Secure Trust Evaluation Using Multipath and Referral Chain Methods

Mohammad G. Raeini and Mehrdad Nojoumian

Department of Computer & Electrical Engineering and Computer Science
Florida Atlantic University, Boca Raton, FL 33431, USA
{mghasemineja2017,mnojoumian}@fau.edu

Abstract. The notions of trust and reputation have been well studied and integrated into computer networks and internet-based services, e.g., Amazon and eBay websites. Using trust and reputation as social mechanisms can enhance the quality, reliability and trustworthiness of networks or services. These social mechanisms can also be used to provide better security measures. Indeed, trust and reputation can be considered as soft security methods that compliment hard security techniques. However, data security and privacy are among the primary challenges in trust and reputation systems. We therefore propose a secure trust evaluation (STE) method in which privacy of trust values and corresponding weights are preserved. Our proposed method is constructed based on an information theoretic framework for modeling trust and two approaches that propagate trust in a network, i.e., multipath and referral chain techniques. In other words, we utilize secure multiparty computation to provide protocols by which the nodes in a network will be able to evaluate their trust values in a secure fashion. We also provide a fascinating application of our STE method in the context of network routing protocols.

Keywords: Secure trust evaluation; Secure trust measurement; Secure multiparty computation; Secure function evaluation.

1 Introduction

Trust and reputation are common social mechanisms that have been used in different contexts including in human interactions, economics, multiagent systems and computer networks, among others. These social mechanisms are now well-studied and have been integrated into electronic applications and services, e.g., Amazon and eBay websites, search engines such as Google's PageRank algorithm, and social networks. These social mechanisms can also be utilized in any collaborative environments such as mining paradigms of digital currencies [19] as well as cryptographic protocols [21] to provide more trustworthy outcomes.

Trust and reputation are sometimes considered as soft security measures that compliment hard security measures such as cryptographic protocols. It is worth mentioning that, using soft security measures alongside hard security measures, can provide more secure and trustworthy systems and networks [28,35]. In other

words, integrating these social concepts into data and computation infrastructures can provide more reliable, secure and trustworthy platforms [11]. In fact, trusted computing is a term that refers to this idea and has been used in the IT security [11, 37]. However, there exist many challenges in modeling and utilizing these social concepts.

First, these concepts are highly subjective in the sense that different people have different impressions about them. It should also be mentioned that these concepts are very contextual-based and time-dependent [8]. Fortunately, there has been significant attempts for modeling and measuring trust and reputation. In the computer science literature, Marsh [15] is among the first researchers who tried to provide a computational model for trust. Thereafter, other models, methods and metrics have been defined for measuring trust and reputation quantitatively. In a nutshell, there are different theories/approaches for modeling and evaluating trust and reputation concepts. Some of the well-known approaches for measuring trust include subjective logic [9, 10], fuzzy logic [14], entropy-based models [32], and Demster-Shafer theory [33]. Reputation is usually evaluated based on the trust values. Some of the well-known reputation systems use simple summation, average and weighted average of trust values [11]. Other reputation systems utilize the Beta probability density function and Bayesian networks [11].

Second, there are studies [29] discussing that users are usually unwilling to provide honest feedback (ratings) in trust and reputation systems mainly due to fear of retaliation for negative ratings. As such, to have proper trust and reputation systems, it is important that such systems preserve the privacy of users' data while allowing them to perform the desired computations on their private values, e.g., trust values/rating scores of users in one another. There are different approaches for providing such systems. One approach is to use decentralized systems. Such reputation systems do not rely on any centralized authority, and thus, they are more reliable [5]. Another approach is to use cryptographic techniques, e.g., secure multiparty computation (MPC).

1.1 Our Contribution

This paper aims at addressing data security and privacy issues in trust and reputation systems. We use secure multiparty computation to provide a secure trust evaluation method. Our proposed method is based on the information theoretic framework [32] for modeling trust and two approaches that propagate trust in a network. These two approaches are referral chains in social networks and multipath trust propagation in a network. We provide two protocols that enable the nodes in a network to securely evaluate their trust values in one another. As an application, we use our proposed STE method to provide a secure network routing protocol. Our protocols can be based on any secret sharing scheme, e.g., the Shaimr's (t, n)-threshold secret sharing scheme [30]. We would like to emphasize that our protocols do not rely on any trusted third parties. In other words, the nodes in a network can perform the required computations for measuring their trust values securely. Using secure trust evaluation methods will result in more secure and trustworthy network-based systems and services.

The paper is organized as follows. In Section 2, we review existing works related to secure trust and reputation models. In Section 3, we provide the necessary preliminaries for our secure trust evaluation method. These include secure MPC based on secret sharing and an encoding approach that allows performing secure computations on real numbers. We use the floating-point representation of real numbers to perform secure computations on such numbers [1]. Note that, in our model, trust values are real numbers in $[-1, 1]$ interval. In Section 4, we provide our main contribution. We propose two secure protocols that are the building blocks of our STE method. We also provide a secure network routing protocol as an appealing application of the proposed STE method. Our technical discussion is presented in Section 5. The paper is concluded in Section 6.

