Trust Establishment in Distributed Networks: Analysis and Modeling

Yan Lindsay Sun and Yafei Yang
Department of Electrical and Computer Engineering
University of Rhode Island, Kingston, RI 02881
Email: yansun@ele.uri.edu

Abstract—Recently, trust establishment is recognized as an important approach to defend distributed networks, such as mobile ad hoc networks and sensor networks, against malicious attacks. Trust establishment mechanisms can stimulate collaboration among distributed computing and communication entities, facilitate the detection of untrustworthy entities, and assist decision-making in various protocols. In the current literature, the methods proposed for trust establishment are always evaluated through simulation, but theoretical analysis is extremely rare. In this paper, we present a suite of approaches to analyze trust establishment process. These analysis approaches are used to provide in-depth understanding of trust establishment process and quantitative comparison among trust establishment methods. The proposed analysis methods are validated through simulations.

I. INTRODUCTION

Recently, building trust among distributed network entities has been recognized as a new security approach to simulate cooperation and improve security in distributed networks [1], [2]. Briefly speaking, with trust establishment mechanisms, network entities will know whether and how much they can trust other network entities. This trust information guides network entities not to take highly risky actions such as asking the nodes with low trust value to forward packets or perform data aggregation. As a consequence, the performance and robustness of the network will be improved. Furthermore, trust establishment mechanisms provide an incentive for cooperation. The network entities will behave more responsibly with the expectation that their trust value or reputation will be hurt by non-cooperative or destructive behaviors.

The research on the subject of trust in computer networks has been extensively performed for a wide range of application scenarios, including authorization and access control [3]–[11], electronics commerce [1], [12], [13], peer-to-peer networks [14], [15], ad hoc and sensor networks [16]–[22], and pervasive computing [2], [23], [24].

Currently, the methods for trust establishment in computer networks have always been evaluated through simulations for specific applications [2]. Theoretical analysis is extremely rare in this field. Without performing extensive simulations, researchers can hardly compare different trust establishment methods, which are often proposed for different application scenarios. Even with simulations, simulation results cannot be easily decoupled from specific simulation implementations. In this paper, we present a set of methods that can analyze the trust establishment process in different application scenarios. The proposed methods lead to insightful understanding and comparison among existing trust establishment approaches. The analysis methods are developed in five steps.

• Step 1: Understanding basic components in trust establishment/reputation systems;
• Step 2: Developing the architecture for performing theoretical analysis, including specifying inputs, outputs and basic modules;
• Step 3: Design individual modules;
• Step 4: Testing the analysis methods through simulations;
• Step 5: Utilizing the analysis methods to quantitatively compare different trust models, and provide in-depth understanding of trust establishment process.

The rest of the paper is organized as follows. Section II describes related work. Section III presents step 1, and Section IV presents step 2 and 3. Section V describes step 4 and step 5. The conclusion is drawn in Section VI.

II. RELATED WORK

To our best knowledge, the analytical study on trust establishment was only presented in [25] previously. In this work, a simple trust evaluation method for autonomous networks is analyzed. Close form solutions are obtained and interesting results are observed. However, the theories in this work cannot be directly applied to more complicated trust evaluation methods. In fact, close form solutions are only possible for very simple systems. Thus, the method in [25] is not sophisticated enough to provide theoretical comparison among different trust establishment methods.

III. CORE DESIGN OF TRUST ESTABLISHMENT METHODS

Trust can be established in a centralized or distributed manner. The later is more suitable for distributed networks, such as MANET and sensor networks. This work focuses on distributed trust establishment where network participants have similar responsibilities. The basic elements in distributed trust-establishment systems are shown in Figure 1 and described as follows.

Trust Record stores information about trust relationship and associated trust values. A trust relationship is always established between two parties for a specific action. That is, one party trusts the other party to perform an action. In this work, the first party is referred to as the subject and the second party as the agent. A notation \( \{ \text{subject : agent, action} \} \) is introduced to represent a trust relationship. For each trust
relationship, one or multiple numerical value, referred to as trust values, describe the level of trustworthiness.

There are two common ways to establish trust in computer networks. First, when the subject can directly observe the agent’s behavior, direct trust can be established. Second, when the subject receives recommendations from other entities about the agent, indirect trust can be established.

