Technical Report

TR-2011-003

Information sharing across private databases: Secure union revisited

by

Pawel Jurczyk, Li Xiong

Mathematics and Computer Science

EMORY UNIVERSITY

Information Sharing Across Private Databases: Secure Union Revisited

Pawel Jurczyk and Li Xiong

Emory University, Atlanta GA 30322, USA

Abstract. There is a growing demand for sharing information across multiple autonomous and private databases. The problem is usually formulated as a secure multiparty computation problem where a set of parties wish to jointly compute a function of their private inputs such that the parties learn only the result of the function but nothing else. In this paper we analyze existing and potential solutions for secure multiparty computation of union. We also present an alternative random shares based approach and show that the protocol, although quite simple, are more efficient than existing protocols while providing reasonable level of security that can be adjusted by users. We formally analyze the security properties and the cost of our protocols. We also experimentally compare the performance of our approach with the existing solutions.

1 Introduction

The amount of personal or sensitive information stored in multiple distributed databases is constantly growing. Institutions increasingly recognize the critical value and opportunities in sharing such a wealth of information. Due to privacy and security constraints, however, the institutions often cannot completely disclose their private data to others. The problem is usually formulated as a secure multiparty computation (SMC) or distributed privacy preserving data sharing problem [13, 20] where a set of parties wish to jointly compute a function of their private data inputs such that the parties learn only the result of the function but nothing else.

In this paper, we focus on the union problem, in which multiple parties wish to compute and share the union of their data without disclosing anything else. For secure union, all the data records will be revealed as part of the result, however, the owner of a certain data record shall not be disclosed. For this paper, we focus on the *semi-honest* adversary model commonly used in SMC problems. A semi-honest party follows the rules of the protocol, but it can attempt to learn additional information about other nodes by analyzing the data received during the execution of the protocol. The *semi-honest* model, although relatively weak, is realistic for our problem scenario where multiple institutions are collaborating with each other to get the correct result for their mutual benefit. Below we describe an application scenario that motivates the secure union operation under the semi-honest assumption.

Secure Union Scenario: Genomic Information Sharing. Consider a scenario in [21] for genomic data sharing. For research purposes, hospitals forward their DNA records to a research group. While the sequences are only tagged with pseudonyms of patients, the submitting institution of the DNA records are disclosed. As a result, if an adversary knows which hospitals a patient visited, called a *trail*, s/he can track the DNA information by the unique features of the trails. To prevent such a risk, we can use secure union protocol to share the DNA information such that the contributing institutions of the DNA records are not disclosed.

Contributions. In this paper we review and analyze existing representative secure union protocols as well as the anonymous communication protocols as a potential solution for the secure set operations. We propose an alternative simple yet effective protocol based on random shares approach. In contrast to traditional SMC protocols, they achieve sufficient (but not absolute) security for participating parties at much lower cost for practical usage. We present a set of formal analysis evaluating and comparing the protocols in terms of their security characteristics and cost. We also implemented all the protocols including the existing ones and experimentally evaluate their cost. Our goal in this paper is not to promote specific protocols, but to: 1) systematically analyze and experimentally evaluate existing protocols, and 2) demonstrate that simple solutions exist if we make a tradeoff between security, efficiency and accuracy and they may be desirable for certain practical settings.

2 Related Work

In the problem of secure multi-party computation (SMC) [10, 13, 20], a given number of participants, each having a private data, wants to compute the value of a public function. A protocol is *secure* if, at the end of computation, all participants know only their local inputs and the final result. Although a general solution to SMC problems has been proven to exist for any function, its high computational overhead makes it impractical. Specialized protocols have been proposed for various functions such as sum [22], the kth element [2], set intersection [3], set intersection size [3, 25], and set union [15, 7]. A closely related research area is privacy preserving data mining and sharing across distributed data sources [8, 24, 20]. It follows the SMC model and the main goal is to ensure that data is not disclosed among participating parties while allowing certain mining or querying task to be carried out. Specialized protocols are designed for various mining tasks with varying degree of accuracy, security, and cost (e.g. [19, 23, 11, 3, 15, 28, 27, 26]). Most above work assume an honest or semi-honest adversary model [13]. Other works focus on broader threat space including malicious adversaries [30, 4, 16, 14].

Existing and Potential Protocols for Secure Union. Various solutions for computing set union were proposed in the literature. They generally fall into three categories: 1) general circuit-based protocols [1, 29], 2) specialized cryptography-based protocols using commutative encryption schemes[8, 15, 7] or homo-

morphic encryption schemes and polynomial representation of sets [17], and 3) probabilistic protocols [5]. In addition to the above three categories, anonymous communication protocols [9], while not directly designed for secure multi-party computation, can be also used for set operations due to the unique nature of set operations. We experimentally compare the cost of the representative protocols in the experiment section.