2 Related Works

Data security and privacy are important issues in trust and reputation systems. Different approaches have been used to address such issues. Among others, we can point out approaches based on secure MPC techniques and those based on decentralized computation frameworks. In what follows, we review the previous works related to secure trust and reputation models. For comprehensive surveys related to trust and reputation systems, we refer the readers to [8, 11].

In [33], two schemes for preserving the privacy of trust evidence providers were proposed. The proposed schemes use two non-colluding service parties, called authorized proxy and evaluation party, to manage the aggregated evidences and process the collected data in encrypted format. The proposed schemes are based on public key cryptography, e.g., RSA and additive homomorphic encryption such as Paillier scheme [24]. Centralized trust and reputation systems can take advantage of their users' data. To address such an issue, the authors in [2] proposed a privacy-preserving distributed reputation mechanism based on the notion of *mailboxes*. Malicious-k-shares protocol, a decentralized privacy-preserving reputation system, was proposed in [7]. Again, the protocol is based on the Paillier cryptosystem and uses source managers (e.g., the Chord distributed hash table [31]) to share the data among k agents and perform privacy-preserving distributed computations.

The privacy-preserving version of the P2PRep [3], called 3PRep, was proposed in [17]. The 3PRep enhances the P2PRep mechanism by adding two new protocols to preserve votes' privacy using semantically secure homomprphic encryption scheme. Three different schemes for privacy-preserving computations of reputation values were presented in [5]. Two of the proposed schemes use a trusted third party to calculate the reputation. The third scheme does not rely on any trusted third party. Pavlov et. al. [25] argued that supporting perfect privacy in a decentralized reputation system is impossible. They then proposed three probabilistic schemes that are able to support partial privacy in decentralized additive reputation systems. The proposed schemes use secret splitting and secret sharing schemes, e.g., the Pederson secret sharing scheme [26].

There are other works related to privacy-preserving reputation systems. In [6], the authors provided the k-shares protocol, which was inspired by the protocol of [25]. The advantage of k-shares protocol is that it has a lower message complexity compared to the protocol proposed in [25], i.e., $O(n)$ versus $O(n^2)$. Finally, the authors of [4] introduced a dynamic privacy-preserving reputation system. This scheme is able to deal with the dynamic structure of some decentralized reputation systems wherein nodes (users) in the network leave and join the network constantly.

3 Preliminaries

3.1 Secure Multiparty Computation

Secure multiparty computation (MPC) is a computational model in which a group of parties can evaluate a public function on their private data without revealing their data. This idea was first introduced by Andrew Yao [34]. Secure MPC, a.k.a., secure function evaluation (SFE) [16], can be realized using cryptographic primitives such as secret sharing schemes, homomorphic encryption techniques and Yao's Garbled circuits. In secret sharing-based MPC, a secret sharing scheme, e.g., the Shamir's (t, n)-threshold secret sharing [30], is used to generate and distribute the shares of secrets (private data) among the participating parties. The computations are then carried out on the shares of those secrets. At the end of the computations, an appropriate technique, e.g., the Lagrange interpolation, is used to obtain the result of the computation.

Secure MPC based on Secret Sharing. Secure MPC based on the Shamir's secret sharing scheme works as follows. First of all, it should be noted that in secure multiparty computation there are n parties where each has a private value, which can be considered as a secret. Moreover, the computations are performed in a finite field such as Z_p, where p is a prime number. In order to perform a computation (evaluate a function) using secure MPC, each party first selects a polynomial $f(x) \in Z_p[x]$ whose coefficient are random values in Z_p and its constant term is the party's secret/private value. Mathematically speaking, each party P_i selects a polynomial as follows:

$$f_i(x) = \alpha_i + a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,t-1}x^{t-1}.$$

where α_i is the secret of party P_i, for $i = 1, 2, \ldots, n$ and $a_{i,1}$, $a_{i,2}$ $\ldots$, $a_{i,t-1}$ are random numbers in Z_p. Moreover, t is the threshold of the secret sharing scheme. Each party then evaluates its polynomial on n points, such as $1, 2, \ldots, n$, to generate the shares of its secret. The parties then distribute the shares of their secrets among each other. To evaluate a function securely, the parties perform the required computations on the shares of their data. They finally execute Lagrange interpolation on their updated shares to obtain/reconstruct the result of their computation, i.e., the function value. Secure MPC based on secret sharing is illustrated in Figure 1.

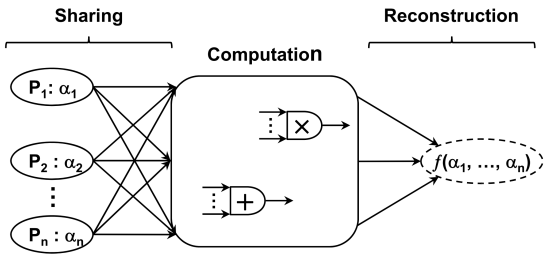

Fig. 1: Secure multiparty computation using secret sharing [18].

3.2 Floating-Point Representation of Real Numbers

Secure computation techniques primary work based on integer numbers, i.e., finite field elements. In our secure trust evaluation method, the trust values are rational numbers in $[-1, +1]$ interval. Therefore, we need to use secure MPC techniques on real numbers. There are different encoding approaches, e.g., floating-point representation that allows secure computation techniques to be used on real numbers. In this paper, we utilize the floating-point representation of real numbers, presented in [1], although other approaches can be used.

