Scalable Private Blocking Technique for Privacy-Preserving Record Linkage

Shumin Han(✉), Derong Shen, Tiezheng Nie, Yue Kou, and Ge Yu

College of Information Science and Engineering, Northeastern University, Shenyang, China
hanshumin_summer@yeah.net,
{shenderong,nietiezheng,kouyue,yuge}@ise.neu.edu.cn

Abstract. Record linkage is the process of matching records from multiple databases that refer to the same entities and it has become an increasingly important subject in many application areas, including business, government, and health. When we collect data which is about people from these areas, integrating such data across organizations can raise privacy concerns. To prevent privacy breaches, ideally records should be linked in a private way such that no information other than the matching result is leaked in the process, this technique is called privacy-preserving record linkage (PPRL). Scalability is one of the main challenges in PPRL, therefore, many private blocking techniques have been developed for PPRL. They are aimed at reducing the number of record pairs to be compared in the matching process by removing obvious non-matching pairs without compromising privacy. However, they vary widely in their ability to balance competing goals of accuracy, efficiency and security. In this paper, we propose a novel private blocking approach for PPRL based on dynamic *k*-anonymous blocking and Paillier cryptosystem. In dynamic *k*-anonymous blocking, our approach dynamically generates blocks satisfying *k*-anonymity and more accurate values to represent the blocks with varying *k*. We also propose a novel similarity measure method which performs on the numerical attributes and combines with Paillier cryptosystem to measure the similarity of two blocks in security, which provides strong privacy guarantees that none information reveals. Experiments conducted on a public dataset of voter registration records validate that our approach is scalable to large databases and keeps a high quality of blocking. We compare our method with other techniques and demonstrate the increases in security and accuracy.

Keywords: Record linkage · Private blocking · *k*-anonymity · Paillier cryptosystem · Scalability

1 Introduction

Nowadays, large amounts of data from several domains like businesses, government agencies and research projects has been generated, collected and stored. Matching records that relate to the same entities from several databases has been recognized to be of increasing importance in many application domains. However, when data about individuals or sensitive attributes is to be integrated across organizations, privacy has to be

F. Li et al. (Eds.): APWeb 2016, Part II, LNCS 9932, pp. 201–213, 2016.
DOI: 10.1007/978-3-319-45817-5_16

considered. Therefore, we need to protect these data from unauthorized disclosure. For example, in a decentralized healthcare system, where the personal medical records are distributed among several hospitals, it is critical to integrate the information of a patient without disclosing his/her sensitive attributes. Thus, making sure that privacy of individuals is maintained whenever databases are linked across organizations is vital.

Privacy-Preserving Record Linkage (PPRL) [1] is the process of identifying records from multiple data sources that refer to the same individuals, without revealing other information besides the matched records. PPRL has been widely used in many fields. For example, Microsoft has acquired Yahoo, by applying record linkage technique on their client databases, we cannot only obtain common clients between them but also acquire the potential new clients from Yahoo, which has significant business value for Microsoft. However, the client databases are confidential, exposing client data to other companies would cause heavy loss. Therefore comparing client databases without data disclosure excepting matched records is crucial.

Considering the growing large volumes of available data, developing PPRL which are scalable to large databases is necessary. Therefore, blocking techniques are developed. Blocking techniques are used to divide records into mutually exclusive blocks and only the records within the same block can be linked. A naive pair-wise comparison of two databases in record linkage has a quadratic complexity in their sizes. Thus, blocking techniques [2] reduce a large number of potential comparisons by removing as many record sets as possible that correspond to non-matches.

Private blocking [3] aims to generate candidate record pairs which are remained to perform PPRL without revealing any sensitive information that can be used to infer individual records and their attribute values. The k-anonymity and Paillier cryptosystem are two main privacy techniques which are applied on private blocking. Although many previous private blocking techniques have used these two privacy techniques [3, 4], there still exist some drawbacks to be solved. In [3], the Two-Party Private Blocking (TPPB) method avoids the use of a third party and cryptographic techniques, and instead, trades off privacy for blocking quality. In [4], Inan et al. suggest creating forming generalized hierarchies (FGH) for reducing the cost of PPRL. However, the forming hierarchies may cause the blocks over-generalization and reduce the accuracy of blocking. We propose a novel private blocking technique based on dynamic k-anonymous blocking and Paillier cryptosystem which can deal with the problems above. Our approach accurately creates blocks without revealing any private information and takes less time than previous approaches which apply cryptographic techniques.

