Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Research article

Privacy preserving IoT-based crowd-sensing network with comparable homomorphic encryption and its application in combating COVID19

Daxin Huang a, Qingqing Gan b, Xiaoming Wang a,∗, Marek R. Ogiela c, Xu An Wang d,e,∗

a Department of Computer Science, Jinan University, Guangzhou, 510632, China
b Department of Cyber Security, Guangdong University of Foreign Studies, Guangzhou 510006, China
c AGH University of Science and Technology, 30 Mickiewicza Ave., 30-059 Krakow, Poland
d State Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China
e Engineering University of PAP, Xi’an, 710086, China

ARTICLE INFO

Keywords:
IoT based crowd-sensing network
Homomorphic encryption
Ciphertext comparison
Combatting COVID 19
Spatial distance

ABSTRACT

IoT-based crowd-sensing network, which aims to achieve data collection and task allocation to mobile users, become more and more popular in recent years. This data collected by IoT devices may be private and directly transmission of these data maybe incur privacy leakage. With the help of homomorphic encryption (HE), which supports the additive and/or multiplicative operations over the encrypted data, privacy preserving crowd-sensing network is now possible. Until now several such secure data aggregation schemes based on HE have been proposed. In many cases, ciphertext comparison is an important step for further secure data processing. However efficient ciphertext comparison is not supported by most such schemes. In this paper, aiming at enabling ciphertext comparison among multiple users in crowd-sensing network, with Lagrange's interpolation technique we propose comparable homomorphic encryption (CompHE) schemes. We also prove our schemes' security, and the performance analysis show our schemes are practical. We also discuss the applications of our IoT based crowd-sensing network with comparable homomorphic encryption for combatting COVID19, including the first example of privacy preserving close contact determination based on the spatial distance, and the second example of privacy preserving social distance controlling based on the spatial difference of lockdown zones, controlled zones and precautionary zones. From the analysis we see our IoT based crowd-sensing network can be used for contact tracing without worrying about the privacy leakage. Compared with the existing CompHE schemes, our proposals can be collusion resistance or secure in the semi-honest model while the previous schemes cannot achieve this easily. Our schemes only need 4 or 5 modular exponentiation when implementing the most important comparison algorithm, which are better than the existing closely related scheme with advantage of 50% or 37.5%.

* Corresponding authors.
E-mail addresses: knightdax@163.com (D. Huang), gan_qingqing@foxmail.com (Q. Gan), twxm@jnu.edu.cn (X. Wang), mogiela@agh.edu.pl (M.R. Ogiela), wangxazjd@163.com (X.A. Wang).

https://doi.org/10.1016/j.iot.2022.100625
Received 5 September 2022; Received in revised form 26 September 2022; Accepted 4 October 2022
Available online 8 October 2022
2542-6605/© 2022 Elsevier B.V. All rights reserved.
1. Introduction

1.1. Background

With the development of wireless sensor network, more and more mobile devices (such as smart phones, smart glasses, wearable devices) are able to be equipped with sensors. In order to collect the sensor data by distributed sensors from a crowd, crowd-sensing network has become popular in recent years. As a new type of sensing mode, crowd-sensing network allows data requesters to assign tasks for complex data collection to a large number of volunteering users, so that users can collaborate to complete the tasks. Compared with the traditional sensor network, IoT based crowd-sensing network, which can be seen from Fig. 1, can achieve tasks with more flexible and resource-saving, since data sensing is distributed to multiple mobile devices or other equipments [1].

However, data collection and transmission in crowd-sensing network may incur some security issues. Some malicious attackers or third-party service provider may obtain the sensitive sensory data for benefit, which will expose the users' privacy. In order to protect data privacy during data transmission, data encryption technology is one of the most commonly used methods in the crowd-sensing network, such as schemes in [2–7]. To reduce the communication overhead, many schemes (e.g., [8,9]) have been proposed based on homomorphic encryption, they can support data aggregation. Although these schemes can achieve privacy-preserving and encrypted data processing with the support of some basic operations, such as ciphertext addition, subtraction and multiplication and so on, but they cannot meet the requirement of data comparison operation in the ciphertext form. However, comparison operation over encrypted data enables the data center to process the collected data from users, such as ciphertext sorting, seeking the top-k value and data segmentation statistics. Thus comparison operations over ciphertexts among multiple users are a quite essential property for the crowd-sensing network.

To support data comparison, order-preserving encryption [10,11] and order-revealing encryption [12–15] have been proposed. The former one can obtain the data order by comparing their ciphertexts directly since the order is preserved after encryption, while the latter one requires a comparing algorithm to reveal the order from the ciphertexts. However, the above schemes only focus on ciphertext comparison and cannot support data homomorphism and data aggregation.

Recently, there are some schemes which can achieve both ciphertext comparison and data homomorphism, such as [16–20]. However these schemes are not aimed at the setting of crowd-sensing network. To the best of our knowledge, there exist very few related schemes in the literature focusing on efficient and privacy-preserving data comparison for the crowd-sensing network. Hence, it is necessary to study privacy-preserving and efficient encrypted data processing scheme with ciphertext comparison in the crowd-sensing network.

Nowadays how to combat COVID19 with information techniques is a very hot topic. Considering the rapidly spread of COVID19, how to prevent its fast spread while simultaneously protect people’s privacy is a very challenge problem. The technologies related with big data and cloud computing have been widely used in combating COVID-19, these technologies have provided strong technical and scientific support for epidemic prevention. However, the non-standard data collection and data management will cause great threaten to users’ privacy. In this paper, we discuss the potential application of our IoT based crowd-sensing network with comparable homomorphic encryption for combatting COVID19.

1.2. Contributions

The main contributions of this paper can be summarized as follows.1

- Based on Bresson et al.’s scheme [21], combined with Lagrange’s interpolation theorem, we construct privacy-preserving comparable homomorphic encryption schemes in the crowd-sensing network. The proposed schemes can not only achieve homomorphic encryption, but also support ciphertext comparison.
- The semantic security of the schemes are proved under the assumption of Partial Discrete Logarithm (PDL) and the Decisional Diffie–Hellman (DDH) assumption over $\mathbb{Z}_N^*$.  
- Compared with the existing homomorphic encryption schemes, the proposed schemes have considerable efficiency on computation and communication overhead.
- Finally we discuss the application of our IoT based crowd-sensing network with comparable homomorphic encryption for combatting COVID19, such as the first example of privacy preserving close contact determination based on the spatial distance, and the second example of privacy preserving social distance controlling based on the spatial difference of lockdown zones, controlled zones and precautionary zones.

---

1 Part of this work has been posted on Cryptology ePrint Archive as Report 2020/806, Cryptology ePrint Archive publishes pre-print papers relevant to the field of cryptology. But the authors still have the right to submit their work to journal.
2. Related works

Privacy-preserving techniques for the crowd-sensing network have been a hot topic in recent years. Focusing on data privacy protection in mobile sensing, Wang et al. [22] introduced a framework called ARTSense. Based on such a framework, two privacy-preserving solutions were proposed to maintain reputation and trust in mobile sensing. In [23], a privacy protection mechanism was built in the crowd-sensing network based on the combination of dynamic trust management with the distribution of key. Soon afterwards, Xiong et al. [9] proposed a privacy-preserving scheme for crowd-sensing network relied on data encryption and game theory. Then based on the additive homomorphic encryption, a secure and efficient data aggregation scheme was constructed to reduce communication overhead. Subsequently, several privacy-preserving solutions are proposed in the crowd-sensing network, such as [24,25]. Recently, some other researches over crowd-sensing network are proposed, including task allocation [26–28], incentive mechanism [29–31] and so on. In 2003, Bresson et al. [21] designed a homomorphic encryption with support for add operations, but their scheme cannot achieve ciphertext comparison.

Until now there are some schemes which can achieve both ciphertext comparison and data homomorphism, such as [16–20,32]. In 2009, Kerschbaum et al. [16] proposed the secure comparison protocols based on the homomorphic encryption scheme or the secret sharing scheme. Although these protocols are very novel, however the protocol on the homomorphic encryption scheme needs heavy encryption workload, while the protocol on the secret sharing technique requires heavy communication among parties. Furthermore their protocols are not aimed at the setting of IoT-edge-cloud computing. In 2017, based on homomorphic re-encryption scheme Ding et al. [18] proposed a novel privacy preserving data processing system, their proposal relies on the data service provider and the access control server to achieve basic operations on ciphertexts. Although this system can be used for solving many interesting practical problems, but it cannot resist collusion attack. In 2019, Zheng et al. proposed privacy preserving top-k query scheme, which is constructed on an encrypted data comparison protocol with a modified Paillier encryption. Although this scheme is very interesting but their encrypted data comparison protocol cannot resist the collusion attack between two cloud servers. In 2019 and 2020, Cheon et al. [33,34] proposed comparable homomorphic encryption (CompHE) schemes based on their approximate (fully) homomorphic encryption named CKKS, although these schemes are excellent on implementing comparison on encrypted numbers, but they are not easily adopted by IoT devices due to the complex lattice parameters and the setting up materials etc. In 2022, based on our paper’s preliminary version [35], Zhao et al. [20]\(^2\) proposed a homomorphic computing toolkit for predictive computation by extending the construction to usual homomorphic calculation and homomorphic function encryption etc.

---

\(^2\) We note the citation of [35] in [20] is with the last author.
but their construction is based on the Paillier encryption, while our scheme is based on Bresson et al.’s homomorphic encryption [21]. Furthermore, these two constructions have some flaws and in this paper we try to give some improvement.

We point out here all these schemes are not discussing the potential application of them for combating COVID19. Nowadays how to combat the COVID19 with advanced computer science techniques is a very challenge problem. Many scientists have given their efforts to tackle this problem. In 2020, Sobti et al. [36] have given their work on time series forecasting for Coronavirus(COVID-19). In that work, they discussed and predicted the spread trend of the COVID19 in India by using time series forecasting. Their predication is very accurate with only a percentage error of 7.57% and 12.85% for different models. This finding can help the government to effective manage the medical resources for combating COVID19. In 2021, Devi and Nayyar [37] discussed the visualization techniques for the COVID19, their techniques relied on unaided dimensional decrease methods, this is a wonderful work to convert crude COVID19 information into visual structure. In 2021, Al-Turjman et al. [38] organize a book named emerging visualization techniques for the COVID19, their techniques relied on unaided dimensional decrease methods, this is a wonderful work to convert crude COVID19 information into visual structure. In 2021, Al-Turjman et al. [38] organize a book named emerging visualization techniques for the COVID19, their techniques relied on unaided dimensional decrease methods, this is a wonderful work to convert crude COVID19 information into visual structure. In 2021, Al-Turjman et al. [38] organize a book named emerging visualization techniques for the COVID19, their techniques relied on unaided dimensional decrease methods, this is a wonderful work to convert crude COVID19 information into visual structure.