Floating-point representation is a method to represent real numbers using a fixed-precision significand v and an exponent p. The exponent p defines how the real number should be scaled in a given base. For instance, when the base is 2, the representation would be $v \,.\, 2^p$. In order to have a proper representation, the authors in [1] used a 4-tuple (v, p, z, s) with base 2 to represent each real value u. In this representation, v is an l-bit significand and p is a k-bit exponent. Moreover, z is a binary value which is 1 if and only if $u = 0$, and s is the sign bit. The sign bit s is set when the value u is negative. For a real value u, the representation will be $u = (1 - 2s)(1 - z)v \,.\, 2^p$.

4 Secure Trust Evaluation (STE)

4.1 Information Theoretic Framework for Modeling Trust

The concept of trust (in human interactions or social networks) is very related to the concept of uncertainty in information theory. This subtle connection was formalized in [32], wherein an information theoretic framework for modeling trust was introduced. Due to the similarity between trust and uncertainty, trust can be measured by entropy, which is a well-accepted concept in information theory. Having said that, two trust models were proposed in [32], an entropy-based trust model and a probability-based trust model. For the probability-based trust model, two approaches were studied, a Binomial distribution and a Bayesian approach. The authors then discussed that the Bayesian approach captures the concept of uncertainty more appropriately.

The information theoretic framework for modeling trust works based on the observations of nodes. In what follows, we briefly explain how trust is evaluated

in this framework. Assume a network is given and node A in the network wants to evaluate its trust (for performing an action, e.g., packet forwarding) in another node, say node X. To do so, the past behaviors of node X regarding that specific action is considered. In the trust model based on the Bayesian approach, first the probability of node X performing that action is calculated. If node X has been asked to perform an action N times, and among them, node X has performed that action k times, the probability of performing that action in the next request, i.e., the $(N+1)$-th request, is defined as follows [32]:

$$Pr(V(N+1)) = \frac{k+1}{N+2} \tag{1}$$

wherein k is the number of times that node X has performed a specific action upon N total requests. In fact, $Pr(V(N+1))$ is the probability that node X will perform that specific action in the $(N+1)$-th request. Note that $V(i)$ is the random variable of performing an action at the i-th request [32]. In the information theoretic framework for modeling trust, trust can also be calculated as entropy, which in fact measures the uncertainty. Having a probabilistic trust value, the entropy-based trust value of node A in node X for performing an action is defined as follows [32]:

$$T(A:X, action) = \begin{cases} 1 - H(p), & \text{for } 0.5 \leq p \leq 1 \\ H(p) - 1, & \text{for } 0 \leq p < 0.5 \end{cases} \tag{2}$$

where $H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$ and p is the probability as defined in Equation 1. The information theoretic framework [32] is an elegant way of modeling the concept of trust. There are a few points that should be emphasized. The trust values in the information theoretic framework can be represented as probability-based values or entropy-based values. Equation 2 shows the relation between these two types of trust values and how they can be converted to each other. It is also important to note that probability-based trust values are in $[0, 1]$ interval, whereas entropy-based trust values vary within $[-1, 1]$ interval. In our STE method, the trust values are in $[-1, 1]$ interval.

4.2 Secure Trust Evaluation Using Multipath Trust Propagation

Trust in a network can propagate in different ways. In this section, we briefly discuss how a node can evaluate its trust in another node using the multipath trust propagation approach. In the multipath trust propagation, a node (say node A_1) wants to evaluate its trust in another node (say node B). To this end, node A_1 asks other nodes, say nodes A_2, A_3, ..., A_n, in the network to reveal their opinions about node B. Figure 2 shows a sample multipath trust propagation in a network.

After receiving the trust values (from other nodes, i.e., A_2, A_3, ..., A_n), node A_1 calculates its trust in node B as follows:

$$T_{A_1 B} = Trust(A_1 : B) = \sum_{i=1}^{n} w_i T_i \tag{3}$$

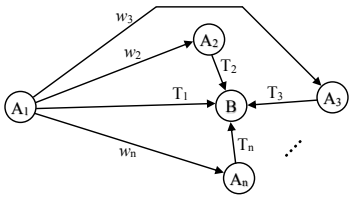

Fig. 2: Multipath trust propagation [32].

where T_1 is the trust value of node A_1 in node B (i.e., based on direct observation) and w_1 is the weight that node A_1 considers for its direct trust value in node B. Moreover, T_i, for $i = 2, \ldots, n$, is the trust value (opinion) of node A_i in node B, which is returned from node A_i to node A_1. Also, w_i is the weight that node A_1 considers for its trust in node A_i for $i = 2, \ldots, n$; see Figure 2. Note that w_i-s are selected by node A_1 such that $0 \leq w_i \leq 1$ and $\sum_{i=1}^{n} w_i = 1$. Since $0 \leq w_i \leq 1$ and $-1 \leq T_i \leq 1$, we will have $-1 \leq T_{A_1B} \leq 1$. The maximum value that T_{A_1B} can get is when $T_i = 1$. In that case, $T_{A_1B} = \sum_{i=1}^{n} w_i T_i \leq \sum_{i=1}^{n} w_i = 1$. The minimum value that T_{A_1B} can get is when $T_i = -1$, where we have $T_{A_1B} = \sum_{i=1}^{n} w_i T_i \geq \sum_{i=1}^{n} w_i(-1) = -1 \times \sum_{i=1}^{n} w_i = -1$. Note that it is assumed $\sum_{i=1}^{n} w_i = 1$.