3 Random Shares Based Union

Problem definition for secure set union. Given n ($n \geq 2$) sites, each site holding a local set of data tuples or items x_i, we wish to compute $X = \bigcup x_i$ while minimizing the probability of a node revealing its ownership of x_i to other nodes. Note that secure union does not have practical sense when there are only 2 parties.

In this section we present a simple set union protocol that uses a random shares approach. To facilitate the discussion, we first describe a simple secure sum protocol to illustrate the random shares idea and then present the details of our random shares secure union protocol.

Secure sum. The secure sum protocol works as follows [8]. Assume that the sum value is known to lie in the range $[0..m]$ and that all the participating nodes are arranged in a ring. The first node generates a random number r and passes to the second node value $v_1 = (x_1 + r)\ mod\ m$. The second node receives the value v_1 from the first node, computes $v_2 = (v_1 + x_2)\ mod\ m$ and passes v_2 to the third node and so on. The last node sends value v_n to the first node. Then, the first node can use equation $(v_n - r)\ mod\ m$ to find the sum. The protocol can be modified in order to address the problem of colluding nodes. Instead of using only one round, it can use p rounds. In each round the ring is permuted and, instead of adding its x_i to the intermediate result, each node adds a random share of x_i. The random shares have to satisfy the following: $\sum_{j=1}^{p} s_{i_j} = x_i$ (s_{i_j} denotes a share contributed by node i in round j).

Random shares based secure union. Now we present a simple union protocol utilizing a similar random shares based approach. Our main design goal for the protocol is to be able to make a tradeoff between security and efficiency so that it can achieve reasonable and probabilistically bounded security at a much lower cost. There are three key ideas to the protocol. First, each node introduces random items so that it will not suffer from a provable exposure of its ownership of items. Second, a starting node is randomly selected so that nodes close to the starting node on the ring will not suffer from a high probability of data disclosure. Finally, the protocol uses multiple rounds and for each round the nodes are permuted and each node participates with a random share of its data items. This random shares based approach further minimizes the effect of potential collusion of the nodes. We describe the protocol as follows.

Phase 1. Random item addition. First, each node i generates a random set r_i and leading site l is chosen randomly. Next, the p rounds of the protocol begin

Algorithm 1 Random shares secure bag union protocol.

INPUT: x_i: local subset of node i
Each node i generate random set r_i, choose leader node, set $IR \leftarrow \emptyset$
Phase1
for round 1 to p **do**
  Arrange nodes in a ring topology randomly
  if leader node **then**
    Send $IR \cup_B x_{i_k} \cup_B r_{i_k}$ to *successor*
    Receive IR from *predecessor*
  else
    Receive IR from *predecessor*
    Send $IR \cup_B x_{i_k} \cup_B r_{i_k}$ to *successor*
  end if
end for
Phase 2
Arrange nodes in a ring topology randomly
if leader node **then**
  Send $IR -_B r_i$ to *successor*
  Receive IR from *predecessor*
  $Result \leftarrow IR$
else
  Receive IR from *predecessor*
  Send $IR -_B r_i$ to *successor*
end if

where each node adds its random share to the intermediate result. In each round, all the nodes are arranged in a ring randomly. This can be done for instance by selecting a random number t_i by each node. Then, the nodes can be arranged in the order indicated by growing values of t_i using a secure k-th element algorithm [2]. Once the nodes are arranged, the leading site adds a random share of its local set x_l and a random share of its random set r_l to the intermediate result from the previous round and passes the result to its successor. The other nodes perform the computation similarly. When node l receives the result from its predecessor, the next round begins. Note that each node has to choose random shares so that: $\cup_{j=1}^{p} x_{i_j} = x_i$ and $\cup_{j=1}^{p} r_{i_j} = r_i$ where x_{i_j} and r_{i_j} denote a random share of the data items and random items added to the intermediate result by node i in round j. When p rounds are completed, the protocol moves to the next phase.

Phase 2. Random item removal. A new random ring topology is generated (note that the leading site remains the same though). Next, the leading site l subtracts its random items r_l from intermediate results received in the previous phase and passes the result to the successor. Then, each node i subtracts its local random set r_i from the intermediate result and passes the result along the ring. When node l receives the result from its predecessor, the protocol finishes and the union is found. To further enhance the security of the protocol, one could also use p rounds for the second phase.

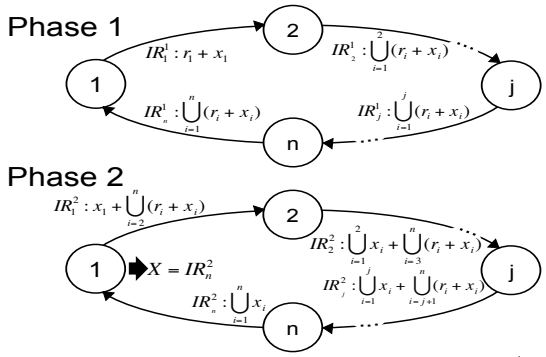

Fig. 1. Illustration of random shares set union protocol (single-round)

A sketch of the algorithm for bag union is presented in Algorithm 1 and Figure 1 presents an illustration of the protocol when only one round is used ($p = 1$). IR^1 and IR^2 represent intermediate result in Phase 1 and 2 respectively.

