

Information Sharing Across Private Databases: Secure Union Revisited

Pawel Jurczyk
Emory University
Atlanta, Georgia 30322
Email: pjurczyk@gmail.com

Li Xiong
Emory University
Atlanta, Georgia 30322
Email: lxiong@emory.edu

Abstract—There is a growing demand for sharing information across multiple autonomous and private databases. The problem is usually formulated as a secure multi-party computation problem in which 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 perform an analysis and an experimental evaluation of existing and potential solutions for secure multi-party computation of union. We also present an alternative random shares based protocol and show that the protocol, although quite simple, is efficient while providing reasonable level of security that can be adjusted by users. We formally analyze the security properties and the cost of our protocol. We also experimentally compare the various existing and potential solutions and show the tradeoff between different protocols in terms of security, efficiency and accuracy.

I. 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 [12], [22] in which 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 secure union problem, in which multiple parties wish to compute the union of their data items without disclosing the ownership of the data. For secure union, all data items will be revealed as part of the result, however, the owner of a certain data item shall not be disclosed. Formally, given n ($n \geq 2$) nodes, each node holding a local set of data items x_i from domain M , we wish to compute $X = \bigcup x_i$ without revealing a node's ownership of x_i to other nodes. We focus on the *semi-honest* adversary model commonly used in SMC problems. A semi-honest party follows 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 in many scenarios where multiple organizations are collaborating in order to get the correct result for their mutual benefit.

Contributions. In this paper, we review and analyze existing representative secure union protocols as well as anonymous communication protocols as a potential solution for the secure set operations. We propose an alternative simple yet effective protocol based on a random shares approach. In contrast to traditional SMC protocols, it achieves 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 evaluated 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.

II. RELATED WORK

In the problem of secure multi-party computation (SMC) [10], [12], [22], 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 [23], the k th element [2], set intersection [3], set intersection size [3], [27], set union [15], [7], and dot product [16]. A closely related research area is privacy preserving data mining across distributed data sources [8], [26], [22]. 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 tasks to be carried out. Specialized protocols are designed for various mining and tasks (some examples include [25], [3], [15], [29], [28], [21], [20], [13]). Most above work assume an honest or semi-honest adversary model [12]. Other works focus on broader threat space including malicious adversaries [31], [4], [17], [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], [30], 2) specialized cryptography-based protocols using commutative encryption schemes[8], [15], [7] or homomorphic encryption schemes and polynomial representation of sets [18], 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.

III. ANALYSIS OF EXISTING PROTOCOLS

In this section, we include a brief discussion and analysis of existing representative protocols. We briefly discuss their adaptability from set union to bag union or vice versa. When available, we cite the analysis results from the original papers. Otherwise, we present our analysis results in terms of security and performance. As the protocols are based on different principles and parameters, the analytical results are not directly comparable. However, we believe they still provide a formal understanding of the complexity and security characteristics of each approach. More importantly, we experimentally compare the cost of the representative protocols in the experiment section.

A. Circuit-Based Secure Union

The secure union can be implemented using secure circuit evaluation [1]. First, each node creates a bit vector with as many bits as there are items in the domain. Next, the nodes generate a circuit that computes bitwise OR operation on all the vectors. The algorithm can be modified to compute a bag union by calculating sum instead of OR.

Security and Cost. The circuit-based protocol is indeed *provably secure*. However it is computationally prohibitive in practice. First, the size of a circuit depends on the domain size for the data items. For larger domains the circuit calculation can be costly. Second, the size of data being transferred between nodes does not depend on size of the result, but on the domain size. As a result, the cost of secure circuit generation and evaluation add significant overhead. We estimated the cost of communication and computation for a semi-honest variant of Yao’s protocol using a similar analysis as the one presented in [3]. The number of gates the protocol requires is $n(|M|)G_e$ and the corresponding communication and computation costs are $4k_cn(|M|)G_e$ and $2C_r n(|M|)G_e$, respectively, where n is the number of nodes, $|M|$ is the domain size of the items, k_c is the size (in bits) of keys used for circuit gates, G_e is the number of gates required to compare 2 numbers, and C_r is the cost of pseudorandom function evaluation.

