Data Sci. Eng. (2017) 2:187–196
DOI 10.1007/s41019-017-0041-5

Private Blocking Technique for Multi-party Privacy-Preserving Record Linkage

Shumin Han[1] · Derong Shen [1] · Tiezheng Nie[1] · Yue Kou[1] · Ge Yu[1]

Received: 12 January 2017 / Revised: 10 April 2017 / Accepted: 25 April 2017 / Published online: 26 May 2017

Abstract The process of matching and integrating records that relate to the same entity from one or more datasets is known as record linkage, and it has become an increasingly important subject in many application areas, including business, government and health system. The data from these areas often contain sensitive information. 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, and this technique is called privacy-preserving record linkage (PPRL). With the increasing data, scalability becomes the main challenge of PPRL, and 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, most of them are designed for two databases and 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 which can be applied on two or multiple databases. 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 or more blocks in security, which provides strong privacy guarantees that none information reveals even collusion. 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 Privacy-preserving record linkage · Private blocking · k-anonymity · Paillier cryptosystem · Scalability

1 Introduction

As the world is moving into the Big Data era, large amounts of data from several organizations require to be integrated. Due to privacy and confidentiality concerns, these organizations are not willing or allowed to reveal their sensitive and personal data to other database owners. Therefore, we need to protect these data from unauthorized disclosure. For example, in a decentralized health-care system, where the personal medical records are distributed among several hospitals, it is critical to integrate the information belonging to 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 two or more data sources that refer to the same individuals, without revealing any private or sensitive information. PPRL has been widely used in many fields. For example, Microsoft has acquired Yahoo, by applying record linkage technique on their client databases, and we can not only obtain common clients between them, but also acquire the potential new clients from Yahoo, which has significant business value for

✉ Shumin Han
hanshumin_summer@yeah.net

[1] College of Computer Science and Engineering, Northeastern University, Shenyang, China

Microsoft. However, the client databases are confidential, and 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 and the increasing number of parties, blocking [2] is a possible solution aimed at improving scalability, which is used to divide records into mutually exclusive blocks, and only the records within the same block can be linked. A naive pair-wise comparison across P databases of n records is n^P. The computation and communication complexities increase significantly with multiple parties. Thus, concentrating on the study of multi-party blocking techniques is the key to improve scalability.

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. So far, there have been many private blocking techniques proposed for two or more databases, and there still exist some drawbacks to be solved. As to the approaches between two databases: 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. As to the approaches among multiple parties: In multiple parties, the risk of collusion increases, where a subset of parties collude in order to learn about other parties' sensitive data. In [5] and [6], the degree of privacy preserving cannot against collusion among the database owners. 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, and the values are called representative values (RVs). (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 or more blocks in security. (3) We propose a multi-party private blocking approach which can against collusion among multiple owners and reduce time cost by multi-thread concurrent mechanism. (4) Experimental evaluation conducted on a real-world dataset shows our method has an advantage of keeping a high accuracy even k becoming very large. 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. As to the methods between two databases, most methods rely on the use of a third party. Al-Lawati et al. [7] 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. [8] in 2011 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. [9] for three-party PPRL techniques. Durham [10] proposed a framework for PPRL using Bloom filters in 2012. Recently, Karakasidis [11] 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. [12] 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. [13] in 2012. Vatsalan [3] proposed an efficient two-party private blocking based on privacy techniques k-anonymous clustering and public reference values. As to the methods among multiple parties, there only few have been proposed. The latest two methods [5, 6] were, respectively, proposed in 2015 and 2016. In these two papers, they preserve the privacy of records by applying Bloom filters.

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, and 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. So 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. The approaches applying Bloom filters in [5, 6] protect the privacy of records to some degree, but they still cannot against collusion among multiple owners.

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 against collusion, which takes less time than previous approaches that apply cryptographic techniques.

3 Preliminaries

3.1 Problem Formulation

We assume P databases of n records $D_1, D_2, \ldots, D_P$ are to be matched, and potentially each record from D_i $(1 \le i \le P)$ needs to be compared with each record from D_j $(1 \le j \le P)$, resulting in a maximum number of n^P comparisons among P 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 [14]:

Blocking Key Selection 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 among the RVs, we can decide whether the records in multiple blocks compare; then, the candidate record pairs would be generated.

3.2 k-anonymity

We now give the definitions of k-anonymity [15].

- 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 consist of sensitive person-specific information such as disease, salary and disability status;
- Non-Sensitive Attributes contain all attributes that do not fall into the previous three categories.