Random data item generation. An important issue in the protocol is the random data item generation. The questions we need to answer are: 1) how to generate a good random set r that look legitimate to other nodes and are indistinguishable from real data, and 2) what should be the size of r? We defer the second question to the next subsection when we analyze the protocol in detail and briefly discuss the first question here. There are a number of factors that need to be considered for generating legitimate items. First, the random item has to come from a legitimate domain. For numeric attributes, we assume the domain range is known to all the nodes. For discrete attributes with closed set of values (such as geographic entities), well-known dictionaries can be exploited. Second, if the distribution of an attribute is known, a node can generate random attribute values such that the distribution of the intermediate result (real items combined with random items) is sufficiently close to the global distribution using metrics such as Kullback-Leibler (KL) divergence [18] in order to protect its own distribution. Finally, if there is a correlation between attributes, a node needs to consider the correlations and generate attribute values based on their dependencies.

4 Analysis of Random Shares Union

While the protocol we just described is quite simple, the analysis is not trivial and is important to understand its security complications. We formally analyze the protocol in this section and believe this is one of the important contributions of the paper. We first introduce the security metric that we use for evaluating how well we achieve our security goal and present a formal analysis using this metric. We will plot the analytical bounds derived in this section along with our experimental results in the experiment section.

4.1 Security Metric and Attack Models

Our security goal is to prevent an adversary from being able to determine the ownership of items from the final result. Given the goal, we need to quantify the degree of data exposure for each node. We adopt the loss of privacy (LoP) metric [27] for this purpose. Let R denote the final result of the algorithm and IR denote the intermediate result during execution of the protocol. Suppose an adversary, as an attack, makes a claim C about the data at a node, we define two probabilities. The first, $P(C|IR, R)$, is the probability of claim C being true when node has both IR and R. The second, $P(C|R)$, is the probability of claim C being true when node has only the final result R. The loss of privacy is defined as follows:

$$LoP = P(C|IR, R) - P(C|R) \tag{1}$$

It essentially measures the difference between the posterior probability (with intermediate result) and the prior probability (without intermediate result) with respect to an attack. This, in spirit, is similar to the metric presented in [12, **?**] which measures the information disclose for anonymization using the notion of posterior probability (with published dataset) and the prior probability (with published dataset).

We consider two kinds of data exposure (or attacks), namely, *set exposure* and *item exposure*. For set exposure, an adversary is able to make a claim on the whole set of items a node contributes to the final union result (C = *node i contributed subset a_i to the final result*). For item exposure, an adversary is able to make a claim on a particular item a node contributes to the final result (e.g. C = *node i contributed item v_i to the final result*). In addition, *negative item exposure* is also possible, in which an adversary is able to make a claim on particular node not contributing a given item to the final result. Below we analyze the security for all these attacks respectively.

4.2 Security Analysis

We focus our analysis on single-round versions of the protocol ($p = 1$). Increasing the number of rounds will only increase the security of our solution.

Theorem 1. If there is no collusion, the Loss of Privacy (LoP) for the random shares based union protocol with respect to set exposure attack is bounded by:

$$LoP \leq \frac{1}{n-1} * \left(\frac{m - c + c/n}{m}\right)^{|r|} \tag{2}$$

Proof. As the worst case scenario, we consider the starting node (we assume node 1 is the starting node) as the victim and the second node (node 2) as the adversary. To begin with, we also assume that node 2 is aware that node 1 is the starting node. This is the worst case because node 2 only has to identify a set of real data items (not randomly generated items) from the intermediate result it receives from node 1 while any further adversary nodes will have to identify not

only the real items, but also the owner of the items. Suppose node 2 is trying to identify the whole set of items of node 1 by making a claim C: $x_1 = a_1$. We derive $P(C|R)$ and $P(C|IR, R)$ below and compute the LoP.

We start by computing $P(C|R)$, the probability of the claim being true given only the final result X. Given only X, the best an adversary (node 2) can do for guessing x_1 is to select a random number of items from X which are not among his own items x_2. If we assume that X contains c distinct items, the probability the claim is true is given by (note that x_1 can contain any number of items, so the adversary has to guess the size of this set and the items):

$$P(C|X) = \frac{1}{\sum_{a=0}^{c} \binom{|X - x_2|}{a}} \approx 0 \tag{3}$$

We are limiting the analysis above to the case when the result set contains only distinct items. As having duplicates helps an adversary, we are actually finding a lower bound for the probability $P(C|R)$ (and an upper bound for the LoP).

