applied sciences

MDPI

*Article*

# PPDC: A Privacy-Preserving Distinct Counting Scheme for Mobile Sensing

**Xiaochen Yang [1], Ming Xu [1], Shaojing Fu [1,2,*] and Yuchuan Luo [1]**

[1] College of Computer, National University of Defense Technology, Changsha 410073, China
[2] Sate Key Laboratory of Cryptology, Beijing 100878, China
* Correspondence: shaojing1984@163.com; Tel.: +86-155-2640-4186

Received: 7 June 2019; Accepted: 4 September 2019; Published: 5 September 2019

check for updates

**Abstract:** Mobile sensing mines group information through sensing and aggregating users' data. Among major mobile sensing applications, the distinct counting problem aiming to find the number of distinct elements in a data stream with repeated elements, is extremely important for avoiding waste of resources. Besides, the privacy protection of users is also a critical issue for aggregation security. However, it is a challenge to meet these two requirements simultaneously since normal privacy-preserving methods would have negative influence on the accuracy and efficiency of distinct counting. In this paper, we propose a Privacy-Preserving Distinct Counting scheme (PPDC) for mobile sensing. Through integrating the basic idea of homomorphic encryption into Flajolet-Martin (FM) sketch, PPDC allows an aggregator to conduct distinct counting over large-scale datasets without disrupting privacy of users. Moreover, PPDC supports various forms of sensing data, including camera images, location data, etc. PPDC expands each bit of the hashing values of users' original data, FM sketch is thus enhanced for encryption to protect users' privacy. We prove the security of PPDC under known-plaintext model. The theoretic and experimental results show that PPDC achieves high counting accuracy and practical efficiency with scalability over large-scale data sets.

**Keywords:** distinct counting; privacy-preserving; mobile sensing; flajolet-martin sketch; secure bitwise XOR

## 1. Introduction

With the rapid development of information technology and modern manufacturing, mobile devices are almost ubiquitous nowadays and have occupied an indispensable position in daily lives of many. Especially, those devices, like smartphones, which are equipped with ROMs, CPUs, and a variety of sensors such as GPS, camera and so on, are used not only for their traditional functions, but also for sensing, data transmission, and calculation. As a result, these features make these devices ideal mobile carriers favored by researchers as they study many issues. The mobile sensing problem is one of these issues. In recent years, a considerable amount of mobile sensing projects have been developed using different mobile devices, like [1–3].

The process of mobile sensing can be described in the following steps: the sensing task publisher (or the aggregator) issues tasks to users with mobile devices, then mobile devices of users collect sensing data and send them to the aggregator. After that, the aggregator processes all the data to draw valid conclusions. In general situation, the aggregator needs to collect and monitor users' data continuously, which means that the scale of collected sensing data is considerable. The data can be in various forms, including camera images, location data, etc.

There are two essential challenges in actual mobile sensing projects. One is that whether users with mobile devices are willing to give the original sensing data to the aggregator. As the original data may contain users' private information such as physical location, consumption habits, physical health, etc., most users would give a negative reaction to such a mobile sensing application lacking reliable privacy protection. The aggregation result would be incomplete and lack representativeness. The other one is, for the aggregator, how to solve the distinct counting problem [4] when facing the huge sensing dataset with a large amount of duplicate data in various forms. If the aggregator does not have a good understanding of the cardinality of users' data, then a lot of meaningless computing resources will be wasted to handle duplicate data during the whole aggregation. In addition, excessive repetitive elements in an aggregated dataset may result in characteristics of data being inconspicuous. For example, in the vehicular sensing network, users may transmit sensing information about road congestion to aggregators, and information about the same intersection from different users can be highly repetitive. Aggregators should not waste time and resources on duplicate data when they count congestion and expect subsequent analysis, such as optimizing path selection. At this point, the aggregator should first make cardinal statistics of the original data, and then analyse the traffic congestion degree. In other words, a solution which can ensure the safety of users' privacy as well as solve the distinct counting problem is in urgent need.

Exiting studies about the distinct counting problem in mobile sensing mainly focus on researching various algorithms (such as the Flajolet–Martin sketch [5] and LogLog [6]); few works have considered the privacy of users during data aggregation. Han et al. [7] propose a secure data aggregation scheme, while their security goal is to enable the traffic monitoring center to verify whether an aggregate sensing report is correct or not. Their security refers to the aggregator's aggregated security rather than the user's privacy protection.

In this paper, we propose a scheme, Privacy-Preserving Distinct Counting scheme (PPDC for short), to solve the distinct counting problem with privacy protection of users. PPDC is based on a semi-honest model and it can complete distinct counting over large datasets with various forms of elements in the mobile sensing scenario. Through expanding each bit of the hashing values of users' original data added to the FM sketch, PPDC enhances the FM sketch to apply the bitwise XOR homomorphic encryption algorithm as an encryption method, so that users' privacy gets protected even under a known plaintext model. We conduct theoretical analysis and experiments, and the results show that our scheme achieves practical counting accuracy and efficiency.

The remainder of this paper is organized as follows. Section 2 defines system and security models and introduces several necessary preliminaries. In Section 3, we present the main idea and essential module of PPDC. Section 3 also analyzes the correctness and security of PPDC. After that, Section 4 provides the experimental results about the evaluations of accuracy rate and efficiency of PPDC. Section 5 discusses the related work. Finally, we conclude the paper in Section 6.

## 2. Problem Statements and Preliminaries

In this section, aiming at the privacy protection and distinct counting problem, we conduct the system and security models in detail. Then we introduce the encryption algorithm and aggregation sketch applied in our scheme.

### *2.1. System and Security Model*

***System model.*** We consider the system model in this paper as follows: there is a group of users with mobile devices who are providing data to a sensing task publisher or an aggregator to do some sensing task. Assume that sensing data of each user is a set of data with various forms of elements, including images and location data and so on. The aggregator needs to find the number of distinct data of all users' data, a big dataset composed by plenty of sub datasets. When transmitting sensing data, all users would not reveal their original data to the aggregator. We discuss a general network model in mobile sensing, in which there is a direct communication channel between the aggregator and every mobile device user. That is to say, the aggregator and all users form a star network topology. The communication channels could be 3G/4G, wifi, or other kinds of channels that are supported by the mobile devices and the aggregator in practical applications. Besides, as for each user's device, it has the ability to do the hash operation and bitwise XOR encryption to its sensing data and transmit worked data to the aggregator. The whole process of data aggregation in our scheme PPDC is described in Figure 1.

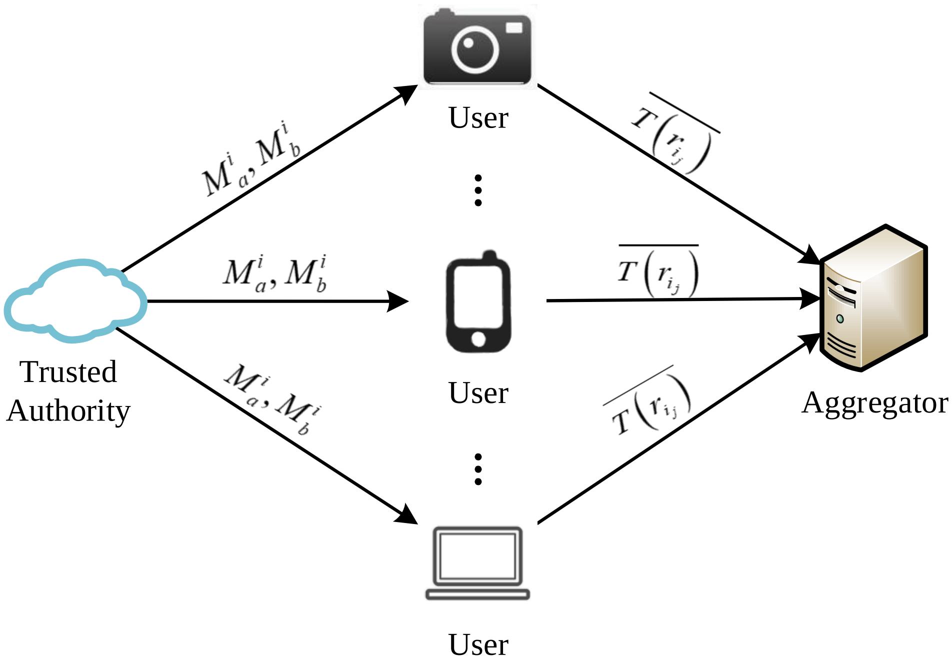

**Figure 1.** Overview of PPDC.

***Security model.*** In this paper, we assume that it is a semi-honest model. All the aggregators and users observe the data transmission and collection process described above. However, they may attempt to derive extra information about other participators' private inputs during the whole execution, which they should not know. Therefore, the scheme is believed to be secure if it guarantees that every participator can learn no more information from the process than the information that this participator is entitled to know. For the users, they should not be able to get the values of data of each other without permission. While for the aggregator, except for the encrypted data from users and the calculating result of these aggregated data, no extra knowledge about users ought to be acquired or speculated from the data he/she aggregates.