**Direct Trust** is established upon observations on whether the previous interactions between the subject and the agent are successful. The observation is often described by two variables: \( s \) denoting the number of successful interaction and \( f \) denoting the number of failed interactions. That is, the direct trust value can be calculated as \( f_{DT}(s,f) \). The notation \( T_{AB}^{d} \) denotes the direct trust values between node \( A \) and \( B \). Thus, \( T_{AB}^{d} = f_{DT}(s,f) \), and \( T_{AB}^{d} \) can be a vector.

Recommendation trust is a very special direct trust. It is for trust relationship \{subject; agent\}, making correct recommendations. Assume that the subject can judge whether a recommendation is correct or not, and this subject receives \( s_{r} \) good recommendations and \( f_{r} \) bad recommendation from the agent. Then, the recommendation trust value can be calculated as \( f_{DT}(s_{r},f_{r}) \). Recommendation trust is important for establishing indirect trust. The recommendation trust between node \( A \) and \( B \) is denoted by \( T_{AB}^{r} \).

**Indirect Trust:** Trust can transit through third parties. For example, if \( A \) and \( B \) have established a recommendation trust relationship and \( B \) and \( C \) have established a direct trust relationship, then \( A \) can trust \( C \) to a certain degree if \( B \) tells \( A \) its trust opinion (i.e. recommendation) about \( C \). This phenomenon is called trust propagation. Indirect trust is established through trust propagations.

Two key factors determine the indirect trust establishment. The first is to determine when and from whom the subject can collect recommendations. For example, in an ad hoc network, a node may collect recommendations only from its one-hop neighbors, or from all nodes in the network. This affects the number of recommendations can be collected and the overhead of collecting recommendations. This first factor is referred to as the recommendation mechanism.

The second is to determine how to calculate indirect trust value based on recommendations. This calculation is governed by trust models. A trust model usually contains two parts: concatenation model and multipath model, as illustrated in Figure 1.

![Fig. 1. Basic Elements in Trust Establishment System](image)

There are two common ways to establish trust in computer networks. First, when the subject can directly observe the agent’s behavior, direct trust can be established. Second, when the subject receives recommendations from other entities about the agent, indirect trust can be established.

**Direct Trust** is established upon observations on whether the previous interactions between the subject and the agent are successful. The observation is often described by two variables: \( s \) denoting the number of successful interaction and \( f \) denoting the number of failed interactions. That is, the direct trust value can be calculated as \( f_{DT}(s,f) \). The notation \( T_{AB}^{d} \) denotes the direct trust values between node \( A \) and \( B \). Thus, \( T_{AB}^{d} = f_{DT}(s,f) \), and \( T_{AB}^{d} \) can be a vector.

Recommendation trust is a very special direct trust. It is for trust relationship \{subject; agent\}, making correct recommendations. Assume that the subject can judge whether a recommendation is correct or not, and this subject receives \( s_{r} \) good recommendations and \( f_{r} \) bad recommendation from the agent. Then, the recommendation trust value can be calculated as \( f_{DT}(s_{r},f_{r}) \). Recommendation trust is important for establishing indirect trust. The recommendation trust between node \( A \) and \( B \) is denoted by \( T_{AB}^{r} \).

**Indirect Trust:** Trust can transit through third parties. For example, if \( A \) and \( B \) have established a recommendation trust relationship and \( B \) and \( C \) have established a direct trust relationship, then \( A \) can trust \( C \) to a certain degree if \( B \) tells \( A \) its trust opinion (i.e. recommendation) about \( C \). This phenomenon is called trust propagation. Indirect trust is established through trust propagations.

Two key factors determine the indirect trust establishment. The first is to determine when and from whom the subject can collect recommendations. For example, in an ad hoc network, a node may collect recommendations only from its one-hop neighbors, or from all nodes in the network. This affects the number of recommendations can be collected and the overhead of collecting recommendations. This first factor is referred to as the recommendation mechanism.

The second is to determine how to calculate indirect trust value based on recommendations. This calculation is governed by trust models. A trust model usually contains two parts: concatenation model and multipath model, as illustrated in Figure 1.

**Indirect Trust:** Trust can transit through third parties. For example, if \( A \) and \( B \) have established a recommendation trust relationship and \( B \) and \( C \) have established a direct trust relationship, then \( A \) can trust \( C \) to a certain degree if \( B \) tells \( A \) its trust opinion (i.e. recommendation) about \( C \). This phenomenon is called trust propagation. Indirect trust is established through trust propagations.