	r_1	x_1	X	$IR_1{}^1$	r_{alg}	$x1_{alg}$
Scenario 1	bbc	abb	abbdef	abbbbc	bbc	abb
Scenario 2	bbc	abb	abbbde	abbbbc	bc	abbb
Scenario 3	bbc	abb	abbcde	abbbbc	bb	abbc

r_{alg}: guessed random items set contributed by the first node

$x1_{alg}$: guessed set contributed by the first node

Fig. 2. Examples of data exposure of node 1 to node 2

We now compute $P(C|IR, R)$, the probability of the claim being true given the intermediate result IR_1^1 and the final result X. IR_1^1 contains the random set r_1 generated by node 1 and the subset x_1 contributed by node 1. If it happens that no item from r_1 appears in X, node 2 can determine r_1 using $r_1 = IR_1^1 - X$ and consequently determine x_1 using $x_1 = IR_1^1 - r_1$. Thus the best an adversary can do for guessing x_1 is to make a claim C: $x_1 = IR_1^1 - (IR_1^1 - X)$. Figure 2 presents a few possible scenarios for the claim. The probability of this claim being true can be derived as follows:

$$\begin{aligned} P(C|X, IR_1^1) &= P(x_1 = r_1 \cup x_1 - (r_1 \cup x_1 - X)) \\ &= P(x_1 = r_1 \cup x_1 - (r_1 - (X - x_1))) \\ &= P(r_1 \cap (X - x_1) = \emptyset) \end{aligned} \tag{4}$$

We will call the probability above $P_\emptyset$. If we assume that the domain M of items contains m distinct items and if we again assume that the result of union algorithm contains c distinct items and that nodes contribute in average c/n distinct items to the final set (a node does not have any knowledge on how

many records other nodes contribute, so a common assumption an adversary would use is a uniform distribution in that each node on average contribute c/n records; the analysis can be easily modified under non-uniform distributions in terms of number of records contributed by each node and it will not impact the result significantly), the probability of r_1 not containing any item from the set $X - x_1$ is given by:

$$P_\emptyset = \left(\frac{m - c + c/n}{m}\right)^{|r_1|} \tag{5}$$

Now we can derive the LoP for the starting node (given the knowledge of the starting node by the adversary). If the intersection $r_1 \cap (X - x_1)$ is empty, the adversary can identify whole subset contributed by the first node. If this subset is not empty, then any subset of the items from identified x_1 can actually be provided by the first node. Therefore, the probability of identifying the true subset x_1 is:

$$P(C|IR, R) = P_\emptyset + (1 - P_\emptyset)\frac{1}{\sum_{i=0}^{|x_1|}\binom{|x_1|}{i}} = P_\emptyset + (1 - P_\emptyset)\frac{1}{2^{|x_1|}} \tag{6}$$

The factor $2^{|x_1|}$ grows very fast, the second number in the equation above will be very small. Therefore, we can assume that $P(C|IR, R) = P_\emptyset$. Moreover, the above analysis assumed that node 2, the adversary node, is aware that node 1 is the starting node. As our protocol utilizes a randomized starting scheme, the probability of a node being the starting node is $\frac{1}{n-1}$ (assuming an adversary node is not the starting node). Note also that the attack associated with $P_\emptyset$ is possible only if the attacker is the second node. Thus we derive the bound of LoP as presented in equation 2. □

Theorem 2. If k ($k >= 2$) nodes collude, the LoP for the union protocol with respect to set exposure attack is bounded by:

$$LoP \leq max(\frac{2(k-1)(n-k)}{(n-1)(n-2)} * \left(\frac{m - c + c/n}{m}\right)^{|r|}, \frac{2k(k-1)^2}{n^3}) \tag{7}$$

Proof. We will assume that in the ring nodes $i-1$ and $i+1$ collude in order to identify items provided by node i. To alleviate the problem of nodes collusion, the fact that each node generates random items helps to limit the LoP. If the nodes collude in the first phase of the protocol, they can identify set $(x_i + r_i)$ using the following formula: $(x_i + r_i) = IR^1_{i+1} - IR^1_i$. If in the second phase nodes do not end up in the same positions in the ring, the best they can do is to proceed according to the algorithm discussed above. As the ring is generated randomly, the probability of two colluding nodes being arranged in the way that makes this attack possible is $\frac{2}{n-1}$ (for the attack to be possible the first colluding node can be placed in any place in the ring; then the second colluding node can be placed two positions before the first one or two positions after). The analysis is similar to the analysis above with respect to the attack from the second node

guessing the set provided by the first node, and LoP can be estimated using equation 6 as:

$$LoP = \frac{2}{n-1} * \left(\frac{m-c+c/n}{m}\right)^{|r|} \tag{8}$$

If colluding nodes surround the same node in the first and second rounds, the privacy of the surrounded node can be clearly compromised, as the whole set provided by this node can be identified. The probability of such scenario can be estimated as follows: $\frac{2}{n-1} * \frac{2}{n-1} * \frac{1}{n-2}$ (the colluding nodes have to surround the same node in the first and second phases). For larger n the probability can be approximated as $\frac{4}{n^3}$ and the LoP can be estimated as:

$$LoP = max(\frac{2}{n-1} * \left(\frac{m-c+c/n}{m}\right)^{|r|}, \frac{4}{n^3}) \tag{9}$$