B. Cryptography-Based Secure Union

As the general circuit-based solution is extremely expensive, specialized cryptography-based protocols are proposed for the union operation based on commutative encryption schemes[8], [15], [7] or homomorphic encryption schemes and polynomial representation of sets [18]. We will use the protocol based on commutative encryption [15] as a representative in our analysis for this category. The basic idea is that each node

encrypts its own items as well as received items and decrypts them due to the commutative property of the encryption. The algorithm finds a bag-union without revealing which item was contributed by which node. To calculate the set union, one can remove the duplicates in the fully-encrypted set before the decryption phase.

Security. As proved in [15], the discussed protocol securely computes union, *revealing* a bounded set of innocuous information such as size of the intersection of the data items and number of items at the nodes.

Cost. Using a similar analysis as the one presented in [3], we conducted a cost analysis for the protocol. The estimate for the communication cost is $n^2 dk_e(2n+1)$ and the computation cost is $2n^2 C_e d$, where n is the number of nodes, d is the average number of items provided by each node, k_e is the size of encrypted item (in bits) and C_e is the cost of encryption/decryption of an item.

C. Probabilistic Secure Union

A probabilistic secure union algorithm was proposed in [5] to address the concerns of high overhead associated with traditional SMC protocols. The protocol uses a bit vector V_i to represent the data items at each node and calculates the logical OR of the bit vectors. The main idea is to use multiple rounds with randomization in each round. The algorithm finds a set union and its modification to calculate a bag union can be problematic. In case of a bag union, the intermediate vector V should store counts of items, and thus the probabilistic bit flipping approach is not easily applicable.

Correctness. As shown in [5], the protocol is not deterministic and the result is correct only with certain probability guarantee. For a given number of rounds, $p \geq \max(3, -\log[1 - \{\frac{8}{7}(1-\epsilon)\}^{1/(n-1)}])$, the probability of having an error in each bit of the result vector is at most ϵ .

Security. The protocol is not absolutely secure and does reveal information about the local data. [5] proved that the probability of one node deducing that its successor has a given data item is 0.71. Unfortunately, when nodes collude, this probability is much higher (however, no details are given in the paper).

Cost. We also conducted a cost analysis of the protocol. The estimate for the communication cost is $pn|M|$ and the computation cost is $rn|M|C_c$, where n is the number of nodes, p is the number of rounds of the protocol, C_c is the cost of evaluating if statements in the protocol.

D. Anonymous Communication-Based Secure Union

Due to the nature of set union operation and its main goal to protect the anonymity of the data owners, anonymous communication techniques [9] are particularly suitable for implementing secure union computations. We could simply adopt an anonymous communication protocol using a random path way, circuit, to ship all the data items to a single node. The protocol finds a bag union and it can be modified to remove duplicates, similarly as the cryptography-based approach.

Security. The protocol described above guarantees security provided that no nodes collude, revealing the size of intersection between nodes (the intersection can be calculated using encrypted items sent to the node computing union) and size of subsets owned by other nodes (protecting the identity of those nodes). If the recipient of data items colludes with some nodes from the communication circuit, the risk of corrupting security increases. Such a risk can be greatly minimized by using longer circuits. In the description above, even though the exact node is unknown, the recipient gains knowledge about a given set of items owned by some node. To minimize this exposure, the nodes can ship data in a few random packets. In the case of set union which removes duplicates, the recipient node learns exact duplicate items (not only the encrypted values).

Cost. The estimate for the communication cost of the protocol is $ndk_e(c + 1)$ and the computation cost is $2nC_e dc$, where n , d , k_e , and C_e have the same meaning as in the previous subsections and c is the number of nodes in a circuit.

IV. RANDOM SHARES BASED UNION

In this section we present a simple set union protocol that uses a random shares approach inspired by the simple secure sum protocol [8]. 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.