### 2.2. XOR Homomorphic Encryption

We choose the bitwise XOR homomorphic encryption as the encryption algorithm in this paper, of which the main idea is very similar to that of an additively homomorphic encryption scheme proposed in [8]. A trusted third party, the authority, is needed during the process of key generation. Let $f_{m,\alpha,\beta}()$ denote a function in the pseudo-random function family $F_{m,\alpha,\beta} = \{f_{m,\alpha,\beta} : \{0,1\}^{\alpha} \rightarrow \{0,1\}^{\beta}\}_{m \in \{0,1\}^{\gamma}}$, where $\alpha, \beta, \gamma \in \mathbb{N}$. Let $t \in \{0, ..., 2^{v} - 1\}$ denote the nonce information of data. The following process shows the details of the encryption algorithm.

a. **Key generation:**

(1) The trusted authority uniformly and independently picks $m_1, ..., m_n \in \{0,1\}^{\gamma}$. Then the authority computes $M_a^i = m_i$ and $M_b^i = m_{(i+1)\ mod\ n}$ for each user $i(i = 1, ..., n)$, and sends them to user $i$.

(2) For each dataset with the nonce information $t$ which is different in each time of transmission and all the user are synchronized, user $i$ computes its secret key by

$$k_i = F_{M_a^i,v,l}(t) \oplus F_{M_b^i,v,l}(t) \tag{1}$$

b. **Encryption:**

Denote by $x_i \in \{0,1\}^l$ a bit-string. The user $i$ encrypts it by computing

$$\overline{x_i} = x_i \oplus k_i \tag{2}$$

c. **Decryption:**

Denote by $\overline{x_i}$ a ciphertext of user $i$. The user $i$ decrypts it by computing

$$x_i = \overline{x_i} \oplus k_i \tag{3}$$

d. **Aggregation:**

Anyone can decrypt the bitwise XOR of all users' plaintexts without any user's secret key by computing

$$x_1 \oplus ... \oplus x_n = \overline{x_1} \oplus ... \oplus \overline{x_n} \tag{4}$$

Because keys are obtained from Equation (1) which means that the total number of seeds for all users' keys are even and the times of the value of each seed are even, a conclusion can be drawn that the bitwise XOR of all users' keys equals 0. As a result, the bitwise XOR of all users' ciphertexts is equal to the bitwise XOR of all users' plaintexts, which is Equation (4). In other words, this encryption algorithm is homomorphic on the bitwise XOR computation.

In this paper, the aggregator does not have any user's private key, so that it cannot decrypt any user's plaintext. Instead, it decrypts the bitwise XOR of all users' plaintexts and uses this information to solve the distinct counting problem. Therefore, when we talk about the aggregator's decryption operation, it refers to the decryption of the bitwise XOR of all users' plaintexts.

### *2.3. FM Sketch*

A FM sketch is a data structure for probabilistic counting of distinct elements that has been introduced in [9]. It is widely used in network applications, such as data dissemination [10] and probabilistic aggregation [11,12].

FM sketch represents an approximation of a positive integer by a bit field $S = s_1, s_2, ..., s_w$ of length $w$, where $w \geq 1$. The bit field is initialized to zero at all positions. To add an element $x$ to the sketch, it is hashed by a hash function $h$ with geometrically distributed positive integer output, where the probability is $P(h(x) = i) = 2^{-i}$. The entry $s_{h(x)}$ is then set to 1. With probability $2^{-w}$, we have $h(x) > w$ and no operation is performed in this case. A hash function with the necessary properties can easily be derived from a common hash function with equidistributed bit string output by using the position of the first 1-bit in the output string as the hash value.

According to [9], an approximation $C(S)$ of the number of distinct elements added to the sketch can be obtained by locating the end of the initial, uninterrupted sequence of ones.

$$Z(S) := \min(\{i \in \mathbb{N}_0 \mid i < w \wedge s_{i+1} = 0\} \cup \{w\}) \tag{5}$$

$$C(S) := \frac{2^{Z(S)}}{\varphi}, \varphi \approx 0.775351 \tag{6}$$

Since the variance of $Z(S)$ is pretty significant, the approximation $C(S)$ in Equation (6) is not very accurate. To avoid this situation, a set of sketches will be used to represent a single value instead of only one sketch. [9] proposes the respective technique called Probabilistic Counting with Stochastic Averaging (PCSA). With PCSA, before being added there, each element is first mapped to one of the sketches by using an equidistributed hash function. If $d$ sketches are used, denoted by $S_1, ..., S_d$, the estimation for the total number of distinct elements added is then calculated through

$$C(S_1, ..., S_d) := d \cdot \frac{2^{\sum_{i=1}^{m} \frac{Z(S_i)}{d}}}{\varphi} \tag{7}$$

However, in [9] it also points out that Equation (7) is rather inaccurate as long as the number of elements is below approximately $10 \cdot d$. According to [13], we modify Equation (7) in the following way:

$$C(S_1, ..., S_d) := d \cdot \frac{2^{\sum_{i=1}^{d} \frac{Z(S_i)}{d}} - 2^{-\kappa \cdot \sum_{i=1}^{d} \frac{Z(S_i)}{d}}}{\varphi}, \kappa \approx 1.75 \tag{8}$$

This alleviates the initial inaccuracies, while otherwise being asymptotically equivalent to Equation (7). PCSA with $d$ sketches yields a standard error of approximately $0.78/\sqrt{d}$ [9,14]. For many mobile sensing projects, it can achieve sufficiently good approximations when the sizes of dataset are reasonable.

The FM sketch can be merged to obtain the total number of distinct elements added to any of them by a simple bitwise OR. It is important to note that, by their construction, repeatedly combining the same sketches or adding already present elements again will not change the results, no matter how often or in which order these operations occur. This makes FM sketches ideally suited for the distinct counting scheme in mobile sensing.

## 3. Privacy-Preserving Distinct Counting Computation

In this section, we describe a specific process of operating on the sensing data of users based on FM sketch. Here, we employ a knack on FM sketch to greatly reduce the overall computing time. And then, the important part in PPDC, operations of encryption and decryption(i.e., calculation based on ciphertexts), are presented in detail. After that, in Section 3.4, the correctness and the security of PPDC will be discussed. Assume that the space of users' data is $[0, N-1] (N \geqslant 2)$, and $w = \lceil \log_2 N \rceil$.

### *3.1. Overview of PPDC*

At a high level, PPDC works as follows. First, each user independently prepares his dataset and transforms the sensing data into a specific form through their smart devices. Then, the trusted authority calculates and distributes a pair of key seeds to each user. Combining with key seeds and a nonce information which is different in each time of aggregation process, each user encrypts his transformed sensing data for this process and sends the ciphertexts to the aggregator. Since all the data are encrypted and there is no information of keys published, the aggregator has no way to decrypt received data, thus, the privacy of users is protected. In the end of PPDC, FM sketches are applied by the aggregator to acquire the count of distinct elements in all users' sensing datasets based on the judgement of ciphertexts. The main idea and essential part of PPDC will be described below.

### *3.2. Main Idea*

From Section 2.3, it is obvious that while dealing with the distinct counting problem, FM sketch cannot provide the protection of users' privacy during the transmission and calculation process. Therefore, we provide a method that expands each bit of the string to make the sketch suitable for the encryption operation, where the string is the calculating result of each user's dataset.

As mentioned above, there are $n$ users in total. In the FM sketch, the bit field $S = s_1, s_2, ..., s_w$ of length $w$ is initialized to zero at all positions. In the meanwhile, each user's sensing data is a dataset with various forms of elements, ranging in size from small to large. Assume that user $i (i = 1, ..., n)$ has $L_i$ elements in his sensing dataset. Let the set $\{x_l^{(i)}\}$ denote the original sensing dataset of user $i$, where $l = 1, ..., L_i$. While putting user $i$'s sensing data $\{x_l^{(i)}\}$ into the FM sketch, PPDC determines the bit field $S$ bit by bit, from the Most Significant Bit (MSB) to the Least Significant Bit (LSB). The MSB refers to the last bit of $S$, and the LSB is the first bit relatively.