To prevent record linkage through QI, Samarati and Sweeney proposed [15] 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 [16], 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, and 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_{kpub} and Dec_{kpriv} 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 $c(m_1) = Enc_{kpub}\ (m_1)$, $c(m_2) = Enc_{kpub}$ (m_2). So homomorphic addition can be expressed by operators "$\cdot$" and "$+$" as follows:

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

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.

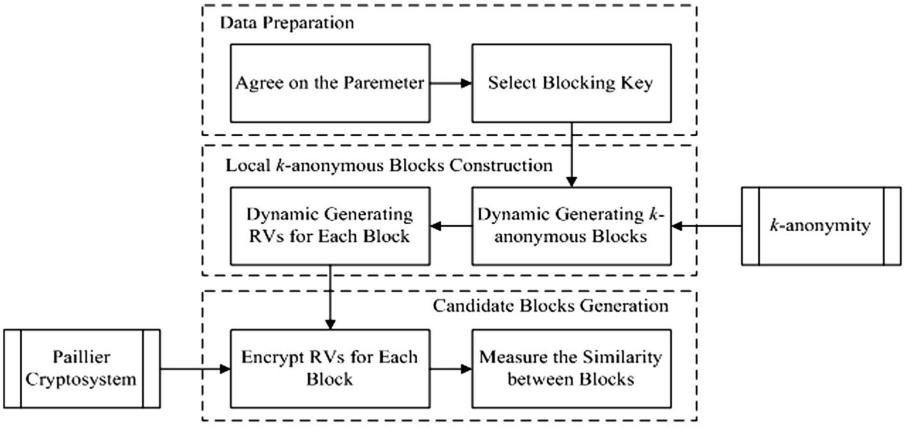

Data Preparation
Agree on the Paremeter
Select Blocking Key
Local k-anonymous Blocks Construction
Dynamic Generating RVs for Each Block
Dynamic Generating k-anonymous Blocks
k-anonymity
Candidate Blocks Generation
Paillier Cryptosystem
Encrypt RVs for Each Block
Measure the Similarity between Blocks

Fig. 1 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 p ($p \geq 2$) participants in our method $P_1, P_2, \ldots, P_p$ who participate in the protocol to perform private blocking on their databases. Decision unit (DU) is used to generate candidate blocks or in other words decide whether to compare the records among n blocks. P_1, P_2, ..., and P_p agree on the parameter k 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 satisfying k-anonymity and protects the RVs by Paillier cryptosystem. 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. The RVs also expose some information about corresponding blocks. To avoid the deficiency above, our approach selects the numerical attributes such as age, zip code (consisting of numbers) or salary as the blocking key. The numerical attributes represent the blocks more accurately and flexibly with varying k. And when we apply Paillier cryptosystem, the computational demand for numeric attributes is much less than for string attributes. Thus, selecting numerical attributes as blocking key can improve the scalability and be applied in many real-world scenarios.

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 P_1, P_2, ..., and P_p ($p \geq 2$), respectively. The blocks are divided by the values of blocking key, and each value of blocking key constructs 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 p databases) shows the main steps involved in the merging of equivalence classes to create k-anonymous blocks (Algorithm 1, lines 4–7).

```
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 $L_{Am}$
1:  i=1; j=1; $L_{Aj}$=Ø;
2:  while $i \leq n$ do:
3:      Kset= Ø
4:          while $|L_{Aj}| \leq k$ do:
5:              $L_{Aj} = L_{Aj} \cup E_i$
6:              Kset.add($E_i . A_N$)
7:              i++
8:      V[$L_{Aj}$]= [Kset[0], Kset[size-1]]
9:      j++
```

Dynamic Generating RVs for Each Block We assume L is a block satisfying 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 are 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, lines 8).

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. Firstly, in Sect. 4.3.1, to protect the privacy of RVs and generate candidate blocks, we encrypt the RVs with Paillier and propose a novel measure method on the encrypted RVs to measure the similarity between two blocks. Then, we extend the method to measure the similarity among multiple blocks and reduce the time cost by using the multi-thread concurrent mechanism in Sect. 4.3.2. At last, we take an example between two blocks to illustrate our method.

4.3.1 Approach for Two Datasets

In this part, we assume two participants in our method Alice (A) and Bob (B) who are the owners of databases D_A and D_B. Decision unit (DU) is used to decide whether to compare the records between two blocks.