3. Organization

The remainder of this paper is organized as follows. Section 4 presents some concepts related in this paper and then Section 5 describes the framework of CompHE. Thereafter, concrete constructions of CompHE are proposed in Section 6. Security analysis and performance analysis are discussed in Sections 7 and 8. Section 9 discusses their application in combating COVID19. Finally, Section 10 concludes the paper.

4. Preliminaries

In this section, we first review some background knowledge, such as Lagrange’s interpolation theorem, complexity assumptions and homomorphic encryption. Then we briefly review Bresson et al.’s scheme and analyze the additive homomorphic property of their scheme.

4.1. Complexity assumptions

Suppose \( G \) be a cyclic group with prime order \( \text{ord}(G) \) and generator \( g \). Let \( p, q \) be two big primes and \( N = pq \). The complexity assumptions are defined as follows.

**Definition 4.1.** Partial Discrete Logarithm (PDL) Assumption Over \( \mathbb{Z}_N^* \) [40]: Given a triple \( (N, g, g^a \mod N^2) \) where \( a \) is a random value from \([1, \text{ord}(G)]\), any probabilistic polynomial-time (PPT) algorithm has only negligible probability in computing \( a \).

**Definition 4.2.** Decisional Diffie–Hellman (DDH) Assumption Over \( \mathbb{Z}_N^* \) [21]: Given a triple \( (N, g, g^a, g^\beta) \in G \) where \( a, \beta \) are a random value from \([1, \text{ord}(G)]\), any PPT algorithm has only negligible probability in computing \( g^{a\beta} \).

4.2. Homomorphic encryption

The concept of homomorphic encryption (HE) was first introduced in [41] by which the operations on ciphertexts equal to that on plaintexts. It can be classified as three types: additive HE [21], multiplicative HE [41] and fully HE [42].

- **additive HE:** If there exists an efficient algorithm for \( \forall m_1, m_2 \in M \), and the equation \( E(m_1 + m_2) = E(m_1) + E(m_2) \) is satisfied, then the algorithm can be called as additive HE algorithm.
- **multiplicative HE:** If there exists an efficient algorithm for \( \forall m_1, m_2 \in M \), and the equation \( E(m_1 \times m_2) = E(m_1) \times E(m_2) \) is satisfied, then the algorithm can be called as multiplicative HE algorithm.
- **fully HE:** If there exists an efficient algorithm that can satisfy both additive and multiplicative HE properties, then the algorithm can be called as fully HE algorithm.
4.3. Brief introduction to Bresson et al.’s scheme

Bresson et al. proposed a homomorphic encryption scheme called DT-PKC scheme. Here we only give a brief introduction of their scheme, more details can be seen in [21].

1. Setup($\lambda$) → ($sp$): takes a security parameter $\lambda$ as input, and chooses two big primes $p, q$ with length $k$, sets $N = pq$ and $g$ be the generator of the group $G$ with order $(p-1)(q-1)/2$. It outputs public parameters $sp = (p, q, N, g)$.

2. KeyGen($sp$) → ($pk, sk$): takes $sp$ as input, and randomly picks $\theta$, computes $h = g^\theta \mod N^2$. It outputs the public key $pk = h$ and secret key $sk = \theta$.

3. Enc($pk, m$) → $C$: takes as inputs the public key $pk$ and the message $m$, selects a random number $r$. It outputs the ciphertext $[m]_pk = (C_1, C_2)$ as $C_1 = h^r (1 + m N) \mod N^2, C_2 = g^r \mod N^2$.

4. Dec([$m]_pk, sk$) → $M$: takes as inputs the ciphertext $[m]_pk$ and the secret key $sk$. It outputs the message $m = \frac{(C_1/C_2^s)^{-1} \mod N^2}{N}$.

Given $m_1, m_2 \in Z_N$ encrypted under the same $pk$, we have the additive homomorphic property as $[m_1]_pk \cdot [m_2]_pk = (h^{1\cdot r_1} (1 + (m_1 + m_2) N) \mod N^2, g^{r_1 + r_2} \mod N^2) = [m_1 + m_2]_pk$.

5. Framework

This section introduces the framework of the proposed comparable homomorphic encryption (CompHE) scheme, including the system model, lemma definition, formal definition and security model [35].

5.1. System model

For crowd-sensing network, the architecture of the CompHE scheme contains four entities, which is demonstrated in Fig. 2.

- **Trust authority** (TA): as a fully-trust authority, it performs the system initialization and secret key distribution.
- **Data center** (DC): has substantial computational power. It executes the data calculation, such as data aggregation or data comparison. It is semi-trusted, which means it will perform the operation honestly but also curious about the information gathered from users.
- **Data request** (DR): is responsible to assign task request for data collection. It can usually be government organization, hospital and so on.
- **User**: is responsible to collect data and upload the gathering date to DC after encryption. We assume the users are semi-honest, and they do not collude with DC.

We here point out the system model can be extended to an extended one. In the extended system model, the user can be categorized as the users who are aided to help the DC to complete the Compare algorithm and the users who are responsible to collect data and upload the gathering date to DC after encryption. We denote the first kind of users as the Aided Users and the second kind of users as the Crowd-sourcing Users. In this way, only the first kind of users can be assigned the aided secret key for helping the DC to complete the Compare algorithm while the second kind of users need not be assigned any secret key. The advantage of this system model is the second kind of users needing not be assigned any secret key, in this way the crowd-sourcing users cannot collude with DC to obtain more information on data, such as decrypt other user’s encrypted data, and also the extended system model is more simple.
5.2. Lemma

In this subsection we give an important lemma, which is introduced and used by [20,35], but here we point out \( |M_1 - M_2| < N^2/2 \) should hold, which is not explicitly described in the original version.

**Lemma 5.1.** Suppose \( M_1, M_2 \) be two positive integers, \( |M_1 - M_2| < N^2/2 \) holds and \( N \) be a large positive integer. Let \( M = (M_1 - M_2) \mod N^2 \), if \( 0 < M < N^2/2 \) holds, then \( M_1 > M_2 \); else if \( N^2/2 < M < N^2 \) holds, then \( M_1 < M_2 \).

**Proof.** As we know, the property of modular operation determines that no matter what the relationship of \( M_1, M_2 \) is, the mod of their difference as

\[
M = (M_1 - M_2) \mod N^2.
\]

will always be positive integer. If \( M_1 = M_2 \), the modular result \( M = 0 \). Therefore, we only consider the case where \( M_1 \neq M_2 \). In order to determine the relationship between \( M_1 \) and \( M_2 \), we set \( |M_1 - M_2| < N^2/2 \). Since \( N \) is a large positive integer, the above range can satisfy the requirement in practice. Now we consider the following two cases.

**Case 1:** If \( 0 < (M_1 - M_2) < N^2/2 \), then \( 0 < (M_1 - M_2) < N^2/2 \). The Eq. (1) is equivalent to

\[
M = (M_1 - M_2).
\]

For \( 0 < (M_1 - M_2) < N^2/2 \), that is to say, \( M_1 > M_2 \). Therefore, from Eq. (2) and the condition \( 0 < (M_1 - M_2) < N^2/2 \), we obtain \( 0 < M < N^2/2 \).

**Case 2:** If \( -N^2/2 < M_1 - M_2 < 0 \), then \( N^2/2 < (M_1 - M_2 + N^2) < N^2 \). The Eq. (1) is equivalent to

\[
M = (M_1 - M_2 + N^2).
\]

For \( -N^2/2 < M_1 - M_2 < 0 \), that is to say, \( M_1 < M_2 \). Therefore, from Eq. (3) and the condition \( N^2/2 < (M_1 - M_2 + N^2) < N^2 \), we obtain \( N^2/2 < M < N^2 \).

To conclude, for \( M = (M_1 - M_2) \mod N^2 \), we have

\[
\begin{align*}
\begin{cases}
M_1 > M_2, & 0 < (M_1 - M_2) < N^2/2. \\
M_1 < M_2, & N^2/2 < (M_1 - M_2) < N^2.
\end{cases}
\end{align*}
\]

5.3. Lagrange's interpolation theorem

In this subsection we review of the Lagrange’s interpolation theorem, which are used by our constructions. Concretely it is the following:

For a polynomial function \( f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \) with order \( (n-1) \), if we choose \( n \) different points as \((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\), then it can construct a unique Lagrange’s polynomial function \( L(x) \) satisfies

\[
L(x) = \sum_{i=0}^{k-1} y_i \prod_{j=0, j \neq i}^{k-1} \frac{x - x_j}{x_i - x_j}
\]

and \( f(x) = L(x) \) holds, then we call \( L(x) \) as Lagrange's interpolation [43,44].

5.4. Formal definition

A CompHE scheme consists of five probabilistic polynomial-time (PPT) algorithms: Setup, Encrypt, Compare, Add, Decrypt.

1. **Setup(\(1^t, t) \rightarrow (\text{params}, pk, sk)\):** is run by TA and takes a security parameter \( 1^t \) and the number of users \( t \) as input, and outputs public parameters \( \text{params} \), the public key \( pk \) and the secret key \( sk \). We omit public parameters \( \text{params} \) in the following description.
2. **Encrypt(\(pk, m \rightarrow C\):** is run by the user and takes as inputs the public key \( pk \) and the message \( m \), and outputs the ciphertext \( C \).
3. **Compare(C_A, C_B) \rightarrow b:** involves an interaction between DC and user, takes as inputs two ciphertexts \( C_A, C_B \), and outputs a bit \( b \in \{0, 1\} \) as the comparison result.
4. **Add(C_i, C_j) \rightarrow C_{sum}:** is run by DC and takes as inputs \( C_i, C_j \), and outputs an aggregate result \( C_{sum} \).
5. **Decrypt(C, sk) \rightarrow M:** is run by DR and takes as inputs the ciphertext \( C \) and the secret key \( sk \), and outputs the message \( m \).
5.5. Security model

In the communication process, all the ciphertext comparison algorithms incur information leakage such as the relationship between the ciphertexts reveals the relationship of underlying plaintexts. To find a tradeoff between security and efficiency, it is quite essential to construct ciphertext comparison algorithms with as less leakage as possible. Therefore, we define the security of the proposed scheme based on the leakage function $[12,15]$. Assume $\mathcal{L}$ is quite essential to construct ciphertext comparison algorithms with as less leakage as possible. Therefore, we define the security model.

\[ \text{Setup} \]

\[ \text{Decrypt} \]

\[ \text{Add} \]

\[ \text{Compare} \]

\[ \text{Encrypt} \]

\[ \text{CompHE} \]

\[ \text{Adversary} \]

\[ \text{Security} \]

\[ \text{Semantic Security} \]

\[ \text{L-Semantic Secure} \]

\[ \text{Correctness} \]

\[ \text{Efficiency} \]

\[ \text{Implementation} \]

\[ \text{Conclusion} \]
3. Compare(C_A, C_B) \rightarrow b:

- When requiring to compare the U_A's ciphertext C_A with U_B's ciphertext C_B, DC first computes

\[ \tilde{C} = C_A / C_B, 1 \]

\[ \inf = \prod_{i=1}^{n-2} R_i^{x_i} \mod N^2 \]

Finally, DC sends comparison request Inf = \{ inf, x_A, g^{x_A}, \tilde{C} \} to U_B. Similarly, DC sends Inf_B = \{ inf, x_B, g^{x_B}, \tilde{C}_B \} to U_A. Note that \tilde{C}_A and \tilde{C}_B are used as a flag to distinguish the subtractor and the minuend.