When more than two sites collude, the probability of one of the scenarios analyzed above is higher. Let's assume a general case of k colluding sites. The first scenario we analyzed above was when colluding nodes surrounded the attacked node only in the first phase. This probability can be calculated as $\frac{2(k-1)(n-k)}{(n-1)(n-2)}$ (any 2 of the k nodes can surround a node that will be attacked). On the other hand, the probability of surrounding attacked node by colluding sites in both phases of the union algorithm can be estimated as follows: $\frac{2(k-1)(n-k)}{(n-1)(n-2)} * \frac{k}{n-1} * \frac{k-1}{n-2}$ which for larger n can be estimated as $\frac{2k(k-1)^2}{n^3}$. Thus, the LoP when k sites collude is bounded by eq. 7. □

Theorem 3. If there is no collusion, the LoP of the protocol with respect to item exposure is as follows:

$$LoP \leq \frac{\sum_{i=1}^{n-1} \frac{1}{i}}{n-1} * \frac{2}{1+|r| * \frac{n-1}{m}} - \frac{1}{n-1} \tag{10}$$

Proof. We again consider the starting node (node 1) as the victim and the second node (node 2) as an adversary. Suppose node 2 is trying to identify a single item contributed by node 1, x_1, by making a claim C: $v_1 \in x_1$.

We first compute $P(C|R)$. Given only X, the best an adversary can do for guessing a single item in x_1 is to select an item from X. The probability this claim being true is: $P(C|R) = \frac{1}{n-1}$.

We now compute $P(C|IR, R)$, the probability of the claim being true given the intermediate result IR_1^1 and the final result X, and analyze how the intermediate result can help this node to find the owner of some items. Using a similar approach as in the set exposure scenario, the adversary can try to find items contributed by its predecessor. Simply put, the less items from random set r_1 are in the algorithm's final result, the easier the second node can identify items contributed by the first node. Specifically, observing again the scenarios presented in Figure 2, and assuming that node 2 is aware of node 1 being the leader, the

probability $P(C|IR, R)$ is given by (we assume that each node contributes in average $\frac{c}{n}$ items and generates $|r|$ random items):

$$P(C|IR,R) = \frac{|x_1| + |x_1 \cap (r_1 \cap (X - x_2))|}{|x_1| + |(r_1 \cap (X - x_2))|} \leq \frac{2 * \frac{c}{n}}{\frac{c}{n} + |r| * \frac{c - \frac{c}{n}}{m}} = \frac{2}{1 + |r| * \frac{n-1}{m}} \tag{11}$$

The equation above considers only the case when adversary is the second node in the ring. The attack is also possible when adversary is in third or any other position. In this case, the attack is successful if the adversary identifies a real item in the intermediate result, and if it can guess the real owner of the real item. If adversary is at the third position, the probability of guessing an owner is $\frac{1}{2}$, if it is at 4th position, the probability is $\frac{1}{3}$ and so on. As the adversary can be located at any position in the ring, the overall probability $P(C|IR, R)$ can be thus estimated as sum of all those factors. Considering this fact and randomized startup scheme, the item exposure is presented in equation 10. □

Theorem 4. If k ($k >= 2$) nodes collude, the LoP of the protocol with respect to item exposure is as follows:

$$LoP \leq max(\tfrac{2(k-1)(n-k)}{(n-1)^2(n-2)} * \tfrac{2}{1+|r|*\frac{n-1}{m}}, \tfrac{2k(k-1)^2}{n^3}) - \tfrac{1}{n-1} \tag{12}$$

Proof. The analysis of a scenario with collusion between nodes is quite similar to that for the set exposure. If colluding nodes surround a victim node only in the first phase, the fact of generating random items by each node brings the analysis to a similar scenario as discussed above. On the other hand, if colluding nodes surround the same node in both rounds, all the items of the attacked node can be compromised. If there are k colluding nodes, the overall LoP can be estimated as in equation 12. □

Theorem 5. If there is not collusion, the negative item exposure is bounded by the following:

$$LoP_{neg} \leq \frac{1}{n-1}(1 - \prod_{i=1}^{c-c/n} \frac{|r| - i}{m - |r| + i}) \tag{13}$$

Proof. We will assume a case when node 1 is a victim and node 2 is an attacker, as this case is the worst case scenario. Following a similar attack scenario as in previous analysis, the adversary is sure that the items from set $X - IR_1^1$ do not belong to node 1. The probability of this set not being empty can be estimated as:

$$P_{ne} = 1 - \frac{\binom{m}{|r_1| - (c - c/n))}}{\binom{m}{|r|}} = 1 - \prod_{i=1}^{c-c/n} \frac{|r| - i}{m - |r| + i} \tag{14}$$

When a node knows only the final result, the probability of identifying item not belonging node 1 by node 2 is $P(C|R) = 1 - \frac{1}{n-1}$. Therefore, the negative LoP can be estimated as $P_{ne} * (1 - (1 - \frac{1}{n-1}))$, and is presented in equation 13. □

Comparison with probabilistic secure union. It is worth comparing our analysis with the probabilistic union protocol since both give a probabilistic security bound. The probability bound derived in [5] for the probabilistic union protocol, in fact, corresponds to our definition of $P(C|IR, R)$ with respect to item exposure when there is no collusion. So the LoP for the item exposure of probabilistic union protocol without collusion can be estimated as $0.71 - \frac{1}{n-1}$.