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. DU generates Paillier public–private key and sends 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) are $[a, b]$ and the RVs of block L_B (from B) are $[c, d]$. The RVs are encrypted as follows:

$$c(-a) = Enc_{kpub}(-a); c(b) = Enc_{kpub}(b) \tag{2}$$

$$\begin{aligned} &c(-c) = Enc_{kpub}(-c); c(d) = Enc_{kpub}(d) \\ &c(-d) = Enc_{kpub}(-d) \end{aligned} \tag{3}$$

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 follows:according to

$$\begin{array}{l} b < c \text{ or } d < a, \\ b < d, \\ \text{ but } L_A \text{ does not match with other blocks in } B \\ \text{otherwise,} \end{array} \tag{4}$$

According to the Homomorphic addition in Paillier cryptosystem:

$$Dec_{kpriv}(c(m_1) \cdot c(m_2)) = m_1 + m_2 \tag{5}$$

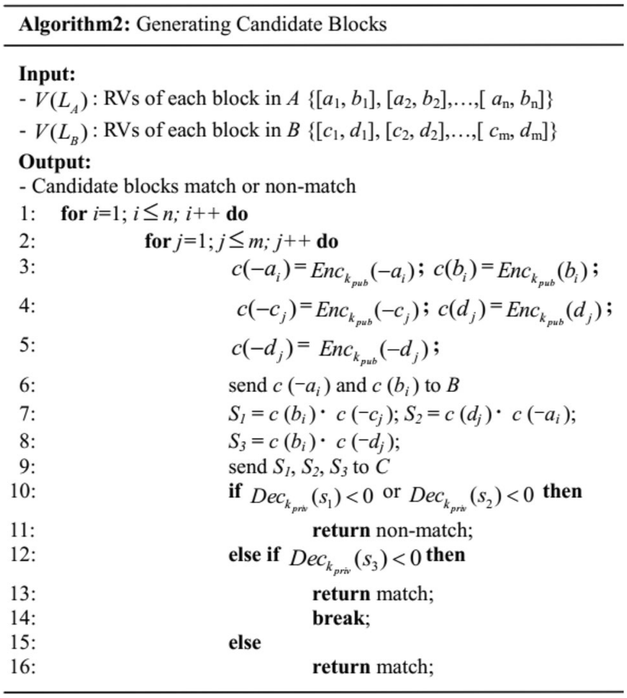
We can express our measure method as:

$$\begin{aligned} &Dec_{kpriv}(c(b) \cdot c(-c)) = b - c \\ &Dec_{kpriv}(c(d) \cdot c(-a) = d - a \\ &Dec_{kpriv}(c(b) \cdot c(-d) = b - d \end{aligned} \tag{6}$$

As Eq. (4) shows, if $b <\ c$ or $d <\ a$, it means L_A and L_B have no intersection. So, they are non-match. Otherwise, L_A and L_B are match. If we assume $b_1 <\ d_1$ in L_{A1} and L_{B1}, as algorithm 1 shows, we know $c_2 gt\ d_1$, so we can infer $b_1 <\ c_2$, L_{A1} and L_{B2} are non-match. By that analogy, L_{A1} also does not match with $L_{B3}, L_{B4}, \ldots, L_{Bm}$.

Our novel similarity measure method combines well with the Paillier cryptosystem. We perform the secure computation $c(m_1)c(m_2)$ which is designed in (6) in party B and send the results to DU. Then, DU 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 unconditioned safe with none of the 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 $\{[a_1, b_1], [a_2, b_2],\ldots,[a_n, b_n]\}$
- $V(L_B)$: RVs of each block in B $\{[c_1, d_1], [c_2, d_2],\ldots,[c_m, d_m]\}$

Output:
- Candidate blocks match or non-match

```
for $i$=1; $i \le n$; $i$++ do
    for $j$=1; $j \le m$; $j$++ 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;
```

4.3.2 *Approach for Multiple Datasets*

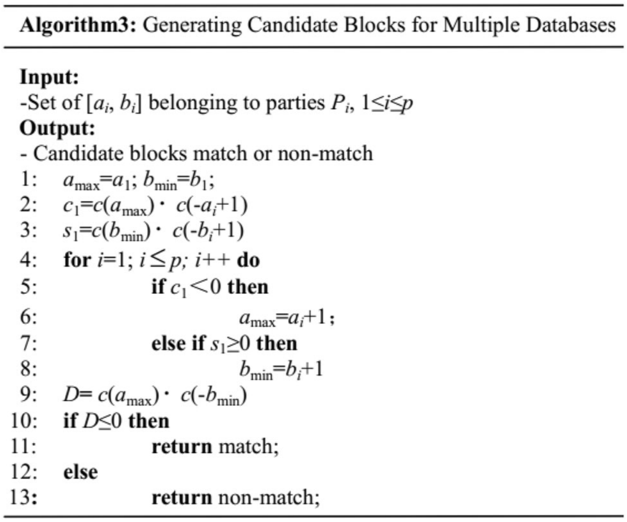
In this part, we assume p participants in our method $P_1, P_2, \ldots, P_p$ who are the owners of databases $D_1, D_2, \ldots, D_p$. Decision unit (DU) is used to decide whether to compare the records among multiple blocks.