- The user U_B receives the comparison request Inf_A, and chooses a random value k_B satisfies

\[ k_B \in \left[ 0, \frac{N}{\sqrt{2 \left( N^{\frac{5}{2}} + 1 \right)}} \right) \]

to blind the difference. Then U_B computes req_B as

\[ \tilde{C}_B = r_B \cdot \tilde{C} \cdot \left( g^{x_A} \right)^{-\Delta_B} \cdot (inf_B)^{x_B} \cdot \left( g^{x_B} \right)^{-\Delta_B} \cdot \left( \frac{x_A}{x_B} \right) \]

Finally, U_B sends req_B to DC. In the similar way, the user U_A also chooses k_A with the same range of k_B and generates req_A as

\[ \tilde{C}_A = r_A \cdot \tilde{C} \cdot \left( g^{x_B} \right)^{-\Delta_A} \cdot (inf_A)^{x_A} \cdot \left( g^{x_A} \right)^{-\Delta_A} \cdot \left( \frac{x_B}{x_A} \right) \]

- When receiving req_A and req_B, DC will compare the two ciphertexts C_A and C_B and obtain the result by computing

\[ M = \tilde{C} \cdot (req_A)^{-1} \cdot req_B \]

\[ = k_A k_B \left( 1 + (m_A - m_B) N \right) \mod N^2. \]  \hspace{1cm} (4)

Based on Lemma 5.1, we require that

\[ \left| k_A k_B \left( 1 + (m_A - m_B) N \right) \right| < \frac{N^2}{2}. \]  \hspace{1cm} (5)

Since m_A \in \left[ 0, N^{\frac{1}{2}} \right], then the range for (m_A - m_B)

\[ (m_A - m_B) \in \left[ -N^{\frac{1}{2}}, N^{\frac{1}{2}} \right]. \]

When solving Eq. (5), we get

\[ k_A k_B \leq \frac{N^2}{2 \left( N^{\frac{5}{2}} + 1 \right)} \leq \frac{N^2}{2 \left( N^{\frac{5}{2}} - 1 \right)}. \]

Finally, we have

\[ k_A k_B \leq \frac{N^2}{2 \left( N^{\frac{5}{2}} + 1 \right)}. \]

As k_A, k_B be positive integers, it satisfies

\[ 0 < k_A k_B < \frac{N}{\sqrt{2 \left( N^{\frac{5}{2}} + 1 \right)}}. \]

This is consistency with the range of k_A, k_B. Therefore, for

\[ k_A, k_B \in \left[ 0, \frac{N}{\sqrt{2 \left( N^{\frac{5}{2}} + 1 \right)}} \right), \]

\[ (m_A - m_B) \in \left[ -N^{\frac{1}{2}}, N^{\frac{1}{2}} \right] \]

we have the following conclusion:

\[ \begin{cases} 
  m_A \geq m_B, & 0 < M < N^2/2. \\
  m_A < m_B, & N^2/2 < M < N^2. 
\end{cases} \]

To sum up, if 0 < M < N^2/2, the algorithm will output 1; else if N^2/2 < M < N^2, output 0.
4. Add \((C_i, C_j) \rightarrow C_{\text{sum}}\): when receiving \(C_i\) and \(C_j\), DC executes the homomorphic add operation as 
\[
C_{\text{sum}} = (C_{\text{sum}_1}, C_{\text{sum}_2}) = \left\{
\begin{array}{l}
C_{\text{sum}_1} = g^{i(r_i r_j)} (1 + (m_i + m_j)) N \text{ mod } N^2 \\
C_{\text{sum}_2} = g^{r_i r_j} \text{ mod } N^2
\end{array}
\right..
\]

DC sends \(C_{\text{sum}}\) to DR.

5. Decrypt \((C, s_k) \rightarrow M\): After obtaining the aggregate ciphertext \(C_{\text{sum}}\), DR uses \(s_k = s\) to decrypt and get the aggregate message \(M\) as
\[
M = \left(\frac{C_{\text{sum}_1}}{C_{\text{sum}_2}} - 1\right) / N. \tag{6}
\]

6.2. Discussion on the first construction

Below we first discuss the construction's correctness. The correctness analysis includes two aspects: the correctness of the Decrypt and Compare. On one hand, the correctness of Decrypt can ensure using the corresponding secret key, DC can recover the correct plaintext from the ciphertext, shown by Eq. (6). Since we adopt DT-PKC as the encryption method, the correctness of Decrypt can be deduced to the correctness of DT-PKC scheme [21]. Therefore, we have omitted here. On the other hand, the correctness of the Compare can guarantee the correct relationship between two plaintexts by execute comparison operation over their ciphertexts, shown by Eq. (4).

At first glance, the correctness of Eq. (4) can be analyzed as follows.

\[
M = \tilde{C} \cdot (\text{req}_A)^{-1} \cdot \text{req}_B
\]

Since (we omit \(\text{mod } N^2\) here)
\[
\tilde{C} = g^{i(r_A r_B)} (1 + (m_A - m_B)) N
\]

\[
\text{req}_A = k_A^\omega_A b^\Delta_A \left(\frac{m_A}{m_A - m_B}\right) \cdot (\text{inf})^{r_A} \cdot (g^r)^{-\Delta_A} \left(\frac{m_A}{m_A - m_B}\right)
\]

\[
\text{req}_B = k_B^\omega_B b^\Delta_B \left(\frac{m_B}{m_B - m_A}\right) \cdot (\text{inf})^{r_B} \cdot (g^r)^{-\Delta_B} \left(\frac{m_B}{m_B - m_A}\right)
\]

\[
\Delta_i = f(x_i) \cdot \prod_{k=1}^{n-2} \frac{-x_k}{x_i - x_k}
\]

Let
\[
V = \sum_{i=1}^{n-2} f(w_i) \cdot \prod_{k=1, k \neq i}^{n-2} \frac{w_k}{w_i - w_k} \cdot \frac{-x_A}{w_i - x_A} \cdot \frac{-x_B}{w_i - x_B},
\]

then we have
\[
\text{inf} = \prod_{i=1}^{n-2} R_i^{\frac{-x_A}{x_i - x_A}}{\frac{-x_B}{x_i - x_B}} = e^V.
\]

So we get
\[
(\text{req}_A)^{-1} \cdot \text{req}_B = k_A^\omega_A b^\Delta_A \left(\frac{m_A}{m_A - m_B}\right) \cdot (\text{inf})^{r_A} \cdot (g^r)^{-\Delta_A} \left(\frac{m_A}{m_A - m_B}\right)
\]

\[
\cdot g^{(r_A - r_B)} \cdot (g^r)^{-\Delta_B} \left(\frac{m_B}{m_B - m_A}\right)
\]

\[
= k_A^\omega_A b^\Delta_A \left(\frac{m_A}{m_A - m_B}\right) \cdot (\text{inf})^{r_A} \cdot (g^r)^{-\Delta_B} \left(\frac{m_B}{m_B - m_A}\right)
\]

\[
= k_A^\omega_A b^\Delta_A (r_A - r_B)
\]

Finally, we can obtain
\[
M = \tilde{C} \cdot (\text{req}_A)^{-1} \cdot \text{req}_B
\]

\[
= k_A^\omega_A b^\Delta_A (r_A - r_B) \cdot (1 + (m_A - m_B)) N \cdot g^{i(r_A r_B)}
\]

\[
= k_A^\omega_A b^\Delta_A (1 + (m_A - m_B)) N \text{ mod } N^2
\]

Here we give our discussion on this construction:

- First, we point out here DC and the users cannot compute \(\frac{-x_A}{w_i - x_A} \text{ mod } ord(g), \frac{-x_B}{w_i - x_A} \text{ mod } ord(g), \frac{-x_A}{x_B - x_A} \text{ mod } ord(g), \frac{-x_A}{x_A - x_A} \text{ mod } ord(g), \frac{-x_A}{x_B - x_A} \text{ mod } ord(g), \frac{-x_A}{x_A - x_A} \text{ mod } ord(g)\) directly, due to \(ord(g)\) is unknown to DC and the users. Thus the algorithm \(\text{Compare}(C_A, C_B) \rightarrow b\) cannot be easily implemented. We note this problem also exists in [20]. This problem can be solved by...
using TA to generates these values and secure send them to the DC or users, as the below second construction demonstrating. This variant of the first construction can be easily obtained by following the second construction, and later in the section of security proof and analysis, we shall also prove this variant’s security.

- Second, we also note in this construction, when implementing the algorithm Compare($C_A, C_B$) → $b$, the users $A$ and $B$ need to remember the randomness $r_A, r_B$ used for generation of ciphertexts $C_A, C_B$, which is not practical for the users. If the user $A$ generates many ciphertexts such as 10,000 ciphertexts, he needs to remember 10,000 different randomness $r$, which is not practical.

- Third in this construction, there are rounds of interaction between DC and the users $U_A$ and $U_B$, such as, DC sends comparison request $Inf_A = \{inf, x_A, g^{r_A}, \omega_A = 1\}$ to $U_B$, DC also sends $Inf_B = \{inf, x_B, g^{r_B}, \omega_B = -1\}$ to $U_A$, and users $A$ and $B$ also sends $req_A$ and $req_B$ to DC. We assume DC and the users $A$ and $B$ are semi-honest, that is, they will follow the steps of the algorithms. If they can behave arbitrarily, DC can send the users $A$ and $B$ maliciously constructed $Inf_A$ and $Inf_B$ and maybe recover $m_A, m_B$, or the malicious user $U_A$ and $U_B$ maybe recover other user's encrypted data or other partial information. There maybe exist verifiable mechanisms to improve these rounds of interaction to be secure in the malicious setting, we left this as the open problem.

- Finally in this construction, any two users such as $A$ and $B$ cannot collude with DC to decrypt $C$, $D$ and other users’ ciphertexts such as $C_C$, $C_D$. But in the below second construction, if all the other users are assigned secret keys by TA as $A$ and $B$'s secret key, then any two users such as $A$ and $B$ can collude with DC to decrypt $C$, $D$ and other users’ ciphertexts such as $C_C$, $C_D$. So in the below second construction, for all the other users such as $C$ and $D$ the TA does not assign any secret keys to them.

