RESEARCH ARTICLE

LiPISC: A Lightweight and Flexible Method for Privacy-Aware Intersection Set Computation

Wei Ren[1,2]*, Shiyong Huang[1], Yi Ren[3], Kim-Kwang Raymond Choo[1,4]

1 School of Computer Science, China University of Geosciences, Wuhan, Hubei, P.R. China, **2** Hubei Key Laboratory of Intelligent Geo-Information Processing (China University of Geosciences (Wuhan)), Wuhan, Hubei, P.R. China, **3** Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R. O. China, **4** School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, South Australia, Australia

* weirencs@cug.edu.cn

OPEN ACCESS

Citation: Ren W, Huang S, Ren Y, Choo K-KR (2016) LiPISC: A Lightweight and Flexible Method for Privacy-Aware Intersection Set Computation. PLoS ONE 11(6): e0157752. doi:10.1371/journal.pone.0157752

Editor: Muhammad Khurram Khan, King Saud University, Kingdom of Saudi Arabia, SAUDI ARABIA

Received: March 29, 2016

Accepted: June 4, 2016

Published: June 21, 2016

Data Availability Statement: All relevant data are within the paper.

Funding: This work was supported by National Science Foundation China, No. 61170217 (http://npd.nsfc.gov.cn) (WR). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Privacy-aware intersection set computation (PISC) can be modeled as secure multi-party computation. The basic idea is to compute the intersection of input sets without leaking privacy. Furthermore, PISC should be sufficiently flexible to recommend approximate intersection items. In this paper, we reveal two previously unpublished attacks against PISC, which can be used to reveal and link one input set to another input set, resulting in privacy leakage. We coin these as Set Linkage Attack and Set Reveal Attack. We then present a lightweight and flexible PISC scheme (LiPISC) and prove its security (including against Set Linkage Attack and Set Reveal Attack).

1 Introduction

Online social networks (including enterprise social networks such as Yammer) and e-commerce websites are increasingly popular with both individual and organizational users. In these applications, recommendations are a frequently used mechanism for users and service providers to suggest potential friends or interested commodities to other users. For example, in online social networks such as Tencent QQ or Facebook, service providers recommend potential friends to another user, by relying on the intersection set of the user's friends' friends. More specifically in this context, B and C are A's friends; thus, the intersection set of B's friends and C's friends may be A's potential friends. The service provider needs to conduct regular intersection set computation (ISC) to find the set D that excludes A's friends. Users in set D are then recommended to A as potential friends. Similarly, services or other contacts (e.g., prospective partners for dating apps) are recommended using this (recommendation computation) approach.

Recommendation computation can be generalized as the calculation of an intersection function, where the input is two or more sets and the output is the intersection set of the input sets. We remark that the entity (e.g., online social network service providers) for computing intersection function may not always be trustworthy. Hence, the privacy of users may be leaked (or compromised) during the computations. Therefore, privacy-aware computation of intersection set has attracted the attention of security and privacy researchers (see [1–7]), and is key to the widespread adoption of online social networks and e-commence websites.

Most existing studies address specific privacy protection problems in the context of a particular application. Relatively few research formalize the privacy-aware computation of intersection set problem as an abstract problem (i.e., privacy-aware intersection set computation (PISC)) or focus on designing schemes that are lightweight and flexible. This is the gap that we seek to address in this paper. We also reveal two previously published security attacks against PISC, namely: set reveal attack and set linkage attack.

In this paper, we propose a lightweight and flexible PISC (hereafter referred to as LiPISC) scheme, which satisfies the following properties:

1. A basic requirement for privacy protection is for intersection computation to be conducted over input sets with concealed members instead of plain members. This is more difficult than traditional ISC.
2. An enhanced requirement for privacy protection is unlinkable. In other words, computation entities cannot infer the existence of the same items using the received input sets after two computation operations.
3. A basic requirement for flexibility is the capability to compute rough set. That is, PISC can return approximate intersection set even though matched common members do not exist in input sets.
4. Lightweight and scalable for large scale applications.
5. Security against set reveal attack, set linkage attack, and other common attacks.

We then prove the security of LiPISC.

The remainder of this paper is organized as follows. Section 2 describes related work. In Section 3, we present the basic assumption and models used in the paper. Section 4 details our proposed LiPISC scheme and the security proof. Finally, Section 5 concludes this paper.

2 Related Work

Ensuring the privacy of recommendations in social networks is a current research focus (see [8–15]). Wicker and Schrader [8] surveyed the philosophical, legal, moral, and epistemological literature on privacy in information networks, and introduced privacy-aware design principles. In the same year, Pentafronimos, Karantjias and Polemi [9] identified privacy requirements in collaborative workspaces, and suggested a number of guidelines for privacy-aware identity and access management systems. Attempting to fulfill the privacy requirements in online social networks, Akcora, Carminati and Ferrari [10] proposed a privacy risk measure to provide users a way of estimating risk (via the use of a Facebook prototype application). In 2013, Shoaran, Thomo and Weber-Jahnke [11] studied privacy issue in big data. They used graphs to model big data and proposed privacy-aware release of graph summarization using zero-knowledge privacy. Similarly, Vidyalakshmi et al [12] proposed a privacy aware information dispersal method in social networks, where they employed a supervised learning model to assist user in spotting unintended audience for a post. A year later in 2014, Li [13] posited that users should be able to indicate different comfort levels to share their friendships in social networks, and this can be done by limiting the number of their friends returned in response to queries through the friend search engine. They also proposed a new attack model that may infer more friendship information based on the friendships detected in the query results, as well as defining a model to safely display friends of individual users in response to queries. However, these studies only focused on specific applications, rather than seeking to solve the underlying challenge using a generalized approach.

