Collusion-resilient Quality of Information Evaluation Based on Information Provenance

Xinlei (Oscar) Wang, Kannan Govindan and Prasant Mohapatra
Department of Computer Science
University of California, Davis, CA 95616
Email: {xlwang, kgovindan, pmohapatra}@ucdavis.edu

***Abstract*—The quality of information is crucial for decision making in many dynamic information sharing environments such as sensor and tactical networks. Information trustworthiness is an essential parameter in assessing the information quality. In this paper, we present a trust model to evaluate the trustworthiness of information as well as information publishing entities based on information provenance. In our trust model, decision makers can give an evaluation on the information they receive and further adjust the evaluation result to a more accurate value by considering two factors: information similarity and path difference. We introduce Collusion Attacks that may bias the computation and present a mechanism to detect and reduce the effect of Collusion Attacks. Based on the final adjusted information trust, feedback is given to the information publishing nodes to adaptively update their trust scores. Therefore, our collusion-resistant scheme can dynamically assess the trustworthiness of information as well as participating entities in a network and thus effectively enhance the network security. Detailed analysis of the proposed approach is presented along with simulation results.**

I. Introduction

With the advance of networking and information technologies, dynamic network environments are often used by individual people, business corporations as well as governmental and military organizations to obtain information from different sources. In such network environments, information transmissions and sharing are the essential activities. Information from different sources makes it possible to extract more accurate and complete knowledge and thus support more informed decision making. Hence, high quality and trustworthiness of received information is crucial for the decision makers.

Plenty of work has been done on the protection from data tampering, e.g., digital signature techniques, to ensure data integrity when the information is routed through multiple nodes. However, they do not address the problem of information trustworthiness. Untrustworthy information may be introduced because of two different reasons: *unintentional errors* and *intentional misbehavior* [1]. Unintentional errors are caused by malfunction of the hardware (e.g., broken or obstructed sensors), mispositioning of the node or exhausted batteries. Intentional misbehaviors are caused by malicious attackers, providing false data on purpose through compromised nodes.

To assess the quality and trustworthiness of information, a trust evaluation mechanism for the entire network is required. It will not only allow us to assess the quality of information, but also enhance the overall security level of the network. In a multi-hop network, information can be generated from multiple source nodes, e.g., sensors or satellites. It may then go through a series of other intermediate nodes before reaching the destination, i.e., the decision maker. In order to assess the trustworthiness of such information, we need to take the provenance of the information into consideration. We define information provenance as follows:

DEFINITION 1: **Information Provenance:** The location history of the information starting from its creation, which has the details about the publisher of the information, and details about various nodes which has passed/processed the information before it reaches the destination.

Another notable observation is that the terms "trust" and "reputation" are generally used interchangeably or only one of them is used in a network trust or reputation model. However, in this paper, we consider them as different concepts and both of them are highly important in our proposed framework. Here we define the trust of information items, trust of nodes and reputation of nodes as follows:

DEFINITION 2: **Trust of Information Items:** The trust of an information item i, denoted as $T(i)$, is the probability of i being true, as perceived by the receiver.

DEFINITION 3: **Trust of Nodes:** The trust of a node N perceived locally by node M, denoted as $T_M(N)$, is the probability that information items N owns and sends to M are true.

DEFINITION 4: **Reputation of Nodes:** The reputation of a node N, denoted as $R(N)$, is a single global value which represents the synthesized probability that information items owned by N are true, as perceived by all other nodes which have received information items and given feedback to N.

In this paper, we address the problem of assessing trustworthiness of information flowing around the network by taking advantage of the fact that usually multiple sources of information are available in a network. Our approach is based on assessing the similarity of multiple information items about the same event from different sources and then adjusts the trust scores of each such information. Based on the information trust evaluation, we can also dynamically update the trust scores of nodes in the network.

There have been several attack models introduced which can undermine the accuracy of trust and reputation systems, e.g., Bad Mouthing Attack, On-off Attack, Conflicting Behavior Attack and Sibyl Attack. They have been well studied and multiple countermeasures have been proposed [2], [3]. Our scheme is also resilient to most of them. Other than these attacks, Collusion Attack remains a major problem in trust and reputation systems and is much harder to deal with. Two or more malicious nodes may collaborate to disrupt the network severely by covering up malicious behavior of each other from the remaining part of the network. The effects of these attacks can dramatically affect the accuracy of the trust evaluation and thus result in large security degradation. Therefore, a scheme to alleviate the effects of Collusion Attack is a significant component on top of the trust model. In this paper, we present an Anti-collusion Mechanism to detect and reduce the effects of Collusion Attack.