### 6.3. Our second construction

Based on the above discussion, based on the extended system model described in subsection System Model, we here give our second construction as the following:

1. Setup($i, t$) → $(params, pk, sk)$:
   - TA inputs the security parameter $i, t$, generates $\lambda$-bit prime number $p, q$, and computes $N = pq$. Then TA chooses a group with order $(p - 1)(q - 1)/2$ and generator $g$. TA selects a random number $s \in Z_{N^2}^\ast$, $a_i \in Z_{N^2}^\ast$ ($i = 1, 2, \ldots, n - 1$) and constructs a polynomial function $f(x)$ with order $(n - 1)$ as follows.
     $$f(x) = s + a_1 x + \cdots + a_n x^{n - 1} \bmod ord(g).$$
   - TA sets $pk_i = g^s \bmod N^2$ as DR’s public key and then sends $sk_i = s$ to DR via secret channel.
   - For $t$ users, TA will choose $(n - 2 + t)(n > t)$ points from $f(x)$. Assume the first $t$ points as the users’ anonymous identities are included in a set $\{x_i\}_{i=1,\ldots,t}$, so the remaining points are gathered in the set $\{w_i\}_{i=n-1,\ldots,n-2}$. TA generates the public parameters $params$ as
     $$params = \{N, g\}$$
   - and master secret key
     $$msk = \{s, p, q, f(x)\}$$
   - TA calculates $R_i = f\left(w_i\right) \prod_{k=1}^{n-2} \frac{-w_k}{w_i - w_k} \bmod ord(g)$ and sends the set $\{R_i\}_{i=1,\ldots,n-2}$ to DC.
   - Here we categorized the users as the Aided User and the Crowded-sourcing User.

   - For the Aided User $U_i$ (In practice for simple we set $U_i$ be $A$ and $B$ and $i = 1, 2$, that is, there are only two users $U_A$ and $U_B$), TA chooses values $x_A$ and $x_B$ from the set $\{x_i\}_{i=1,\ldots,t}$ and computes $f(x_A)$ and $f(x_B)$. Note that $U_A$ and $U_B$ will get a different value from TA. Finally, TA sends
     $$\Delta_A = f(x_A) \prod_{k=1}^{n-2} \frac{-w_k}{x_A - w_k} \cdot \frac{-x_B}{x_A - x_B} \bmod ord(g)$$
     $$\Gamma_A = \frac{-x_A}{w_i - x_A} \bmod ord(g)(1 \leq i \leq n - 2)$$
   to the user $U_A$ as the secret key $sk_A$ via a secret channel. And also TA sends
     $$\Delta_B = f(x_B) \prod_{k=1}^{n-2} \frac{-w_k}{x_B - w_k} \cdot \frac{-x_A}{x_B - x_A} \bmod ord(g)$$
     $$\Gamma_B = \frac{-x_B}{w_i - x_B} \bmod ord(g)(1 \leq i \leq n - 2)$$
   to the user $U_B$ as the secret key $sk_B$ via a secret channel.

   - For the other Crowded-sourcing Users, TA sends nothing to them. They just complete the below Encrypt algorithm for the outsourced data.
2. Encrypt($pk$, $m$) $\rightarrow$ C:
   - User $U_i$ encrypts data and uploads to DC. $U_i$ first picks a random number $r_i \in Z_N^*$, then calculates $C_i = (C_{i,1}, C_{i,2})$
     \[
     C_i = \begin{cases} 
     C_{i,1} = pk_i^{\omega_i} \left(1 + m_i N\right) \mod N^2 \\
     C_{i,2} = g^{\omega_i} \mod N^2 
     \end{cases}
     \]
   where $pk_i = g^i \mod N^2$ is DR’s public key. Finally, $U_i$ sends $(x_i, C_i)$ to DC.

3. Compare($C_A, C_B$) $\rightarrow$ b:
   - When requiring to compare the $U_A$’s ciphertext $C_A$ with $U_B$’s ciphertext $C_B$, DC first computes
     \[
     \tilde{C} = C_{A,1}/C_{B,1}
     \]
     \[
     \text{inf} = \prod_{i=1}^{n-2} (g^{r_i \cdot m_i})^{R_i} \mod N^2
     \]

   Finally, DC sends comparison request $Inf_A = \{\text{inf}, g^{x_A}, \omega_A = 1\}$ to $U_B$. Similarly, DC sends $Inf_B = \{\text{inf}, g^{x_B}, \omega_B = -1\}$ to $U_A$. Note that $\omega_A$ and $\omega_B$ are used as a flag to distinguish the subtractor and the minuend.
   - The user $U_B$ receives the comparison request $Inf_A$, and chooses a random value $k_B$ satisfies
     \[
     k_B \in \left(0, \frac{N}{\sqrt{2 \left(N \frac{2}{5} + 1\right)}} \right)
     \]
     to blind the difference. Then $U_B$ computes $req_B$ as
     \[
     T1 = k_B^\omega_A (g^{x_A})^{2b} \cdot \left((g^{x_A})^{-1}\right)^{A_B}
     \]
     \[
     T2 = (\text{inf})^{\frac{n \cdot x_A}{m \cdot s_B}}
     \]
     where $\frac{x_A}{n \cdot s_B}$ is from $sk_A$, and $(g^{x_A})^{-1} \mod N^2$ can be computed with high probability, for $g^{x_A}$ and $N^2$ are co-prime with high probability, otherwise $N^2$ will be factored which is impossible. Finally, $U_B$ sends $T1$ to DC, $U_B$ sends $T2$ to $U_A$.
   - In the similar way, the user $U_A$ also chooses $k_A$ with the same range of $k_B$ and generates $req_A$ as
     \[
     T1' = k_A^{\omega_B} (g^{x_B})^{2b} \cdot \left((g^{x_B})^{-1}\right)^{A_A}
     \]
     \[
     T2' = T2^{\frac{n \cdot x_B}{m \cdot s_B}}
     \]
     where $\frac{x_B}{n \cdot s_B}$ is from $sk_B$, and $(g^{x_B})^{-1} \mod N^2$ can be computed with high probability, for $g^{x_B}$ and $N^2$ are co-prime with high probability, otherwise $N^2$ will be factored which is impossible. Finally, $U_A$ sends $req_A$ to DC.
   - When receiving $req_A$ and $req_B$, DC will compare the two ciphertexts $C_A$ and $C_B$ and obtain the result by computing
     \[
     M = \tilde{C} \cdot (T1)^{-1} \cdot T1' \cdot T2'
     \]
     \[
     = k_A k_B (1 + (m_A - m_B) N) \mod N^2.
     \]

Based on Lemma 5.1, we require that
\[
|k_A k_B (1 + (m_A - m_B) N)| < \frac{N^2}{2}.
\]
Since $m_i \in [0, N \frac{2}{5}]$, then the range for $(m_A - m_B)$
\[
(m_A - m_B) \in \left[-N \frac{1}{5}, N \frac{1}{5}\right].
\]
Thus we can get
\[
k_A k_B \leq \frac{N^2}{2 \left(N \frac{2}{5} + 1\right)} \leq \frac{N^2}{2 \left(N \frac{1}{5} - 1\right)}.
\]
Finally, we have
\[
k_A k_B \leq \frac{N^2}{2 \left(N \frac{2}{5} + 1\right)}.
\]
As \( k_A, k_B \) be positive integers, it satisfies
\[
0 < k_A \cdot k_B < \frac{N}{\sqrt{2(N^2 + 1)}}.
\]

This is consistency with the range of \( k_A, k_B \). Therefore, for
\[
k_A, k_B \in \left(0, \frac{N}{\sqrt{2(N^2 + 1)}}\right).
\]
\[
(m_A - m_B) \in [-N^2, N^2]
\]
we have the following conclusion:
\[
\begin{cases}
  m_A \geq m_B, & 0 < M < N^2/2, \\
  m_A < m_B, & N^2/2 < M < N^2.
\end{cases}
\]

To sum up, if \( 0 < M < N^2/2 \), the algorithm will output 1; else if \( N^2/2 < M < N^2 \), output 0.

4. Add\((C_j, C_j) \rightarrow C_{sum}\): when receiving \( C_i \) and \( C_j \), DC executes the homomorphic add operation as \( C_{sum} = (C_{sum, 1}, C_{sum, 2}) = (C_{i1}, C_{j1}, C_{i2}, C_{j2}) \).
\[
\begin{align*}
C_{sum, 1} &= g^{(r_i + r_j) (1 + (m_i + m_j)) N} \\
C_{sum, 2} &= g^{(r_i + r_j)} N^2
\end{align*}
\]
DC sends \( C_{sum} \) to DR.

5. Decrypt\((C, sk_r) \rightarrow M\): After obtaining the aggregate ciphertext \( C_{sum} \), DR uses \( sk_r = s \) to decrypt and get the aggregate message \( M \) as
\[
M = \frac{(C_{sum, 1} - 1)}{N}.
\]

6.4. Discussion on the second construction

Here we first discuss the correctness of the Compare which can guarantee the correct relationship between two plaintexts by execute comparison operation over their ciphertexts. Hence, the correctness of Compare can be analyzed as follows.
\[
M = \hat{C} \cdot (T1)^{-1} \cdot T1' \cdot T2'
\]
Since (we omit mod \( N^2 \) here)
\[
\hat{C} = g^{(r_A - r_A)} (1 + (m_A - m_B) N)
\]
\[
(T1) = k_A g^{r_A - \Delta_A} \cdot (g^{r_A})^{-\Delta_A}
\]
\[
(T1') = k_B g^{r_B - \Delta_B} \cdot (g^{r_A})^{-\Delta_B}
\]
Let
\[
\begin{align*}
\nabla &= \sum_{i=1}^{n-2} f(w_i) \prod_{k=1, k \neq i}^{n-2} \frac{-w_k - w_i}{w_i - w_k} \cdot \frac{-x_A}{w_i - x_A} \cdot \frac{-x_B}{w_i - x_B},
\end{align*}
\]
then we have
\[
T2' = g^{\nabla (r_B - r_A)}.
\]
So we get
\[
\begin{align*}
(T1)^{-1} \cdot T1' \cdot T2' &= k_A k_B g^{(r_A - r_A) \Delta_A} \cdot T2' \\
&= k_A k_B g^{(r_A - r_A) (\Delta_A + \nabla + \Delta_B)} \\
&= k_A k_B (1 + (m_A - m_B) N) \cdot g^{(r_A - r_A)}
\end{align*}
\]
Finally, we can obtain
\[
M = \hat{C} \cdot (T1)^{-1} \cdot T1' \cdot T2' = k_A k_B g^{(r_A - r_A) \Delta_A} \cdot T2' = k_A k_B (1 + (m_A - m_B) N) \mod N^2
Thus, the correctness of the proposed scheme has been proved.