Privacy protection of recommendations in e-commerce systems has also been the subject of extensive research. For example, Bunea et al [16] presented a privacy-aware collaborative filtering recommender framework, where they used a probabilistic matrix factorization technique to mitigate the sparsity and a dynamic privacy inference model to ensure privacy. In the dynamically personalized recommendation algorithm of Tang and Zhou [17], information in ratings and profile contents were used, without a strong focus on privacy protection. He, Ren and Zhang [18] presented an approach to increase the conversion rate from browsing to buying. Their method uses a behavior-based inference of a customer's propensity to purchase from a product category. Celdran et al [19] proposed a middle-ware to provide users with custom context-aware recommendations. However, these studies neither focused on privacy protection nor presented a more general method that can be deployed in a wide range of applications. We also observe that there is no suggestion that PISC can be modeled as secure multi-party computation.

PISC problem has been addressed using different approaches in the literature (see [1, 3–7]). For example, Shao, Yang and Yu [1] used searchable encryption to fulfill private set intersection, in which public key encryption with multiple keywords search is used as the basic tool. Zhao and Luo proposed a two-party private set intersection protocol based on negative database [3]. The negative database is a new technique for preserving privacy, and it stores information in the complementary set of a traditional database. The security foundation of this technique is that reversing the negative database to recover the corresponding database is NP-hard. In the study of how to extract common sensitive information from encrypted sets, Liu et al [4] argued that existing methods are not suitable for cloud deployment. Hence, they designed the Encrypted Set Intersection Protocol (ESIP) that allows server and users to perform collaborative operations to obtain the correct set intersection with privacy-preserving. In the attempt to solve privacy-preserving intersection of regular languages instead of finite sets, Guanciale, Gurov and Laud [5] proposed an approach based on minimal deterministic finite automata. Wang, Zhu and Luo [6] proposed a scheme which allows any entity to publicly verify the correctness of set intersection query, without requiring any secret key. More recently in 2015, Thapa et al [7] used asymmetric social proximity to design different private matching protocols, designed to provide different privacy levels. We observe that the literature rarely requires a PISC solution to be both lightweight and flexible. In the big data era, lightweight and flexible are two critical properties for privacy-aware PISC [20, 21].

3 Problem Formulation

3.1 Network Model

There are three entities in the intersection set computation, namely: an entity A, an entity B, and a computation server C. Upon receiving the sets from A and B, C will compute and return the intersection sets to A and B. For privacy protection, A and B conduct their respective computations to conceal the original set prior to sending to C. C conducts the computation of PISC over the concealed sets and returns information on the intersection of original sets.

We denote the set from A to C, the set from B to C, the the intersection set from C to A and B, as Set_A, Set_B, and Set_C, respectively. Thus, the basic logic flow in network model can be simplified as follows:

Msg1) $A \to C : Set_A$, where $x \to y : z$ denotes a message z being sent from entity x to entity y.

Msg2) $B \rightarrow C : Set_B$;

Msg3) $C \rightarrow A : Set_C$, $C \rightarrow B : Set_C$.

It is worth noting that Set_A and Set_B are concealed sets of the original sets, and Set_C can be redirected to the intersection of the original sets only at A and at B upon receipt.

3.2 Attack Model

We assume that the communication channels between entities and the server are protected by standard security mechanisms (e.g., encryption and integrity protection) at link layers (e.g., IEEE802.11i and CDMA); thus, attackers who can sniff packets are beyond the scope of this paper. In this paper, we focus on adversaries for privacy leakage at server side. In this context, we point out the following potential attacks, and we denote the adversary as Adv.

Definition Set Reveal Attack (Adv_{sra}). The adversary reveals the privacy of entity A and entity B from observing Set_A and Set_B. Roughly speaking, $H(x) = H(x|Set_A, Set_B)$, where H is the entropy function and x is any information about A or B.

In other words, entity A and entity B present Set_A and Set_B to server C for discovering of the intersection members. As the server C is not entirely trustworthy, the presenting sets should not reveal the privacy of A or B. Thus, Set_A and Set_B must be transformed from the original sets to the concealed sets for further intersection computation. That is, Set_A and Set_B that are presented to the server C must not leak the privacy of entity A and entity B, respectively. In the tradition reductionist approach [22–24], the adversary we consider in this paper has an upper bound in computational capabilities (i.e., a probabilistic polynomial-time turing machine—PPTM).

Adv_{sra} captures the basic security notion, and the security can be achieved by some transformations guaranteeing computational secrecy. However, we observe that if sets presented to untrustworthy servers are "linkable", it may also compromise the privacy of entity A (or B). This is because the adversary at the untrustworthy server can discover that entity A (or B) has conducted the same behavior more frequently than others, the interests or preference at entity A (or B) can then be inferred by the adversary (e.g., purchasing certain merchandise such as milk powder more frequently than others). We coin such an attack against the PISC as a Set Linkage Attack.

Definition Set Linkage Attack (Adv_{sla}). The adversary can successfully guess that at least one item in Set_A (or Set_B) in one round is the same as another entity in a previous round. Roughly speaking, $\Pr\{\text{Find } x, x \in Set_A \bigwedge x \in Set'_A | \text{ Observing } Set_A, Set'_A\} > 1/2$, where Set_A and Set'_A are in two distinct rounds.