The contributions of this paper are: (1) we propose a novel dynamic k-anonymous blocking algorithm which generates k-anonymous blocks and more accurate values to represent the blocks with varying k, the values are called representative values (RVs) in the following text. (2) We apply a cryptographic technique Paillier cryptosystem on the RVs of each block without revealing any information, which provides stronger privacy than previous approaches. And we propose a novel measure method which performs on the numerical attributes and combines with Paillier cryptosystem to measure the similarity of two blocks in security. (3) Experimental evaluation conducted on a real-world dataset shows our method has an advantage of keeping a high accuracy even k becoming very large. This advantage is meaningful because it is acknowledged that blocks become more secure with the increasing *k*. We compare our method with other techniques and demonstrate the increases in security and accuracy.

The remainder of this paper is organized as follows. In the following section we mention some previous works related to ours. In Sect. 3 we introduce definitions and background. In Sect. 4 we describe our approach. In Sect. 5 we analyze the privacy of our approach. In Sect. 6 we show its experimental evaluation. Finally we summarize our findings in Sect. 7.

2 Related Work

Due to the growing size of databases, various private blocking methods have been developed in recent years. Most methods rely on the use of a third party. Al-Lawati et al. [5] proposed a secure three-party blocking protocol in 2005 which achieves high performance PPRL by using secure hash encoding for computing the TF-IDF distance measure in a secure fashion. Inan et al. [4] proposed a hybrid approach that combines generalization and cryptographic techniques to solve the PPRL problem in 2008. An approach to PPRL was proposed by Karakasidis et al. [6] in 2011 that a secure blocking based on phonetic encoding algorithms. The records that have similar (sounding) values are divided into the same block. In 2012 a *k*-anonymous private blocking approach based on a reference table was proposed by Karakasidis et al. [7] for three-party PPRL techniques. Durham [8] proposed a framework for PPRL using Bloom filters in 2012. Recently, Karakasidis [9] proposed a novel privacy preserving blocking technique based on the use of reference sets and Multi-Sampling Transitive Closure for Encrypted Fields (MS-TCEF). As to the two-party techniques, Inan et al. [10] in 2010 presented an approach for PPRL based on differential privacy. The approach combines differential privacy and cryptographic methods to solve the PPRL problem in a two-party protocol. A two-party approach based on the use of Bloom filters for approximate private matching was developed by Vatsalan et al. [11] in 2012. Vatsalan [3] proposed an efficient Two-party private blocking based on privacy techniques *k*-anonymous clustering and public reference values.

The methods in [3, 4] are closest to our approach. However the approach in [3] uses public reference values as the RVs, although the attributes values of records are not revealed, to a certain degree, public reference values also expose some information about corresponding block. And when *k* becomes very large, the public reference values cannot sufficiently represent the blocks causing the quality of blocking reduces heavily. The approach in [4] uses forming generalized hierarchies to generate *k*-anonymous blocks, which may make the RVs over-generalization and reduces the accuracy of generating candidate pairs. We create blocks using dynamic *k*-anonymous blocking instead of forming hierarchies, which generates the RVs more accurately and flexibly. Applying Paillier cryptosystem provides a stronger guarantee of privacy, which takes less time than previous approaches that apply cryptographic techniques.

3 Preliminaries

3.1 Problem Formulation

We assume two databases D_A and D_B are to be matched, potentially each record from D_A needs to be compared with each record from D_B, resulting in a maximum number of $|D_A| \times |D_B|$ comparisons between two databases. Private blocking contributes to removing obvious non-matching pairs and generating candidate record pairs without revealing any information about the originating plaintexts, which reduces the complexity of comparisons. Considering the privacy, the process of private blocking is different from the traditional blocking. In private blocking, the records of one database should not be exposed to other parties. Further details involved in private blocking are outlined as follows [12]:

Blocking Key Selection. The blocking key is the criteria by which the records are partitioned.
Block Partitioning. Once a blocking key has been selected, this blocking key is as an input to partition each database *respectively* by the same principle where the output is a set of blocks and their RVs.
Candidate Blocks Generation. Given the blocks of each database, through measuring the similarity between the RVs, we can decide whether the records in two blocks compare, then the candidate record pairs would be generated.