Here we give our discussion on this construction:

• Note in the second construction, \( U_B \) needs sending \( T2 \) to \( U_A \), that is, there is a round of interaction between \( U_B \) and \( U_A \), this can be avoid by publishing

\[
\Gamma_A = \frac{-x_A}{w_i - x_A} \mod \text{ord}(g)(1 \leq i \leq n-2)
\]

and

\[
\Gamma_B = \frac{-x_B}{w_i - x_B} \mod \text{ord}(g)(1 \leq i \leq n-2)
\]

In this way, the DC can compute \( T2 \) itself. And also this variant is secure due to \( x_A, x_B, w_i \) are all unknown to all users and DC but only the TA.

• Note in this construction, TA only assigns secret keys to users \( U_A \) and \( U_B \), and other users have not been assigned any secret key. \( U_A \) and \( U_B \) can be seen as the aided helper of DC for completing the Compare algorithm. For implementing Compare algorithm on other users’ ciphertexts, \( U_A \) and \( U_B \) and DC can also complete this via following the Compare algorithm due to the Compare algorithm now not needing the users to remember the randomness used for encryption.

• However, we point out in this construction, users \( U_A \) and \( U_B \) and DC can collude to derive the underlying plaintext \( m_A \) or \( m_B \), but this construction is secure in the extended system model which assuming the aided helper \( U_A \) and \( U_B \) is not colluding with DC. We leave how to design a construction simultaneously satisfying users needing not remember randomness used for encryption and being secure even if the aided helper \( U_A \) and \( U_B \) is colluding with DC as an important open problem.

• Finally we note in the second construction also there are rounds of interaction between DC and the users \( U_A \) and \( U_B \). We also assume DC and the users \( U_A \) and \( U_B \) are semi-honest, that is, they will follow the steps of the algorithms. If they can behave arbitrarily, DC can send the users \( U_A \) and \( U_B \) malicious constructed value and maybe recover \( m_A \), \( m_B \), or the malicious user \( U_A \) and \( U_B \) maybe recover other user’s encrypted data or other partial information. There maybe exists verifiable mechanisms to improve these rounds of interaction to be secure in the malicious setting, we left this also as the open problem.

7. Analysis

For this section, we give the analysis for the proposed CompHE schemes based on the security model in preliminaries. And then we show the proposed schemes can resist attacks and maintain privacy.

7.1. Security proof for the first construction and its variant

Based on the model in the section of preliminaries, we will prove that the probability for a PPT adversary \( A \) to break the semantic security of the first construction or its variant is negligible. Inspired by [45,46], the first proposed construction introduces a parameter called ciphertext comparison history \( ch(m) \). Note that \( ch(m) \) records all the comparison queries, containing \( inf, \text{req} \) and comparison result. Now we prove the security of the proposed scheme by following the theorem below.

**Theorem 7.1.** If the PDL and DDH problem over \( Z^*_N^2 \) are hard, and the leakage function \( L = (L_{\text{Setup}}, L_{\text{Encrypt}}, L_{\text{Compare}}, L_{\text{Add}}, L_{\text{Decrypt}}) \) is defined as

\[
\{ \begin{align*}
L_{\text{Setup}} &= \perp, \\
L_{\text{Encrypt}} &= \perp, \\
L_{\text{Compare}} &= ch(m) \\
L_{\text{Add}} &= \perp, \\
L_{\text{Decrypt}} &= \perp
\end{align*} \}
\]

Then the proposed CompHE is \( L \)-semantic secure.

**Proof.** In order to prove Theorem 7.1, we try to construct several indistinguishable games. Specifically, the first game is the real world game \( \text{REAL}_{A}^{\text{CompHE}}(\lambda) \) and the second game can be deduced to the PDL problem over \( Z^*_N^2 \). In the third game, the security is deduced to the semantic security of DT-PKC scheme, while in the fourth game, the security is based on the random number chosen in the CompHE scheme. Finally, the last game comes to the ideal world game \( \text{IDEAL}_{A}^{\text{CompHE}}(\lambda) \). By analyzing the indistinguishable distribution between each games, we can obtain an efficient simulator \( S \). Hence, we have \( \text{IDEAL}_{A}^{\text{CompHE}}(\lambda) \) executed by \( S \) cannot be distinguished from \( \text{REAL}_{A}^{\text{CompHE}}(\lambda) \) and the security proof can get through. The detailed process is shown as follows.

**Game \( G_0 \):** this game is defined as the real world game \( \text{REAL}_{A}^{\text{CompHE}}(\lambda) \), so we have

\[
\Pr \left[ \text{REAL}_{A}^{\text{CompHE}}(\lambda) = 1 \right] = \Pr \left[ G_0 = 1 \right]
\]

**Game \( G_1 \):** this game is almost the same as \( G_0 \), except the generation method of \( inf \). To be specific, during the ciphertext request process in \( G_0, \text{inf} \) can be obtained as

\[
\text{inf} = \prod_{i=1}^{n-2} R_i^{-\frac{x_A}{w_i - x_A} - \frac{x_B}{w_i - x_B}} \mod N^2.
\]
However, we introduce a table $T_{\inf}$ to store the entry $(x_A, x_B, \inf)$ in $G_1$. In Compare algorithm, when $\inf$ is required to be returned, the experiment will first check whether $T_{\inf}$ contains the entry $(x_A, x_B, \inf)$ or not. If does, $\inf$ will be returned directly; otherwise, a group element $\inf$ will be randomly chosen from $G$ as the returned result and store $(x_A, x_B, \inf)$ into $T_{\inf}$. Since the PDL problem over $Z_{N^2}$ is hard to be solved, we can define an efficient adversary $B_1$ so that

$$\Pr [G_1 = 1] - \Pr [G_0 = 1] \leq Adv^{PDL}_{G,B_1} (\lambda)$$

**Game $G_2$:** this game is similar to $G_2$, except the generation method for ciphertext. To be exact, the real message $m_i$ is encrypted in $G_1$ while a string $0^2$ is encrypted instead in $G_2$. Therefore, the form of the ciphertext in $G_2$ is shown as follows.

$$C_j = \begin{cases} C_{i,1} = g^{x_A} \left( 1 + 0^2 N \right) \mod N^2 \\ C_{i,2} = g^{x_B} \mod N^2 \end{cases}$$

The above ciphertext is output by the encryption algorithm in DT-PKC [21]. Since DT-PKC is proved to be semantic security (SS) based on the DDH assumption over $Z_{N^2}$. Suppose $\text{ploy}(\lambda)$ denote the times of encryption, there exists an efficient adversary $B_2$ such that

$$\Pr [G_2 = 1] - \Pr [G_1 = 1] \leq \text{ploy}(\lambda) \cdot Adv^{SS}_{DT-PKC,B_2} (\lambda)$$

**Game $G_3$:** this game is quite identical to the previous game $G_2$, except that the generation of req. In $G_2$, req is obtained by

$$\text{req}_A = k_A^{x_A} \cdot (\inf)_A \cdot (g^{x_A})^{-1} \cdot (g^{x_B})^{-\lambda}$$

$$\text{req}_B = k_B^{x_B} \cdot (\inf)_B \cdot (g^{x_B})^{-\lambda}$$

By calculating

$$M = C \cdot (\text{req}_A)^{-1} \cdot \text{req}_B = k_A k_B \left( 1 + (m_A - m_B) N \right) \mod N^2$$

we can get the comparison result from the range of $M$.

However, we introduce a global counter $\text{cnt}$ to record the times of comparison in $G_1$. Note that $\text{cnt}$ is defined in Setup and initialize as 0, then it will increase by 1 in Compare. Furthermore, req is randomly selected from group $G$ and $M$ is randomly chosen from $Z_{N^2}$. Since $k_A, k_B$ are random value, the game $G_3$ and $G_2$ is indistinguishable, that is

$$\Pr [G_3 = 1] = \Pr [G_2 = 1]$$

**Simulator $S$:** the simulator $S$ has only slightly difference from $G_1$ at the input of Compare. To be exact, simulator $S$ obtains the ciphertext comparison history $\text{ch}(m)$ from $L_{\text{Compare}}$, as the input of Compare algorithm. Since the above operation will not change the distribution of the whole algorithm, the simulator $S$ is indistinguishable from $G_1$ as

$$\Pr [\text{\text{IDEAL}}^{\text{CompHE}}_{A,S} (\lambda) = 1] = \Pr [G_3 = 1]$$

Combining $G_0, G_1, G_2, G_3$ and $S$, we can have the following conclusion:

$$\Pr [\text{\text{REAL}}^{\text{CompHE}}_{A,S} (\lambda) = 1] - \Pr [\text{\text{IDEAL}}^{\text{CompHE}}_{A,S} (\lambda) = 1] \leq Adv^{PDL}_{G,B_1} (\lambda) + \text{ploy}(\lambda) \cdot Adv^{SS}_{DT-PKC,B_2} (\lambda)$$

That is to say,

$$\Pr [\text{\text{REAL}}^{\text{CompHE}}_{A,S} (\lambda) = 1] - \Pr [\text{\text{IDEAL}}^{\text{CompHE}}_{A,S} (\lambda) = 1] \leq \text{negl}(\lambda)$$

Therefore, we have completed the proof of Theorem 7.1.

### 7.2 Security proof for the second construction and its variant

The security proof for the second construction and its variant is almost the same as the first construction due to the two construction almost share the same structure. We give the following theorem.

**Theorem 7.2.** If the PDL and DDH problem over $Z_{N^2}$ are hard, and the leakage function $L = (L_{\text{Setup}}, L_{\text{Encrypt}}, L_{\text{Compare}}, L_{\text{Add}}, L_{\text{Decrypt}})$ is defined as

$$\left\{ \begin{array}{ll} L_{\text{Setup}} = 1, L_{\text{Encrypt}} = 1, L_{\text{Compare}} = ch(m) \\ L_{\text{Add}} = 1, L_{\text{Decrypt}} = 1 \end{array} \right\}$$

Then the proposed CompHE is $L$-semantic secure.
Proof. In order to prove the above Theorem, we try to construct several indistinguishable games. Specifically, the first game is the real world game $\text{REAL}_{A}^{\text{CompHE}}(\lambda)$ and the second game can be deduced to the PDL problem over $Z_{N^2}^*$. In the third game, the security is deduced to the semantic security of DT-PKC scheme while in the fourth game, the security is based on the random number chosen in the CompHE scheme. Finally, the last game comes to the ideal world game $\text{IDEAL}_{A,S}^{\text{CompHE}}(\lambda)$. By analyzing the indistinguishable distribution between each games, we can obtain an efficient simulator $S$. Hence, we have $\text{IDEAL}_{A,S}^{\text{CompHE}}(\lambda)$ executed by $S$ cannot be distinguished from $\text{REAL}_{A}^{\text{CompHE}}(\lambda)$ and the security proof can get through. The detailed process is shown as follows.