For example, after the adversary observes Set_A in one round and Set'_A in another round, the adversary can link one member in Set_A to another member in Set'_A. That is, the adversary can successfully guess there exists a same member in both Set_A and Set'_A, which can compromise the privacy of that particular entity in some situations.

A more relaxed notion is to assume server C is semi-trustworthy (also known as honest-but-curious adversary). That is, the computation on server C for intersection set computation (mathematically) is trustworthy (i.e., correctly computed), but the adversary on server C may seek to infer the privacy of entity A and entity B by observing Set_A and Set_B. Here, the computation functionality for intersection set has to be trustworthy as a prerequisite condition; otherwise, we will not be able to achieve privacy-aware intersection set computation.

Thus, the privacy requirements are stated as follows:

Definition $Pri_{PISC}^{Adv_{sra}, Adv_{sla}}$. In the presence of Adv_{sra} and Adv_{sla} at a semi-trustworthy server C with a computational bound, the intersection set of Set_A and Set_B uploaded by entity A and entity B can be correctly computed without compromising the privacy of entity A and entity B. (Note that, the intersection set is for the original sets, rather than the concealed sets.)

3.3 Design Goals

The design of PISC should also consider flexibility, as Set_A and Set_B may not have the exact intersection set. In other words, we can also compute the rough intersection set even though Set_A and Set_B are not exactly or approximately equal. The flexibility will be particularly helpful in social network and e-commerce applications, as approximate intersection set is adequate for the required functionalities (e.g., identifying similar goods that may be of interest to an e-commerce user). Flexibility is also necessary when exact intersection set for the uploaded sets does not exist.

To achieve flexible PISC in a lightweight manner is also important, particularly due to the scale of data involved and the real-time nature of such computations. The lightweight computation in one time computation of PISC will significantly influence the scalability of the proposed method.

Therefore, the design goals are lightweight and flexible PISC.

4 Proposed Scheme - LiPISC

Table 1 outlines the notation used in the proposed Lightweight and Flexible PISC (LiPISC) scheme.

4.1 Abstract Model

LiPISC consists of the following functions:

1) $f_{conceal}$. It takes as input the original set, Set_{AO}, and outputs a concealed set, Set_{AC}. That is,

$$f_{conceal} : Set_{AO} \Rightarrow Set_{AC},$$

where $Set_A \Rightarrow Set_B$ means that $\forall x \in Set_A$, compute $y = f_{conceal}(x)$ and include y in Set_B.

Table 1. Notation.

Adv	Adversary
Adv_{sra}	Set Reveal Attack
Adv_{sla}	Set Linkage Attack
$f_{conceal}$	Concealing Function
$f_{intersect}$	Intersection Set Computation Function
f_{reveal}	Reveal Function
Set_{AC}	Concealed Set from A
Set_{AO}	Original Set from A
Set_{BC}	Concealed Set from B
Set_{BO}	Original Set from B
Set_{IC}	“Intersection set” for Concealed Set
Set_{IAO}	Indeed Intersection Set in Original Set from A
Set_{IBO}	Indeed Intersection Set in Original Set from B

doi:10.1371/journal.pone.0157752.t001

2) $f_{intersect}$. It takes as input two sets, Set_{AC} and Set_{BC}, and outputs a concealed intersection set, Set_{IC}. That is,

$$f_{intersect} : Set_{AC} \times Set_{BC} \Rightarrow Set_{IC}.$$

3) f_{reveal}. It takes as input a set, Set_{IC}, and outputs an original intersection set, Set_{IO}. That is,

$$f_{reveal} : Set_{IC} \Rightarrow Set_{IAO}, Set_{IC} \Rightarrow Set_{IBO},$$

where $Set_{IO} = Set_{AO} \wedge Set_{BO}$, $Set_{IAO} = Set_{IO} \wedge Set_{AO}$, $Set_{IBO} = Set_{IO} \wedge Set_{BO}$. Certainly, $Set_{IO} = Set_{IAO} = Set_{IBO}$.

Therefore, in LiPISC,

1. A and B conceal the original sets Set_A and Set_B using $f_{conceal}$, and obtain concealed sets Set_{AC} and Set_{BC}, respectively.
2. A and B respectively send concealed sets Set_{AC} and Set_{BC} to C.
3. C computes the intersection set Set_{IC} from the received concealed sets using $f_{intersect}$. C then returns Set_{IC} to both A and B.
4. A finds the indeed intersection set (Set_{IAO}) of the original sets (i.e., $Set_{AO} \wedge Set_{BO}$) from Set_{IC} using f_{reveal}. Similarly, B finds the indeed intersection set (Set_{IBO}) of the original sets (i.e., $Set_{AO} \wedge Set_{BO}$) from Set_{IC} using f_{reveal}.

Thus, LiPISC can be formulated as an abstract model:
1) Entity A:
1.1) $f_{conceal} : Set_{AO} \Rightarrow Set_{AC}$;
1.2) $A \rightarrow C : Set_{AC}$.
2) Entity B:
2.1) $f_{conceal} : Set_{BO} \Rightarrow Set_{BC}$;
2.2) $B \rightarrow C : Set_{BC}$.
3) Server C:
3.1) $f_{intersect} : Set_{AC} \times Set_{BC} \Rightarrow Set_{IC}$;
3.2) $C \rightarrow A : Set_{IC}$;
3.3) $C \rightarrow B : Set_{IC}$.
4)
Entity A:
4.1) $f_{reveal} : Set_{IC} \Rightarrow Set_{IAO}$;
Entity B:
4.2) $f_{reveal} : Set_{IC} \Rightarrow Set_{IBO}$.
The requirement is as follows: $f_{conceal}$.