***Step 1*** This step is taken by users. For a user $i (i = 1, ..., n)$, every element $x_l^{(i)} (l = 1, ..., L_i)$ in his original dataset $\{x_l^{(i)}\}$ is hashed by a hash function $h$ with a $w$-bit string output. Let $\Lambda_l^{(i)} = (r_l^{(i)})_1, ..., (r_l^{(i)})_w$ denote this bit string output of length $w$, where $(r_l^{(i)})_j$ is the $j$th bit in the string and the probability is $P((r_l^{(i)})_j = 1) = 2^{-j}$. That is to say, $\Lambda_l^{(i)} = h(x_l^{(i)})$. Then a bitwise OR operation is taken to get

$$\Lambda_i = \Lambda_1^{(i)} \vee ... \vee \Lambda_{L_i}^{(i)} \tag{9}$$

The string $\Lambda_i = r_{i_1}, ..., r_{i_w}$ represents all elements in user $i$'s original dataset.

According to the knowledge mentioned in Section 3.3, Equation (10) should be correct.

$$S = \Lambda_1 \vee ... \vee \Lambda_n \tag{10}$$

However, $S$ should not be calculated out straightforward. Because according to Equation (9), if the aggregator could receive $\Lambda_i$ directly, $\Lambda_i$ would reveal the original data of user $i$, especially when the size of his dataset is small. Therefore, a series of operations should be carried on the $\Lambda_i$.

***Step 2*** This step is also done on the user's side. The user $i$ operates on each bit of the string $\Lambda_i$ in order to avoid any damage caused on the privacy. In PPDC, we design a kind of specific coding scheme for these bits. Let $T(r_{i_j})$ denote the corresponding code of $r_{i_j}$ in the coding scheme, where $j = 1, ..., w$. The coding scheme is defined as follows:

$$T(r_{i_j}) \begin{cases} = 0^q, & if\ r_{i_j} = 0 \\ \overset{\circ}{=} \{0,1\}^q \setminus 0^q, & if\ r_{i_j} = 1 \end{cases} \tag{11}$$

where $q \in N$ is the accuracy controlling parameter and $\overset{\circ}{=}$ denotes to sample uniformly at random.

Figure 2 shows an example of the process user 1 deals with his original dataset.

***Step 3*** The aggregator takes this step after aggravation all users' coded data. Let $G(j) = T(r_{1_j}) \oplus ... \oplus T(r_{n_j})$ with bitwise XOR operation. Then there is a judgement rule designed to determine each bit of FM sketch $S$, corresponding to the coding scheme (11). We define the rule as follows:

$$s_j = \begin{cases} 0, & if\ G(j) = 0^q \\ 1, & if\ G(j) \neq 0^q \end{cases} \tag{12}$$

where $s_j$ is the $j$th-LSB, or $(w - j)$th-MSB in the bit field $S$. Notice here that when PPDC judging each bit in FM sketch, it starts from the MSB to the LSB.

***Step 4*** The calculation work is done by the aggregator. Based on the FM sketch $S$, the aggregator can get a significant parameter $Z(S)$, the position of the last bit in $S$ that is 1, according to Equation (5). As mentioned in Section 3.3, the approximation of distinct counting needs several more FM sketches in which the hash functions are different. After taking Step 1 to Step 3 for $d$ times and according to Equation (8), the aggregator can get the final result $C(S)$, the number of distinct elements in the sensing dataset.

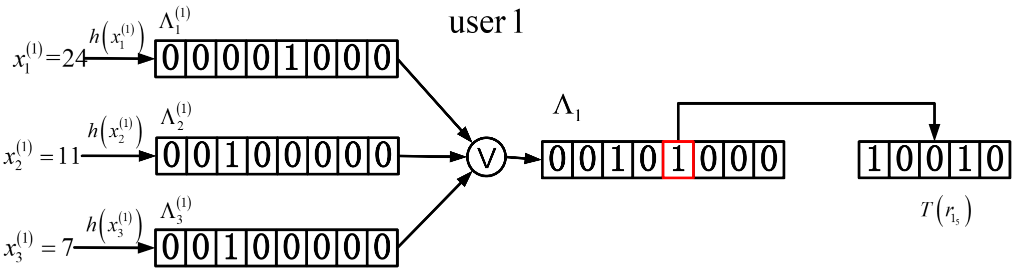

**Figure 2.** An example of the process that user 1 deals with his original dataset. Here, for example, the elements in the dataset are all integers. User 1 hashes elements of his dataset into 8-bit strings and do the bitwise OR operation to get a string as the representation of his dataset. Then each bit of this string is coded to a 5-bit string.

**Remark 1.** *In Step 2, it is worth noting that there is a probability of* $1 - 1/2^q$ *to occur such a situation, where* $r_{i_j}$ *equals to 1 but* $T(r_{i_j})$ *is coded to be* $0^q$*. Thus in Equation (11), the coding scheme requires that if this situation happened,* $r_{i_j}$ *should be recoded until* $T(r_{i_j})$ *is not* $0^q$*. In that step, each bit of the string* $\Lambda_i$*, from MSB to LSB, would be expanded into a q-bit string* $T(r_{i_j})$ *under the action of our coding scheme (11), which is suitable for encryption operation.*

**Remark 2.** *Notice that during the whole process, in order to reduce the computing time, we employ a knack here which is that PPDC determines bits of FM sketch S from the last bit to the first bit. When the aggregator applies FM sketches, the purpose is to find out the position of the last bit in S that is 1 and regard it as an index. This is to find out the position of the first bit in S that is 1, when PPDC starts finding from the last bit of S. This transformation means the aggregator does not have to determine all bits in S, after all, what the aggregator needs is the index to calculate the number of distinct counting of the dataset rather than the whole S. Through this knack, PPDC can leave out a lot of computing steps, thus improving the efficiency.*

### 3.3. Privacy-Preserving Distinct Counting Scheme

In Section 3.2, we calculated the number of distinct counting through PPDC. The specific operation towards users' data is prepared for the homomorphic encryption to protect users' privacy. In this subsection, we highlight the modules of encryption and decryption(i.e., calculation based on ciphertexts) in PPDC that allow the aggregator to solve the distinct counting problem and to avoid acquiring each user's data privacy at the same time. In our assumption, there is a trusted authority as a third party who helps users and the aggregator to establish a key system each time.

(1) ***Setup.*** The protection mechanism of PPDC is based on the bitwise XOR homomorphic encryption introduced in Section 2.2. The trusted authority has $m_1, ..., m_n \in \{0,1\}^\gamma$ privately and he computes $M_a^i = m_i$ and $M_b^i = m_{(i \ mod \ n)+1}$ for each user $i(i = 1, ..., n)$. Then the two seeds are sent to the corresponding user. The user $i$ does a bitwise XOR operation on the seeds as well as a nonce number $t$ according to Equation (1) to acquire his own key $k_i$. Notice that the nonce number $t$ used for calculating $k$ is different in each transmission.

(2) ***Encrypt.*** The data encryption is operated on the user's side. The user $i$ regards the coding string $T(r_{i_j})$ for $j$th bit of his data representation $\Lambda_i$ as the plaintext and encryptes it with the bitwise XOR homomorphic encryption algorithm to get the ciphertext

$$\overline{T(r_{i_j})} = T(r_{i_j}) \oplus k_i, (j = 1, ..., w) \tag{13}$$

where the user $i$'s key $k_i$ is generated as introduced above. Then the ciphertext $\overline{T(r_{i_j})}$ is sent to the aggregator as user $i$'s sensing data.

(3) ***Aggregate.*** On the side of the aggregator, he collects all the $n$ users' data about the $j$th-LSB and then does the bitwise XOR computations. Denote by $\overline{G(j)}$ the bit string result. According to Equation (4), it can be drawn that

$$\overline{G(j)} = \overline{T(r_{1_j})} \oplus ... \oplus \overline{T(r_{n_j})} \tag{14}$$

It is easy to see that if the $j$th-LSBs of the $n$ users are all 0, then the bitwise XOR of the corresponding strings $\overline{G(j)}$ is always a $q$-bit string of 0s. If there is any user whose data $\Lambda$ is 1 on the $j$th-bit, the bitwise XOR of all reports' corresponding strings is not a $q$-bit string of 0s with a probability of $1 - 1/2^q$. However this situation has little influence on the accuracy of PPDC which will be proved in Section 3.4.