Algorithm 1 Random shares secure bag union protocol.

```

1: INPUT:  $x_i$ : local subset of node  $i$ 
2: Each node  $i$  generate random set  $r_i$ , choose leader node, set  $IR \leftarrow \emptyset$ 
3: Phase 1
4: for round  $j$  from 1 to  $p$  do
5:   Arrange nodes in a ring topology randomly
6:   if leader node then
7:     Send  $IR \cup_B x_{i_j} \cup_B r_{i_j}$  to successor
8:     Receive  $IR$  from predecessor
9:   else
10:    Receive  $IR$  from predecessor
11:    Send  $IR \cup_B x_{i_j} \cup_B r_{i_j}$  to successor
12:   end if
13: end for
14: Phase 2
15: Arrange nodes in a ring topology randomly
16: if leader node then
17:   Send  $IR -_B r_i$  to successor
18:   Receive  $IR$  from predecessor
19:   Result  $\leftarrow IR$ 
20: else
21:   Receive  $IR$  from predecessor
22:   Send  $IR -_B r_i$  to successor
23: end if

```

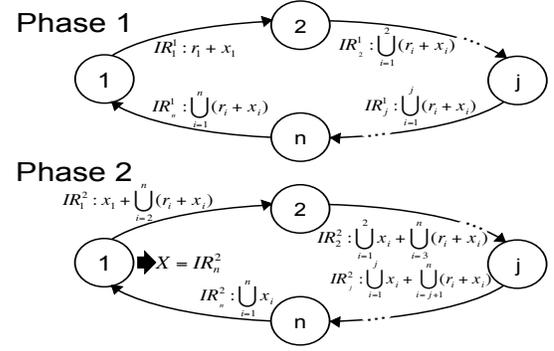


Fig. 1. Illustration of random shares set union protocol (single-round)

Phase 1. Random item addition. First, each node i generates a random set r_i and leading node l is chosen randomly. Next, the p rounds of the protocol begin where each node adds its random share to the intermediate result. In round j , all 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 ascending order of t_i using a secure k -th element protocol [2]. Once the nodes are arranged, the leading node adds a random share of its local set x_{l_j} and a random share of its random set r_{l_j} 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 such 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 node remains the same). Next, the leading node l subtracts its random items r_l from the intermediate result 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 final result gives the union. To further enhance the security of the protocol, one could also use p rounds for the second phase.

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 [19] in order to protect its own distribution. Finally, if there is a correlation between attributes, a node needs to consider the correlation and generate attribute values based on their dependencies.

V. 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 attack model and a 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.

A. Attack Model and Security Metric

Our security goal is to prevent an adversary from being able to determine the ownership of items from the final result. We consider two kinds of data exposure or attacks, namely, *set exposure* and *item exposure*. For set exposure attack, an adversary makes a claim C on the whole set of items a node i contributes to the final union result X ($C: x_i = a_i$). For item exposure attack, an adversary makes a claim C on a particular item a node i contributes to the final result ($C: v_i \in x_i$). Please note that *negative item exposure* is also possible, in which an adversary makes a claim on particular node not contributing a given item to X . We omit the analysis of negative item exposure in this paper due to space limitations and focus on the item exposure and set exposure in the rest of the analysis.

In order to quantify the degree of information exposure, we measure the change of belief of an adversary with respect to an attack or claim due to the intermediate results observed during the execution of the protocol [29]. Let X denote the final result of a protocol and IR denote the intermediate result observed during execution of the protocol. Suppose an adversary node amounts an item exposure or set exposure attack by making a claim C , we denote $P(C|IR, X)$ as the probability of C being true when the node has access to both IR and X , and $P(C|X)$ as the probability of C being true when node has access to only X , as if the nodes are using a Trusted Third Party to do the computation. We define the change of belief as follows:

$$LoP = P(C|IR, X) - P(C|X) \quad (1)$$

The metric measures the difference between the posterior probability (with intermediate results) and the prior probability

(without intermediate results). In spirit, it is similar to the adversarial privacy metric [11], [24] that measures the information disclose due to the publishing of an anonymized dataset by the change of belief or the difference between the posterior probability (with published dataset) and the prior probability (without published dataset).