We now propose a protocol that allows a node in a network, e.g., node A_1, to evaluate its trust in another node, e.g., node B, using the multipath trust propagation approach. In fact, the nodes on the multipath network perform their computations using secure multiparty computation. To this end, the nodes use the Shamir's secret sharing scheme to share their secrets (in this case, their trust values in each other) and perform computations in a secure fashion. Note that the nodes on the multipath, illustrated in Figure 2, need to securely evaluate the function represented in Equation 3. In this equation, w_i's are private values of A_1 while T_i is the private value of node A_i, for $i = 2, \ldots, n$. Protocol 1 shows our secure trust evaluation method using the multipath trust propagation approach.

4.3 Trust Evaluation Using Referral Chains

The idea of using referral chains (referral graphs) in trust and reputation systems was introduced in [35] and further studied in [20, 36]. Yu and Singh [35] defined a referral chain as follows. Given the graph representation of a network (e.g., a social network), a referral chain from node A_0 to node A_n is basically a path between the two nodes. Such a referral chain is represented as $\chi = \langle A_0, A_1, \ldots, A_n \rangle$, where A_i is a neighbor of A_{i+1}.

The concept of referral chain in a network can capture the notion of trust propagation in a good way. In [35], the authors used this concept for estimating the quality of nodes in a trust network, in which the trust value of a node (say node A) in another node (say node B) is measured based on three factors [20, 35]: A's direct observation of B, the B's neighbors opinion about B, and the A's opinion about the neighbors of B. Having the trust values of the nodes on a referral chain, the trust over the referral chain propagates according to the

Protocol 1: Secure Trust Evaluation Using the Multipath Approach

Input: Trust values $\{T_1, T_2, \ldots, T_n\}$ and weights $\{w_1, w_2, \ldots, w_n\}$.
Output: Calculates $T_{A_1B} = \sum_{i=1}^{n} w_i T_i$ using secure MPC.

1 Each party (node) A_i, for $i = 1, 2, \ldots, n$, uses floating-point representation to encode its input into a single finite field element.

2 Each party A_i uses the Shamir's secret sharing scheme to generate the shares of its input T_i. Party A_i selects a polynomial as follows:

$$f_i(x) = T_i + a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,t-1}x^{t-1}.$$

where T_i is the trust value of node A_i in node B.

3 Party A_1 uses the Shamir's secret sharing scheme to generate the shares of its weights, i.e., w_i's. A_1 selects a polynomial as follows:

$$g_i(x) = w_i + b_{i,1}x + b_{i,2}x^2 + \cdots + b_{i,t-1}x^{t-1}.$$

where w_i is the weight that party A_1 considers for node A_i.

4 Each party distributes the shares of its input among all parties. The share-exchange matrix [23] (wherein party A_i generates the i-th row and receives the i-th column) is as follows:

$$E_f = \begin{bmatrix} f_1(1) & f_1(2) & \ldots & f_1(n) \\ \vdots & \vdots & \ddots & \vdots \\ f_n(1) & f_n(2) & \ldots & f_n(n) \end{bmatrix} \begin{matrix} \leftarrow \text{Shares of } T_1 \text{ generated by } A_1 \text{ using } f_1(x) \\ \vdots \\ \leftarrow \text{Shares of } T_n \text{ generated by } A_n \text{ using } f_n(x) \end{matrix}$$

5 Party A_1 distributes the shares of its weights w_i's, for $i = 1, \ldots, n$. The share-exchange matrix is as follows:

$$E_g = \begin{bmatrix} g_1(1) & g_1(2) & \ldots & g_1(n) \\ \vdots & \vdots & \ddots & \vdots \\ g_n(1) & g_n(2) & \ldots & g_n(n) \end{bmatrix} \begin{matrix} \leftarrow \text{Shares of } w_1 \text{ generated by } A_1 \text{ using } g_1(x) \\ \vdots \\ \leftarrow \text{Shares of } w_n \text{ generated by } A_1 \text{ using } g_n(x) \end{matrix}$$

6 Party A_i, for $i = 1, 2, \ldots, n$, performs the following computation:

$$T^i_{A_1B} = \sum_{k=1}^{n} g_k(i) \times f_k(i).$$

where $g_k(i)$ is the share of a weight that party A_i has received from party A_1 and $f_k(i)$ is the share of T_k that party A_i has received from party A_k. Moreover, $T^i_{A_1B}$ means the share of party A_i of trust value T_{A_1B}. Note that after each multiplication, $g_k(i) \times f_k(i)$, the participating parties must execute a degree reduction protocol, as explained in [22].