Encrypt RVs for Each Block As similar with the measure method between two blocks, at first, DU generates Paillier public–private key and sends the public key to each block. Then, each block, respectively, encrypts its RVs with the public key. We assume that the RVs of block L_i (from P_i) are $[a_i, b_i]$ $(1 \le i \le p)$.

Measure the Similarity between Blocks After getting encrypted RVs in each block, we transmit the encrypted RVs to next part starting from P_1. Gaining the encrypted RVs from P_1, P_2 performs the secure computations on the encrypted RVs (Algorithm 3, lines 1–3). Then, we send the computation results c_1 and s_1 to the DU. By receiving the R_1 from DU, we transmit the bigger a and the smaller b to the next part. And so on, we obtain $a_{\max}$ and $b_{\min}$ until the last part (Algorithm 3, lines 4–8). At last, we design a novel similarity measure method which combines with Paillier cryptosystem to measure the similarity among multiple blocks (Algorithm 3, lines 9–13). The whole process is shown in Fig. 2, and the novel similarity measure method is expressed as follows:

$$\begin{aligned} c(a_{\max})\cdot c(-b_{\min}) \le 0, &\quad L_1\ L_2,\ldots,\text{and } L_p \text{ match} \\ \text{otherwise}, &\quad L_1\ L_2,\ldots,\text{and } L_p \text{ non}-\text{match} \end{aligned} \tag{7}$$

As Eq. (7) shows, if $c(a_{\max})\cdot c(-b_{\min}) \le 0$, it means the p blocks have intersection, so we decide the p blocks match. Otherwise, they are non-match.



Algorithm3: Generating Candidate Blocks for Multiple Databases

Input:
-Set of $[a_i, b_i]$ belonging to parties P_i, $1 \le i \le p$

Output:
- Candidate blocks match or non-match

```
1:  $a_{max}=a_1$; $b_{min}=b_1$;
2:  $c_1=c(a_{max})\cdot c(-a_i+1)$
3:  $s_1=c(b_{min})\cdot c(-b_i+1)$
4:  for $i$=1; $i \le p$; $i$++ do
5:          if $c_1<0$ then
6:                  $a_{max}=a_i+1$;
7:          else if $s_1 \ge 0$ then
8:                  $b_{min}=b_i+1$
9:  $D= c(a_{max})\cdot c(-b_{min})$
10: if $D \le 0$ then
11:         return match;
12: else
13:         return non-match;
```

Reduce the Time Cost From the algorithm 3, we know that there are p–1 secure computations in it. Each secure computation takes much more time than the time used to compare the plaintexts. Therefore, we propose using the multi-thread concurrent mechanism to deal with the algorithm 3 as shown in algorithm 4 which can reduce half of time than the algorithm before.

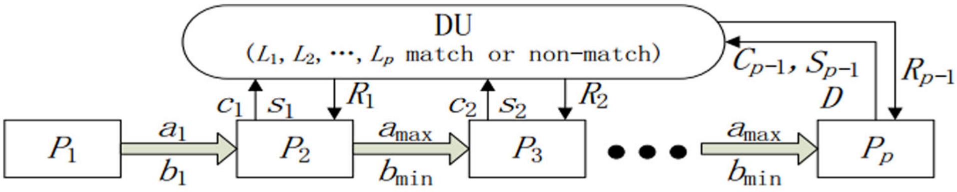


Fig. 2 Process of generating candidate blocks among multiple parties

Algorithm4: Generating Candidate Blocks for Multiple Databases

```
Input:
-Set of [$a_i$, $b_i$] belonging to parties $P_i$, $1 \leq i \leq p$
Output:
- Candidate blocks match or non-match
findmami (1, p);
g=2/p
if n>1 then
        Thread a = new Thread (findmami(1, g)).start();
        Thread b = new Thread (findmami(g+1, p)).start();
Thread.join();
else return $a_{max}$, $b_{min}$;
```