3.2 *K*-anonymity

We now give the definitions of *k*-anonymity [13].

- Explicit Identifier is a set of attributes, such as name and social security number (SSN), containing information that explicitly identifies record owners;
- Quasi Identifier (*QI*) is a set of attributes that could potentially identify record owners;
- Sensitive Attributes consists of sensitive person-specific information such as disease, salary, and disability status;
- Non-Sensitive Attributes contains all attributes that do not fall into the previous three categories.

To prevent record linkage through *QI*, Samarati and Sweeney proposed [13] the notion of *k*-anonymity:

***k*-anonymity:** If one record in table *T* has some value *QI*, at least $k - 1$ other records also have the value *QI*. Table *T* is *k*-anonymity with respect to the *QI*.

In other words, the minimum group size on *QI* is at least *k*. In a *k*-anonymous table, each record is indistinguishable from at least $k - 1$ other records with respect to *QI*. Consequently, the probability of linking a victim to a specific record through *QI* is at most $1/k$. Consider a table *T* contains no sensitive attributes (such as the voter list). An attacker could possibly use the *QI* in *T* to link to the sensitive information in an external source. A *k*-anonymous *T* can still effectively prevent this type of record linkage without revealing the sensitive information. In this paper, the RVs are *QI*.

3.3 Paillier Cryptosystem

The Paillier cryptosystem [14], named and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public-private key cryptosystem. The scheme is an additive homomorphic cryptosystem, this means that, given only the public key and the encryption of m_1 and m_2, one can compute the encryption of $m_1 + m_2$. More formally, let $Enc_{k_{pub}}$ and $Dec_{k_{priv}}$ be the Paillier encryption and decryption functions with keys k_{pub} and k_{priv}, m_1 and m_2 be messages, $c(m_1)$ and $c(m_2)$ be ciphertexts such that $c(m_1) = Enc_{k_{pub}} c(m_1)$, $c(m_2) = Enc_{k_{pub}} c(m_2)$. So Homomorphic addition can be expressed by operators "·" and "+" as follow:

$$Dec_{k_{pri}}(c(m_1) \cdot c(m_2)) = m_1 + m_2. \tag{1}$$

4 Proposed Solution

Our proposed solution conducts private blocking by dynamic *k*-anonymous blocking and Paillier cryptosystem. It is composed of three parts: Data Preparation, Local *k*-anonymous Blocks Construction and Candidate Blocks Generation. The framework is described in Fig. 1.

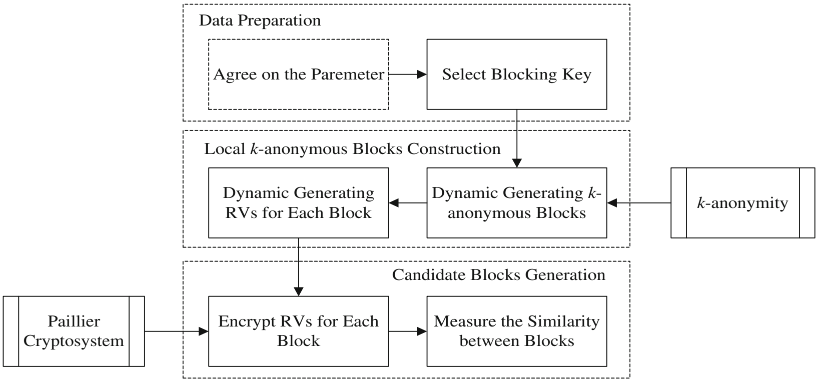

Data Preparation
Agree on the Paremeter
Select Blocking Key
Local *k*-anonymous Blocks Construction
Dynamic Generating RVs for Each Block
Dynamic Generating *k*-anonymous Blocks
k-anonymity
Candidate Blocks Generation
Paillier Cryptosystem
Encrypt RVs for Each Block
Measure the Similarity between Blocks

Fig. 1. The framework of our approach

4.1 Data Preparation

In data preparation, we agree on the parameters used in our approach and select one or more attributes as blocking keys.