(4) ***Judge.*** Just like the rule mentioned above, we define the rule as follows:

$$s_j = \begin{cases} 0, & if\ \overline{G(j)} = 0^q \\ 1, & if\ \overline{G(j)} \neq 0^q \end{cases} \tag{15}$$

where $s_j$ is the $j$th-LSB in the bit field $S$.

In Figure 3, a detailed example of aggregation in the FM sketch using above described transformation and corresponding bitwise XOR computations is shown. And the formal description of our entire scheme is shown in Algorithm 1.

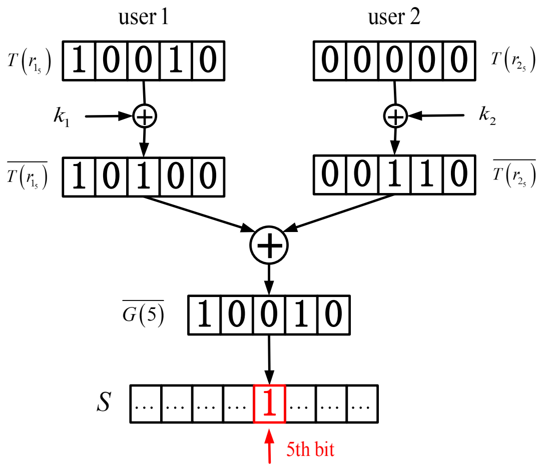

**Figure 3.** An example of determining the 5th bit in FM sketch $S$ with the bitwise XOR homomorphic encryption, where the coding scheme defined the length of the bit string $T(r_{i_j})$ is 5. The result shows that the aggregation result is not influenced by the operations of user's privacy protection.

**Remark 3.** *The operation of homomorphic encryption causes no damage on the accuracy of PPDC. According to the property of the bitwise XOR homomorphic encryption, there is* $\overline{G(j)} = \overline{T(r_{1_j})} \oplus ... \oplus \overline{T(r_{n_j})} = T(r_{1_j}) \oplus ... \oplus T(r_{n_j}) = G(j)$. *Thus, we can say that Equation (15) is equal to Equation (12), which means that the calculated aggregation result is not influenced by the encryption and decryption operations for user's privacy protection. Then the final result of distinct counting problem is calculated by Step 4 in Section 3.2.*

**Remark 4.** *The correctness and security of PPDC are credible. According to Equation (10), we have*

$$s_j = r_{i_1} \vee ... \vee r_{i_w} \tag{16}$$

In PPDC, Equation (15) is equal to Equation (16) with a probability of $1 - 1/2^q$, which means that the operations in PPDC have nearly no effect on the final result when the parameter $q$ is appropriate. We will prove it in Theorem 1. The security of PPDC will later be formally proved in Theorem 2.

**Algorithm 1** Privacy-preserving Distinct Counting Scheme

```
Input:
  ${x_i^l}$: User $i$'s dataset with various elements, $l \in [0, L_i], x_i^l \in [0, N-1],\ i = 1, ..., n$;
  $h$: A hash function with a $w$-bit string output;
  $d$: the number of FM sketches;
  $M_a^i$ and $M_b^i$: two secret seeds of User $i$;
  $t \in [0, 2^v - 1]$: A public known nonce number;
  $q \in \mathbb{N}$: An accuracy controlling parameter.
Output: The number of distinct elements in $\{x_i\}, i = 1, ..., n$.
for $k = 1$ to $d$ do
    for $i = 1$ to $n$ do
        User $i$: $\Lambda_i \leftarrow h(x_i), len(\Lambda_i) = w$;
        User $i$: $k_i = F_{M_a^i, v, q}(t) \oplus F_{M_b^i, v, q}(t)$;
    end for
    for $j = w$ to 1 do
        for $i = 1$ to $n$ do
            User $i$: $T(r_{i_j}) \leftarrow r_{i_j}$ in $\Lambda_i$, $len(T(r_{i_j})) = q$;
            User $i$: $\overline{T(r_{i_j})} = T(r_{i_j}) \oplus k_i$;
        end for
        Aggregator $P$: $\overline{G(j)} = \overline{T(r_{1_j})} \oplus ... \oplus \overline{T(r_{n_j})}$.
        if $\overline{G(j)} = \{0\}^q$ then
            continue;
        else
            break;
        end if
    end for
end for
return $C(S) = d \cdot \frac{2^{\sum_{k=1}^{d} \frac{Z(S_k)}{d}} - 2^{-\kappa \cdot \sum_{k=1}^{d} \frac{Z(S_k)}{d}}}{\varphi}$;
```

### 3.4. Scheme Analysis

We present analysis of PPDC in terms of correctness and security.

**Theorem 1** (Correctness)**.** *The probability that the result of Equation (15) equals to that of Equation (16) is greater or equal to* $1 - 1/2^q$*. The correctness of PPDC is greater or equal to* $1 - (w/2^q)^d$*.*

**Proof.** According to the definition in Section 3.3, it is obvious that the sketch constructed by Equation (10) is the correct result of our problem. Equation (16) is one of Equation (10)'s mutually independent $w$ parts to determine the $j$th-LSB bit. While in PPDC, Equation (15) represents the result. Actually, Equation (15) is equal to Equation (16) with a probability of $1 - 1/2^q$ on the calculation.

On the basis of a regular bitwise OR, only when all the numbers on that bit are 0, the result bit is 0, which is $0 \vee ... \vee 0 = 0$. Otherwise, that bit should be 1. If our scheme is 100 percent accurate, when $s_j = 0$, it means $G(j)$ should be $0^q$ where all users' $r_{i_j}$ should be 0. However there is a special case that a user $y$ whose $r_{y_j}$ is 1 while $y$'s encrypted bit string equals the bitwise XOR of all other users' encrypted strings. Since the encoding function in our scheme is random and the encoding string has $2^q$ different choices, the probability for the result of our scheme being not accurate is $1/2^q$. Therefore, we have:

$$\begin{aligned} P(the\ jth\ bit\ is\ accurate) &= P(all\ users'\ jth\ bits\ are\ 0) \times 1 + P(any\ user's\ jth\ bit\ is\ 0) \times (1 - 1/2^q) \\ &\geqslant 1 - 1/2^q \end{aligned}$$

Because Equation (15) has to be independently calculated for $w$ times to achieve the goal of Equation (10) and there are $d$ FM sketches used, it is obvious that the correctness of PPDC is greater or equal to $1 - (w/2^q)^d$.

As in most cases, a malicious aggregator can only have the knowledge of ciphertext in privacy-preserving mobile sensing schemes. This belongs to the ciphertext-only attacks, which correspond to an attaker of minimal capability. However, we still analyze the security of PPDC under a more stronger model, known as the plaintext model, which assumes that the attackers may obtain a certain number of plaintext-ciphertext pairs through extra channels. The security of the proposed PPDC is summarized in the following theorem. □

**Theorem 2** (Security). *For the homomorphic operations in PPDC, there is no probabilistic polynomial time (P.P.T.) adversary that can break the data confidentiality of user's data under the known plaintext model.*

**Proof.** In PPDC, for the aggregator, we prove that there is no extra knowledge revealed to him in PPDC. We consider the situation that the aggregator could acquire most information. To calculate out the final result, all the $w$ bits in $S$ should be confirmed, which means that there must be $w$ times communication between each user and the aggregator and each time the aggregator could get $n$ cipertexts from all users. Let $I = (I_1, ..., I_w)$ denote the aggregator's information received from all users for $w$ times, where $I_j = (\overline{T(r_{1_j})}, ..., \overline{T(r_{n_j})})$ $(j = 1, ..., n)$ is the ciphertexts of $q$-bit bit strings from all users to decide the $j$th bit in $S$. The aggregator calculates $G(j)$ according to Equation (14) and then determines $s(j)$ in $S$ by Equation (15).

If the result is $s_j = 1$, then $G(j) \neq 0^q$, which can help the aggregator speculate that in $I_j$ there is at least one bit string $\overline{T}$ from some user which is not $0^q$. The aggregator wants to speculate $T$ from $\overline{T}$. Since the aggregator has no corresponding key, the probability that he guesses the correct plaintext is $1/2^q$. Under known plaintext model, the aggregator could get $a$ plaintext-ciphertext pairs to calculate $a$ keys. Then, the probability of knowing a certain user's original data rises to $1/A_n^a$. However, the fact that users' keys are different each time, leads to low probability and the attacking time $O(nq)$ is over P.P.T. Therefore, the aggregator could not conjecture any extra knowledge about the users.