Random data item generation for guaranteed security. One remaining issue we have left from previous subsection is how many random items a node should generate in Phase 1. Based on the previous theorems, we can derive the minimum number of random items that are required to guarantee a given LoP bound.

Set exposure. Given a desired bound $LoP_{expected}$ for set exposure, we can obtain the minimum required number of random items when there is no collusion based on Theorem 1 as in Equation (15). When there is collusion, it can be similarly estimated using Theorem 2. In the collusion case, however, the LoP cannot be smaller than the factor $\frac{k(k-1)}{n^3}$ which does not depend on the size of the random set.

$$|r| = \lceil \log_{\frac{m-c+c/n}{m}} (n-1) * LoP_{expected} \rceil \tag{15}$$

Item exposure. Given a desired bound $LoP_{expected}$ for item exposure, we can obtain the minimal number of random items when there is no collusion based on Theorem 3 as in Equation (16). The minimal number of random items in the case of collusion can be similarly derived using Theorem 4.

$$|r| = \lceil \frac{m}{n-1} * \left(\frac{2\sum_{i=1}^{n-1} \frac{1}{i}}{(n-1)LoP_{expected} + 1} - 1 \right) \rceil \tag{16}$$

4.3 Cost Analysis

The communication and computation costs of the protocol are $kn(\frac{d}{2} + \frac{3}{2}nd + nr)$ and $nC_s(2p+1)$, respectively, where k is the average size of an item, d is the average number of items owned by a node, r is the size of random set generated by each node and C_s is the cost of a set operation (union/intersection/difference) on two input sets. To obtain the cost of the protocol for a desired LoP value, one can use Equations (15) and (16) to calculate the required r for given LoP and apply those values in estimating the cost.

5 Experimental Evaluation

In this section we will present a set of experimental evaluations of the proposed protocols. The questions we attempt to answer are: 1) How do the proposed protocols perform in terms of security in various settings and how does the result compare with the analytical results, and 2) What is the cost of our protocols in comparison to other options?

Parameter name	Description	Default value
m	Size of domain	100,000
n	Number of participating nodes	20
c	Size of algorithm result	1000
r	Number of generated random items	varies
p	Number of rounds	1

Table 1. Experiment parameters

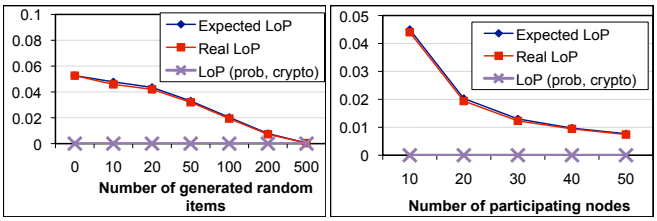

Fig. 3. Set exposure for union (single round)

5.1 Security of Random Shares Union

We have implemented the random shares based protocols for both secure union and secure intersection. To answer the first question above, we prepared a simulation of a distributed environment and used synthetically generated data with varying parameters which allowed us to test and evaluate the protocol in multiple scenarios and settings. A summary of the set of simulation parameters is presented in Table 1. In all the experiments the default values are used unless otherwise specified. We have assumed that nodes contribute in average $\frac{c}{n}$ items to the final result. We report the results for both set exposure and item exposure attacks.

Set exposure without Collusion. We first evaluate the set exposure when there is no collusion and the impact of the number of random items used in the protocol and the number of participating nodes.

Figure 3 present the analytical LoP bound (recall Equation (6)) and the actual LoP obtained from the experiments for a single round protocol ($p = 1$) with varying number of random items and participating nodes respectively. We observe that LoP decrease as number of random items increases. Given the default number of nodes, even when no random items are generated, the algorithm provides quite reasonable security (LoP is around 0.05) by utilizing the inherent anonymity of the network. Given a smaller number of nodes, generation of random items becomes more essential. On the other hand, when the number of nodes in the network increases (100 random items are used by each node), both expected and actual LoP decrease due to the increased anonymity of the network. Both plots verify that the value of actual LoP is lower than, though close to, the analytical bound. The LoP of the proposed random shares union is also compared with LoP for cryptographic and probabilistic secure union protocols. While the cryptographic protocols do not reveal additional information