Agree on the Parameter. We assume three participants in our method *Alice*, *Bob* and *Charlie*. *Alice* and *Bob* are the owners of databases D_A and D_B who participate in the protocol to perform private blocking on their databases. *Charlie* is used to generate candidate blocks or in other words decide whether to compare the records in two blocks. *Alice* and *Bob* agree on the parameter k that the minimum number of elements in a block.

Select Blocking Key. Blocking key is used to partition the records into blocks. Selecting an appropriate blocking key is necessary. To protect the privacy of blocks, our approach generates blocks satisfied k-anonymity, in other words each block contains at least k records. The method in [3] also uses k-anonymity and select Given name and Surname as blocking keys. However, when k becomes large, the RVs in method [3] cannot sufficiently represent the blocks causing the quality of blocking reduces heavily. To avoid the deficiency above, our approach selects the numerical attributes such as age, zip code or salary as the blocking key. The numerical attributes represent the blocks more accurately and flexibly with varying k. They also take less time than other attributes.

4.2 Local *k*-anonymous Blocks Construction

The local blocks construction phase partitions the records into blocks by blocking key. To construct blocks on distinct data sources without leaking any private information, our approach utilizes k-anonymity and Paillier cryptosystem privacy techniques. We generate k-anonymous blocks and obtain the RVs of each block using dynamic k-anonymous blocking algorithm.

Dynamic Generating *k*-anonymous Blocks. We suppose A_N (numerical attribute) is selected to be the blocking key, then we form blocks on the databases of *Alice* and *Bob* respectively. The blocks are divided by the values of blocking key, and each value of blocking key construct one block. After this, we obtain equivalence classes and sort them by the blocking key values (BKVs). Considering privacy, we merge equivalence classes until the number of records in a block being at least k. It provides k-anonymous privacy characteristics, as each record in the database can be seen as similar to at least $k-1$ other records. Algorithm 1 (which is executed independently by *Alice* and *Bob*) shows the main steps involved in the merging of equivalence classes to create k-anonymous blocks (Algorithm 1, lines 4–7).

Dynamic Generating RVs for Each Block. We assume L is a block satisfied k-anonymity, and x, y are the smallest and biggest BKVs in L. The RVs are composed by $[x, y]$. Then, the BKVs of each record in block L is replaced by $[x, y]$, more specifically each record in block L has at least $k-1$ records with the same BKVs. Therefore, the block L is k-anonymity respecting to $[x, y]$ and $[x, y]$ is the RVs of the block L.

Comparing the approach in [4], which uses forming generalized hierarchies may lead to the RVs over-generalization and reduce the accuracy of generating candidate blocks, our approach dynamically adjusts the RVs with the change of k and has a good influence on keeping high accuracy even k becoming very large. Algorithm 1 shows the main steps involved in dynamic generating the RVs of each block (Algorithm 1, line 8).

Algorithm 1: Dynamic *k*-anonymity Blocking

```
Input:
- E: Equivalence classes divided and sorted by $A_N$ $\{E_1, E_2, E_3,\ldots, E_n\}$
- Minimum number of elements in a block $k$
Output:
- $L_A$: Set of $k$-anonymous blocks $\{ L_{A1}, L_{A2}, L_{A3},\ldots, L_{Am} \}$
- $V[L_{Am}]$: RVs of $L_{Am}$
i=1; j=1; $L_{Aj}=\emptyset$;
while $i \le n$ do:
  $Kset = \emptyset$
    while $|L_{Aj}| \le k$ do:
        $L_{Aj} = L_{Aj} \cup E_i$
        $Kset.\mathrm{add}(E_i . A_N)$
        $i{+}{+}$
  $V[L_{Aj}] = [Kset[0], Kset[size-1]]$
  $j{+}{+}$
```

4.3 Candidate Blocks Generation

After generating *k*-anonymous blocks and corresponding RVs, we need to decide candidate blocks to eliminate record pairs that are expected to be non-matches. To protect the privacy of RVs and generate candidate blocks, Algorithm 2 shows the process that encrypts the RVs with Paillier and performs a novel measure method on the encrypted RVs to measure the similarity between blocks. And Fig. 2 shows the process of generating candidate blocks in privacy, from which we know that our approach is absolute security with none information revealing.