Two key factors determine the indirect trust establishment. The first is to determine when and from whom the subject can collect recommendations. For example, in an ad hoc network, a node may collect recommendations only from its one-hop neighbors, or from all nodes in the network. This affects the number of recommendations can be collected and the overhead of collecting recommendations. This first factor is referred to as the recommendation mechanism.

The second is to determine how to calculate indirect trust value based on recommendations. This calculation is governed by trust models. A trust model usually contains two parts: concatenation model and multipath model, as illustrated in Figure 1.

**Trust Establishment System**

- **Trust Record**
- **Direct Trust Calculation**
- **Indirect Trust Calculation (Trust Models)**
- **Recommendation Manager**
  - Provide rec. to others
  - Request and process rec. from others

**Fig. 2. Trust Propagation for Indirect Trust Establishment**

Figure 2(a) and 2(b), respectively. In Figure 2(a), \( B \) observes the behavior of node \( C \) and establishes direct trust in \( C \) with trust value \( T_{BC}^{d} \). Node \( A \) has recommendation trust in \( B \) with trust value \( T_{AB}^{r} \). Node \( B \) provides recommendation about \( C \) by telling \( A \) the value of \( T_{BC}^{d} \). The concatenation model is a function that calculates the indirect trust values between \( A \) and \( C \), denoted by \( T_{AC}^{ind} \) from \( T_{BC}^{d} \) and \( T_{AB}^{r} \). This function is denoted by \( f_{ctp}(\cdot) \), and

\[
T_{AC}^{ind} = f_{ctp}(T_{BC}^{d}, T_{AB}^{r})
\]  

(1)

In Figure 2(b), \( A \) receives recommendations from multiple nodes. The multipath model is a function that combines trust established through multiple paths. This function is denoted by \( f_{mtp}(\cdot) \), and

\[
T_{AC}^{ind} = f_{mtp}\left\{f_{ctp}(T_{BC}^{d}, T_{AB}^{r})\right\}_{i=1,2,...}
\]  

(2)

As a summary, a trust establishment system needs to specify at least three key elements: (1) direct trust calculation, (2) recommendation mechanism, and (3) trust models.

**IV. TRUST ESTABLISHMENT ANALYSIS**

**A. Overview**

The overall structure of the proposed analysis method is illustrated in Figure 3. It contains four types of components. Type 1 components describe specific design of trust evaluation methods. They include direct trust calculation, recommendation mechanism, and trust models.

Type 2 components are abstractions that describes the interaction between applications and trust establishment. They allow us to take the application context into consideration but not restrict the analysis to detailed network setups. Rectangles in Figure 3 represent type 2 components.

Type 3 components are the core of trust establishment analysis. They include direct trust analysis, indirect trust analysis and recommendation trust analysis. The design and implementation of type 3 elements is one of the main contributions of this work.

Type 4 component generates a set of metrics describing the performance of trust evaluation methods. The performance analysis module is type 4.

The details of these components are presented in the following subsections.

**B. Inputs**

In this section, we introduce type 1 and type 2 components, i.e. the inputs to the trust analysis module. These inputs describe the application scenarios and the design of trust establishment methods.
1) Direct Interaction Decision-making Model and Virtual Network Topology: The probability that one entity will interact with another entity is an important aspect that will influence trust establishment. This aspect is closely related with application scenarios. This probability is described by the Direct Interaction Decision-making (DID) Model. To simplify the analysis, we assume that this probability is determined by two factors: closeness among entities and trust.

We introduce the concept virtual distance. The virtual distance between entity A and entity B, denoted by \(d_{AB}\), represents the closeness between A and B in terms of having direct interactions. Generally speaking, smaller is \(d_{AB}\), more likely A and B will directly interact with each other, when no trust information is available. In an ad hoc network or a sensor network, virtual distance might be directly related with the real physical distance. The virtual network topology model specifies the virtual distance among all network participants.

