Privacy-Preserving Record Linkage via Bilinear Pairing Approach

Chih-Hsun Lin and Chia-Mu Yu
National Chung Hsing University, Taiwan

***Abstract*—In the era of big data, people are increasingly focusing on the useful information of various sources and looking for potential relation hidden in the data. Privacy-preserving record linkage (PPRL) is a means for finding the correspondence of records from different datasets with the guarantee of no privacy leakage from individuals. Here, we propose a simple yet effective PPRL protocol as a platform for the information mining in the real world. We perform an implementation to test the feasibility and efficiency of our proposed protocol.**

I. Introduction

With the big data revolution, more and more organizations collect data and perform analysis so that they can mine interesting patterns or information, in order to promote scientific, industrial and social benefits. Such practices often require data from various parties to link and integrate. However, analyzing and even finding the relation among big data from various parties will bring security and privacy issues.

Anonymizing and publishing data are a strategy. However, without a careful design, it can cause harm. For example, the Netflix releases the anonymized data, looking for the performance improvement, but such a data with insufficient consideration of privacy leakage has also become a classic example of personal information leakage. Therefore, PPRL which can establish the linkage between records from two privacy-preserving datasets plays an important role.

In many cases, the use of non-interactive means to release public information is still difficult to achieve. Even if we can make a trade-off between privacy and utility, it still has to consider the attackers' background knowledge. Therefore, our PPRL adopts interactive query by restricting objects and the number of requests to discourage the leakage of information.

A. Recent Work on PPRL

Many efforts have been devoted to developing techniques for PPRL. The goals of the research can be generally divided into three directions: privacy, efficiency and correctness.

Several studies focus on pursuing efficiency of PPRL, by using blocking design to reduce the number of comparisons required. Common practices include Bloom filter, clustering, and tree structure. These designs are mainly based on the dataset structure. However, these techniques, while effective in suppressing huge amounts of computation, have affected the correctness.

In addition, several studies focus on the correctness on PPRL, through the similarity (distance) calculation to determine whether match/non-match. These studies mainly deal with the processing and analysis of data types, such as weight evaluation, missing value processing.

Very recently, the notion of *differential privacy* (DP) has been adopted to strike the balance among privacy, efficiency and correctness. In this research, we focus only on the exact match of records. For the user, he or she does not want the wrong record linkage to cause serious real harm (privacy invaded). In this direction, S. Ma et al. [2] propose a method that can test equality with flexible authorization. The interactive operation from [2] inspires us to design the protocol.

B. Our Contribution

We propose a simple yet effective PPRL. Despite its burden in terms of computation overhead, we use the blocking mechanism to reduce the computation time. To demonstrate the feasibility and efficiency of the proposed PPRL, we make a prototype implementation to show the improved efficiency.

II. System Model

System model of our proposed scheme is shown in Figure 1. First, both parties register with the third party and generate secret/token. In addition, they also need to make key pairs (PK, SK) for communication via any *public-key cryptography* (PKC). Then, they encrypt their data (each value of the attribute) with the secret, and send it to the semi-trusted server. When they through the secure channel to coordinate the time, the process of authorization-generating will add the object, token, and time stamp. Finally, they send it to the server to perform the linkage test.

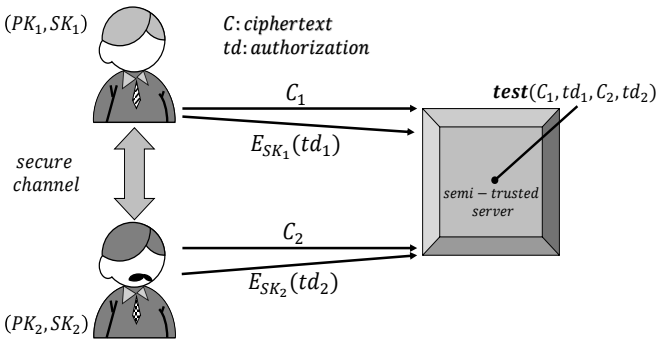

Fig. 1: System model for privacy-perserving record linkage
C: ciphertext; td: authorization

III. Preliminaries And Definitions

In this section, we outline the bilinear pairing technique [3], which will serve as the basis of our proposed scheme.

Bilinear pairing is a property of elliptic curve. Let $\mathbb{G}_1$ be an additive cyclic groups, and $\mathbb{G}_2$ be a multiplicative group. They have same prime order q. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ be an admissible bilinear mapping that satisfies the following properties:

1) ***Bilinearity****:* $\forall a, b \in Z_q^*$, $\exists P$ which is a generator of group $\mathbb{G}_1$, such that $e(aP, bP) = e(P, P)^{ab}$.

2) ***Non-degeneracy****:* There exists $P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1_{\mathbb{G}_2}$, where $1_{\mathbb{G}_2}$ is an identity element of $\mathbb{G}_2$.

3) ***Computability****:* For any $P, Q \in \mathbb{G}_1$, there must be an efficient algorithm, which can compute $e(P, Q) \in \mathbb{G}_2$.