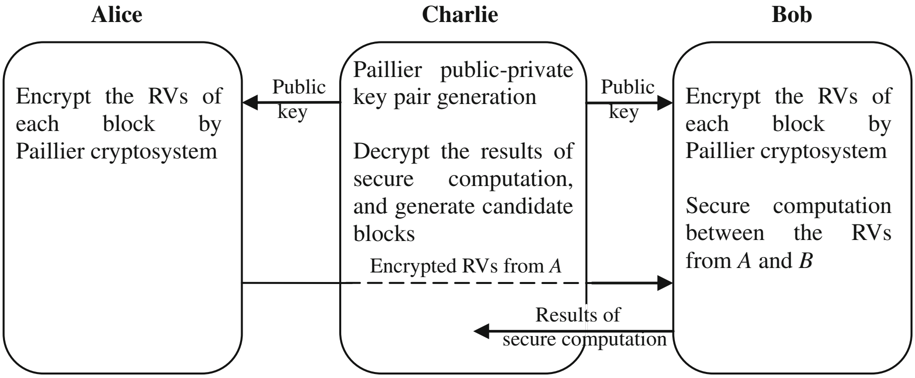

Alice
Charlie
Bob
Encrypt the RVs of each block by Paillier cryptosystem
Public key
Paillier public-private key pair generation
Decrypt the results of secure computation, and generate candidate blocks
Public key
Encrypt the RVs of each block by Paillier cryptosystem
Secure computation between the RVs from A and B
Encrypted RVs from A
Results of secure computation

Fig. 2. The process of generating candidate blocks in privacy

Encrypt RVs for Each Block. To measure the similarity between blocks, the RVs of blocks should be released by at least one data owner. Before releasing, the RVs in both *A* and *B* are encrypted by Paillier to guarantee privacy.

Charlie generates Paillier public-private key and send the public key to *A* and *B*. Then, *A* and *B* respectively encrypt their RVs with the public key (Algorithm 2, lines 3–5). We assume that the RVs of block L_A (from *A*) is [*a, b*] and the RVs of block L_B (from *B*) is [*c, d*]. The RVs are encrypted as follow:

$$c(-a) = Enc_{k_{pub}}(-a); \quad c(b) = Enc_{k_{pub}}(b) \tag{2}$$

$$c(-c) = Enc_{k_{pub}}(-c); \quad c(d) = Enc_{k_{pub}}(d); \quad c(-d) = Enc_{k_{pub}}(-d) \tag{3}$$

Measure the Similarity Between Blocks. After getting encrypted RVs in *A* and *B*, we pass the encrypted RVs in *A* to part *B*. In part *B* who lacks the private key, Bob cannot infer the plaintexts of records in *A*.

As to the party *B*, Bob has gained the encrypted RVs from *A*, then he uses the encrypted RVs of two blocks from *A* and *B* to decide whether two blocks match. We design a novel similarity measure method which combines with Paillier cryptosystem to measure the similarity between blocks (Algorithm 2, lines 7–16). The novel similarity measure method is expressed as follow:

$$\begin{cases} b < c \text{ or } d < a, & L_A \text{ and } L_B \text{ non - match} \\ b < d, & L_A \text{ and } L_B \text{ match, but } L_A \text{ does} \\ & \text{not match with other blocks in } B \\ \text{otherwise} & L_A \text{ and } L_B \text{ match} \end{cases} \tag{4}$$

According to the Homomorphic addition in Paillier cryptosystem:

$$Dec_{k_{pri}}(c(m_1) \cdot c(m_2)) = m_1 + m_2. \tag{5}$$

We can express our measure method as:

$$\begin{cases} Dec_{k_{pri}}(c(b) \cdot c(-c)) = b - c \\ Dec_{k_{pri}}(c(d) \cdot c(-a)) = d - a \\ Dec_{k_{pri}}(c(b) \cdot c(-d)) = b - d \end{cases} \tag{6}$$

Our novel similarity measure method combines well with the Paillier cryptosystem. We perform the secure computation $c(m_1) \cdot c(m_2)$ which designed in (6) in party *B* and send the results to *C*. Then *C* decrypts the results by the private key to get real results. Through judging the real results by (4), we could decide whether two blocks become candidate blocks. Therefore, in the whole process, our approach is absolute safe with none information revealing.