For the DID model, let \(f_1(d_{AB}, T_{AB})\) be the probability that the node A will interact with node B in some time interval, where \(d_{AB}\) denotes the virtual distance and \(T_{AB}\) denotes the trust value between A and B. Furthermore, it is assumed that \(f_1(d_{AB}, T_{AB}) = g(d_{AB}) \cdot h(T_{AB})\). That is, the influence of virtual distance and trust values can be considered separately. The DID model should specify \(g(.)\) and \(h(.)\).

2) Direct Interaction Model and Recommendation Behavior Model: These two models describe the behavior of good and bad entities when they interact with other entities and when they provide recommendations.

In the simplest model, there are only two types of nodes: good and bad, and only two outcomes after an interaction: success and failure. When a node interacts with a good node, the interaction will succeed with probability \(\theta_a\). When a node interacts with a bad node, the interaction will succeed with probability \(\beta_a\). This simple direction interaction behavior model can be described by two parameters: \(\theta_a\) and \(\beta_a\).

Similarly, in the simplest recommendation behavior model, there are only two types of recommendations: correct and incorrect. If A’s recommendation about X agrees with X’s behavior, the recommendation is correct. Otherwise, the recommendation is incorrect. This recommendation behavior model is also described by two parameters: \(\theta_r\), which is the probability that the recommendation is correct when it is from a good node, and \(\beta_r\), which is the probability that the recommendation is correct when it is from a bad node.

3) Direct Trust Calculation and Trust Models: As discussed in Section III, the function \(f_{DT}(\cdot), f_{ctp}(\cdot)\) and \(f_{nttp}(\cdot)\) represent the core design of trust establishment methods. As the inputs to our analysis, these three functions need to be specified.

4) Recommendation Mechanism: A brief procedure of collecting and using recommendations is shown in Procedure 1.

\[\text{Procedure 1} \ A \text{ wants to establish indirect trust in B}\]
1: A requests the nodes in A’s recommendation range for recommendations about B. (If node X is in A’s recommendation range and has established direct trust in B, node X will send its recommendation to A.)
2: A puts all received recommendations in a buffer, and calculates indirect trust in B using the trust model.
3: If A observes B’s behavior after establishing indirect trust in B, A can compare B’s behavior with the recommendations in the buffer, and then update the recommendation trust in the nodes that have provided recommendations about B.

From this procedure, one can see that the recommendation mechanism needs to specify two things: recommendation range and how to update recommendation trust.

Recommendation range can be described by the recommendation distance \(d_{rec}\). Node X is in node A’s recommendation range if \(d_{AX} \leq d_{rec}\). A node can only collect recommendations from the nodes in its recommendation range.

To establish and update recommendation trust, the first step is to estimate whether a particular recommendation is correct or incorrect. For example, if the difference between the recommendation about B and A’s observation about B’s behavior is smaller than a threshold, this recommendation is considered to be correct. Then, the recommendation trust can be calculated based on \(s_r\) and \(f_r\) values as described in Section III.

In the process of modeling application scenarios and trust establishment methods, there are many simplifications for the purpose of making the analysis feasible. Even with simplifications, the most important features of trust establishment methods are maintained. The input parameters of the analysis modules are summarized in Table I.

<table>
<thead>
<tr>
<th>Inputs Blocks</th>
<th>Parameters or functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual network topology</td>
<td>Virtual distance among entities</td>
</tr>
<tr>
<td>Direct interaction decision-making</td>
<td>(f_1(d_{AB}, T_{AB}))</td>
</tr>
<tr>
<td>Direct interaction behavior</td>
<td>(\theta_a, \beta_a)</td>
</tr>
<tr>
<td>Recommendation behavior</td>
<td>(\theta_r, \beta_r)</td>
</tr>
<tr>
<td>Direct trust calculation</td>
<td>(f_{DT}(\cdot), f_{ctp}(\cdot)) and (f_{nttp}(\cdot))</td>
</tr>
<tr>
<td>Trust Models</td>
<td>(f_{ctp}(\cdot), f_{nttp}(\cdot))</td>
</tr>
<tr>
<td>Recommendation mechanisms</td>
<td>(d_{rec}), way to determine (s_r) and (f_r)</td>
</tr>
</tbody>
</table>

C. Direct Trust Establishment

A node may continuously observe other nodes’ behavior. However, it is not necessary to update trust record whenever
new observations are made. Thus, we introduce the concept of **round**. Assume that there are total $N$ nodes in the network and let $k$ denote the index of the round. The trust establishment process in round $k$ is as follows.