Proposition 4.1 *$f_{conceal}$ should achieve the functionality to defend against Adv_{sra} and Adv_{sla}.*

Proof $f_{conceal}$ is the only transformation of the original set before the observation of the adversary. After this transformation, the adversary at C can observe the transformation result of $f_{conceal}$. Thus, $f_{conceal}$ should defend against Adv_{sra} and Adv_{sla}. □

$f_{conceal}$ should make it possible to find the corresponding $f_{intersect}$ and f_{reveal}. That is, corresponding $f_{intersect}$ and f_{reveal} should be found easily and should perform efficiently after the transformation of $f_{conceal}$. Thus, $f_{conceal}$ is the most important of three functions. Besides, these three functions must be lightweight in terms of computation. Therefore, finding proper $f_{conceal}$ has the highest priority in searches.

4.2 Basic Construction

The basic construction is described as follows:

1) Entity A:

1.1) $f_{conceal}: Set_{AO} \Rightarrow Set_{AC}$. Let $f_{conceal} = Hash(\cdot) : \{0,1\}^m \rightarrow \{0,1\}^n, m, n \in \mathbb{N}$; That is, $Set_{AC} \Leftarrow Hash(x \in Set_{AO})$, where $Hash(\cdot)$ could be a cryptographic hash function. In other words, $Set_{AC} = \{y|y = Hash(x), \forall x \in Set_{AO}\}$.

1.2) $A \rightarrow C : ID[i], Set_{AC}[i]$, where $i = 1, \ldots, |Set_{AC}|$; $|*|$ returns the number of items in the set $*$; $ID[i]$ is a temporary sequence number for i-th item in Set_{AC}, which is denoted as $Set_{AC}[i]$;

2) Entity B:

2.1) $B : f_{conceal} : Set_{BO} \Rightarrow Set_{BC}$; Similar to 1.1), let $f_{conceal} = Hash(\cdot)$; $Set_{BC} \Leftarrow Hash(Set_{BO})$, where $Hash(\cdot)$ could be a one-way cryptographic hash function. In other words, $Set_{BC} = Hash(x \in Set_{BO}) = \{y|y = Hash(x), \forall x \in Set_{BO}\}$.

2.2) $B \rightarrow C : ID[i], Set_{BC}[i]$, where $i = 1, \ldots, |Set_{BC}|$; $ID[i]$ is a temporary sequence number for i-th item in Set_{BC}, which is denoted as $Set_{BC}[i]$;

3) Server C:

3.1) $f_{intersect} : Set_{AC} \wedge Set_{BC} \Rightarrow Set_{IC}$;

3.2) $C \rightarrow A : ID[i]$, where $Set_{AC}[i] \in Set_{IC}$, $i \in [1, |Set_{AC}|]$; The total number of i is $|Set_{IC}|$; That is, $|ID[i]| = |Set_{IC}|$;

3.3) $C \rightarrow B : ID[j]$, where $Set_{BC}[j] \in Set_{IC}$, $j \in [1, |Set_{BC}|]$; The total number of j is $|Set_{IC}|$; That is, $|ID[j]| = |Set_{IC}|$;

4)

Entity A:

4.1) $f_{reveal} : ID[i] \Rightarrow Set_{IAO}$; As A has computed $Set_{AC} \Leftarrow Hash(Set_{AO})$ before, A can retrieve corresponding i-th item in Set_{AO} that will compose final Set_{IAO};

Entity B:

4.2) $f_{reveal} : ID[j] \Rightarrow Set_{IBO}$; As B has computed $Set_{BC} \Leftarrow Hash(Set_{BO})$ before, A can retrieve corresponding j-th item in Set_{BO} that will compose final Set_{IBO};

Remarks

1) $f_{conceal}$ is pre-deployed or negotiated in advance at A and B, or distributed by C instantly and publicly. Even though $f_{conceal}$ is publicly known, Adv_{sra} is still defended against due to the cryptographic properties of $Hash(\cdot)$.

2) Note that $Hash(\cdot)$ could be any function with dedicated requirements, although a computationally efficient cryptographic hash is usually preferred. Next, we will define the one-way and collision resistant requirements.

Proposition 4.2 $f_{conceal} : \{0,1\}^m \rightarrow \{0,1\}^n, m \geq n \in \mathbb{N}$ *should be one-way.*

Proof Due to 4.1, $f_{conceal}$ should have functionality of defending against Adv_{sra}. $H(x|f_{conceal}(x)) = H(x)$ for PPTM adversary. $Pr\{x|y = f_{conceal}(x)\} = 1/2^{n*}1/2^{m-n} = 1/2^m = negl(m)$ should be negligible when m is sufficient large. $negl(m)$ is a negligible function in m. Given y compute x is hard; thus, $f_{conceal}$ must be one-way. □

Proposition 4.3 $f_{conceal}$ *should be collision resistant.*

Proof $f_{conceal}$ should be collision resistant, which is not for security but for the soundness of resulting intersection set returned from C. If so, the probability will be negligible that $y_1 \in Set_{AC}$ and $y_2 \in Set_{BC}$ are equal but $x_1 \in Set_{AO}$ and $x_2 \in Set_{BO}$ are not equal. □

Note that it is non-trivial to be aware that $f_{conceal}$ is not second pre-image resistant, because server C has no idea of the pre-image of Set_{AC} as well as Set_{BC}.