4.3.3 Example of Our Approach Between Two Blocks

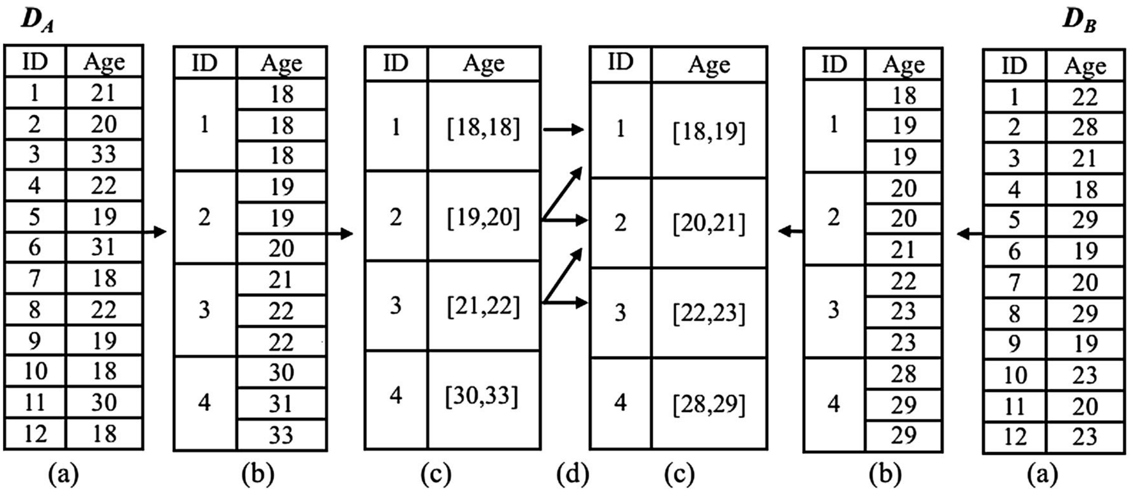
In this part, we take an example to illustrate our approach except the part of Paillier cryptosystem and describe the process of generating candidate blocks in privacy. We select age as blocking key and $k = 3$. In Fig. 3a, we choose twelve records in D_A and D_B to perform private blocking. In Fig. 3b, A and B sort by BKVs and generate k-anonymous blocks, respectively. In Fig. 3c, we obtain the RVs in A and B. Then in Fig. 3d, we apply proposed similarity measure method to decide which blocks match. For example, as shown in Fig. 3c, we choose the block L_A which ID = 2 to compare with the block L_B which ID = 2. The RVs are respective [19, 20] and [20, 21] of L_A and L_B, so $a = 19$, $b = 20$, $c = 20$, $d = 21$. Firstly, we compute $c(r_1) \ = c(20)c(-20), c(r_2) = c(21)c(-19), c(r_3) = c(20)$ $c(-21)$ according to 4.3 (6) in B. Then, we send the results to C and decrypt them. We would get $r_1 = 0, r_2 = 2$ $> 0, r_3 = -1 < 0$, and through judging by 4.3 (4), we decide L_A and L_B match, but L_A does not match with other blocks in B.

5 Privacy Analysis

In this section, we will discuss the privacy guarantees offered by our approach. We assume that all parties will follow the protocol honestly, but may try to infer private information based on messages they receive during the process or collusion [15]. Next we summarize the information that our approach discloses to each of the participants. Firstly, a pair of private and public keys is generated for encrypting and decrypting the RVs.The public key is known to all parties while the private key is known only to the DU. $P_i(1 \leq i \leq n)$: Each party receives encrypted RVs of blocks and sends the encrypted results of secure computation to the DU. Without knowing the private key, a party cannot decrypt the received RVs, and therefore, colluding with a party to learn another party's RVs would be impossible. DU: This party only receives the encrypted results of secure computation from $P_i(1 \leq i \leq p)$. After decrypting the encrypted results with private key, the real results only show the final results without revealing the specific information from each part. Thus, we can conclude Paillier cryptosystem can guarantee our approach is unconditioned safe.

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) [17]. We selected attribute age as the blocking key. To evaluate the scalability of multi-part blocking method, we need to generate two different sizes of datasets which are 10,000 and 100,000 for 2, 3, 5, 7 and 10 parties. Therefore,



(a) D_A

ID	Age
1	21
2	20
3	33
4	22
5	19
6	31
7	18
8	22
9	19
10	18
11	30
12	18

(b)

ID	Age
1	18
	18
	18
2	19
	19
	20
3	21
	22
	22
4	30
	31
	33

(c)

ID	Age
1	[18,18]
2	[19,20]
3	[21,22]
4	[30,33]

(c)

ID	Age
1	[18,19]
2	[20,21]
3	[22,23]
4	[28,29]

(b)