B. 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.

Set Exposure Attack. We consider an adversary who attempts to make a claim about the set of items of its predecessor based on the intermediate result and the final result it receives. Assuming node 1 is the leader node, the attack will be most successful when the adversary is node 2 following the leader node. The reason is that 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. For node 2, 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 we assume the above attack strategy for an adversary and it makes the following claim about its predecessor's items: $x_i = IR_i - (IR_i - X)$. Figure 2 presents a few possible scenarios for the claim. Without the intermediate result, the best attack strategy for an adversary is to select a random number of items from X which are not among his own items. Hence the claim takes the form: $x_i = a$, where $a \subset X - x_{i+1}$.

Theorem 1. If there is no collusion, the change of belief with respect to the above set exposure attack for the random shares based union protocol is bounded by:

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

where n is the number of nodes, m is the size of the item domain M , c is the number of distinctive items in the final result R , and r is the random set.

Proof. Suppose the adversary is node 2. We start by computing $P(C|X)$, the probability of the claim being true given only the final result X . 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):

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

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 $P(C|X)$ (and an upper bound for the LoP).

	r_1	x_1	X	IR_1^1	r_{alg}	x_{1alg}^1
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
 x_{1alg}^1 : guessed set contributed by the first node

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

We now compute $P(C|IR, X)$, the probability of the claim being true given the intermediate result IR_1^1 and the final result X . It can be derived as follows:

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

If we assume that the item domain M contains m distinct items, X contains c distinct items, and the nodes contribute on 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:

$$P(C|X, IR_1^1) = \left(\frac{m - c + c/n}{m} \right)^{|r_1|} \quad (5)$$

As our protocol utilizes a randomized starting scheme, the probability of a node being node 2 following the starting node is $\frac{1}{n-1}$ (assuming an adversary node is not the starting node). Thus we derive the bound of LoP as presented in equation 2. ■

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

$$LoP \leq \max\left(\frac{2(k-1)(n-k)}{(n-1)(n-2)} * \left(\frac{m - c + c/n}{m} \right)^{|r_1|}, \frac{2k(k-1)^2}{n^3} \right) \quad (6)$$

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: $(x_i + r_i) = IR_{i+1}^1 - IR_i^1$. If in the second phase nodes do not end up in the same positions in the ring, the best attack strategy is for them 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 5 as:

$$LoP = \frac{2}{n-1} * \left(\frac{m - c + c/n}{m} \right)^{|r_1|} \quad (7)$$

If colluding nodes surround the same node in the first and second rounds, the data at 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:

$$LoP = \max\left(\frac{2}{n-1} * \left(\frac{m - c + c/n}{m} \right)^{|r_1|}, \frac{4}{n^3} \right) \quad (8)$$

When more than two nodes collude, the probability of one of the scenarios analyzed above is higher. Let's assume a general case of k colluding nodes. 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 nodes 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 nodes collude is bounded by eq. 6. ■

Item Exposure Attack. We consider an adversary who attempts to make a claim about one particular item of its predecessor based on the intermediate result and the final result it receives. Using a similar attack strategy as in the set exposure attack, the adversary makes the following claim: $v \in x_i$, where $v \in IR_i - (IR_i - X)$. Without the intermediate result, the best attack strategy for an adversary is to select a random item from X which are not among his own items. Hence the claim takes the form: $v \in x_i$, where $v \in X - x_{i+1}$.

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

$$LoP \leq \frac{\sum_{i=1}^{n-1} \frac{1}{i}}{n-1} * \frac{2}{1 + |r| * \frac{n-1}{m}} - \frac{1}{n-1} \quad (9)$$

Proof. We again consider the starting node (node 1) as the victim and the second node (node 2) as an adversary. We first compute $P(C|X)$. Given the attack strategy, the probability of a random item from $X - x_2$ belongs to node 1 is $P(C|X) = \frac{1}{n-1}$.