7 Each party A_i, for $i = 2, 3, \ldots, n$, sends the result of the computation, in the previous step, to party A_1.

8 A_1 uses Lagrange interpolation to obtain the final result, i.e., T_{A_1B}, as follows:

$$T_{A_1B} = \sum_{i=1}^{n} \Big(\prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{k}{k-i} \times T^i_{A_1B} \Big)$$

trust propagation operator; see Definitions 5 and 6 of [35]. In our secure trust evaluation method, we consider a general case of a referral chain consisting of n nodes as illustrated in Figure 3:

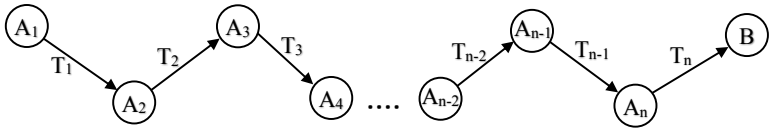

Fig. 3: A referral chain in a network [35].

The trust value of node A_1 in the last node on the referral chain, i.e., node B, is calculated as follows [35]:

$$T_{A_1B} = T_{A_1A_2} \otimes T_{A_2A_3} \otimes \cdots \otimes T_{A_nB} \tag{4}$$

where $T_{A_iA_{i+1}}$, for $i = 1, 2, \ldots, n-1$, is the trust value of node A_i in node A_{i+1} and it is represented as T_i in Figure 3. Moreover, $\otimes$ represents the trust propagation operator, which is defined as follows:

Definition 1. $x \otimes y :=$ if $(x \geq 0 \wedge y \geq 0)$ then $x \times y$; else $-|x \times y|$ [35].

The trust propagation on a referral chain is then defined as follows:

Definition 2. For any k, where $k \in \{1, 2, \ldots, n\}$, the trust of A_1 in A_k is defined as: $T_{A_1A_k} = T_{A_1A_2} \otimes \cdots \otimes T_{A_{k-1}A_k}$ [35].

In the following, we propose a protocol that enables a node in a network to evaluate its trust in another node through a referral chain. The main idea is that the nodes on the referral chain use secure MPC based on secret sharing to carry out the trust evaluation computations, i.e., to evaluate Equation 4 in a secure fashion. The procedure of secure trust evaluation on a referral chain is described in Protocol 2. Note that, in Protocol 2, the trust value of node A_i in node A_{i+1} is represented as T_i, where $i = 1, \ldots, n$. That is, $T_i = T_{A_iA_{i+1}}$.

To execute the trust propagation operator, i.e., $\otimes$ in Definition 1 and Equation 4, two trust values are compared with zero (i.e., if $x \geq 0 \wedge y \geq 0$) before the multiplication of each pair of trust values. Thus, in order to carry out the trust propagation operator in Protocol 2, each pair of trust values need to be securely compared with zero. This can be done in different ways. One solution is to use a secure comparison protocol, e.g., a protocol from Table IV of [27]. Another approach is to use secure MPC for determining the sign of the final trust value, i.e., T_{A_1B}, as follows. Each party (node) A_i encodes and shares the sign of its trust value T_i: If A_i's trust value is positive (i.e., $0 \leq T_i < 1$), then A_i shares 0 among all parties. If A_i's trust value is negative (i.e., $-1 \leq T_i < 0$), then A_i shares 1 among all parties. Parties then exchange and add their shares and send the results to party A_1. By obtaining the final result (using the Lagrange interpolation), party A_1 can determine the sign of the final trust value as follows: If the final result is 0, the sign of the final trust value (i.e., T_{A_1B}) is positive. Otherwise, it is negative.

Protocol 2: Secure Trust Evaluation Using the Referral Chain Approach

Input: Trust values $\{T_1, T_2, \ldots, T_n\}$, where $T_i = T_{A_i A_{i+1}}$,
Output: Calculates $T_{A_1 B} = T_1 \otimes \cdots \otimes T_n$ using secure MPC.

1 Each party A_i, i.e., each node on the referral chain, uses floating-point representation to encode its input T_i into a single finite field element.

2 Each party A_i uses the Shamir's secret sharing scheme to generate the shares of its trust value T_i. Party A_i selects a polynomial as follows:

$$f_i(x) = T_i + a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,t-1}x^{t-1}.$$

where T_i is the trust value of node A_i in node A_{i+1} on the chain.

3 Each party A_i distributes the shares of its trust value among all parties. The share-exchange matrix [23] (wherein party A_i generates the i-th row and receives the i-th column) is as follows:

$$E_f = \begin{bmatrix} f_1(1) & f_1(2) & \ldots & f_1(n) \\ \vdots & \vdots & \ddots & \vdots \\ f_n(1) & f_n(2) & \ldots & f_n(n) \end{bmatrix} \begin{matrix} \leftarrow \text{Shares of } T_1 \text{ generated by } A_1 \text{ using } f_1(x) \\ \vdots \\ \leftarrow \text{Shares of } T_n \text{ generated by } A_n \text{ using } f_n(x) \end{matrix}$$

4 Each party A_i multiplies its received shares:

$$T^i_{A_1 B} = \prod_{k=1,\ldots,n} f_k(i)$$

where $f_k(i)$ is the share that party A_i has received from party A_k where $k = 1, 2, \ldots, n$. Moreover, $T^i_{A_1 B}$ means the share of party A_i of trust value $T_{A_1 B}$. Note that, after each multiplication, the participating parties must execute a degree reduction protocol as shown in [22].