Proposition 4.4 *The false probability of basic construction for PISC is* $1/2^{m-n}$. *That is,* $\Pr\{x_1 \in Set_{AO} \neq x_2 \in Set_{BO}|y = f_{conceal}(x_1) \in Set_{AC}, y = f_{conceal}(x_2) \in Set_{BC}\} = 1/2^{m-n}$.

Proof There are $1/2^{m-n}$ $x \in Set_{AO}$ mapping into the same $y \in Set_{AC}$. Thus, even y is the same x is different. The false probability of basic construction is $1/2^{m-n}$. □

3) Only last k, $k < |Hash(\cdot)| = n$ bits can be selected in $Hash(\cdot)$ during comparison to further improve the efficiency, but it may induce extra false members in intersection set with false probability (namely, $1/2^{m-k}$).

The privacy strength for the basic construction is as follows:

Proposition 4.5 *Basic construction can defend against Adv_{sra} but not Adv_{sla}.*

Proof As $f_{conceal}$ is one-way, the adversary can reveal neither Set_{AO} from Set_{AC} nor Set_{BO} from Set_{BC}. Thus, Adv_{sra} is defended against. However, the same $y \in Set_{AC}(Set_{BC})$ in different rounds can be linked to the same $x \in Set_{AO}(Set_{BO})$. Thus, Adv_{sla} is not defended against. □

4.3 An Enhanced Basic Construction

To defend against Adv_{sla}, we propose the following enhancement by adding a random number to be used only once (i.e., nonce) in $f_{conceal}$ to make it unlinkable when the adversary observes Set_{AC} (or Set_{BC}). More specifically, the enhancement is at steps 1.1) and 2.1) with the addition of the nonce, respectively. The enhanced steps 1.1) and 2.1) are as follows:

1.1) $f_{conceal} : Set_{AO} \Rightarrow Set_{AC}$. Let $f_{conceal} = Hash(\cdot)$; That is, $Set_{AC} \Leftarrow H(nonce_A||Set_{AO})$, where $Hash(\cdot)$ could be a cryptographic hash function; *nonce* is a number used once. In other words, $\forall x \in Set_{AO}$, $y \Leftarrow Hash(nonce_A||x)$, $y \in Set_{AC}$.

2.1) $f_{conceal} : Set_{BO} \Rightarrow Set_{BC}$; Similar to 1.1), let $f_{conceal} = Hash(\cdot)$; $Set_{BC} \Leftarrow Hash(nonce_B||Set_{BO})$, where $H(\cdot)$ is a cryptographic hash function with one-wayness. In other words, $\forall x \in Set_{BO}$, $y \Leftarrow Hash(nonce_B||x)$, $y \in Set_{BC}$.

Remarks

1) $Nonce_A$ and $Nonce_B$ could be timestamps, in which A and B need to be synchronized in advance.

2) $Nonce_A$ and $Nonce_B$ could be a value from a pseudorandom number generator, which is generated from a shared secret key (seed) and synchronized at A and B. That is, $Nonce = PRNG(k)$, where k is a shared secret key between A and B (e.g., generated using a key establishment protocol [25]); $PRNG(\cdot)$ is a pseudorandom number generator.

3) $Nonce_A$ and $Nonce_B$ could be a synchronized counter.

Proposition 4.6 *The enhanced basic construction can defend against Adv_{sra} and Adv_{sla}.*

Proof The enhanced basic construction can defend against Adv_{sra} inherently due to basic construction. The discussion, thus, only concentrates on Adv_{sla}. As $Nonce_A$ and $Nonce_B$ vary in each round, the same $y \in Set_{AC}(Set_{BC})$ in different rounds usually cannot be identified by the adversary due to the collision resistance property of $f_{conceal}$. Thus, the linkage will not be drawn and Adv_{sla} is defended against. □

4.4 Advanced Construction with Rough Intersection for Flexibility

In a real-world deployment, such as on-line social networks and e-commerce websites, entity A and entity B are unlikely to have the exact number or same members in the submitted sets. Although exact intersection of Set_{AO} and Set_{BO} dose not exist, server C is still required to return approximate intersection of Set_{AO} and Set_{BO}. In this scenario, server C has to recommend the most approximate intersection of Set_{AO} and Set_{BO} from Set_{AC} and Set_{BC}. The basic construction and the enhanced basic construction are not able to address such a situation. Thus, we propose an advanced method that can satisfy this requirement.

Intuitively, if $Hash(\cdot)$ is "fuzzy" in that $\|x_1 - x_2\| \in \Delta \Rightarrow \|y_1 - y_2\| \in \Delta$, $y_1 = Hash(x_1)$, $y_2 = Hash(x_2)$, the rough set will be returned. Here, $\|\cdot\|$ returns either the absolute value or the hamming distance. However, this fuzzy property may compromise the one-wayness.