The last step PPRL conducts on each candidate record pairs individually by using a private matching technique, which should not reveal any information regarding the sensitive attributes and non-matches (this step is outside of our approach).

Algorithm2: Generating Candidate Blocks

```
Input:
- $V(L_A)$: RVs of each block in $A$ {[$a_1$, $b_1$], [$a_2$, $b_2$],…,[ $a_n$, $b_n$]}
- $V(L_B)$: RVs of each block in $B$ {[$c_1$, $d_1$], [$c_2$, $d_2$],…,[ $c_m$, $d_m$]}
Output:
- Candidate blocks match or non-match
for $i$=1; $i \leq n$; $i$++ do
    for $j$=1; $j \leq m$; $j$++ do
        $c(-a_i) = Enc_{k_{pub}}(-a_i)$; $c(b_i) = Enc_{k_{pub}}(b_i)$;
        $c(-c_j) = Enc_{k_{pub}}(-c_j)$; $c(d_j) = Enc_{k_{pub}}(d_j)$;
        $c(-d_j) = Enc_{k_{pub}}(-d_j)$;
        send $c\ (-a_i)$ and $c\ (b_i)$ to $B$
        $S_1 = c\ (b_i) \cdot c\ (-c_j)$; $S_2 = c\ (d_j) \cdot c\ (-a_i)$;
        $S_3 = c\ (b_i) \cdot c\ (-d_j)$;
        send $S_1$, $S_2$, $S_3$ to $C$
        if $Dec_{k_{priv}}(s_1) < 0$ or $Dec_{k_{priv}}(s_2) < 0$ then
            return non-match;
        else if $Dec_{k_{priv}}(s_3) < 0$ then
            return match;
            break;
        else
            return match;
```

5 Privacy Analysis

In this section we will discuss the privacy guarantees offered by our approach. We assume *Alice*, *Bob* and *Charlie* will follow the protocol honestly, but may try to infer private information based on messages they receive during the process without collusion [15]. Next we summarize the information that our approach discloses to each of the participants.

Alice: This party does not receive any messages regarding *Bob*'s database.

Bob: This party receives encrypted RVs of blocks from *A*. With the protection of Paillier cryptosystem, *B* cannot infer the real values from *A*.

Charlie: This party does not receive any messages regarding the RVs of blocks in *A* or *B* but receives the encrypted results of secure computation from *B*. After decrypting the encrypted results with private key, the real results only show final results without revealing the specific information from *A* and *B*. For example, *C* only knows the result of *b-c* and does not know the respective value of *b* and *c*.

6 Experiments

To perform the experimental analysis, we selected a publicly available dataset of real personal identifiers, derived from the North Carolina voter registration list (NCVR). The database NCVR contains 375,314 records. We selected attribute Age as the blocking key. For blocking evaluation, we need to generate two different sizes of datasets which are 10,000 and 100,000. Therefore, we respectively sampled 10,000 and 100,000 number of records randomly drawing from NCVR for *Alice*. Then we generated datasets for *B* composed of 10,000 and 100,000 records as well. Of these records, 8000 (80000) were randomly selected from NCVR (excluding those in *A*), while 2000 (20000) were randomly selected from *A*. The goal was to privately identify the 2000 (20000) matching records between *A* and *B*. Our experiments also perform on datasets of different sizes, we sampled 0.1 %, 1 %, 10 % and 100 % of records in the full database twice each for *A* and *B*. All tests were conducted on a computer server with a 64-bit, 8.0G of RAM Intel Core (3.30 GHz) CPU.