IV. Proposed PPRL Scheme

In this section, we describe our proposed PPRL scheme in more details.

1) ***System Initialization****:* Given a parameter λ, trusted third party (TTP) first generates (q, P, G_1, G_2, e) by running *Setup(λ)*. With λ, data provider can generate paremeters (r, rg) and encrypt their datasets with secret r, and send it to the server.

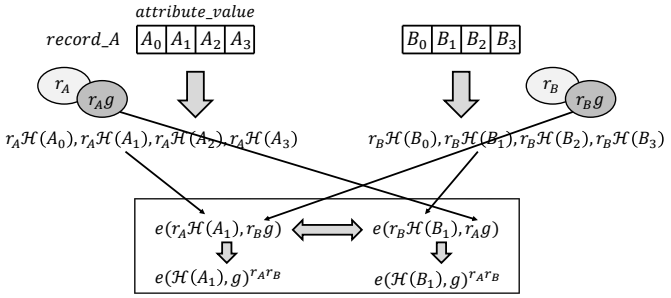

Fig. 2: An example of running our proposed PPRL scheme.

2) ***Authorization Mechanism****:* After coordinating the time through the secure channel, they individually generate their own authorization. The content of authorization is $E_{SK_A}(ID_A, ID_B, r_Ag, timestamp)$ where SK_A is A's communicational private key.

3) ***Linkage Test****:* As shown in Figure 2, the algorithm outputs 0 if they are equal, and 1 otherwise. The test is only for one attribute because the weight (importance) for each attribute is not the same. After testing all attributes then we can compute similarity (distance) to determine whether these two records are the same or not.

Since the number of record comparisons is $|A| \times |B|$, which could be high for resource-constrained devices, we still need a blocking mechanism (in the sense of a divide-and-conquer approach) to reduce the computational cost. As shown in Figure 3, we can use *locality sensitive hashing* (LSH) to build the hash table. An alternative is that we can compute the key-score to rank records, which proposed in [4].

This architecture has the desired properties. For example, in addition to drastically reducing the amount of computation, it allows us to parallelize and increase the efficiency with GPU acceleration.

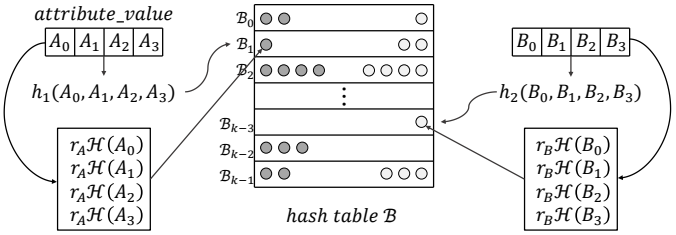

Fig. 3: Blocking mechanism.

V. Results and Comparisons

With different parameters in our test, when box setting in blocking is larger, comparison and time cost will be dramatically reduced. Nonetheless, in contrast, the recall will be affected, as shown in Figure 4.

records*attribute(5)	Boxnum	PPRL(securehash+ keyGen+compare)	compare times	recall	comparison reduction
100	1	551.988 (sec)	50000	100%	0.00%
100	3	357.670	31745	100%	36.51%
100	5	239.205	20990	100%	58.02%
100	10	131.806	11000	100%	78.00%
100	50	29.190	2320	100%	95.36%
400	1	too long	null	null	null
400	3	too long	null	null	null
400	5	3488.654	317170	100%	60.35%
400	10	1941.339	173935	100%	78.26%
400	50	283.776	35470	100%	95.57%
400†	50	287.825	24710	36.25%	96.91%
1000	50	2485.842	227300	100%	95.45%

Fig. 4: PPRL with datasets that there has exactly 20% true. Note that the box interval should be consistent, else will make a bad result as 11^{th} row record.

Although the blocking mechanism indeed improves efficiency, once the records become larger, the time cost will also become heavier. The secure hash stage, which converts strings to elliptic points, its heavy computation ($O(n)$) needs to be alleviated. On the other hand, the comparison stage, which has to do the elliptic calculation with the large number (near $O(n^2)$ times) also needs to be taken care.

VI. Conclusion and Future Work

The proposed PPRL scheme can achieve acceptable accuracy and privacy. We plan to run on GPU to test the computational efficiency of our proposed PPRL scheme.

References

[1] X. He, A. Machanavajjhala, C. Flynn, and D. Srivastava. Composing differential privacy and secure computation: A case study on scaling private record linkage. *ACM SIGSAC*, 2017.

[2] S. Ma, Q. Huang, M. Zhang, and B. Yang. Efficient public key encryption with equality test supporting flexible authorization. *IEEE TIFS*, 10(3):458–470, 2015.

[3] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *Advances in Cryptology - CRYPTO*, volume 2139 of *LNCS*, pages 213–229. Berlin, Germany: Springer, 2001.

[4] M.A. Hernndez and S.S. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data mining and knowledge discovery*, 2(1), 9-37, 1998