Game $G_0$: this game is defined as the real world game $\text{REAL}_{A}^{\text{CompHE}}(\lambda)$, so we have

$$\Pr[\text{REAL}_{A}^{\text{CompHE}}(\lambda) = 1] = \Pr[G_0 = 1]$$

Game $G_1$: this game is almost the same as $G_0$, except the generation method of $inf$. To be specific, during the ciphertext request process in $G_0$, $inf$ can be obtained as

$$T2' = \prod_{i=1}^{n-2} g^{R_i - \frac{r}{i+1} - \frac{r}{i \cdot (b-r)} \cdot (g^{b-r} \cdot M^*)} \mod N^2.$$  

However, we introduce a table $T_{T2'}$ to store the entry $(x_A, x_B, T2')$ in $G_1$. In Compare algorithm, when $T2'$ is required to be returned, the experiment will first check whether $T_{T2'}$ contains the entry $(x_A, x_B, T2')$ or not. If $T2'$ will be returned directly; otherwise, a group element $T2'$ will be randomly chosen from $G$ as the returned result and store $(x_A, x_B, T2')$ into $T_{T2'}$. Since the PDL problem over $Z_{N^2}^*$ is hard to be solved, we can define an efficient adversary $B_1$ so that

$$\Pr[G_1 = 1] - \Pr[G_0 = 1] \leq Adv_{G,B_1}^{\text{PDL}}(\lambda).$$

Game $G_2$: this game is similar to $G_2$, except the generation method for ciphertext. To be exact, the real message $m_i$ is encrypted in $G_1$ while a string $0^t$ is encrypted instead in $G_2$. Therefore, the form of the ciphertext in $G_2$ is shown as follows.

$$C_i = \begin{cases} C_{i,1} = g^{\Delta_i} \cdot \left(1 + 0^t \cdot (g^{b-r}) \right) \mod N^2 \\ C_{i,2} = g^{\gamma} \mod N^2. \end{cases}$$

The above ciphertext is output by the encryption algorithm in DT-PKC [21]. Since DT-PKC is proved to be semantic security (SS) based on the DDH assumption over $Z_{N^2}^*$. Suppose $\text{poy}(\lambda)$ denote the times of encryption, there exists an efficient adversary $B_2$ such that

$$\Pr[G_2 = 1] - \Pr[G_1 = 1] \leq \text{poy}(\lambda) \cdot Adv_{DT-PKC,B_2}^{\text{SS}}(\lambda).$$

Game $G_3$: this game is quite identical to the previous game $G_2$, except that the generation of $T1,T1'$. In $G_2$, $T1,T1'$ is obtained by

$$(T1) = k_A g^{-\Delta} \cdot (g^{b-r})^{-\Delta}$$

$$T1' = k_B g^{-\Delta} \cdot (g^{b-r})^{-\Delta}.$$  

By calculating

$$M = \hat{C} \cdot (T1)^{-1} \cdot T1' \cdot T2'$$

$$= k_A k_B g^{(X_{A-B})} \cdot \left(1 + (m_A - m_B) \cdot N \right) \cdot g^{(b-r) \cdot \Delta}$$

$$= k_A k_B \left(1 + (m_A - m_B) \cdot N \right) \mod N^2.$$  

we can get the comparison result from the range of $M$.

However, we introduce a global counter $cnt$ to record the times of comparison in $G_3$. Note that $cnt$ is defined in $\text{Setup}$ and initialize as 0, then it will increase by 1 in $\text{Compare}$. Furthermore, $\text{req}$ is randomly selected from group $G$ and $M$ is randomly chosen from $Z_{N^2}^*$. Since $k_A, k_B$ are random value, the game $G_3$ and $G_2$ is indistinguishable, that is

$$\Pr[G_3 = 1] = \Pr[G_2 = 1].$$

Simulator $S$: the simulator $S$ has only slightly difference from $G_3$ at the input of $\text{Compare}$. To be exact, simulator $S$ obtains the ciphertext comparison history $\text{ch}(m)$ from $\mathcal{L}_{\text{Compare}}$, as the input of $\text{Compare}$ algorithm. Since the above operation will not change the distribution of the whole algorithm, the simulator $S$ is indistinguishable from $G_3$ as

$$\Pr[\text{IDEAL}_{A,S}^{\text{CompHE}}(\lambda) = 1] = \Pr[G_3 = 1].$$

Combining $G_0, G_1, G_2, G_3$ and $S$, we can have the following conclusion:

$$\Pr[\text{REAL}_{A}^{\text{CompHE}}(\lambda) = 1] - \Pr[\text{IDEAL}_{A,S}^{\text{CompHE}}(\lambda) = 1] \leq Adv_{G,B_1}^{\text{PDL}}(\lambda) + p\text{oy}(\lambda) \cdot Adv_{DT-PKC,B_2}^{\text{SS}}(\lambda).$$

That is to say,

$$\Pr[\text{REAL}_{A}^{\text{CompHE}}(\lambda) = 1] - \Pr[\text{IDEAL}_{A,S}^{\text{CompHE}}(\lambda) = 1] \leq \text{negl}(\lambda).$$

Therefore, we have completed the proof of Theorem 7.2.
7.3. More analysis

Besides security proof, we will analyze the security of the CompHE schemes from two aspects: collusion attacks resistance and privacy protection on ciphertext difference.

We first distinguish the property of **Collusion attacks resistance**.

(1) In the first proposed CompHE scheme and its variant, we define a polynomial function \( f(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \) and choose \( (n-2+t) \) points from \( f(x) \). Assume the first \( t \) points as the users' anonymous identity set \( \{ x_i \} \in [1, ..., t] \), the remaining points as the set \( \{ w_i \}_{i \in [1, ..., n-2]} \), TA calculates and sends the set \( \{ R_i \}_{i \in [1, ..., n-2]} \) to DC, where

\[
R_i = g^{f(w_i)\prod_{k=1}^{n-2} \frac{w_k}{w_i - w_k}} \mod ord(g)(1 \leq i \leq n - 2)
\]

Based on the Lagrange's interpolation theorem, it requires \( n \) points from \( f(x) \) to recover the secret parameter \( s = f(0) \). Since we define \( n > t \), where \( t \) denotes the number of users, users' collusion attack cannot recover \( f(x) \). Therefore, our scheme can resist users' collusion attack.

When DC receives \( \{ R_i \}_{i \in [1, ..., n-2]} \), \( f(w_i) \) is embedded at the exponent. Since the PDL problem over \( \mathbb{Z}^{*}_{N^2} \) is hard to solve, DC cannot obtain \( f(w_i) \). Based on this, DC cannot collude with users to get \( s = f(0) \). Hence, the proposed scheme can resist collusion attack from DC and users.

Considering the case where a malicious user colludes with DC, the malicious user may send out various comparison requests with the target user and can obtain the comparison results from DC. Then the malicious user may guess out the underlying value of the target user. However, since \( x_i \) is an anonymous identity label, the real identity will not be exposed, thus the malicious user and DC cannot know the real identity of the target user.

To sum up, the first proposed CompHE scheme can resist collusion attack from users and collusion attack from DC and users.

(2) In the second proposed CompHE scheme and its variant, the security analysis of collusion attacks resistance is almost the same as the first proposed CompHE scheme but with the below difference:

\[
R_i = f(w_i)\prod_{k=1}^{n-2} \frac{w_k}{w_i - w_k} \mod ord(g)(1 \leq i \leq n - 2)
\]

and \( \{ x_i \}_{i \in [1, ..., t]} \), \( \{ w_i \}_{i \in [1, ..., n-2]} \) are not the public parameters and known only to the TC. Due to \( ord(g) \) is unknown and \( \{ x_i \}_{i \in [1, ..., t]} \), \( \{ w_i \}_{i \in [1, ..., n-2]} \) are also unknown to the adversary, it cannot derive \( f(x) \) and thus the master secret key \( f(0) \) from the theory of linear algebra. Thus in this way, the second proposed scheme and its variant can resist collusion attack for deriving the master secret key from DC and users.

But in the second construction and its variant, we also note the users need not remember the randomness such as \( r_A \) and \( r_B \) used for encryption, this improvement also make it possible for \( A, B \) and DC has the ability to decrypt the ciphertexts of other users, but the second construction and its variant is secure in the extended system model introduced in the preliminaries, which do not allow the collusion between \( A, B \) and DC. We left how to design a construction simultaneously satisfying needing not remember the randomness for encryption and collusion resistance as the open problem.

We then discuss the security property of **Privacy protection on ciphertext difference**. The proposed first and second construction and their variant can determine the relationship of ciphertexts, as well as not leaking the ciphertext difference. This is achieved by the blindness in ciphertext difference. Specifically, DC obtains the comparison result

\[
M = k_Ak_B\left(1 + (m_A - m_B)N\right) \mod N^2
\]

where the random value \( k_A, k_B \) satisfies

\[
0 < k_Ak_B \leq \frac{N^2}{2(N^\frac{3}{2} + 1)}.
\]

That is to say, we can define \( k_Ak_B \) as a random value from the pseudo-random function \( F_q \) where

\[
q = \frac{N^2}{2(N^\frac{3}{2} + 1)}.
\]

As it can be seen that \( k_Ak_B \) is random, \( M \) cannot be distinguished thus \( (m_A - m_B) \) is not revealed to DC. Therefore, the proposed first and second CompHE scheme and their variants can realize the ciphertext comparison and the privacy of ciphertext difference.

8. Performance analysis

In this section, we analyze the performance of our proposed CompHE schemes by two parts: making comparison with several existing schemes, and showing experimental results.
second CompHE scheme and its variant are stateless, but they cannot resist the collusion attack between servers. In reality, two servers may collude with each other to recover the secret so we cannot define both schemes [18,19] can resist against collusion attack. Hence, our proposed first CompHE scheme and its variant assumes there is no collusion between two servers. And the scheme [18] also involves two servers: data service provider and access control server, and their scheme also assumes the secret key is divided into two parts and distributed to two servers, and the comparison is achieved by cooperation between two servers. As can be seen from Table 1, schemes [18,19,21] and our scheme all can protect data privacy and enable homomorphic encryption, ciphertext comparison and collusion resistance. The result is shown in Table 1.

As can be seen from Table 1, schemes [18,19,21] and our scheme all can protect data privacy and enable homomorphic encryption. However, the scheme [21] cannot support ciphertext comparison operation. Although the scheme [19] allows to compare ciphertext, their scheme introduces two servers and assumes there is no collusion between two semi-trust servers. To be specific, the secret key is divided into two parts and distributed to two servers, and the comparison is achieved by cooperation between both servers. And the scheme [18] also involves two servers: data service provider and access control server, and their scheme also assumes there is no collusion between two servers. In reality, two servers may collude with each other to recover the secret so we cannot define both schemes [18,19] can resist against collusion attack. Hence, our proposed first CompHE scheme and its variant is superior to schemes [18,19,21] on the functionality aspect, however our first CompHE scheme and its variant are stateful. Our second CompHE scheme and its variant are stateless, but they cannot resist the collusion attack between A, B and DC.