If $s_j = 0$, then $I_j$ are all $0^q$ or it happens to such a case that some bit strings, $\{0,1\}^q \setminus 0^q$, equal to $0^q$ under the effect of XOR operations. However the aggregator could not distinguish these two situations by calculation in P.P.T.

Moreover, because the keys are pseudo-random for each user and each time, $I_1, ..., I_w$ are independent. As a result, there is no probabilistic polynomial time (P.P.T.) adversary that can break the data confidentiality of a user's data under known plaintext model. PPDC ensures the privacy protection of users. □

## 4. Performance Evaluation

In this section, we conduct experiments to evaluate the accuracy of PPDC as well as its efficiency compared with the situation lacking of privacy protection. In our experiments, the schemes are implemented with Python. The sensing data of users are randomly formed by the Python programs in the uniform random distribution, where we use difference values of integers to represent users' data.

### *4.1. Accuracy Evaluation of PPDC*

In PPDC, $q$ is an accuracy controlling parameter, the length of encoded bit string in *Step 2* of Section 3.2. Figures 4 and 5 show the relationship between the value of $q$ and the accuracy rate when the length $w$ of FM sketch is changing, where the total amounts of users are $n$ = 15,000 and $n$ = 25,000 respectively. It can be concluded that no matter what value $w$ is, as the value of $q$ approaches $w$, the accuracy rate gradually increases to nearly 100%, which is in accord with theoretical analysis above. At the same time, through comparing Figures 4 and 5, we can see that the trend of PPDC's accuracy rate is not affected by the number of users, but related to the corresponding value relationship between $q$ and $w$. When the difference between the value of $q$ and $w$ is reduced, the total accuracy is much more close to 100%. This conclusion reflects that as long as the parameters of PPDC are set appropriately, PPDC can be applied in the mobile sensing situation where no matter what the scale of users is.

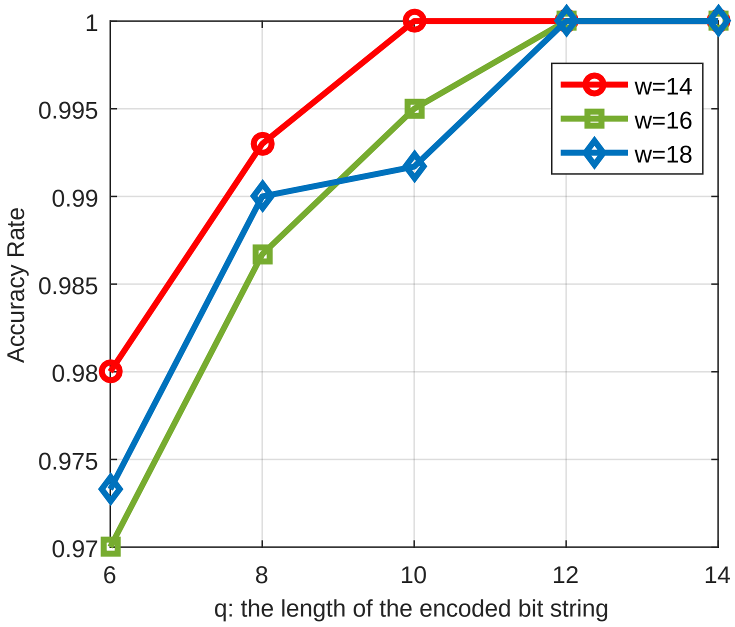

**Figure 4.** Accuracy rate of estimated data influenced by the value of $q$, where $n$ = 15,000.

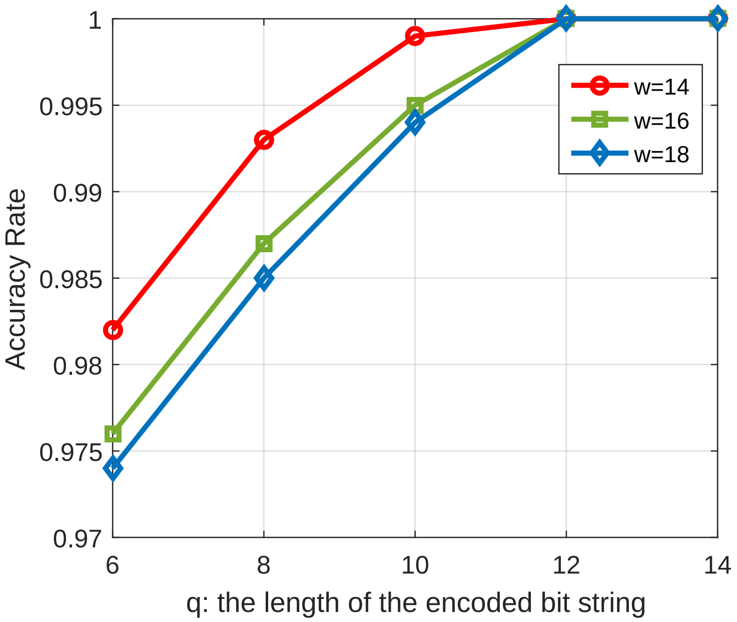

**Figure 5.** Accuracy rate of estimated data influenced by the value of $q$, where $n$ = 25,000.

Since there is a significant error in applying only one FM sketch, $d$, the number of FM sketches, must be discussed. From Figure 6, it is observed that the error rate of estimated data decreases dramatically along with the increase of the number of repeat times in the beginning, then keeps relatively stable after a specific threshold, like $d = 4$ in this experiment. In the face of different sizes of datasets, the threshold will be different. We set experiments with different numbers of users participating in the program. In the meantime, the corresponding number of distinct counting in each dataset is independent and irregular since the elements which present users' sensing data are generated entirely randomly. The calculated values of PPDC are contrasted with true values in Figure 7. There is a difference between the two values, and as the size of the dataset improves, the overall difference tends to decrease but still shows fluctuation. The fluctuation is associated with the result of the FM sketch which has a close relationship with the multiplies of 2.

According to the results in Figure 7, we calculate the accuracy rate of PPDC presented in Figure 8 to evaluate the correctness of PPDC more intuitively. It is obvious that the accuracy rate of PPDC is gradually raising close to 100% along with the increase of the size of datasets, where, even in the case of a small dataset, the accuracy rate can still reach 97%. When the amount of users is huge and corresponding cardinal number is big, PPDC can perform much better.