5 Each party A_i for $i = 2, 3, \ldots, n$ sends its result of the multiplication, in the previous step, to party A_1.

6 Party A_1 uses Lagrange interpolation to obtain the final result, i.e., $T_{A_1 B}$:

$$T_{A_1 B} = \sum_{i=1}^{n} (\prod_{\substack{k=1 \\ k \neq i}}^{n} \frac{k}{k-i} \times T^i_{A_1 B})$$

It is worth mentioning that a disadvantage of the referral chain approach is that, on long chains, the trust propagation operator fades the trust value of node A_1 in node B [20]. An alternative solution for the referral chain approach is to use a weighted average of the trust values on the chain, where the weights decrease monotonically, i.e., $1 \geq w_1 > w_2 > \cdots > w_n \geq 0$. Note that w_i's are selected such that $\sum_{i=1}^{n} w_i = 1$. In such a monotonically-decreasing weighted referral chain, the trust value $T_{A_1 B}$ can be securely evaluated using Protocol 1.

4.4 Secure Network Routing

The concept of trust, as a soft security measure, can be used for improving the quality of network services in different ways. For instance, trust models can improve network routing protocols and provide malicious-node-detection capability [32]. An important thing in most networks is the security and privacy of the nodes' data. It is important for the nodes in a network to not reveal their private data, e.g., their trust values [29, 32]. This is because, if trust values are revealed, nodes with high trust values may be compromised by adversaries. This can reduce the trustworthiness of the whole network.

In this section, we use our proposed protocols to provide a secure network routing protocol. By using the secure network routing protocol, a node in a network can find a high quality route in a network while the nodes' private data is not revealed. Secure network routing protocols can provide a better networking platform in the sense that adversaries will not be able to figure out how an action, e.g., packet forwarding in a network, is carried out. We first need to define the quality of a route in a network.

The Quality of a Route in a Network. Assume a network is given and node A and node N_{dest} are two nodes in that network. Moreover, suppose node A intends to perform an action in the network, e.g., to forward a packet to node N_{dest}. There are usually different routes in the network for performing such an action. In order to execute the packet forwarding action with a higher chance of success, node A can determine the quality of each route prior to forwarding its packet to the destination. One approach for defining the quality of a route in a network is based on the trust values of nodes on that route [32]. Suppose R is a route in a network and $\{N_i\}$ represents the set of all nodes on route R. Similar to [32], we define and calculate the quality of route R as follows:

$$Quality(R) = \begin{cases} \prod_i T_i & \text{if } T_i > 0 \ \forall \text{ nodes } N_i \text{ on route } R \\ \min\{T_i\} & \text{otherwise} \end{cases} \tag{5}$$

where T_i is the trust value of node A in node N_i on route R. Equation 5 is basically multiplications of trust values on route R. In cases that there are nodes with negative trust values on the route, we define the quality of route as the minimum trust value, i.e., the smallest negative trust value.

We now propose a protocol that enables a node in a network to evaluate the quality of a route in a secure manner. Our proposed secure network routing protocol works as follows. Assume node A intends to evaluate the quality of route R. Node A evaluates the trust value of each node on the route using the secure trust evaluation protocols (Protocol 1 and Protocol 2). Then, node A calculates the quality of the route using Equation 5. To find a high quality route, node A must calculate the quality of different possible routes (to its desired destination) and find the route with the highest quality. The secure network routing protocol is provided in Protocol 3. Note that we assumed each node, including node A, has a trust record on which the trust values are stored [32].

Protocol 3: Secure Network Routing Protocol

Input: Nodes' trust records, i.e., nodes observations or opinions.
Output: A high quality route in the network from node A to node N_{dest}.

1 Let $\{S_i\}$ denote the set of all nodes on all possible routes between node A and node N_{dest} in the network.
2 **for** *any node* S_i **do**
3 **if** *node A has a trust record about node* S_i **then** Node A uses that trust record.
4 **else** Node A sends trust recommendation request about node S_i to other nodes. Node A collaboratively with other nodes use Protocol 1 and Protocol 2 to securely evaluate its trust value in node S_i.
5 Let R denote a particular route in the network and let $\{N_i\}$ denote the set of all nodes on route R. Let T_i denote the trust value of node A in node N_i. Node A calculates the quality of route R as follows:

$$Quality(R) = \begin{cases} \prod_i T_i & \text{if } T_i > 0 \ \forall \text{ nodes } N_i \text{ on } R \\ \min\{T_i\} & \text{otherwise} \end{cases}$$

Note that the above multiplication is performed locally by node A. However, each T_i is computed securely when node A does not have a trust record about node N_i; see step 4.
6 Let $\{R_i\}$ denote the set of routes from node A to node N_{dest} in the network among which A wants to find a good quality route. Node A selects a route which has a good quality, e.g., larger than a threshold or the route with the maximum quality, as follows:

$$R^* = argmax_{R_i}\{Quality(R_i)\}$$

7 Node A updates its trust records using the recent observations and calculated trust values.
8 Node A initiates its desired action on the high quality route, i.e., route R^*.