Then we compare the proposed schemes with schemes [18,19] in terms of the computation, communication overhead. Specifically, the comparison contains the computational cost for the Encrypt, Compare algorithm, the communication cost during the comparison process, as shown in Table 2. Note that $T_{exp}$ denotes the time cost for an exponentiation operation in $Z_{N^2}^*$. Since the time cost for the add and multiplication operation in $Z_{N^2}^*$ can be very low, we have neglected here.

Since all the four schemes adopt DT-PKC scheme for data encryption, the computation cost for Encrypt in all schemes are the same, requiring 2 exponentiation operations to perform data encryption (Table 2). Here we only consider that each user encrypt and upload the data once. For the Compare algorithm, the computation overhead includes all the operations by DC and users. Note that Inf in the first CompHE scheme can be pre-calculated and stored at the DC side before an comparison request happens. Therefore, scheme [18] costs 8 exponentiation operations and scheme [19] incurs 3 exponentiation operations, while the first proposed CompHE scheme has the computation cost between both schemes for one comparison operation. We can see that the computation overhead is acceptable for mobile users in crowd-sensing network.

Since all of the four schemes have the same cost for the encrypted data transmission, we only consider the communication cost for the ciphertext comparison process in Table 2. Note the communication includes the interaction between DC and users. Suppose $Z_{N^2}^*$ be $L(n)$-bit integer group. In our first proposed scheme, the ciphertext comparison process requires that DC sends $Inf$ to user $U_A$ and $U_B$, then $req_A$ and $req_B$ are returned to DC respectively. Therefore, the total communication cost during ciphertext comparison are $8L(n)$ bits. In our second proposed scheme, the total communication cost during ciphertext comparison are $6L(n)$ bits. For the scheme [18], the communication overhead are $12L(n)$ bits. While the ciphertext comparison process in the scheme [19] needs two servers to interact with each other. Specifically, server A sends ($c_1, c_2$) to server B and then the latter calculates and returns the result, thus the total communication cost are $2L(n)$ bits.

Above all, our CompHE schemes makes a tradeoff between the security and the efficiency in comparison with schemes [18,19].

## 8.1. Theoretical analysis

First, we make functionality comparison with schemes [18,19,21] including whether enables data privacy, homomorphic encryption, ciphertext comparison and collusion resistance. The result is shown in Table 1.

### Table 1

| Schemes | Data privacy | Hom encryption | Ciphertext comparison | Collusion resistance | Stateless |
|---------|--------------|----------------|-----------------------|----------------------|-----------|
| Scheme [21] | √ | √ | × | √ | √ |
| Scheme [18] | √ | √ | √ | – | √ |
| Scheme [19] | √ | √ | √ | – | √ |
| CompHE1 and Variant | √ | √ | √ | √ | – |
| CompHE2 and Variant | √ | √ | √ | – | √ |

## 8.2. Experimental evaluation

To show the efficiency of our CompHE schemes, we conduct an experiment comparing with existing works. The experiment is implemented under 64-bit Windows-7 operating system with an Intel Core i3 CPU @ 1.80 Hz and 6.00 GB RAM. We complete the programming in JAVA and record the average time of ciphertext comparison phase in the first proposed CompHE scheme and
schemes [18,19]. Note that for consistency in parameters, we choose to encrypt the same message in these three schemes. The experiment results are illustrated in Fig. 3.

From Fig. 3, we can see that the average time for the \texttt{Compare} algorithm in scheme [18] is larger than that in our CompHE scheme and scheme [19]. While with the increase in the length of security parameter, scheme [19] tends to be more efficient in ciphertext comparison. However, scheme [19] does not consider the collusion attacks resistance of two servers. Therefore, our first CompHE scheme has found a balance between security and efficiency.

9. Application in combating COVID19

Since year 2019 we saw a major global outbreak of COVID19, causing catastrophic damage to people’s lives and property. In the process of fighting the epidemic, people are often exploring information techniques for epidemic prevention, such as IoT, cloud computing and big data techniques, which make epidemic prevention more scientific and efficient.

For example in China everyone is associated with a health code, by assigning different colors to different persons, the government can easily manage the social activities of people. If some one is assigned with red health code, then this person is prohibiting to take any social event and should be isolated in special hospitals. And if he is assigned with yellow code, then this person is a potential contact for COVID19 and should be isolated in special hospitals or mobile cabin hospitals. While if some one is assigned with green code, then he is a health person and can take social events as usual. The wide application of “red, yellow and green” health codes play an important role in epidemic prevention and control in China, which make epidemic prevention be more accurate and efficient.

However, along with the development of epidemic and the strain changing, the epidemic prevention has become more complicated. The recent practice of epidemic prevention in China tells us the following rules:

- Only by accurately distinguishing persons as the close contacts, potential contacts, no contacts and associating each kind with different social control methods, the epidemic prevention can be effective.
- Also only by accurately distinguishing the social regions as the high, medium and low risk ones and associating each one with different social control methods, the epidemic prevention and control can be effective.
- Furthermore, if there is a burst of epidemic in one city, the city should be categorized as the lockdown zones, controlled zones and precautionary zones according to the spatial difference from the point of burst. For different zones we implement different control measures, only in this way we can do a good job of epidemic prevention and control without affecting social order and economic development.

But a fact we must note is that, in the epidemic prevention, the leaking of people’s privacy information should be given more attention. This privacy information maybe be used by the criminals, which will led to negative impact to the epidemic prevention. To solve this problem, we should not only rely on the law, but also should rely on fundamental techniques. There is a paradigm called “privacy computing” which can help to solve this problem. The most fundamental way is to encrypt important personal data before the processing, so even if criminals obtaining encrypted personal data they cannot get any useful information. Below we give two examples to demonstrate how our proposed CompHE schemes can be used for privacy preserving IoT-based crowd-sensing network for combating COVID19.

- The first example is the privacy preserving close contact determination based on the spatial distance. Contact tracing in China via the spatial distance determination from mobile phone’s signal is now a very common technique. If the government wants to figure out whether a person is the close contact, it can first set up a distance from the COVID19 infected person’s location such as 800 meters as the threshold for determining close contacts. If this person is smaller than 800 meters far from the COVID19 infected person, which can be measured by mobile phone service provider (which can also be the cloud service provider) by detecting the signal distance from this person’s mobile phone and the COVID19 infected person’s mobile phone, then this person is determined as a close contact. Otherwise this person is determined as a not close contact.
However, in this way people maybe worry about their privacy is not preserving. To privacy preserving implementing this method, government can take our privacy preserving IoT-based crowd-sensing framework with CompHE schemes. Concretely, this example must satisfy the below conditions:

1. Institutions which authorized by government can secure and efficient collect of personal location information, and with data mining of these information they can aid the government for formulating measures for COVID19 epidemic prevention. These institutions can be the Data Requestor in our privacy preserving IoT-based crowd-sensing framework.

2. People need to release their location information to the government for contact determination, but also worry about their personal location information is abused. By using our privacy preserving IoT-based crowd-sensing framework, they can encrypt their personal location information by mobile phone APP before outsourcing it to the cloud centers administrated by government. So government can be the TA, people can be the Users, the mobile phone service provider can be the Data Center or the Cloud Service Provider in our privacy preserving IoT-based crowd-sensing framework.

3. The cloud service provider can implement efficient storage and secure homomorphic comparison on the encrypted personal location information, and meet the computational requirements of homomorphic comparison. In our privacy preserving IoT-based crowd-sensing framework, the Data Center can complete this task well. Furthermore, there can exist aided helpers for helping the Data Center for implementing homomorphic comparison, these aided helpers can be the proxy delegated by the government in the mobile phone service provider. These aided helpers can be $U_A$ and $U_B$ in our privacy preserving IoT-based crowd-sensing framework.

In this example, the administrator for epidemic prevention can grasp the basic information of the user based on the distance between him and the effected COVID19 users, adjust the color of the user's health code according to the distance, and notify all the relevant close contacts to take epidemic control measures. As shown in Fig. 4, the users can be categorized as close contact, secondary close contact, and non-contacting user. This categorization is judged by the distance between the user and the COVID19 infected person. We can first set up two circles with the center point being the COVID19 infected person, the small circle is with radius $D_m$ and the large circle is with radius $D_l$. If the users lie in the small circle, that is, the distance between them and the infected one is less than $D_m$, then they will be determined as close contact. If the users lie in the large circle, that is, the distance between them and the infected one is less than $D_l$ but greater than $D_m$, then they will be determined as secondary close contact. If the users lie outside the large circle, that is, the distance between them and the infected one is greater than $D_m$, then they will be determined as non-contacting user. By using our privacy preserving IoT-based crowd-sensing framework, the users can first encrypt their location information by using mobile phone APP and then outsource the encrypted results to the cloud server, and the cloud server then implements homomorphic comparison with the help of aided users on these ciphertexts, and the cloud server can quickly determine whether a user is a close contact, secondary close contact, or non-contacting user without knowing the location information of the users. Thus it is privacy preserving.

• The second example is the privacy preserving social distance controlling based on the spatial difference. Now in China if there is a trend for COVID19 break in some city at the initialization stage, then the city will probably take some measures to control social activities. Now it is common for city to categorized the zones as the lockdown zones, controlled zones and precautionary zones due to the order of severity. For example, in January 2022, Xi'an city in China has suffered with a COVID19 break, to quickly control the COVID19 spread, Xi'an city quickly categorized the zones as the lockdown zones, controlled zones and precautionary zones. For example, as shown in Fig. 5, according to the number of COVID19 infected persons, the number of close contacts, the number of secondary contacts and their summarization, the zones in the city can be categorized as the lockdown zones, controlled zones and precautionary zones. For different zones, the social controlling measures are different. In the lockdown zones, people are prohibited to take any social events, they can only keep staying in home. In the controlled zones, people are prohibited to take any public social events, they can only keep staying in home and the residential area. In the precautionary zones, people can take some public social events but cannot travel outside this zone. Furthermore, the traffic between these zones are all shutdown. In this way, we can ensure the COVID19 spread is controlled with great effort and also the economy can be retained with the most extension. This method for epidemic prevention in China has achieved great success and now it is more and more common in our cities. However directly adopting this model maybe leak the privacy of users, such as some user do not want others know the location of his social activity. Thus in these cases adopting our privacy preserving IoT-based crowd-sensing framework is a good choice. Users only need to encrypt their location information to the cloud server, and the cloud server implement the homomorphic comparison with the help of aided users on these ciphertexts for determining whether the users are inside the lockdown zones, or the controlled zones or the precautionous zones, and thus determine whether they are following the rules for epidemic prevention.