The improvements are due to the following: at $f_{conceal}$, $Hash(\cdot)$ is replaced by $G(\cdot): g^x \mod p$, where p is a sufficient large prime, and at $f_{intersect}$, the comparison is changed into computation of $(g^{x_1})/(g^{x_2}) \approx g^{\delta}$. More specifically, the method is described as follows:

1) Entity A:

1.1) $A: f_{conceal}: Set_{AO} \Rightarrow Set_{AC}$. Let $f_{conceal} = g^x \mod p$, where $x \in Set_{AO}$; p is a large prime. That is, $Set_{AC} \Leftarrow G(Set_{AO})$, where $G(Set_*)$ means to compute $f_{conceal}(x)$ for $\forall x \in Set_*$. In other words, $Set_{AC} = \{y|\forall x \in Set_{AO}, y = g^x \mod p\}$.

1.2) $A \rightarrow C: ID[i], Set_{AC}[i]$, where $i = 1, \ldots, |Set_{AC}|$; $|*|$ returns the number of items in the set $*$; $ID[i]$ is a temporary sequence number for i-th item in Set_{AC}, which is denoted as $Set_{AC}[i]$;

2) Entity B:

2.1) $B: f_{conceal}: Set_{BO} \Rightarrow Set_{BC}$; Similar to 1.1), let $f_{conceal} = g^x \mod p$, where $x \in Set_{BO}$; That is, $Set_{BC} \Leftarrow G(Set_{BO})$. In other words, $Set_{BC} = \{y|\forall x \in Set_{BO}, y = g^x \mod p\}$.

2.2) $B \rightarrow C: ID[i], Set_{BC}[i]$, where $i = 1, \ldots, |Set_{BC}|$; $ID[i]$ is a temporary sequence number for i-th item in Set_{BC}, which is denoted as $Set_{BC}[i]$;

3) Server C:

3.1) $f_{intersect}: \forall Set_{AC}[i] \in Set_{AC}, Set_{BC}[j] \in Set_{BC}$, if $Set_{AC}[i]/Set_{BC}[j] < g^{\delta} < p$, let $Set_{AC}[i] \vee Set_{BC}[j] \Rightarrow Set_{IC}$;

3.2) $C \rightarrow A: ID[i]$, where $Set_{AC}[i] \in Set_{IC}$, $i \in [1, |Set_{AC}|]$; The total number of i is $|Set_{IC}|/2$; That is, $|ID[i]| = |Set_{IC}|/2$;

3.3) $C \rightarrow B: ID[j]$, where $Set_{BC}[j] \in Set_{IC}$, $j \in [1, |Set_{BC}|]$; The total number of j is $|Set_{IC}|/2$; That is, $|ID[j]| = |Set_{IC}|/2$;

4)

Entity A:

4.1) $f_{reveal}: ID[i] \Rightarrow Set_{IAO}$; As A computes $Set_{AC} \Leftarrow G(Set_{AO})$, A can reveal corresponding i-th item in Set_{AO} that consists final Set_{IAO};

Entity B:

4.2) $f_{reveal}: ID[j] \Rightarrow Set_{IBO}$; As B computes $Set_{BC} \Leftarrow G(Set_{BO})$, A can reveal corresponding j-th item in Set_{BO} that consists final Set_{IBO};

Remarks

1) To reduce the computation overhead at entity A and entity B, in computing $G(\cdot)$, A and B can only compute the last k bits of x, y, $\forall x \in Set_{AO}$ and $\forall y \in Set_{BO}$. $\log_2 x = \{0,1\}^m = \{0,1\}^{m-k}||\{0,1\}^k$, $\log_2 y = \{0,1\}^n = \{0,1\}^{n-k}||\{0,1\}^k$. It will not compromise the soundness of PISC if k is properly chosen. We state the chosen method in the following proposition.

Proposition 4.7 *If the absolute gap value between* $x \in Set_{AO}$ *and* $y \in Set_{BO}$ *is* $||x - y||$, *we let* $k = \log_2 \log_g ||x - y||$.

Proof $x \in Set_{AO}$, $g^x \bmod p \in Set_{AC}$. $x \in Set_{BO}$, $g^x \bmod p \in Set_{BC}$. $\frac{g^x}{g^y}$ can be computed via $\frac{g^{m+(x-m)}}{g^{m+(y-m)}}$, where $\log_g(x - m) < 2^k$ and $\log_g(y - m) < 2^k$. Thus, only computing the last k bits of x, y will not result in any difference for $\frac{g^x}{g^y}$ at server C. □

2) δ is a system parameter, which measures the approximate strength of the intersection set. More specifically, we state it formally in the following proposition.

Proposition 4.8 *If the absolute gap value between* $x \in Set_{AO}$ *and* $y \in Set_{BO}$ *is* $||x - y||$, *the intersection set produced by PISC will have* $||x - y|| \leq \delta$.

Proof $x \in Set_{AO}$, $g^x \mod p \in Set_{AC}$. $y \in Set_{BO}$, $g^y \mod p \in Set_{BC}$. Server C selects intersection set by $\frac{g^x}{g^y} < g^{\delta}$, thus $||x - y|| \leq \delta + k\varphi(p) = \delta + k(p - 1)$, where $\phi(\cdot)$ is Euler function, and $k \in \mathcal{N}$. $p - 1 >> \delta$ and $x \leq p - 1$, $y \leq p - 1$, thus $||x - y|| \leq \delta$. □

From above two propositions, we can choose $k = \log_2 \log_g ||x - y|| = \log_2 \log_g \delta$ bits at rear of $\log_2 x$, $\forall x \in Set_{AO}$ or $\log_2 y$, $\forall y \in Set_{BO}$ to compute Set_{AC} and Set_{BC}, respectively.