We now compute $P(C|IR_1^1, X)$, the probability of the claim being true given the intermediate result IR_1^1 and the final result X . Given the attack strategy, the less items from random set r_1 are in the final result, the easier the second node can identify items contributed by the first node. Specifically, observing again the scenarios presented in Figure 2, the probability is

given by (we assume that each node contributes on average $\frac{c}{n}$ items and generates $|r|$ random items):

$$P(C|IR_1^1, X) = \frac{|x_1| + |x_1 \cap (r_1 \cap (X - x_2))|}{|x_1| + |(r_1 \cap (X - x_2))|} \leq \frac{2 * \frac{c}{n}}{\frac{c}{n} + |r| * \frac{c - \frac{c}{n}}{m}} = \frac{2}{1 + |r| * \frac{n-1}{m}} \quad (10)$$

The equation above considers 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 $P(C|IR, X)$ can be thus estimated as sum of all those factors. Considering this fact and randomized startup scheme, the item exposure is presented in equation 9. ■

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

$$LoP \leq \max\left(\frac{2(k-1)(n-k)}{(n-1)^2(n-2)} * \frac{2}{1+|r| * \frac{n-1}{m}}, \frac{2k(k-1)^2}{n^3}\right) - \frac{1}{n-1} \quad (11)$$

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 11. ■

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 $P(C|IR, X)$ 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 question we have left from previous section is the size of r , i.e. how many random items a node should generate in Phase 1. Based on the previous theorems, we note that the larger $|r|$, the better security the protocol provides. We can derive the minimum number of random items that are required to generate in order to guarantee a given LoP bound with respect to set exposure and item exposure. We omit the detailed results in this paper due to space restrictions.

C. Cost Analysis

The communication and computation costs of the protocol are $n(\frac{d}{2} + \frac{3}{2}nd + n|r|)$ and $nC_s(2p + 1)$, respectively, where d is the average number of items owned by a node, r is the

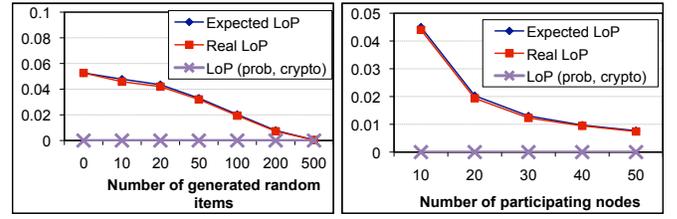


Fig. 3. Set exposure for union (single round)

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 calculate the required $|r|$ for given LoP and apply those values in estimating the cost.

VI. 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 I
EXPERIMENT PARAMETERS

A. Security of Random Shares Union

We have implemented the random shares based protocols. 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 I. In all the experiments the default values are used unless otherwise specified. We have assumed that nodes contribute on average $\frac{c}{n}$ items to the final result. We report the results for both set exposure and item exposure attacks. To measure the actual LoP , we ran the experiments multiple times, and for each run, a randomly selected node acts as an adversary and amounts the set exposure and item exposure attack as discussed earlier. We then measure the overall probability of the claim being true based on the data.

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 (5)) 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

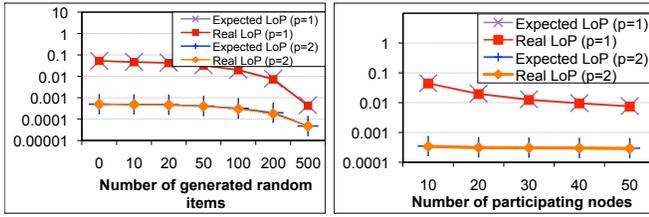


Fig. 4. Set exposure for union (multiple rounds)

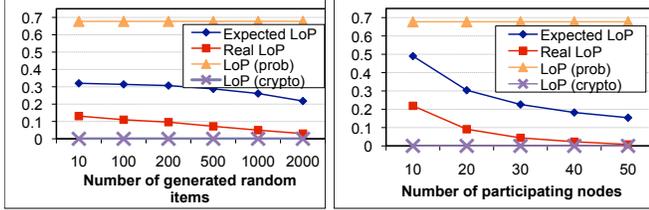


Fig. 5. Item exposure for union

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 ($LoP = 0$), 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. In our analysis, 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