ID	Age
1	18
	19
	19
2	20
	20
	21
3	22
	23
	23
4	28
	29
	29

(a) D_B

ID	Age
1	22
2	28
3	21
4	18
5	29
6	19
7	20
8	29
9	19
10	23
11	20
12	23

Fig. 3 (**a**) Example databases held by $A(D_A)$ and $B(D_B)$ with blocking key values based on age. (**b–d**) Illustrate the protocol, which is described in Sect. 4.3.1, $k = 3$

we, respectively, sampled 10,000 and 100,000 records randomly drawn from NCVR for each party. Of these records, 8000 (80,000) were randomly selected from NCVR (excluding those in other parties), while 2000 (20,000) were selected the same as other parties. The goal was to privately identify the 2000 (20,000) matching records between two or more blocks. Our experiments also perform on datasets of different sizes, and we sampled 0.1%, 1%, 10% and 100% of records in the full database for each part. 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 [18]. 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, $n = |D_A||D_B|$ 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:

$$RR = 1 - c/n \qquad PC = c_m/n_m\,. \tag{8}$$

6.2 Performance Evaluation

As to the two datasets, 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.

As to the multiple databases, we evaluate our multi-party private blocking method by *RR*, *PC* and blocking time against k and the number of participants p. We also evaluate the improved algorithm which uses the multi-thread concurrent mechanism by blocking time.

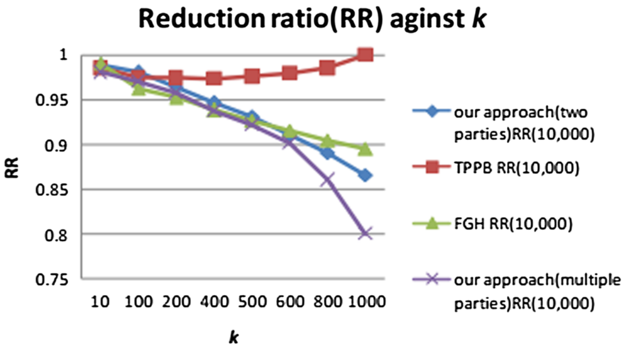


Fig. 4 *RR* with different values for k, dataset size = 10,000

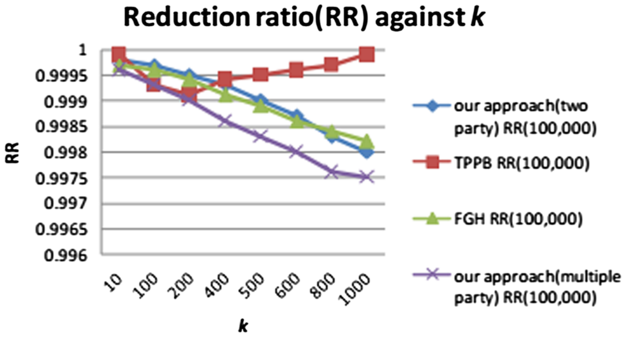


Fig. 5 *RR* with different values for k, dataset size = 100,000

RR with Varying k Figures 4 and 5 show the *RR* with varying k in three approaches and our multi-part blocking method, $P = 3$. Our approach (two parties) 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. Toward 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. The *RR* of our approach (multiple parties) is a little lower than the *RR* of our approach (two parties).

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 Figs. 6 and 7. 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 (two and multiple parties), *PC* is always 1 on both datasets. This owns to our better similarity measure method.

Blocking Time with Varying k To the aspect of *blocking time* in Figs. 8 and 9, the *blocking time* reduces with k in three approaches because the number of resulting blocks