**Procedure 2** Establishing direct trust in round $k$

1: for $A=1:N$
2: Node $A$ determines a set of nodes with which $A$ will interact with in round $k$. This set of nodes, denoted by $\mathcal{B}$, is determined by the direct interaction decision-making model. The model uses trust values, those trust values are from the trust record established at the end of round $k-1$.
3: The interactions between $A$ and the nodes in $\mathcal{B}$ may succeed or fail, depending on the direct-interaction-behavior model.
4: Based on the outcome of the interactions, node $A$ updates its direct trust record about the nodes in $\mathcal{B}$.
5: end for

It is noted that direct trust, as well as indirect trust and recommendation trust, can be updated at the end of round $k$. When studying the direct trust, the possible update in indirect trust and recommendation trust is not considered. Therefore, Procedure 2 does not include the establishment of indirect and recommendation trust.

Figure 4 shows the process of establishing direct trust between two nodes: say $A$ and $B$. In this figure, each rectangle represents a state. Each state is described by two parameters $(s, f)$. As discussed earlier, $s$ is the number of successful interaction, $f$ is the number of failed interaction, and the direct trust value is calculated from $(s, f)$. Thus, the direct trust value between $A$ and $B$, i.e. $T_{AB}^d$, depends only on which state the system is at. In other words, if we can determine the probability of the states at round $k$, we can obtain the probability mass function (pmf) of the direct trust values at round $k$.

The transition between the states is uniquely determined by three inputs to trust analysis: direct interaction decision-making model, direct interaction behavior model, and the calculation of direct trust. As shown in Table I, these three models are represented by $f_I(d_{AB}, T_{AB}), f_{DT}(s, f), \theta_n$ and $\beta_n$. In particular, if the current state is $(s, f)$ at round $k$, the system can possibly transfer to the following states in round $k+1$.

- to state $(s, f + 1)$ with probability $(1 - \theta) \cdot f_I(d_{AB}, f_{DT}(s, f))$;
- to state $(s + 1, f)$ with probability $\theta \cdot f_I(d_{AB}, f_{DT}(s, f))$;
- remain at state $(s, f)$ with probability $1 - f_I(d_{AB}, f_{DT}(s, f))$;

where $\theta = \theta_n$ if $B$ is a good node and $\theta = \beta_n$ if $B$ is a bad node.

Based on Figure 4, one can calculate the pmf of $T_{AB}^d$ for any given round $k$ using a simple program shown in Procedure 3. Here, let $p_{s,f,k}$ denote the probability that $T_{AB}^d = f_{DT}(s, f)$ at round $k$.

**Procedure 3** Calculating the pmf of direct trust values

1: Set $p_{0,0,0} = 1$ and all other $p_{s,f,k}$ values as 0.
2: for $k = 1 : \hat{k}$
3: for all possible $(s, f)$ values do
4: update $p_{s,f,k+1}$ using (3).
5: end for
6: end for

The equations used in Procedure 3 is

$$
p_{s,f,k+1} = p_{s,f,k}(1 - f_I(d_{AB}, f_{DT}(s, f))) + p_{s-1,f,k} \cdot \theta \cdot f_I(d_{AB}, f_{DT}(s-1, f)) + p_{s,f-1,k} \cdot (1 - \theta) \cdot f_I(d_{AB}, f_{DT}(s, f-1)).$$

With $p_{s,f,k}$, it is easy to generate the pmf of $T_{AB}^d$ at any given round. It is noted that if the function $f_{DT}(s, f)$ yields the same value for some different $(s, f)$ values, $p_{s,f,k}$ and the pmf of $T_{AB}^d$ are not exactly same. Some simple conversion is needed.

As a summary, the direct trust establishment module can output the statistical distribution of direct trust value between arbitrary pair of nodes in any given round.

**D. Recommendation Trust Establishment**

In this section, the goal is to calculate the pmf of recommendation trust values. From Procedure 1, one can see that node $B$ can make recommendations to node $A$ about node $Y$ in round $k$ if and only if the following three conditions are satisfied.

(C1) $A$ requests recommendations about node $Y$ in round $k$.
(C2) $B$ has established direct trust in node $Y$ in the previous $k-1$ rounds.
(C3) $B$ is in $A$’s recommendation range, i.e. $d_{AB} \leq d_{rec}$.

