly multi-party computation on the cloud via multi-key fully homomorphic encryption[C]//Proceedings of the forty-fourth annual ACM symposium on theory of computing. 2012: 1219–1234.
18. Hoffstein Jeffrey, Pipher Jill, and Silverman Joseph H. Ntru: A ring-based public key cryptosystem. In Buhler Joe, editor, ANTS, volume 1423 of Lecture Notes in Computer Science, pages 267–288. Springer, 1998. https://doi.org/10.1378/chest.114.2_supplement.133s PMID: 9725800

19. Vijayakumar P, Naresh R, Jegatha Deborah L, et al. An efficient group key agreement protocol for secure P2P communication[J]. Security and Communication Networks, 2016, 9(17): 3952–3965.

20. Vijayakumar P, Azees M, Kannan A, et al. Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4):1–14.

21. Audithan S, Murunya TS, Vijayakumar P. Anonymous Authentication for Secure Mobile Agent Based Internet Business[J]. Circuits & Systems, 2016, 07(8):1421–1429.

22. Huiyong Wang, Yong Feng, Lngzhong Zhao, et al. A secure multi-Party computation protocol based on multi-key homomorphic technique [J]. Journal of South China University of Technology (Natural Science Edition),2017, 45(07):69–76.

23. Kim E, Lee H S, Park J. Towards round-optimal secure multi-party computations: Multikey FHE without a CRS[C]//Australasian Conference on Information Security and Privacy. Springer, Cham, 2018: 101–113.

24. Zengpeng Li. Lattice-based fully homomorphic encryption and its applications [D]. Journal of Harbin Engineering University.

25. Chunming Tang, Yezhou Hu, Xixi Li. Secure multi-party computation without CRS based on the multi-party fully homomorphic encryption scheme [J]. Journal of Cryptologic Research, 2021, 8(02): 273–281.

26. A D M, DD B, Py C, et al. A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party—ScienceDirect[J]. Journal of Information Security and Applications, 55.

27. Vinoth R, Deborah LJ, Vijayakumar P, et al. Secure Multi-factor Authenticated Key Agreement Scheme for Industrial IoT[J]. IEEE Internet of Things Journal, 2020, PP(99):1–1. 

28. Chunming Tang, Yezhou Hu. Secure multi-party computation based on multi-bit fully homomorphic encryption [J]. Chinese Journal of Computers, 2021, 44(04): 836–845.

29. Xixi Li, Chunming Tang, Yezhou Hu. Multi-bit fully homomorphic encryption mechanism supporting multi-bit encryption [J/OL]. Journal of Cryptologic Research: 1–11[2021-10-18].

30. Xia X, Ji S, Vijayakumar P, et al. An efficient anonymous authentication and key agreement scheme with privacy-preserving for smart cities[J]. International Journal of Distributed Sensor Networks, 2021, 17(6):155014772110268.

31. Li Chen, Yang Zhou, Ran Duan. The design of a fully homomorphic encryption mechanism supporting multi-bit encryption[J]. Application Research of Computers, 2021, 38(02): 579–583.

32. Regev O. On lattices, learning with errors, random linear codes, and cryptography[J]. Journal of the ACM (JACM), 2009, 56(6): 1–40.

33. Micciancio D, Peikert C. Trapdoors for lattices: simpler, tighter, faster, smaller[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2012: 700–718.

34. Zhu Z-W, Huang R-W. Efficient SMC Protocol Based on Multi-Bit Fully Homomorphic Encryption. Applied Sciences. 2021; 11(21):10332. https://doi.org/10.3390/app112110332