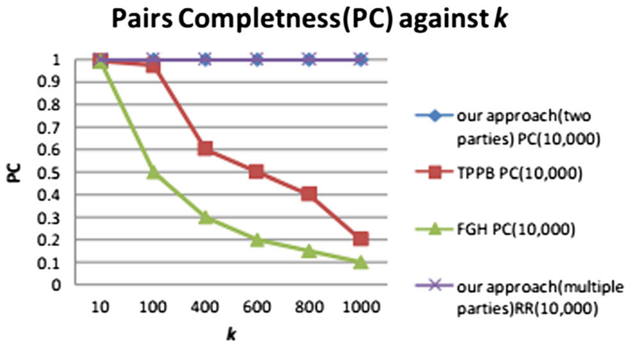


Fig. 6 *PC* with different values for *k*, dataset size = 10,000

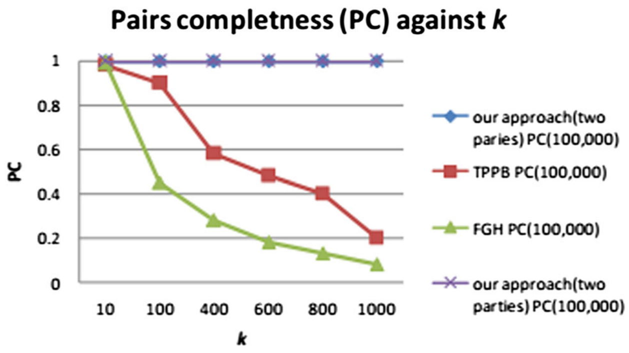


Fig. 7 *PC* with different values for *k*, dataset size = 100,000

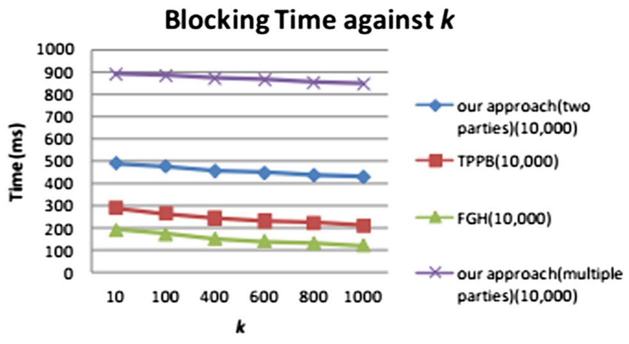


Fig. 8 *Blocking time* with different values for *k*, dataset size = 10,000

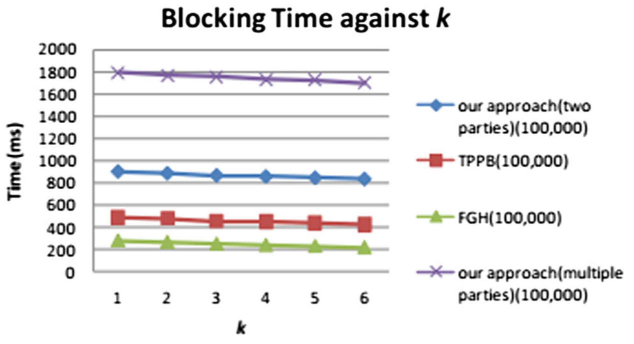


Fig. 9 *Blocking Time* with different values for *k*, dataset size = 100,000

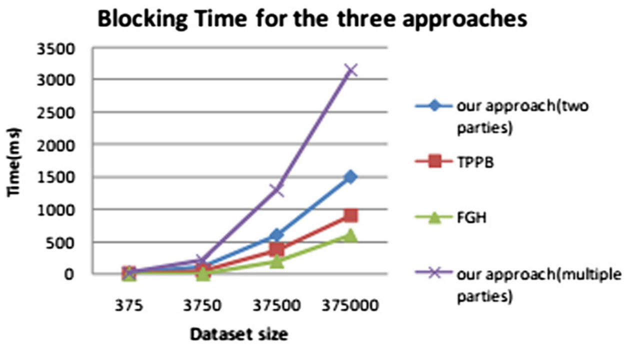


Fig. 10 *Blocking Time* with different dataset sizes for the four approaches

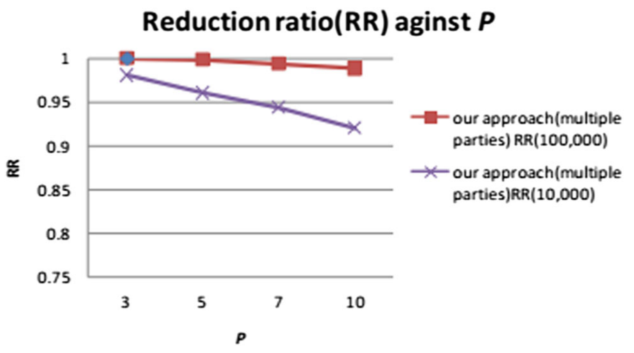


Fig. 11 (*RR* with varying *p* for multi-part private blocking

(*n*/*k*) becomes less as *k* gets bigger. As shown in Figs. 8 and 9, the blocking time of our approach is more than the other two approaches. It is because that our approach applies Paillier cryptosystem.

Blocking Time with Varying Database Sizes In Fig. 10, 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 are far less than the communication cost of previous approaches applying cryptographic techniques.

RR with Varying p In Fig. 11, we assume *k*=10 and measure the *RR* of our multi-part private blocking approach with the change of *p*. As Fig. 11 shows, *RR* reduces with the change of *p*.

PC with Varying p In Fig. 12, we assume *k*=10 and measure the *PC* of our multi-part private blocking approach with the change of *p*. As Fig. 12 shows, *PC* always keeps 1 with the change of *p*.