Through $B$’s recommendation about $Y$, $A$ can establish recommendation trust in $B$ in round $k$ if one additional condition is satisfied:

(C4) $A$ can judge whether $B$’s recommendation about $Y$ is correct or not.

To make the analysis manageable, the following assumptions are made.

(A1) $A$ requests recommendations about node $Y$ in round $k$ if $A$ selects node $Y$ to interact with in round $k$.
(A2) $A$ makes accurate judgment on whether $B$’s recommendation about $Y$ provided in round $k$ is honest or not, if $A$ establishes direct trust with $Y$ in round $k$ or in the previous rounds.
The recommendation mechanism follows Procedure 1. One direct interaction is sufficient to establish direct trust. For example, if $B$ has one direct interaction with node $Y$, then $B$ can provide recommendations about node $Y$. Here, the first assumption (A1) is made because we have to know when condition 1 will be satisfied. This assumption can greatly simplify the analysis. The second assumption (A2) can simplify the handling of condition 4. In particular, with (A1) (A2) and (A3), condition 4 is satisfied whenever condition 1 is satisfied. For condition 3, we only need to multiply whatever probability results we obtained by $1(d_{AB} \leq d_{rec})$, where $1$(statement) equals to 1 when the statement is true and equals to 0 otherwise. To make a concise presentation, we omit the term $1(d_{AB} \leq d_{rec})$ in the rest of this section.

From above discussion, one can see that

$$Pr\{B makes recommendation to A about Y in round k\}$$

$$= Pr\{condition 1 is satisfied\} \cdot Pr\{condition 2 is satisfied\}$$

(4)

Let $T_{AY,k}^d$ denote the value of $T_{AY}$ in round $k$. From the direct interaction model, one can see that

$$Pr\{condition 1 is satisfied\} = f_I(d_{AY}, T_{AY,k-1}^d),$$

(5)

$$Pr\{condition 2 is satisfied\} = 1 - \prod_{i=1}^{k-1} (1 - f_I(d_{BY}, T_{BY,i-1}^d)),$$

(6)

From (4) (5) and (6), we get

$$Pr(B makes recommendation to A in round k)$$

$$= \sum_Y f_I(d_{AY}, T_{AY,k-1}^d) \cdot \left[ 1 - \prod_{i=1}^{k-1} (1 - f_I(d_{BY}, T_{BY,i-1}^d)) \right]$$

(7)

We assume that $\sum_Y f_I(d_{AY}, T_{AY,k-1}^d) < 1$. That is, a node is unlikely to interact with more than one node in one round. Due to this assumption and (A1)-(A4), the probability calculated in equation (7) also equals to the probability that $A$ observes whether $B$ makes one honest or dishonest recommendation in round $k$. We denote this probability as $f_R(A, B, k)$, where $f_R(\cdot)$ is a function of node index ($A$ and $B$), round index ($k$), and trust values among many nodes in round $k - 1$, as indicated in (7).

With $f_R(A, B, k)$ and the recommendation behavior model, we are ready to calculate the pmf of the recommendation trust. Figure 5 show the model for calculating recommendation trust. Similar as in Figure 4, the rectangles represent states, which are described by $(s_r, f_r)$. If the current state is $(s_r, f_r)$ at round $k$, the systems can transfer to the following states in round $k + 1$:

- to state $(s_r + 1, f_r)$ with probability $\theta \cdot f_R(A, B, k)$;
- to state $(s_r, f_r + 1)$ with probability $(1 - \theta) f_R(A, B, k)$;
- remain at state $(s_r, f_r)$ with probability $1 - f_R(A, B, k)$;

where $\theta = \theta_r$ if $B$ is a good node and $\theta = \beta_r$ when $B$ is a bad node.

Let $T_{AYB}^d$ represent the recommendation trust between $A$ and $B$. Let $p_{s_r,f_r,k}^r$ denote the probability that $T_{AYB}^d = f_R(T_{AY}^d, 1)$ at round $k$. Based on Figure 5, it is not difficult to calculate $p_{s_r,f_r,k}^r$, using a procedure similar to Procedure 3.