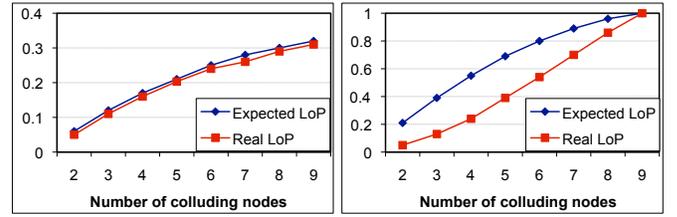


Fig. 6. Set exposure and item exposure for union with collusion

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 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.

B. 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.

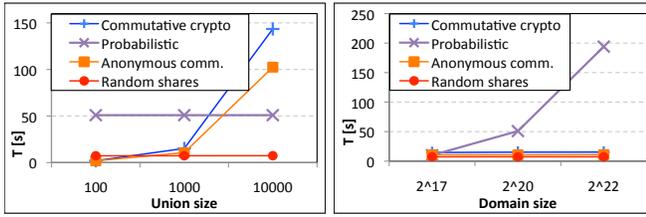


Fig. 7. Cost comparison of union protocols

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 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.

VII. 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 efficient 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 more extensive security analysis for various attack strategies and generalization of the random shares approach to other set operations and computation tasks.

ACKNOWLEDGEMENT

The research is supported by a Career Enhancement Fellowship from Woodrow Wilson Foundation and a Cisco Research Award. The authors would like to thank the anonymous reviewers for their valuable comments which helped improve the final version of the paper.

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 k th 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] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 2003.
- [12] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004.
- [13] X. He, H. Lu, J. Vaidya, and N. R. Adam. Secure construction and publication of contingency tables from distributed data. *Journal of Computer Security*, 19(3), 2011.
- [14] W. Jiang, C. Clifton, and M. Kantarcioglu. 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] M. Kantarcioglu, R. Nix, and J. Vaidya. An efficient approximate protocol for privacy-preserving association rule mining. In *PAKDD*, 2009.
- [17] H. Kargupta, K. Das, and K. Liu. Multi-party, privacy-preserving distributed data mining using a game theoretic framework. In *PKDD*, 2007.
- [18] L. Kissner and D. Song. Privacy-preserving set operations. In *CRYPTO*, 2005.
- [19] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 1951.
- [20] T. Léauté and B. Faltings. Privacy-preserving multi-agent constraint satisfaction. In *CSE (3)*, pages 17–25, 2009.
- [21] A. J. Lee, K. Minami, and N. Borisov. Confidentiality-preserving distributed proofs of conjunctive queries. In *ASIACCS*, 2009.
- [22] Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1), 2009.
- [23] B. Schneier. *Applied Cryptography*. John Wiley & Sons, 2nd edition, 1996.
- [24] Y. Tao, X. Xiao, J. Li, and D. Zhang. On anti-corruption privacy preserving publication. *ICDE*, 2008.
- [25] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *SIGKDD*, 2002.
- [26] J. Vaidya and C. Clifton. Privacy-preserving data mining: Why, how, and when. *IEEE Security & Privacy*, 2(6):19–27, 2004.
- [27] J. Vaidya and C. Clifton. Secure set intersection cardinality with application to association rule mining. *J. Comput. Secur.*, 13(4), 2005.
- [28] J. Vaidya, M. Kantarcioglu, and C. Clifton. Privacy-preserving naïve bayes classification. *VLDB J.*, 17(4):879–898, 2008.
- [29] L. Xiong, S. Chitti, and L. Liu. Preserving data privacy for outsourcing data aggregation services. *ACM Transactions on Internet Technology (TOIT)*, 7(3), 2007.
- [30] 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.
- [31] N. Zhang and W. Zhao. Distributed privacy preserving information sharing. In *VLDB*, 2005.