10. Conclusion

In this paper, we introduced the framework of comparable homomorphic encryption (CompHE), enabling comparison operation over ciphertexts. Thereafter, we constructed novel CompHE schemes for crowd-sensing network, achieving data comparison with lightweight computation overhead on DC and user side. Based on the PDL and DDH over $\mathbb{Z}_p^*$, assumption, the proposed CompHE schemes were proved to be semantic secure against malicious attackers. Security analysis shows the CompHE schemes can resist collusion attacks and protect the privacy of the difference on ciphertext. Performance analysis showed the CompHE schemes with considerable efficiency comparing with related schemes. Our schemes only need 4 or 5 modular exponentiation when
implementing the most important comparison algorithm. Finally we discuss the application of our IoT based crowd-sensing network with comparable homomorphic encryption for combatting COVID19, such as the first example of privacy preserving close contact determination based on the spatial distance, and the second example of privacy preserving social distance controlling based on the spatial difference of lockdown zones, controlled zones and precautionary zones.

This paper only give very preliminary results on how to construct comparable homomorphic encryption scheme and its application for combating COVID19. There are many interesting open problems needed to be solved, here we list some of them:

- How to design a comparable homomorphic encryption scheme simultaneously satisfying users needing not remember randomness used for encryption and being secure even if the users are colluded with DC. On the one hand, this construction needs the data collector to encrypt the data freely, and later when implementing the comparison algorithm, this encrypter needs to contribute his aided computation part without remembering the randomness used for encryption. On the other hand, without remembering the randomness the user can give correct aided computation part to the DC, and the DC also cannot
collude with other user to correct recover the underlying message. We try to tackle this challenge problem, but some difficult problems need to be solved.

- How to design comparable homomorphic encryption schemes based on different mathematica problem is also a very interesting problem. Although there are existing some constructions on lattice [33,34], but they are relying on the approximate (fully) homomorphic encryption techniques and thus not easily understood by engineers. How to design clean comparable homomorphic encryption scheme without any interaction between DC and user, and also it does not rely on lattice is a valuable open problem. Furthermore, the existing constructions on lattice are also not very efficient, how to improve the efficiency of them is also needed for further research.

- How to combine comparable homomorphic encryption with other techniques on outsourced data for solving the security problems when combating COVID19 is also an interesting open problem. We think many interesting techniques such as homomorphic authenticators, cloud storage auditing, verifiable outsourced computing can be used. For example, the communication between the users and DC are not verifiable in our constructions, that is, our schemes are only secure in the semi-honest model, verifiable computing techniques can be used to achieve stronger security. Furthermore, there are many security problems when combating COVID19, we only try to solve some of them in some special settings, more cryptographic techniques can be used for solving these security problems in other settings, which are also our future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China under Grant no. 61932010. Research project is supported by program “Excellence initiative – research university” for the AGH University of Science and Technology. It is also supported by the Foundation of State Key Laboratory of Public Big Data (No. 2019BDKFJJ008).

References

[1] L. Wang, D. Zhang, Z. Yan, H. Xiong, B. Xie, EfFense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading, IEEE Trans. Syst. Man Cybern. Syst. 45 (12) (2015) 1549–1563.
[2] Z. Wang, D. Huang, Privacy-preserving mobile crowd sensing in ad hoc networks, Ad Hoc Netw. 73 (2018) 14–26.
[3] M. Dai, Z. Su, Q. Xu, Location privacy preservation scheme based incentive for crowd-sensing networks, in: 2019 IEEE/CIC International Conference on Communications in China, ICCC, IEEE, 2019, pp. 695–699.
[4] G. Sun, S. Sun, J. Sun, H. Yu, X. Du, M. Guizani, Security and privacy preservation in fog-based crowd sensing on the internet of vehicles, J. Netw. Comput. Appl. 134 (2019) 89–99.
[5] W. Zhang, G. Li, An efficient and secure data transmission mechanism for internet of vehicles considering privacy protection in fog computing environment, IEEE Access 8 (2020) 64461–64474.
[6] B. Zhao, S. Tang, X. Liu, X. Zhang, Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing, IEEE Trans. Mob. Comput. (2020).
[7] A. Alamer, S. Basudan, An efficient truthfulness privacy-preserving tendering framework for vehicular fog computing, Eng. Appl. Artif. Intell. 91 (2020) 103583.
[8] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, J. Hu, Appa: An anonymous and privacy preserving data aggregation scheme for fog-enhanced iot, J. Netw. Comput. Appl. 125 (2019) 82–92.
[9] J. Xiong, R. Ma, L. Chen, Y. Tian, Q. Li, X. Liu, Z. Yao, A personalized privacy protection framework for mobile crowdsensing in iiot, IEEE Trans. Ind. Inf. (2019).
[10] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, H. Shi, Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions, in: Annual International Cryptology Conference, Springer, 2005, pp. 205–222.
[11] R. Popa, F. Li, N. Zeldovich, An ideal-security protocol for order-preserving encoding, in: 2013 IEEE Symposium on Security and Privacy, IEEE, 2013, pp. 463–477.
[12] N. Chenette, K. Lewi, S. Weis, D. Wu, Practical order-revealing encryption with limited leakage, in: International Conference on Fast Software Encryption, Springer, 2016, pp. 474–493.
[13] X. Wang, Y. Zhao, Order-revealing encryption: file-injection attack and forward security, in: European Symposium on Research in Computer Security, Springer, 2018, pp. 101–121.
[14] D. Cash, F.-H. Liu, A. O’Neill, M. Zhandry, C. Zhang, Parameter-hiding order revealing encryption, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2018, pp. 181–210.
[15] Y. Li, H. Wang, Y. Zhao, Delegatable order-revealing encryption, in: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, 2019, pp. 134–147.
[16] F. Kerschbaum, D. Biewas, S. de Hoogh, Performance comparison of secure comparison protocols, in: 2009 20th International Workshop on Database and Expert Systems Application, IEEE, 2009, pp. 133–136.
[17] J. Furukawa, Short comparable encryption, in: International Conference on Cryptology and Network Security, Springer, 2014, pp. 337–352.
[18] W. Ding, Z. Yan, R. Deng, Encrypted data processing with homomorphic re-encryption, Inform. Sci. 409 (2017) 35–55.
[19] Y. Zheng, R. Lu, X. Yang, J. Shao, Achieving efficient and privacy-preserving top-k query over vertically distributed data sources, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–6.
[20] K. Zhao, X. Wang, B. Yang, Y. Tian, J. Zhang, A privacy preserving homomorphic computing toolkit for predictive computation, Inf. Process. Manage. 59 (2022) 102880.

[21] E. Bresson, D. Catalano, D. Pointcheval, A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2003, pp. 37–54.

[22] X. Wang, W. Cheng, P. Mohapatra, T. Abdelzaher, Enabling reputation and trust in privacy-preserving mobile sensing, IEEE Trans. Mob. Comput. 13 (12) (2013) 2777–2790.

[23] D. Wu, S. Si, S. Wu, R. Wang, Dynamic trust relationships aware data privacy protection in mobile crowd-sensing, IEEE Internet Things J. 5 (4) (2017) 2958–2970.

[24] X. Wei, Y. Yan, W. Jiang, J. Shen, X. Qiu, A blockchain based mobile crowdsensing market, China Commun. 16 (6) (2019) 31–41.

[25] Z. Noshad, A. Javavid, M. Zahid, I. Ali, N. Javavid, et al., A blockchain based incentive mechanism for crowd sensing network, in: International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Springer, 2019, pp. 568–578.

[26] Y. Nakayama, Network-side task allocation for mobile crowdsensing, in: 2020 IEEE 17th Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2020, pp. 1–6.

[27] X. Gao, H. Huang, C. Liu, F. Wu, G. Chen, Quality inference based task assignment in mobile crowdsensing, IEEE Trans. Knowl. Data Eng. (2020).

[28] Z. Yu, W. Zhu, L. Guo, W. Guo, Z. Yu, Task allocation for crowdsensing based on submodular optimisation, Int. J. Ad Hoc Ubiquitous Comput. 33 (1) (2020) 48–61.

[29] N. Jiang, D. Xu, J. Zhou, H. Yan, T. Wan, J. Zheng, Toward optimal participant decisions with voting-based incentive model for crowd sensing, Inform. Sci. 512 (2020) 1–17.

[30] G. Li, J. Cai, An online incentive mechanism for crowd sensing with random task arrivals, IEEE Internet Things J. 7 (4) (2020) 2982–2995.

[31] Y. Zhan, Y. Xia, J. Zhang, T. Li, Y. Wang, An incentive mechanism design for mobile crowdsensing with demand uncertainties, Inform. Sci. (2020).

[32] A. Chatterjee, M. Kaushal, I. Sengupta, Accelerating sorting of fully homomorphic encrypted data, in: International Conference on Cryptology in India, Springer, 2013, pp. 262–273.

[33] J. Cheon, D. Kim, D. Kim, H. Lee, K. Lee, Numerical method for comparison on homomorphically encrypted numbers, in: ASIACRYPT, (2) 2019, pp. 415–445.

[34] J. Cheon, D. Kim, D. Kim, Efficient homomorphic comparison methods with optimal complexity, in: ASIACRYPT, (2) 2020, pp. 221–256.

[35] D. Huang, Q. Gan, X. Wang, C. Huang, Y. Lin, Toward comparable homomorphic encryption for crowd-sensing network, in: Cryptology ePrint Archive, Report 2020/806.

[36] P. Sobti, A. Nayyar, P. Nagrath, Time series forecasting for coronavirus (COVID-19), in: International Conference on Futuristic Trends in Networks and Computing Technologies, Springer, Singapore, 2020, pp. 309–320.

[37] A. Devi, A. Nayyar, Perspectives on the definition of data visualization: a mapping study and discussion on coronavirus (COVID-19) dataset, in: Emerging Technologies for Battling Covid-19, Springer, Cham, 2021, pp. 225–240.

[38] F. Al-Turjman, A. Devi, A. Nayyar, Emerging technologies for battling Covid-19, Stud. Syst. Decis. Control (2021) 324.

[39] H. Tripathy, S. Mishra, S. Suman, A. Nayyar, K. Sahoo, Smart COVID-shield: an IoT driven reliable and automated prototype model for COVID-19 symptoms tracking, Computing 104 (2022) 1233–1254.

[40] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 1999, pp. 223–238.

[41] R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.

[42] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, 2009, pp. 169–178.

[43] L.-J. Pang, Y.-M. Wang, A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing, Appl. Math. Comput. 167 (2) (2005) 840–848.

[44] A. Shamir, How to share a secret, Commun. ACM 22 (11) (1979) 612–613.

[45] Z. Zhang, J. Wang, Y. Wang, Y. Su, X. Chen, Towards efficient verifiable forward secure searchable symmetric encryption, in: European Symposium on Research in Computer Security, Springer, 2019, pp. 304–321.

[46] S.-F. Sun, J. Liu, A. Sakzad, R. Steinfeld, T. Yuen, An efficient non-interactive multi-client searchable encryption with support for boolean queries, in: European Symposium on Research in Computer Security, Springer, 2016, pp. 154–172.