E. Indirect Trust Establishment

The last module of trust establishment analysis is to obtain the distribution of indirect trust values. As discussed in Section III, indirect trust is calculated from direct trust and recommendation trust. The calculation is governed by trust models.

Figure 2(b) shows the scenario of trust propagation that will be analyzed in this section. It is noted that this analysis only allows one-hop concatenation trust propagation. This limitation is due to the following reasons. As the length propagation chain increases, the cost of collecting recommendations increases rapidly (e.g. exponentially in [20]) and the trust between the subject and the agent degrades quickly. Therefore, many trust establishment methods only allow one-hop trust propagation, such as in [22]. Additionally, the analysis for multi-hop trust propagation is extremely difficult.

As discussed in Section III, the indirect trust between $A$ and $Y$ is calculated from direct trust and recommendation trust using trust models. When the pmf of direct trust ($T_{BY}^d$) and the pmf of recommendation trust $T_{AYB}^d$ are obtained using the procedure in Section IV-C and IV-D, the pmf of $T_{AY}^{ind}$ can be calculated.

It is noted that direct trust values are calculated from $(s, f)$, where $s$ and $f$ are integers. Therefore, the values of $T_{AY}^d$ are discrete. Similarly, the recommendation trust values are also discrete. The fact that $T_{AY}^{ind}$ and $T_{AYB}^{ind}$ are discrete random variables enables simple numerical calculation of the pmf of $T_{AY}^{ind}$. It is also noted that the complexity of the calculation increases rapidly with the number of rounds.

F. Outputs

As shown in Figure 3, for any given pair of nodes, our analysis can generate the pmfs of the direct trust, recommendation trust and indirect trust. Of course, these pmfs are for specific trust establishment methods under certain application scenarios.

Based on these pmfs, more metrics can be calculated by the performance analysis module. For any pair of nodes $(A, Y)$, one can compute the mean and variance of direct/indirect/recommendation trust between $A$ and $Y$ as a function of round index. In addition, when the trust values are used in malicious node detection algorithms, the detection rate and false alarm rate can be computed.
V. DEVELOPMENT, TESTING, AND RESULTS

We implement an analysis tool in Matlab based on the approaches presented in Section IV. This tool takes the inputs described in Table I, and outputs the statistical distribution of direct, recommendation and indirect trust between two arbitrary nodes in the network. To validate the analysis methods proposed in the paper, we also build a simulation testbed that takes similar inputs as the the analysis tool. This testbed simulates the trust establishment process and estimates the pmf of trust values based on a large number of tests. In this section, we first compare the analysis and simulation results, and then utilize the analysis tool to derive important research results.

A. Validating Analysis Through Simulations

The first experiment is to show the process of establishing direct trust values. The inputs are chosen as follows. The 2D virtual network topology is shown in Figure 6. For direct trust calculation, \( f_{DT}(s, f) = \frac{(s + 1)}{(s + f + 2)} \), where \( s \) and \( f \) are defined in Section III. This has been used in many trust establishment methods. In the DID model, \( g(d_{AB}) = \frac{1}{N} \) and \( h(T_{AB}) = t(T_{AB}) \), where \( N \) is the total number of nodes in the network, and function \( t(T_{AB}) = f_{DT}(s, f) \). In the direct interaction model, \( \beta_A = 0.9 \) and \( \beta_B = 0.1 \).

Figure 7 shows the pmf of the direct trust values between two nodes at round 5, 20, 40 and 120. The left four plots are for \( T_{AY_1}^d \) and the right four plots are for \( T_{AY_2}^d \). \( A \) and \( Y_1 \) are good node, and \( Y_2 \) is a bad node. \( Y_1 \) and \( Y_2 \) are two hops away from \( A \). In all plots, the lines marked with \( \diamond \) are for analysis results, and the lines marked with \( * \) are for simulation results. One can see that the analysis and simulation match well. When two nodes do not have direct interaction (i.e. \( s = 0 \) and \( f = 0 \)), the trust value is 0.5. This is why some plots have a peak at 0.5.

Similar to the first experiment, we can verify the analysis of recommendation trust and indirect trust through simulations. For example, Figure 8 shows the mean and variance of the recommendation trust between \( A \) and \( Y_1 \) as a function of round index. In this experiment, \( \theta_r = 0.8 \), \( \beta_r = 0.5 \), and \( g(d_{AB})h(T_{AB}) = \frac{1}{N} \). The trust models are chosen to be the simple model described in Section V-C.