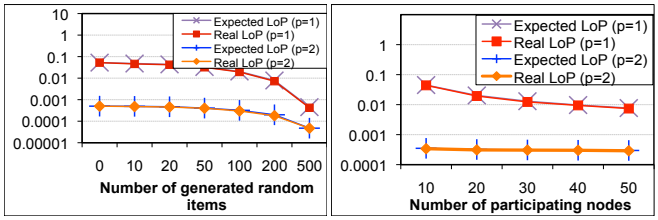

Fig. 4. Set exposure for union (multiple rounds)

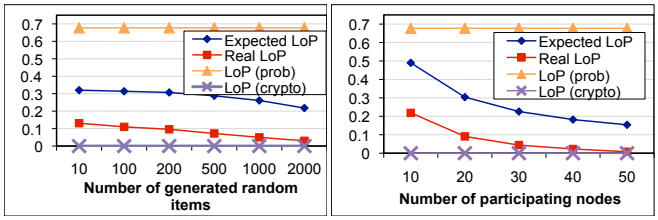

Fig. 5. Item exposure for union

($LoP = 0$), and probabilistic protocol introduces very low $LoP \approx 0$, the LoP introduced by our protocol is also relatively small.

Figure 4 compares the single-round version ($p = 1$) with the multiple-round version ($p = 2$) to show the effect of using multiple rounds and reports the expected and real LoP for both cases. The results demonstrate that increasing number of rounds reduces the average LoP by a significant factor. Similar to single-round protocol, the expected LoP is always lower than the analytical LoP, although the difference is almost invisible in the plots.

Item exposure without collusion. Figure 5 presents the LoP results for item exposure with varying number of generated random items and participating nodes. With varying number of participating nodes, we generated 5,000 random items in total. Similar to the set exposure, an increase in the number of random items and participants leads to a reduction in the value of expected and actual LoP. It is worth noting that the actual LoP value is around half of the analytical value. Such a phenomenon is worth an explanation. When we derived Equation 16, we have assumed worst case scenario and assumed that $|x_1 \cap (r \cap (x - x_2))| = |x_1|$. On the other hand, in most cases the value of $|x_1 \cap (r \cap (x - x_2))|$ will be significantly smaller.

The result also show a comparison with commutative cryptographic and probabilistic protocols. While cryptographic protocols do not incur any item exposure ($LoP = 0$), the probabilistic protocol introduces the constant LoP of item exposure of value ≈ 0.68. In this respect, our protocol performs much better as the LoP is much smaller, and can also be adjusted as desired.

Set exposure and item exposure with collusion. Now we experimentally evaluate the impact of collusion between nodes. We used again 20 nodes in the

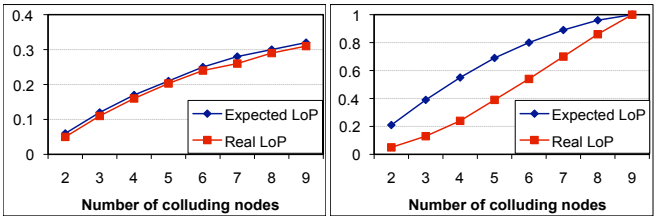

Fig. 6. Set exposure and item exposure for union with collusion

network, and varied number of colluding nodes from 2 up to 9 nodes. Each node generated 50 random items. The results for set and item exposure are presented in Figure 6. It can be observed that an increase in the number of colluding nodes leads to an increase in the expected and real LoP. Unfortunately, when there is half of the nodes colluding in the network, the nodes has a provable exposure ($LoP = 1$). On the other hand, the protocol achieves reasonable security even when there is a small number of nodes colluding with each other.

5.2 Cost of Secure Union Protocols

While the analytical results of our protocol and existing protocols do not allow direct comparison, we experimentally evaluate and compare the proposed protocol with the representative existing protocols in this section. We implemented the circuit-based, commutative cryptography-based, probabilistic, as well as the anonymous communication-based protocols we discussed earlier. The circuit-based protocol was implemented using the FairplayMP [6] framework. The implementation of the commutative cryptography-based protocol was based on RSA cipher. For the anonymous communication protocol, we used a communication circuit of 4 machines (excluding the sender and recipient).

We simulated a distributed environment with $n = 20$ nodes and measured time of execution for each of the protocols except the circuit-based protocol. Due to a large time of execution, the runtime of the circuit-based protocol was estimated based on a performance analysis of the FairplayMP presented in [6]. We ran each of the other protocols for different result sizes and domain sizes. In the random shares protocol, we set the size of the generated random set at each node to 10000 items. The cost of leader/ring selection is not reported. However, this cost will be insignificant and very small if compared with the cost of the protocol itself.