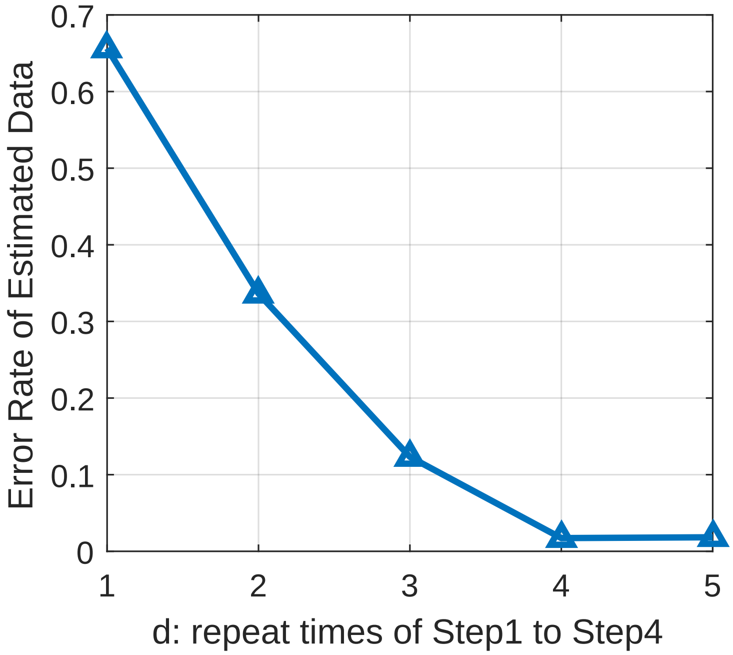

**Figure 6.** Error rate of estimated data affected by the value of $d$, where $n = 20{,}000$.

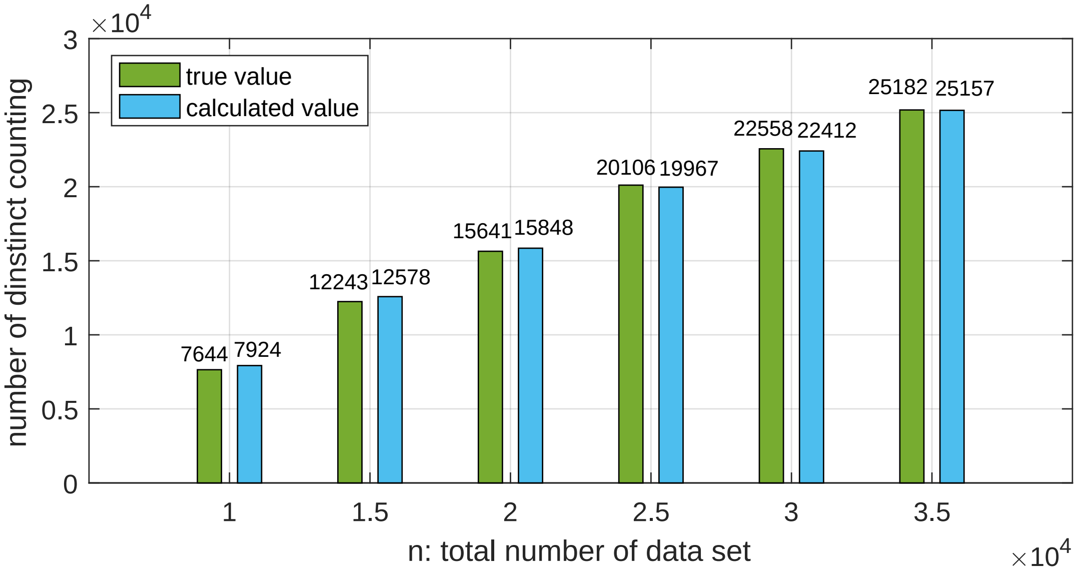

**Figure 7.** The contrast of the numbers of distinct counting coming from the true value and the calculated value.

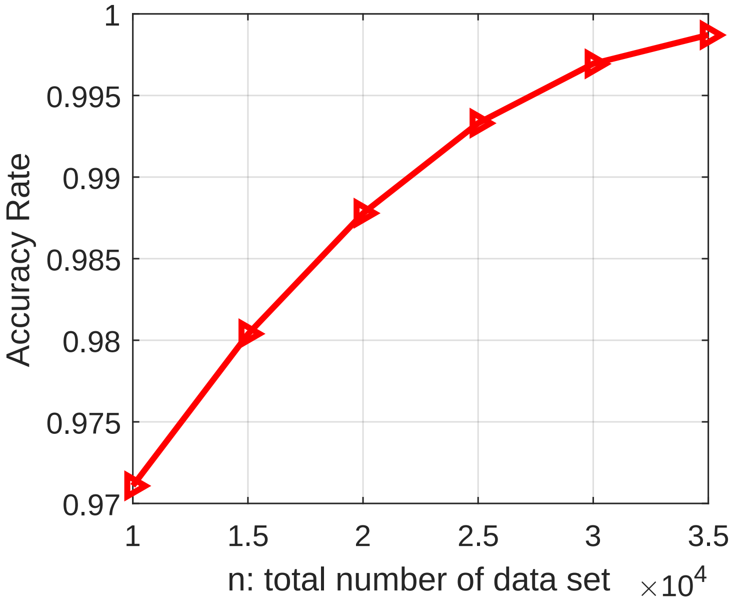

**Figure 8.** Accuracy rate of PPDC on different size of datasets.

### 4.2. Efficiency Evaluation of PPDC

After confirming the accuracy of PPDC through a set of experiments, we explore efficiency of PPDC by testing the communication cost and computing time.

(1) ***Communication Overhead.*** Table 1 shows the comparison of communication cost between the baseline method without privacy protection and PPDC. In Table 1, the total bits sent by a user, as the communication cost of a user, and the total bits received by the aggregator, as the communication cost of the aggregator, are the measured standards, as well as the computation complexity and round complexity of two schemes. The mentioned parameters include: $n$ which is the total number of users, and the range of users' data is $[0, N-1]$, and $w = \lceil \log_2 N \rceil$ is the length of each user's bit string, and $d$ is the number of FM sketches we applied. As the proof of Theorem 1 shows, $1 - (w/2^q)^d$ is the upper limit of the correctness of PPDC. When $q$ is approximately equal to $w$ and not too small, the error rate of PPDC will decrease to an acceptable level (for example, less than 0.001). Meanwhile, the communication cost of PPDC affected by $q$ would also be reduced. Besides, PPDC can send or receive less data than the baseline method when $n$ is not greater than $N$, i.e., $n = O(N)$.

However, the total communication cost is also influenced by the round complexity. The round complexity refers to the amount of time a user has to keep communicating online. Notice that the baseline method needs only one round of communication which is its most significant advantage. Therefore, the cases where PPDC performs better are when the network connection is stable, while when the network connection cannot stay reliable, the baseline method is more suitable.

(2) ***Computation Overhead.*** We discuss the computing time spent during the whole process. Here, the computing time of PPDC includes the time of hash operation and coding, encryption time for each user, and the time of decryption to determine the final FM sketch $S$ and calculating results for the aggregator. Note that 'decryption' is the decryption of the bitwise XOR of all users' plaintexts which guarantees the protection of user's privacy. The data used as a comparison is the computing time of the number of distinct counting calculated without privacy protection in Figure 9. It can be seen that it takes more time for PPDC to calculate the results. Since there are more processes like encoding, encrypting, decrypting and formula calculating than the general method, PPDC is relatively more

time-consuming. However, this consumption is within an acceptable range, as shown in Figure 9; with the dataset expanding, the trend of the increase in the consumption time is slower than the linear increase. Besides, due to solving the distinct counting problem, the following other operations on the aggregated dataset will reduce resources consumption of repetitive process. Furthermore, when the size of a dataset is huge, the probability of the index $Z(S)$ approaching the end of FM sketch $S$ is high. With our knack in Section 3.2, the computing time will gradually decrease accordingly. Therefore, on the whole, PPDC does not waste computing time. This conclusion proves that the efficiency of PPDC is appropriate for large-scale data aggregation processing.

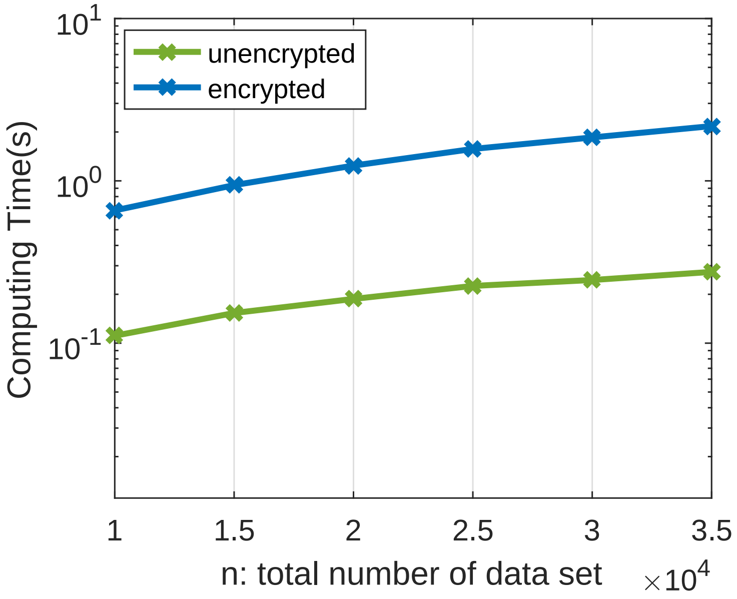

**Figure 9.** Computing time of the unencrypted process and that of PPDC.

When it comes to the computing time, we also conducted assessments of PPDC under different factors. Firstly, the value of $q$ has an effect on the computing time. Figure 10 shows the variation trend of computing time of PPDC with different values of $q$. As the $q$ is bigger, the corresponding time is much more and the difference caused between contiguous different $q$ rises when the size of dataset increases. Therefore, an appropriate value of $q$ is needed. Next, we evaluate the computing time spent by each step in PPDC. The corresponding results are presented in Figure 11. In Figure 11, the encryption and decryption step which is the most important part of achieving users' privacy-preserving in PPDC, is the most costly compared with other steps of hashing and coding. Due to the bitwise XOR operation of encryption and decryption, such result is reasonable. As the size of dataset increases, the increase of encryption and decryption time will slow down, since we employ the knack on FM sketches in PPDC. Besides, the increasing curves of all steps in Figure 11 tend to be lower than the linear increase, which is in accordance with the result in Figure 9.