As expected, the mean of the recommendation trust increases as the round index increase. Initially, one would expect that the variance value decreases with the round index. However, it is not completely true. The indirect trust values are discrete. In the first few rounds, there are only a few possible \( T_{AY_1}^r \) values, which yields a relatively small variance. As the round index increases, \( T_{AY_1}^r \) can take more values, which yields a larger variance value. As the round index further increases, the node \( A \) collects more observations about \( Y_1 \), and more observations lead to smaller variance value in the calculation of \( T_{AY_1}^r \).

B. Probability of Trust Establishment

Trust values are used to assist decision-making in distributed systems. From the application points of view, the earlier trust can be established the better. Therefore, it is important to evaluate how likely trust can be established as the round index increases.
In Figure 9, the probability that direct, recommendation, and indirect trust can be established between node A and Y1 are shown as a function of round index. The simulation setup is similar to that in Section V-A. Several observations are made. First, compared with direct trust, indirect trust is more likely to be established. Thus, with recommendation mechanism, the subject can use indirect trust information before direct trust can be established. Second, it is highly likely that recommendation trust is established long before the direct trust. Thus, when A tries to determine whether Y is trustworthy or not, recommendation trust between A and Y should be used, especially at the beginning of trust establishment (i.e. when the round index is small).

C. Comparison Among Trust models

An important application of the proposed trust analysis tool is to facilitate comparison among different trust establishment methods. As discussed earlier, the design of trust models is critical for indirect trust establishment. Thus, we compare different trust models by examining the pmf of T^{ind}_AY1 and T^{ind}_AY2. Here, T^{ind}_AY1 and T^{ind}_AY2 represent the indirect trust of a good and a bad node, respectively.

Assume that we are going to use the indirect trust to determine whether a node is good or bad. Given a detection threshold Th, if T^{ind}_AY < Th, node Y is detected as a suspicious node.

Based on the pmf of T^{ind}_AY1 and T^{ind}_AY2, for a given Th, we calculate the detection probability as the probability that Y2 is marked as suspicious, and the false alarm probability as the probability that Y1 is marked as suspicious. Furthermore, by changing Th, we can obtain the curve: detection probability vs. false alarm probability. Such curves for several different trust models are shown in Figure 10. The three models evaluated in this experiment are

- The simple model used in many trust evaluation schemes.
  In this model, T_{AB} is a scalar and T_{AB} = (s+1)/(s+f+2), f_{ctp}(x,y) = xy, and f_{mtp}(\{x_i\}) is just the average of \{x_i\}.
- The probability model proposed in [20]. The trust value T_{AB} is a scalar, f_{ctp}(x,y) = xy + (1-x)(1-y), and f_{mtp}(.) is based on a data fusion model.
- The Beta function model proposed in [26]. In this model, T_{AB} is a 2 by 1 vector. One element represents trust, and the other element represents confidence. The detailed equations can be found in [26] and [27].

In Figure 10, we first examine the region when false alarm probability is smaller than 0.05. The malicious node detection algorithms should always work in this region. The probability model performs better than the simple model. This is because the probability model can better handle incorrect recommendations from malicious nodes. The beta function model performs much better than the other two models. This is because the beta function model considers the possible estimation error when calculating the trust values. This observation agree with the qualitative arguments in the current literature. More importantly, The proposed analysis tool provides a quantitative comparison among trust models. The quantitative comparison is not currently available in the literature, but it is important for the future research in this field.

When the false alarm probability is higher than 0.05, the detection probability starts to saturate. For a given round index, the detection probability cannot be higher than the probability that A can establish indirect trust in Y2. In Figure 10, the probability model has the lowest detection probability when the false alarm probability is large. Compared with two other models, the probability model results in the least likelihood of establishing indirect trust. In both regions, the beta function model has the best performance.

VI. CONCLUSION

This paper proposed the methods to analyze the process of trust establishment in distributed networks. The tools for performing the analysis were implemented, and validated by simulations. The proposed analysis methods were utilized to solve an important research problem: quantitative comparison among different trust models. In the future, we will exploit more applications of the analysis tool, such as understanding the effects that application context has upon trust establishment, and guiding the design of better trust establishment methods.
REFERENCES