The results are presented in Figure 7. First, we can observe that the commutative cryptography-based, anonymous communication-based and random shares-based protocols do not depend significantly on the domain size. On the other hand, for the probabilistic protocol, the runtime is strongly determined by this size due to its use of the bit vector. For $m = 2^{22}$ the protocol runtime is significantly larger than the random shares-based protocol even though the security provided by the latter is better. Finally, the costs of commutative cryptography-based and anonymous communication-based protocols increase as the result size

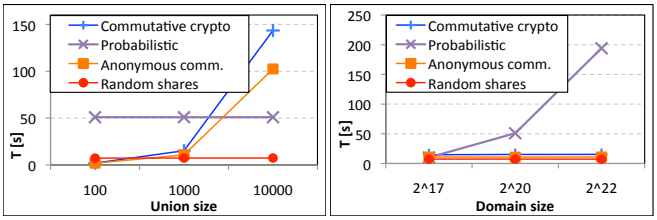

Fig. 7. Cost comparison of union protocols

increases due to their dependence on an encryption. For a small result size, these protocols perform better than the random shares protocol. However, for larger result size, the random shares protocol performs much better.

Runtime of the circuit-based protocol was not placed on the plots due to very large values. For the domains we tested, FairplayMP generated circuits of size $2.8*10^7$, $2.2*10^8$ and $9*10^8$ gates. The estimated runtime for such circuits is 15 days, 127 days and 1.4 years, respectively. This makes the protocol impractical for most real life problems.

6 Conclusion

In this paper we have reviewed different secure union protocols and presented a simple and intuitive secure union protocol based on the random shares approach. Our formal analysis and experimental results indicate that our protocols are more efficient than existing protocols while achieving reasonable level of security that can be adjusted by users. While the circuit-based and probabilistic protocols turned out to be too costly for larger domain size, the cryptography and anonymous communication-based approaches performed quite well. We believe that it is desirable to make a tradeoff between security, efficiency and possibly accuracy for certain practical settings. Our future work include investigating malicious models and generalization of the random shares approach to other operators.

References

1. M. Abadi and J. Feigenbaum. Secure circuit evaluation. *J. Cryptol.*, 2(1), 1990.
2. G. Aggarwal, N. Mishra, and B. Pinkas. Secure computation of the kth ranked element. In *Eurocrypt*, 2004.
3. R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases, 2003.
4. R. Agrawal and E. Terzi. On honesty in sovereign information sharing. In *EDBT*, pages 240–256, 2006.
5. M. Bawa, R. J. Bayardo, Jr., and R. Agrawal. Privacy-preserving indexing of documents on the network. In *VLDB*, 2003.
6. A. Ben-David, N. Nisan, and B. Pinkas. Fairplaymp: a system for secure multi-party computation. In *CCS*, 2008.

7. S. Böttcher and S. Obermeier. Secure set union and bag union computation for guaranteeing anonymity of distrustful participants. *JSW*, 3(1):9–17, 2008.
8. C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, and M. Zhu. Tools for privacy preserving distributed data mining, 2003.
9. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, 2004.
10. W. Du and M. J. Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *New security paradigms workshop (NSPW)*, 2001.
11. W. Du and Z. Zhan. Building decision tree classifier on private data. In *CRPIT '14: Proceedings of the IEEE international conference on Privacy, security and data mining*, 2002.
12. A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 2003.
13. O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.
14. W. Jiang, C. Clifton, and M. Kantarciouglu. Transforming semi-honest protocols to ensure accountability. *Data Knowl. Eng.*, 65(1), 2008.
15. M. Kantarcioglu and C. Clifton. Privacy preserving data mining of association rules on horizontally partitioned data. *IEEE TKDE*, 16(9), 2004.
16. H. Kargupta, K. Das, and K. Liu. Multi-party, privacy-preserving distributed data mining using a game theoretic framework. In *PKDD*, 2007.
17. L. Kissner and D. Song. Privacy-preserving set operations. In *CRYPTO*, 2005.
18. S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 1951.
19. Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3), 2002.
20. Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1), 2009.
21. B. Malin and L. Sweeney. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics*, 37(3):179–192, 2004.
22. B. Schneier. *Applied Cryptography*. John Wiley & Sons, 2nd edition, 1996.
23. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *SIGKDD*, 2002.
24. J. Vaidya and C. Clifton. Privacy-preserving data mining: Why, how, and when. *IEEE Security & Privacy*, 2(6):19–27, 2004.
25. J. Vaidya and C. Clifton. Secure set intersection cardinality with application to association rule mining. *J. Comput. Secur.*, 13(4), 2005.
26. J. Vaidya, M. Kantarcioglu, and C. Clifton. Privacy-preserving naïve bayes classification. *VLDB J.*, 17(4):879–898, 2008.
27. L. Xiong, S. Chitti, and L. Liu. Preserving data privacy for outsourcing data aggregation services. *ACM Transactions on Internet Technology (TOIT)*, 7(3), 2007.
28. Z. Yang, S. Zhong, and R. N. Wright. Privacy-preserving classification of customer data without loss of accuracy. In *SIAM SDM*, 2005.
29. A. C.-C. Yao. How to generate and exchange secrets. In *SFCS '86: Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 1986.
30. N. Zhang and W. Zhao. Distributed privacy preserving information sharing. In *VLDB*, 2005.