6.1 Evaluation Measures

We use the following measures to evaluate the performance of private blocking techniques in terms of complexity and quality of blocking. Complexity is evaluated by the total time required for blocking. We utilize reduction ratio (*RR*) and pair completeness (*PC*) as evaluation metrics for private blocking approaches [15]. Specifically, suppose c is the number of candidate record pairs produced by the private blocking, c_m is the number of true matches among c candidate pairs, $n = |D_A|\cdot|D_B|$ is the number of all possible pairs and n_m is the number of true matches among all pairs. Then, *RR* and *PC* are defined as follows:

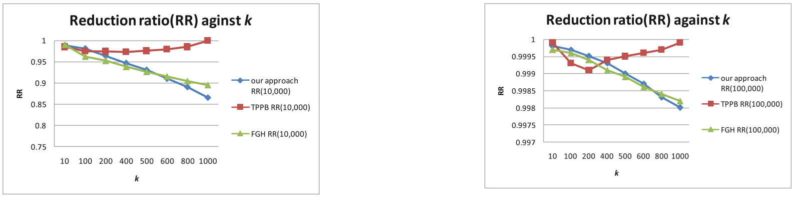
$$RR = 1 - c/n \quad PC = c_m/n_m \tag{7}$$

6.2 Performance Evaluation

We compare our approach with previous two approaches TPPB [3] and FGH [4]. The approach TPPB generates candidate blocks satisfying k-anonymity and uses public reference values as the RVs of blocks. Since each block consists of at least k records, only when revealing one reference value from each block can guarantee k-anonymity privacy. If several reference values are released by a block, the k-anonymity privacy would not be guaranteed. As to FGH, it generates k-anonymous blocks by forming generalized hierarchies.

We set the parameters of two approaches according to the settings provided by the authors [3, 4]. We compared three private blocking techniques on two different sizes of datasets which are 10,000 and 100,000 to measure the change of *RR*, *PC* and *blocking time* against k. The changing trends of *RR*, *PC* and *blocking time* against k are similar in two datasets. We also measure the *blocking time* with different dataset sizes for the three approaches. Then, we discuss the results of our experiments.

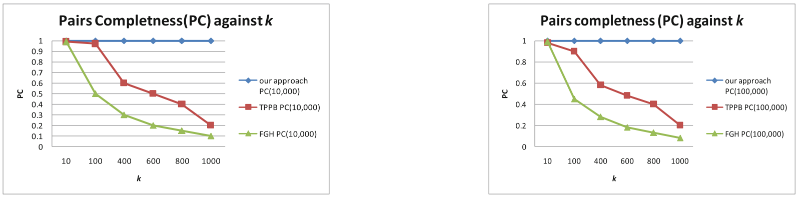
RR* with Varying *k. Figure 3 shows the *RR* with varying *k* in three approaches. Our approach and FGH keep a high *RR* with the increasing *k*. When *k* increases to 1000, *RR* is still above 0.86 in the smaller dataset. Towards TPPB, at first *RR* reduces when *k* is less than 200. Then, with *k* becoming bigger, *RR* increases and at last *RR* almost closes to 1. It can be explained that when *k* becomes larger, in TPPB, representing a block by only one reference value is not sufficient to represent all the values in block, which might lead to the number of candidate blocks reduces and the *RR* increases.



(a) (b)

Fig. 3. *RR* with different values for *k* (a) Dataset Size = 10,000 (b) Dataset Size = 100,000

PC* with Varying *k. Because of the reason above, some true candidate blocks being missed with the increasing *k*, therefore the *PC* reduces heavily in TPPB as shown in Fig. 4. In FGH, *PC* also reduces heavily with the reason that the bigger the *k* the higher level in the VGHs the records are generalized which may cause over-generalization. With regard to our approach, *PC* is always 1 on both datasets. This owns to our good similarity measure method.



(a) (b)

Fig. 4. *PC* with different values for *k* (a) Dataset Size = 10,000 (b) Dataset Size = 100,000

Blocking Time* with Varying *k. To the aspect of *blocking time* in Fig. 5, the *blocking time* reduces with *k* in three approaches because the number of resulting blocks (*n/k*) becomes less as *k* gets bigger. As shown in Fig. 5, the blocking time of our approach is more than the other two approaches. It is because that our approach applies Paillier cryptosystem.