5 Technical Discussion

In this paper, we introduced a secure trust evaluation (STE) method. Our proposed approach is based on the information theoretic framework for modeling trust and two approaches that propagate trust in a given network, i.e., multipath trust propagation and referral chains. The Beta reputation system [12] is a specific case of the information theoretic framework for modeling trust. Note that the trust value in the information theoretic framework is measured using Equation 1 in section 4. In the Beta reputation system, the reputation of a user is calculated as $\frac{r+1}{r+s+2}$ (see [12] and [13]). The Beta reputation system is one of the commonly referred reputation systems in the literature. Thus, our secure trust evaluation method can be used wherever the Beta reputation system is applicable. For instance, our proposed protocols can be used in computer networks and Internet-based services that use the Beta reputation system.

It is worth mentioning that our proposed STE method is a decentralized trust evaluation system. This has its own advantages and makes a network more reliable and trustworthy because the nodes in a network do not reveal their private values to any third party or any other nodes. Recall that the secure protocols (Protocol 1 and Protocol 2) in our trust evaluation method use secure MPC and secret sharing schemes, e.g., the Shamir's (t, n)-threshold secret sharing scheme, which are powerful tools for secure function evaluation.

Another fact in many privacy-preserving trust and reputation systems is that, regardless of using the cryptographic primitives or any other privacy measures, a ratee in a reputation system can figure out the impact of a rater's feedback (rate) on its reputation [13]. This is because a feedback is usually provided after a transaction is completed. Therefore, the ratee knows when the rater has left his feedback. The ratee can then see the impact of that feedback on its reputation. Although the ratee might not be able to figure out the exact feedback rate, he will be able to figure out if the feedback is positive or negative.

Our proposed secure trust evaluation method addresses the aforementioned issue appropriately. In our model, when a node (say node A) in a network intends to evaluate its trust in another node (say node B), node A asks other nodes for their ratings about node B. The process of evaluating the trust value of node A in node B is carried out in such a way that node B may not even notice its reputation has been evaluated by other nodes. This makes sense because, in a decentralized trust and reputation system, the nodes are witnesses for each others' behavior. Recall that in our trust model, the trust value of a node is evaluated as a weighted average of other nodes' ratings; see Equations 3.

Finally, the security analysis of our protocols is inherited from the security of the underlying secret sharing scheme, which is the Shamir's scheme. In our proposed protocols, the parties use this scheme to generate shares of their secrets, i.e., trust values. They then perform their computations on the shares of trust values rather than trust values themselves. Note that our protocols work in a semi-honest (passive) adversarial model. In other words, we assumed that the nodes in the network are honest-but-curious. In a passive adversarial model, the participating parties act honestly and follow the protocols' rules but they are curious to learn other parties' private data. It is worth mentioning that our protocols can deal with active adversaries if we utilize verifiable secret sharing.

6 Concluding Remarks

In this paper, we introduced a secure trust evaluation (STE) method. Our STE method consists of two protocols that allow the nodes in a network to securely evaluate their trust values in one another. The proposed protocols in our STE method use secure multiparty computation based on the Shamir's secret sharing scheme to guarantee the security and privacy of the parties' private data. As an application, we also proposed a secure network routing protocol that shows how our proposed STE method can be used for improving network routing protocols.

Furthermore, our proposed STE method can be used in different networks for providing more reliable and trustworthy services. Our STE method relies on the information theoretic framework for modeling trust, which is a powerful trust model. Besides, our STE method can be utilized in other trust and reputation systems, e.g., the Beta reputation system and the weighted average reputation model. As stated earlier, soft security measures such as trust and reputation mechanisms can compliment hard security measures to provide more reliable and trustworthy networks. Therefore, consideration should be given to further improve trust and reputation systems.

7 Acknowledgment

Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-18-1-0483. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Aliasgari, M., Blanton, M., Zhang, Y., Steele, A.: Secure computation on floating point numbers. In: NDSS (2013)
2. Anceaume, E., Guette, G., Lajoie-Mazenc, P., Prigent, N., Tong, V.V.T.: A privacy preserving distributed reputation mechanism. In: Communications (ICC), 2013 IEEE International Conference on. pp. 1951–1956. IEEE (2013)
3. Aringhieri, R., Damiani, E., Di Vimercati, S.D.C., Paraboschi, S., Samarati, P.: Fuzzy techniques for trust and reputation management in anonymous peer-to-peer systems. Journal of the American Society for Information Science and Technology **57**(4), 528–537 (2006)
4. Clark, M.R., Stewart, K., Hopkinson, K.M.: Dynamic, privacy-preserving decentralized reputation systems. IEEE Transactions on Mobile Computing **16**(9), 2506–2517 (2017)
5. Gudes, E., Gal-Oz, N., Grubshtein, A.: Methods for computing trust and reputation while preserving privacy. In: IFIP Annual Conference on Data and Applications Security and Privacy. pp. 291–298. Springer (2009)
6. Hasan, O., Brunie, L., Bertino, E.: Preserving privacy of feedback providers in decentralized reputation systems. Computers & Security **31**(7), 816–826 (2012)
7. Hasan, O., Brunie, L., Bertino, E., Shang, N.: A decentralized privacy preserving reputation protocol for the malicious adversarial model. IEEE Transactions on Information Forensics and Security **8**(6), 949–962 (2013)
8. Hendrikx, F., Bubendorfer, K., Chard, R.: Reputation systems: A survey and taxonomy. Journal of Parallel and Distributed Computing **75**, 184–197 (2015)
9. Josang, A.: An algebra for assessing trust in certification chains. In: Proceedings of the Network and Distributed Systems Security Symposium (NDSS99). The Internet Society (1999)

10. Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9**(03), 279–311 (2001)
11. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision support systems **43**(2), 618–644 (2007)
12. Jsang, A., Ismail, R.: The beta reputation system. Proceedings of the 15th bled electronic commerce conference. Vol. 5 pp. 2502–2511 (2002)
13. Kerschbaum, F.: A verifiable, centralized, coercion-free reputation system. In: Proceedings of the 8th ACM workshop on Privacy in the electronic society. pp. 61–70. ACM (2009)
14. Manchala, D.W.: Trust metrics, models and protocols for electronic commerce transactions. In: Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No. 98CB36183). pp. 312–321. IEEE (1998)
15. Marsh, S.P.: Formalising trust as a computational concept. Ph.D. thesis, University of Stirling (1994)
16. Micali, S., Rogaway, P.: Secure computation. In: Annual International Cryptology Conference. pp. 392–404. Springer (1991)
17. Nithyanand, R., Raman, K.: Fuzzy privacy preserving peer-to-peer reputation management. IACR Cryptology ePrint Archive **2009**, 442 (2009)
18. Nojoumian, M.: Novel Secret Sharing and Commitment Schemes for Cryptographic Applications. Ph.D. thesis, Department of Computer Science, University of Waterloo, Canada (2012)
19. Nojoumian, M., Golchubian, A., Njilla, L., Kwiat, K., Kamhoua, C.: Incentivizing blockchain miners to avoid dishonest mining strategies by a reputation-based paradigm. In: Computing Conference. pp. 1118–1134. AISC 857, Springer (2018)
20. Nojoumian, M., Lethbridge, T.C.: A new approach for the trust calculation in social networks. In: E-business and Telecommunication Networks: 3rd International Conference on E-Business, Best Papers. CCIS, vol. 9, pp. 64–77. Springer (2008)
21. Nojoumian, M., Stinson, D.R.: Socio-rational secret sharing as a new direction in rational cryptography. In: 3rd International Conference on Decision and Game Theory for Security (GameSec). LNCS, vol. 7638, pp. 18–37. Springer (2012)
22. Nojoumian, M., Stinson, D.R.: On dealer-free dynamic threshold schemes. Advances in Mathematics of Communications (AMC) **7**(1), 39–56 (2013)
23. Nojoumian, M., Stinson, D.R., Grainger, M.: Unconditionally secure social secret sharing scheme. IET Information Security (IFS), Special Issue on Multi-Agent and Distributed Information Security **4**(4), 202–211 (2010)
24. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 223–238. Springer (1999)
25. Pavlov, E., Rosenschein, J.S., Topol, Z.: Supporting privacy in decentralized additive reputation systems. In: International Conference on Trust Management. pp. 108–119. Springer (2004)
26. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Annual International Cryptology Conference. pp. 129–140. Springer (1991)
27. Raeini, M.G., Nojoumian, M.: Comprehensive survey on secure comparison protocols. Tecnical Report (2019)
28. Rasmusson, L., Jansson, S.: Simulated social control for secure internet commerce (position paper). In: Proceedings, New Security Paradigms Workshop, Lake Arrowhead (1996)

29. Resnick, P., Zeckhauser, R.: Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. In: The Economics of the Internet and E-commerce, pp. 127–157. Emerald Group Publishing Limited (2002)
30. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
31. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. ACM SIGCOMM Computer Communication Review **31**(4), 149–160 (2001)
32. Sun, Y.L., Yu, W., Han, Z., Liu, K.R.: Information theoretic framework of trust modeling and evaluation for ad hoc networks. IEEE Journal on Selected Areas in Communications **24**(2), 305–317 (2006)
33. Yan, Z., Ding, W., Niemi, V., Vasilakos, A.V.: Two schemes of privacy-preserving trust evaluation. Future Generation Computer Systems **62**, 175–189 (2016)
34. Yao, A.C.: Protocols for secure computations. In: Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. pp. 160–164. IEEE (1982)
35. Yu, B., Singh, M.P.: A social mechanism of reputation management in electronic communities. In: International Workshop on Cooperative Information Agents. pp. 154–165. Springer (2000)
36. Yu, B., Singh, M.P.: An evidential model of distributed reputation management. In: Proceedings of the first international joint conference on Autonomous Agents and Multiagent Systems: Part 1. pp. 294–301. ACM (2002)
37. Zyskind, G., Nathan, O., et al.: Decentralizing privacy: Using blockchain to protect personal data. In: Security and Privacy Workshops (SPW), 2015 IEEE. pp. 180–184. IEEE (2015)