**Table 1.** Communication Cost and Computation Complexity.

| | A User | | The Aggregator | | Round Complexity |
|---|---|---|---|---|---|
| | **Comm. Cost** | **Comp. Complexity** | **Comm. Cost** | **Comp. Complexity** | |
| Baseline | $N\lceil\log_2 n\rceil$ | $O(N\lceil\log_2 n\rceil)$ | $nN\lceil\log_2 n\rceil$ | $O(nN\lceil\log_2 n\rceil)$ | 1 |
| PPDC | $q\lceil\log_2 N\rceil$ | $O(q\lceil\log_2 N\rceil)$ | $nq\lceil\log_2 N\rceil$ | $O(nq\lceil\log_2 N\rceil)$ | $d$ |

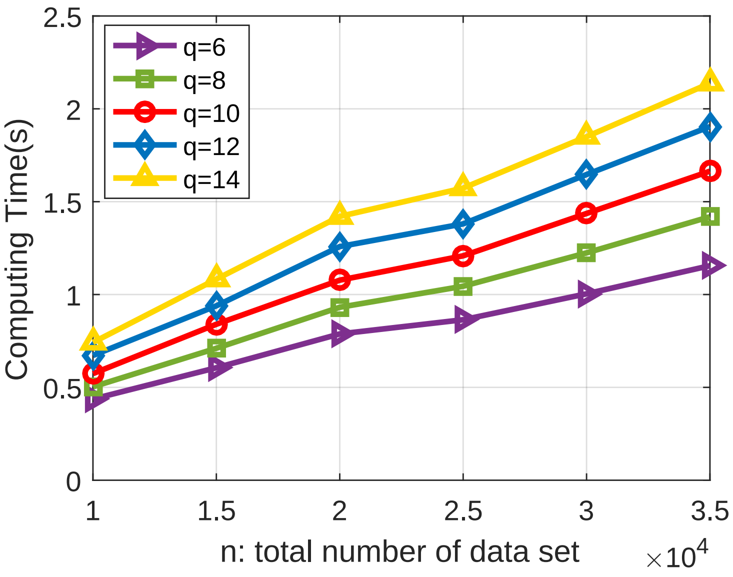

**Figure 10.** Computing time of PPDC with different value of *q*.

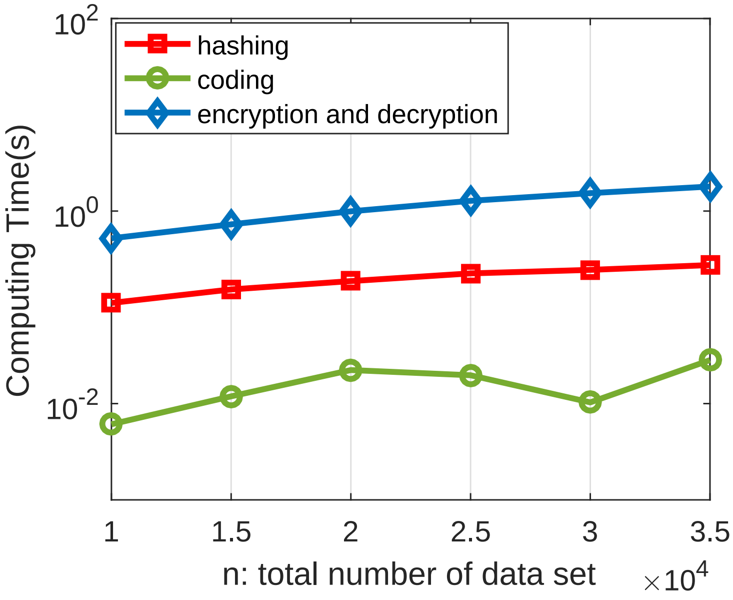

**Figure 11.** Computing time spent by different steps in PPDC.

## 5. Related Work

Recently, both the applications about mobile sensing and the problem of distinct counting have been discussed from a variety of aspects [7,15–18]. However, their scenarios usually include either privacy protection of users or distinct counting without considering them at the same time, which leads to lacking a secure and resource-saving system.

### *5.1. Privacy Preserving in Mobile Sensing Applications*

In terms of the applications about mobile sensing, most works focus on researching the various methods of a user's privacy protection or discussing the operation of aggregated data, like [19–23].

Both [15,24], consider the protection of user's privacy and then to seek the minimum computation in the aggregated dataset. They study how an untrusted aggregator in mobile sensing can periodically obtain desired statistics over the data contributed by multiple mobile users, without compromising the privacy of each user. Their scheme [15], which is based on [24], utilizes the redundancy in security to decrease the communication cost caused by each user's joining and/or leaving activities. Their protocol traverses the entire data space to find the minimum value on the basis of summation protocols rather than bitwise XOR operations.

Xiong et al. in [25] propose a scheme for mobile crowdsensing sevices. In this scheme, a differential privacy mechanism is utilized through allowing different users to add noise data, then employing homomorphic encryption for protecting the sensing data, and finally, uploading ciphertext to the mediator, who is able to obtain the collection of ciphertext of the sensing data without actual decryption. However, the cost of transmission and evaluation is relatively non-negligible.

Miao et al. describe a lightweight framework for mobile crowd sensing systems in [26]. The framework can achieve the protection of each participating worker's sensory data and reliability information, and introduce little overhead to the workers. It is implemented by involving two non-colluding cloud platforms and adopting additively homomorphic cryptosystem, where on the workers' side, their jobs are perturbing the data with some random numbers rather than directly encrypting the data to be uploaded. This method, however, has more requests for the participating clouds which are proposed to be non-colluding but can communicate with each other.

In [27], Zhang and Chen propose semi-honest protocols to calculate the minimum and $k$th minimum values in mobile sensing systems. The data can be a time-series. By using probabilistic coding schemes and a cipher system, they construct two protocols that allow homomorphic bitwise XOR computations for their problems. The homomorphic bitwise XOR algorithm ensures privacy during the whole process. As the interaction times increase, the bits sent or received by users and the aggregator are much more.

### 5.2. Distinct Counting

On the other hand, distinct counting is also of interest to a lot of researches. However, it is mainly discussed in Vehicular Ad Hoc Networks, like [5,7], rather than a more general mobile sensing scenario. In the mean while, in order to solve distinct counting problem, there are many studies [5,28,29], adopting different algorithms, including FM sketch.

In [6], Wangle et al. present a self-adaptive algorithm, Self-Adaptive LogLog, which is proposed based on Refined LogLog, to adapt to cardinalities of different scales automatically. They focus on the accuracy of the method, while the application scene is not clear and they rarely take the collection process into consideration.

Considine et al. in [30], use FM sketches to accomplish a kind of robust in-network aggregation in sensor networks. The application situation is believed to result in packet loss or node failures. They consider the coordinated collection of information towards a sink in the sensor network. However, the security problem is overlooked during the entire aggregation process. In [31], the FM sketch is used to integrate with spatio-temporal indexes to solve the problem: "How many objects were in region $x$ over the time interval $t$?" Like [30], Tao et al. in [31], do not mention the privacy protection of user data.

In [32], Zekri et al. propose an event-exchanging and data-gathering scheme based on FM sketches in vehicular networks. The sketches can be exchanged without loss of information and can be insensitive, so as to allow manipulating the same physical repository for all vehicles. Their aggregation structure reduces the time needed to actually find a parking space and increases the percentage of vehicles finding such a resource in a bounded time in congested situations.

Han et al. propose a secure data aggregation scheme in Vehicular Ad Hoc Networks which is based on the FM sketch in [7]. They also consider the security problem, while their security goal is to enable the traffic monitoring center to verify whether an aggregate sensing report is correct or not. Their security refers to the aggregator's aggregated security rather than the user's privacy protection.