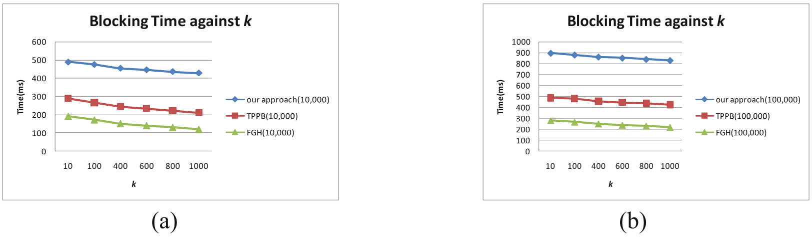


Fig. 5. *Blocking Time* with different values for *k* (a) Dataset size = 10,000 (b) Dataset size = 100,000

***Blocking Time* with Varying Database Sizes.** In Fig. 6, we compare the *blocking time* for three approaches with different dataset sizes. Our approach takes a little more time than the others with different dataset sizes. All the three approaches do not consider the communication cost. Through inferring, we can get the knowledge that all encrypted RVs are totally transmitted at most 500 times in our approach, which far less than the communication cost of previous approaches applying cryptographic techniques.

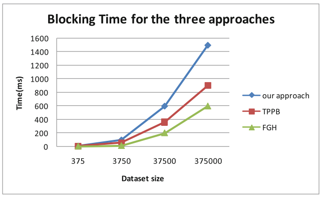


Fig. 6. *Blocking Time* with different dataset sizes for the three approaches

Hence, we conclude that our approach performs better in accuracy and privacy with a little loss of efficiency.

7 Conclusion

We present a novel scalable private blocking technique which is more accurate and secure than previous approaches. Dynamic *k*-anonymity blocking guarantees that each block has at least *k* records and meanwhile generates more accurate RVs with varying *k*. We also propose a novel similarity measure method which combines with Paillier cryptosystem and guarantees absolute security without revealing any information. As experiments show, our approach exhibits high performance both in accuracy and security with a little loss of *blocking time*. A limitation in our approach is the application of Three-Party Private Blocking. In future work, we plan to extend our approach to Multi-Party Private Blocking that is applicable for several datasets.

Acknowledgment. This work is supported by the National Basic Research 973 Program of China under Grant No. 2012CB316201, the National Natural Science Foundation of China under Grant No. 61472070.

References

1. Vatsalan, D., Christen, P., Verykios, V.S.: A taxonomy of privacy-preserving record linkage techniques. Inf. Syst. **38**(6), 946–969 (2013)
2. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE Trans. Knowl. Data Eng. **24**, 1537–1555 (2011)
3. Vatsalan, D., Christen, P., Verykios, V.S.: Efficient two-party private blocking based on sorted nearest neighborhood clustering. In: ACM CIKM (2013)
4. Inan, A., Kantarcioglu, M., Bertino, E., Scannapieco, M.: A hybrid approach to private record linkage. In: ICDE, pp. 496–505 (2008)
5. Al-Lawati, A., Lee, D., McDaniel, P.: Blocking-aware private record linkage. In: IQIS, pp. 59–68 (2005)
6. Karakasidis, A., Verykios, V.S.: Secure blocking + secure matching = secure record linkage. J. Comput. Sci. Eng. **5**, 223–235 (2011)
7. Karakasidis, A., Verykios, V.S.: Reference table based k-anonymous private blocking. In: 27th Annual ACM Symposium on Applied Computing, Trento (2012)
8. Durham, E.: A framework for accurate, efficient private record linkage. Ph.D. Thesis, Vanderbilt University (2012)
9. Karakasidis, A., Verykios, V.S.: Scalable blocking for privacy preserving record linkage. In: ACM KDD, Sydney (2015)
10. Inan, A., Kantarcioglu, M., Ghinita, G., Bertino, E.: Private record matching using differential privacy. In: EDBT, Lausanne, Switzerland, pp. 123–134 (2010)
11. Vatsalan, D., Christen, P.: An iterative two-party protocol for scalable privacy-preserving record linkage. In: Aus DM, CRPIT, Sydney, Australia, vol. 134 (2012)
12. Durham, E.A.: A framework for accurate, efficient private record linkage. Ph.D. thesis, Graduate School of Vanderbilt University, Nashville (2012)
13. Sweeney, L.: *k*-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst 10, 557–570 (2002)
14. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
15. Kuzu, M., Inan, A.: Efficient privacy-aware record integration. In: ACM EDBT (2013)