3) The computation overhead at server C is module exponential computation (for computing $g^{\delta} \mod p$) and one time division (for computing $Set_{AC}[i]/Set_{BC}[j]$). The computation overhead at an entity is module exponential computation (for computing $g^{k} \mod p$ and k is at most $\log_2 \log_g \delta$ bits).

Proposition 4.9 *The advanced construction can defend against Adv_{sra}, but not Adv_{sla}.*

Proof As $f_{conceal}$ is one-way due to the underlying discrete logarithm problem, Adv_{sra} is defended against. As $f_{conceal}$ is a deterministic function, the same image will link to the same pre-image; thus, Adv_{sla} cannot be defended against. □

Proposition 4.10 *In the advanced construction, $f_{conceal}$ is collision resistant.*

Proof The aim is to prove that it is difficult to find $x \neq y$ such at $g^x \equiv g^y \bmod p$. That is, it is difficult to find $x \neq y$ such that $g^{x-y} \bmod p = 1$. If $g^{x-y} \bmod p = 1$, $x - y \bmod p - 1 = 0$ by Fermat's theorem. As $x - y < \delta$ and p is a big prime, it is computational infeasible for $x - y \mod p - 1 = 0$. □

Proposition 4.11 *The advanced construction is sound.*

Proof It is concluded from the previous propositions. $f_{conceal}$ is one-way, collision resistant, and the returned intersection sets are indeed the approximate members. Thus, the advanced construction is sound. □

Finally, to defend against Adv_{sla}, we propose an enhancement method for the advanced construction by adding a nonce in $f_{conceal}$. More specifically, the enhancement is at Steps 1.1) and Step 2.1) by multiplying the nonce, respectively:

1.1). $f_{conceal} = nonce * g^x \bmod p$, where $x \in Set_{AO}$; p is a large prime. That is, $Set_{AC} \Leftarrow G(Set_{AO})$, where $G(Set_*)$ means to compute $f_{conceal}(x)$ for $\forall x \in Set_*$. In other words, $Set_{AC} = \{y | \forall x \in Set_{AO}, y = nonce * g^x \mod p\}$.

2.1) $f_{conceal} = nonce * g^x \bmod p$, where $x \in Set_{BO}$; That is, $Set_{BC} \Leftarrow G(Set_{BO})$.

The discussion on *nonce* is similar to the aforementioned remarks in the last section. The security proof after the inclusion of *nonce* is similar to Proposition 4.6. Until now, we reach the final proposed version of PISC that is the enhancement of the advanced construction. This incremental presentation helps for better understanding and smoothly remembering.

Potential Applications and Further Discussions

1) Example I. In social networks for recommending prospective friends, both users B and C are user A's friends; thus, the intersection set of B's friends and C's friends may also be A's friends. The service provider needs to perform PISC of two inputs—B's friends and C's friends, and those inputs should be concealed for privacy protection.

2) Example II. For recommending potentially goods of interest in an e-commence website, B's purchase history and C's purchase history have overlapping items with A's purchase history (i.e., A, B, C have similar purchasing habits and preferences); thus, the intersection set of user B's purchases and C's purchases may also interest A. The service provider needs to perform PISC of two concealed inputs—B's purchase history and C's purchase history. Since we require that the purchase history of both B and C to be unlinkable, the service provider is unable to guess whether the same item exists in B's or C's purchase history.

3) As in LiPISC, A and B have to establish a shared nonce, which may require synchronization at both A and B. Alternatively, nonce can also be distributed by servers which does not break the security of the scheme as the servers will not know the nonce due to the one-wayness of $f_{conceal}$.

4) As stated in Section 3 (i.e., Problem Formulation), the adversary (trust) model in this paper only assumes an adversary at a central server. That is, entity A and entity B for PISC is trustworthy. Nonetheless, even though A (or B) could be un-trustworthy, A (or B) can only

reveal the intersection set with B (or A) by random guess due to the one-wayness of $f_{conceal}$. The probability of successful guess is negligible due to the selection of $f_{conceal}$.

5) The gap between numeric distance represents the numeric deviation of inputting values, and the gap between Hamming distance represents the property deviation. The former gap can be computed from the latter gap if the major difference comes from the bits at the end, and the latter gap can also be estimated from the former gap. In this paper, we focus on the numeric distance, which is more general than the Hamming gap. In addition, the proposed method is suitable for both two situations, as here only rough intersection set computation is required.

6) The proposed scheme outperforms related work in sever aspects as follows: It is lightweight as only raw discrete logarithm computation is involved instead of cryptographic encryption. It can compute rough intersection set that is flexible in realistic. It tackles set linkage attack that has not been carefully explored in the literatures.

5 Conclusion

In this paper, we introduced two new attacks against PISC, which we coined as set reveal attack and set linkage attack. We then proposed a lightweight and flexible PISC (LiPISC), which achieves approximate intersection set computation and rough intersection set computation in a lightweight manner. We then proved the security of LiPISC.

Future work includes deploying the proposed scheme in a real-world application, such as e-commerce website, with the aims of refining and validating the scheme.

Author Contributions

Conceived and designed the experiments: WR KKRC. Performed the experiments: WR SH. Analyzed the data: WR. Contributed reagents/materials/analysis tools: WR YR. Wrote the paper: WR YR KKRC.