**Remark 5.** *In general, there are few works that have studied both privacy protection of users and distinct counting for mobile sensing simultaneously. Our scheme, PPDC, integrating the idea of homomorphic encryption into FM sketch, provides a useful solution to solve this research problem. PPDC not only reduces the consumption of storage space and other resources by calculating the number of different elements, but also has practicality due to guaranteeing user privacy, which makes it worthy to be studied and applied for practical applications.*

## 6. Conclusions

Both privacy protection of users and the distinct counting problem on large datasets are essential issues in mobile sensing applications. In this paper, we propose a privacy-preserving distinct counting scheme, PPDC, to solve these two problems simultaneously. PPDC expands each bit of the hashing values of the users' original data, so that the FM sketch is enhanced for encryption to protect user privacy. We choose the bitwise XOR encryption algorithm as the encryption algorithm. According to the theoretical analysis, PPDC causes little damage to the accuracy of the FM sketch. Moreover, a set of experiments demonstrates that, with appropriate value of several parameters, PPDC achieves high counting accuracy and practical efficiency with scalability over large-scale datasets.

For future work, we will aim to expend PPDC to support the screening of invalid users and wrong data, so as to make the aggregated conclusions more accurate and facilitate other subsequent analysis operations.

**Author Contributions:** Conceptualization, X.Y. and Y.L.; software, X.Y.; resources, M.X. and S.F.; funding acquisition, M.X. and S.F.; Writing—Original draft preparation, X.Y.; Writing—Review and editing, M.X., S.F. and Y.L.

**Funding:** This work is supported in part by the National Natural Science Foundation of China (NSFC) under grants 61872372, 61379144, 61572026 and 61672195, and Open Foundation of State Key Laboratory of Cryptology (No: MMKFKT201617).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Thiagarajan, A.; Ravindranath, L.; La Curts, K.; Madden, S.; Balakrishnan, H.; Toledo, S.; Eriksson, J. VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, Berkeley, CA, USA, 4–6 November 2009; ACM: New York, NY, USA, 2009; pp. 85–98.
2. Rana, R.K.; Chou, C.T.; Kanhere, S.S.; Bulusu, N.; Hu, W. Ear-phone: An end-to-end participatory urban noise mapping system. In Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, Stockholm, Sweden, 12–16 April 2010; ACM: New York, NY, USA, 2010; pp. 105–116.
3. Huang, K.; Liu, X.; Fu, S.; Guo, D.; Xu, M. A Lightweight Privacy-Preserving CNN Feature Extraction Framework for Mobile Sensing. *IEEE Trans. Dependable Secure Comput.* **2019**. [CrossRef]
4. Bar-Yossef, Z.; Jayram, T.; Kumar, R.; Sivakumar, D.; Trevisan, L. Counting distinct elements in a data stream. In *International Workshop on Randomization and Approximation Techniques in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 1–10.
5. Lochert, C.; Scheuermann, B.; Mauve, M. Probabilistic aggregation for data dissemination in VANETs. In Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks, Montreal, QC, Canada, 10 September 2007; ACM: New York, NY, USA, 2007; pp. 1–8.

6. Wang, L.; Cai, Z.; Wang, H.; Jiang, J.; Yang, T.; Cui, B.; Li, X. Fine-grained probability counting: Refined loglog algorithm. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (Bigcomp), Shanghai, China, 15–17 January 2018.
7. Han, Q.; Du, S.; Ren, D.; Zhu, H. SAS: A secure data aggregation scheme in vehicular sensing networks. In Proceedings of the 2010 IEEE International Conference on Communications (ICC), Cape Town, South Africa, 23–27 May 2010; pp. 1–5.
8. Castelluccia, C.; Chan, A.C.; Mykletun, E.; Tsudik, G. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Trans. Sens. Netw.* **2009**, *5*, 20. [CrossRef]
9. Flajolet, P.; Martin, G.N. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.* **1985**, *31*, 182–209. [CrossRef]
10. Lochert, C.; Rybicki, J.; Scheuermann, B.; Mauve, M. Scalable data dissemination for inter-vehicle-communication: Aggregation versus peer-to-peer (skalierbare informationsverbreitung für die fahrzeug-fahrzeug-kommunikation: Aggregation versus peer-to-peer). *Inf. Technol.* **2008**, *50*, 237–242. [CrossRef]
11. Nadeem, T.; Dashtinezhad, S.; Liao, C.; Iftode, L. TrafficView: traffic data dissemination using car-to-car communication. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2004**, *8*, 6–19. [CrossRef]
12. Garofalakis, M.; Hellerstein, J.M.; Maniatis, P. Proof sketches: Verifiable in-network aggregation. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 996–1005.
13. Scheuermann, B.; Mauve, M. Near-Optimal Compression of Probabilistic Counting Sketches for Networking Applications. In Proceedings of the DIALM-POMC, Portland, OR, USA, 16 August 2007; Citeseer: Princeton, NJ, USA, 2007.
14. Kirschenhofer, P.; Prodinger, H.; Szpankowski, W. How to count quickly and accurately: A unified analysis of probabilistic counting and other related problems. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 211–222.
15. Li, Q.; Cao, G.; La Porta, T.F. Efficient and privacy-aware data aggregation in mobile sensing. *IEEE Trans. Dependable Secure Comput.* **2014**, *11*, 115–129. [CrossRef]
16. Liu, Z.; Li, B.; Huang, Y.; Li, J.; Xiang, Y.; Pedrycz, W. NewMCOS: Towards a Practical Multi-cloud Oblivious Storage Scheme. *IEEE Trans. Knowl. Data Eng.* **2019**. [CrossRef]
17. Liu, Z.; Huang, Y.; Li, J.; Cheng, X.; Shen, C. DivORAM: Towards a practical oblivious RAM with variable block size. *Inf. Sci.* **2018**, *447*, 1–11. [CrossRef]
18. Li, J.; Huang, Y.; Wei, Y.; Lv, S.; Liu, Z.; Dong, C.; Lou, W. Searchable Symmetric Encryption with Forward Search Privacy. *IEEE Trans. Dependable Secure Comput.* **2019**. [CrossRef]
19. Ma, R.; Cao, Z. Serial number based encryption and its application for mobile social networks. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 332–339. [CrossRef]
20. Au, M.H.; Liang, K.; Liu, J.K.; Lu, R.; Ning, J. Privacy-preserving personal data operation on mobile cloud—Chances and challenges over advanced persistent threat. *Future Gener. Comput. Syst.* **2018**, *79*, 337–349. [CrossRef]
21. Bae, M.; Kim, K.; Kim, H. Preserving privacy and efficiency in data communication and aggregation for AMI network. *J. Netw. Comput. Appl.* **2016**, *59*, 333–344. [CrossRef]
22. Lu, R.; Liang, X.; Li, X.; Lin, X.; Shen, X. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1621–1631.
23. Samanthula, B.K.; Jiang, W.; Madria, S. A probabilistic encryption based MIN/MAX computation in wireless sensor networks. In Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management (MDM), Milan, Italy, 3–6 June 2013; Volume 1, pp. 77–86.
24. Li, Q.; Cao, G. Efficient and privacy-preserving data aggregation in mobile sensing. In Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP), Austin, TX, USA, 30 October–2 November 2012; pp. 1–10.

25. Xiong, J.; Ma, R.; Chen, L.; Tian, Y.; Lin, L.; Jin, B. Achieving incentive, security, and scalable privacy protection in mobile crowdsensing services. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 8959635. [CrossRef]
26. Miao, C.; Su, L.; Jiang, W.; Li, Y.; Tian, M. A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
27. Zhang, Y.; Chen, Q.; Zhong, S. Efficient and Privacy-Preserving Min and $k$ th Min Computations in Mobile Sensing Systems. *IEEE Trans. Dependable Secure Comput.* **2017**, *14*, 9–21. [CrossRef]
28. Dietzel, S.; Bako, B.; Schoch, E.; Kargl, F. A fuzzy logic based approach for structure-free aggregation in vehicular ad-hoc networks. In Proceedings of the Sixth ACM International Workshop on VehiculAr InterNETworking, Beijing, China, 25 September 2009; ACM: New York, NY, USA, 2009; pp. 79–88.
29. Lochert, C.; Scheuermann, B.; Mauve, M. A probabilistic method for cooperative hierarchical aggregation of data in VANETs. *Ad Hoc Netw.* **2010**, *8*, 518–530. [CrossRef]
30. Considine, J.; Li, F.; Kollios, G.; Byers, J. Approximate aggregation techniques for sensor databases. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 2 April 2004; pp. 449–460.
31. Tao, Y.; Kollios, G.; Considine, J.; Li, F.; Papadias, D. Spatio-temporal aggregation using sketches. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, USA, 2 April 2004; pp. 214–225.
32. Zekri, D.; Defude, B.; Delot, T. Building, sharing and exploiting spatio-temporal aggregates in vehicular networks. *Mob. Inf. Syst.* **2014**, *10*, 259–285. [CrossRef]