Blocking Time with Varying p In Fig. 13, we assume *k* = 10 and measure the blocking time of our multi-part private

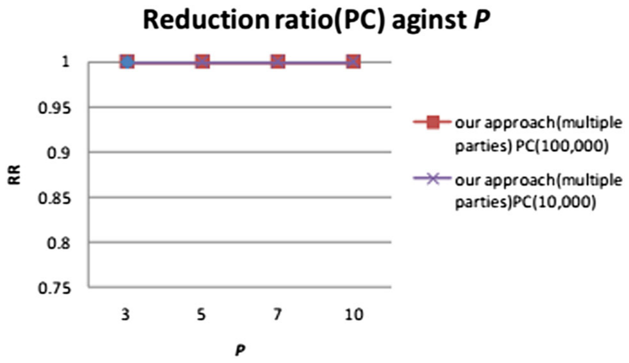


Fig. 12 *PC* with varying *p* for multi-part private blocking

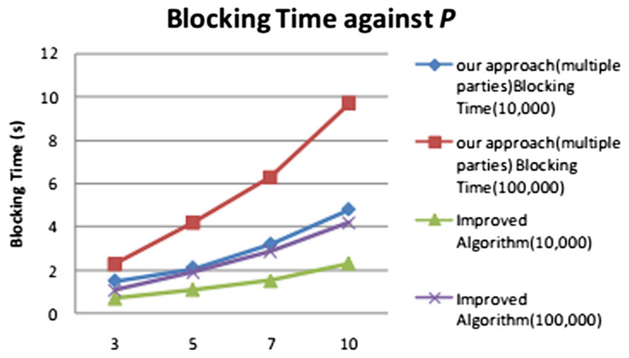


Fig. 13 *Blocking Time* with varying *p* for multi-part private blocking

blocking approach with the change of *p*. We also evaluate the blocking time of improved algorithm which uses the multi-thread concurrent mechanism. As Fig. 13 shows, the improved algorithm reduces the time effectively. 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. We extend this effective measure method to multiple parties which can avoid collusion. As experiments show, our approach exhibits high performance both in accuracy and security with a little loss of blocking time.

Acknowledgements This work is supported by the National Basic Research 973 Program of China under Grant No. 2012CB316201 and the National Natural Science Foundation of China under Grant Nos. (61472070, 61672142).

References

1. Vatsalan D, Christen P, Verykios VS (2013) A taxonomy of privacy-preserving record linkage techniques. Inf Syst 38(6):946–969
2. Christen P (2011) A survey of indexing techniques for scalable record linkage and deduplication. IEEE Trans Knowl Data Eng
3. Vatsalan D, Christen P, Verykios VS (2013) Efficient two-party private blocking based on sorted nearest neighborhood clustering. In: ACM CIKM
4. Inan A, Kantarcioglu M, Bertino E, Scannapieco M (2008) A hybrid approach to private record linkage. In: ICDE. pp 496–505
5. Ranbaduge T, Vatsalan D, Christen P (2015) Clustering-based scalable indexing for multi-party privacy-preserving record linkage. In: PAKDD. pp 549–561
6. Ranbaduge T, Vatsalan D, Christen P, Verykios V (2016) Hashing-based distributed multi-party blocking for privacy-preserving record linkage. In: PAKDD. pp 415–427
7. Al-Lawati A, Lee D, McDaniel P (2005) Blocking-aware private record linkage. In: IQIS. pp 59–68
8. Karakasidis A, Verykios VS (2011) Secure blocking + secure matching = secure record linkage. J Comput Sci Eng 5:223–235
9. Karakasidis A, Verykios VS (2012) Reference table based *k*-anonymous private blocking. In: 27th annual ACM symposium on applied computing. Trento
10. Durham E (2012) A framework for accurate, efficient private record linkage. Ph.D. Thesis, Vanderbilt University
11. Karakasidis A, Verykios VS (2015) Scalable blocking for privacy preserving record linkage. In: ACM KDD. Sydney
12. Inan A, Kantarcioglu M, Ghinita G, Bertino E (2010) Private record matching using differential privacy. In: EDBT. Lausanne, Switzerland, pp 123–134
13. Vatsalan D, Christen P (2012) An iterative two-party protocol for scalable privacy-preserving record linkage. In: Aus DM, CRPIT, vol 134. Sydney, Australia
14. Durham EA (2012) A framework for accurate, efficient private record linkage. Ph.D. Thesis, Graduate School of Vanderbilt University, Nashville
15. Sweeney L (2002) k-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl Based Syst 10:557–570
16. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT'99. pp 223–238
17. Vatsalan D, Christen P (2014) Scalable privacy-preserving record linkage for multiple databases. In: ACM CIKM. Shanghai, pp 1795–1798
18. Kuzu M, Inan A (2013) Efficient privacy-aware record integration. In: ACM EDBT