References

1. Shao Z, Yang B, Yu Y. Private Set Intersection via Public Key Encryption with Multiple Keywords Search. 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS'13), 323–328, Sept., 2013
2. Patsakis C, Solanas A. Privacy as a Product: A Case Study in the m-Health Sector. 2013 4th International Conference on Information, Intelligence, Systems and Applications (IISA'13), 1–6, Jul., 2013
3. Zhao D, Luo W. A Study of the Private Set Intersection Protocol Based on Negative Databases. 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing (DASC'13), 58–64, Dec., 2013
4. Liu F, Ng W, Zhang W, Giang D, Han S. Encrypted Set Intersection Protocol for Outsourced Datasets. 2014 IEEE International Conference on Cloud Engineering (IC2E'14), 135–140, Mar., 2014
5. Guanciale R, Gurov D, Laud P. Private Intersection of Regular Languages. 2014 12th Annual International Conference on Privacy, Security and Trust (PST'14), 112–120, Jul., 2014
6. Wang T, Zhu Y, Luo X. Publicly Verifiable Delegation of Set Intersection. 2014 International Conference on Cloud Computing and Internet of Things (CCIOT'14), 26–30, Dec., 2014
7. Thapa A, Li M, Salinas S, Li P. Asymmetric Social Proximity Based Private Matching Protocols for Online Social Networks. IEEE Transactions on Parallel and Distributed Systems. 2015; 6: 1547–1559. doi: 10.1109/TPDS.2014.2329016
8. Wicker SB, Schrader DE. Privacy-Aware Design Principles for Information Networks. Proceedings of the IEEE. 2011; 2: 330–350. doi: 10.1109/JPROC.2010.2073670
9. Pentafronimos G, Karantjias A, Polemi N. Open Issues on Privacy and Trust in Collaborative Environments. 2011 IEEE Symposium on Computers and Communications (ISCC'11), 876–880, Jun., 2011
10. Akcora CG, Carminati B, Ferrari E. Privacy in Social Networks: How Risky is Your Social Graph? 2012 IEEE 28th International Conference on Data Engineering (ICDE'12), 9–19, Apr., 2012
11. Shoaran M, Thomo A, Weber-Jahnke JH. Zero-knowledge Private Graph Summarization. 2013 IEEE International Conference on Big Data, 597–605, Oct., 2013

12. Vidyalakshmi BS, Wong RK, Ghanavati M, Chi HC. Privacy as a Service in Social Network Communications. 2014 IEEE International Conference on Services Computing (SCC'14), 456–463, Jun., 2014
13. Li N. Privacy-aware Display Strategy in Friend Search. 2014 IEEE International Conference on Communications (ICC'14), 945–950, Jun., 2014
14. Perera C, Ranjan R, Wang L. End-to-End Privacy for Open Big Data Markets. IEEE Cloud Computing. 2015; 4: 44–53. doi: 10.1109/MCC.2015.78
15. Perera C, Ranjan R, Wang L, Khan SU, Zomaya AY. Big Data Privacy in the Internet of Things Era. IT Professional. 2015; 3: 32–39. doi: 10.1109/MITP.2015.34
16. Bunea R, Mokarizadeh S, Dokoohaki N, Matskin M. Exploiting Dynamic Privacy in Socially Regularized Recommenders. 2012 IEEE 12th International Conference on Data Mining Workshops (ICDM'12W), 539–546, Dec., 2012
17. Tang X, Zhou J. Dynamic Personalized Recommendation on Sparse Data. IEEE Transactions on Knowledge and Data Engineering. 2013; 12: 2895–2899. doi: 10.1109/TKDE.2012.229
18. He M, Ren C, Zhang H. Intent-based Recommendation for B2C e-Commerce Platforms. IBM Journal of Research and Development. 2014; 5/6: 1–10.
19. Celdran A, Perez M, Garcia F, Martinez G. Precise: Privacy-aware Recommender Based on Context Information for Cloud Service Environments. IEEE Communications Magazine. 2014; 8: 90–96. doi: 10.1109/MCOM.2014.6871675
20. Jie W, Cai W, Wang L, Procter R. A Secure Information Service for Monitoring Large Scale Grids. Parallel Computing. 2007; 7-8: 572–591. doi: 10.1016/j.parco.2007.05.001
21. Zhao J, Wang L, Tao J, Chen J, Sun W, Ranjan R, Kolodziej J, Streit A, Georgakopoulos D. A Security Framework in G-Hadoop for Big Data Computing across Distributed Cloud Data Centres. Journal of Computer and System Sciences. 2014; 5: 994–1007. doi: 10.1016/j.jcss.2014.02.006
22. Choo K K R. A Proof of Revised Yahalom Protocol in the Bellare and Rogaway (1993) Model. Computer Journal. 2007; 5: 591–601. doi: 10.1093/comjnl/bxm019
23. Choo K K R, Boyd C, Hitchcock Y. Examining Indistinguishability-Based Proof Models for Key Establishment Protocols. Advances in Cryptology (ASIACRYPT'05), 585–604, Dec. 2005
24. Choo K K R, Boyd C, Hitchcock Y. Errors in Computational Complexity Proofs for Protocols. Advances in Cryptology (ASIACRYPT'05), 624–643, Dec. 2005
25. Choo K K R. Secure Key Establishment. Advances in Information Security 41, US: Springer; 2009. doi: 10.1007/978-0-387-87969-7