The rest of the paper is organized as follows. Section II summarizes some of the related work in the field of network trust assessment and countermeasures for Collusion Attack. Section III introduces the fundamental network structure and elements, basic concepts of our provenance-based information trust evaluation, as well as our proposed Centralized Reputation Distributed Trust (CRDT) framework. Section IV presents our Primary Trust Model which is a three-step trust evaluation process and Section V describes the detailed approach and algorithm to handle Collusion Attack. Analysis and simulation results are presented in Section VI. We discuss the limitations of our scheme as well as our future research plans in Section VII. Finally, Section VIII concludes the paper and outlines our future research work.

II. Related work

Approaches of trust computations and analysis have been widely studied in order to establish and quantify the trust of entities in different types of networks [4]–[6]. In addition, there have been lots of work on trust dynamics too including trust prediction [7], [8], propagation [9], [10] and aggregation [10], [11]. These are all important trust and reputation techniques to detect or predict any malicious behaviors and thus enhance the overall security of the network. However, these techniques only consider the trust or reputation of network nodes. None of them addresses the information trustworthiness and hence quality of information. Especially in multi-hop networks, information is processed by a series of intermediate nodes before reaching the destination and even some intermediate node may generate new infused information based on inputs from other nodes. In these cases, it will not be appropriate to make judgments on the information trustworthiness based only on the trust of a particular node. Thus, information provenance has to be taken into account.

The concept of provenance has been studied and used in the field of database and data-centric workflows [12]–[14]. However, to our knowledge, there is limited work which has investigated modeling and analysis techniques to assess the quality of information based on provenance in multi-hop information sharing network environment. A related work is an agent-based approach proposed by Yu et al. [15], in which a computation model is presented to calculate the trustworthiness of information using the framework of Dempster-Shafer theory. However, Dempster-Shafer theory cannot be used to correctly capture information conflicts. In addition, Dempster's rule of combination can only merge independent and uncorrelated evidences, whereas correlation between information items could be highly common in a dynamic network environment, e.g., information items from different paths might originated from the same node or have been processed by some common nodes. A provenance-based data trust model which estimates the trustworthiness of both data and data providers is presented in [16]. Four factors that affect the trustworthiness of data have been taken into account, which are (a) data similarity, (b) path similarity, (c) data conflict and (d) data deduction. They also extended this work and considered countermeasures for Collusion Attack in [17]. In these two papers, the authors presented lots of interesting ideas for provenance-based trust evaluation and collusion handling techniques, which we have borrowed in our trust model in this paper. However, they mainly considered location data only and there are certain drawbacks with respect to network situations. First, the data similarity and data conflict factors may introduce some replicate effects into the information trust computation. The assumption of data deduction method is too simple and fixed which may lead to inaccurate results of trust computation in a complicated network environment. In addition, the collusion detection and mitigation algorithm is specific to location data only, which needs to be revised and generalized for applying to a dynamic multi-hop network.

Our contributions: We have proposed a three-step approach to evaluate the trustworthiness of information and the information providers based on provenance in [18]. Our new trust model in this paper is an extension of this three-step approach, so that it still has the advantages mentioned in [18], i.e., it is taking both information similarity and path correlation into consideration and also evaluating the trust of nodes in a feedback manner. Also, it is independent of techniques used by intermediate nodes to derive/infuse new information, hence intermediate nodes can have many different ways to do so based on different purposes and the end users need not to have any knowledge about it. However, our previous work does not take various types of reputation system attacks into consideration and has pretty large implementation and computation overhead due to every node has to report a trust value on each information item they own. In this paper, we make refinements on [18] and propose our new Primary Trust Model to eliminate the implementation and computation overhead associated with the reported trust, yet it still maintains the ability to accurately capture the trustworthiness of information. Moreover, we propose the Centralized Reputation Distributed Trust framework instead of always using global reputation values as proposed in [18], so that our Primary Trust Model is resilient to basic attack types for reputation systems such as Bad Mouthing Attack, On-off Attack and Conflicting Behavior Attack. Most importantly, in most of the existing work, Collusion Attack can cause severe damage to the network. Here we present our Anti-collusion Mechanism which effectively handles Collusion Attack in the network.

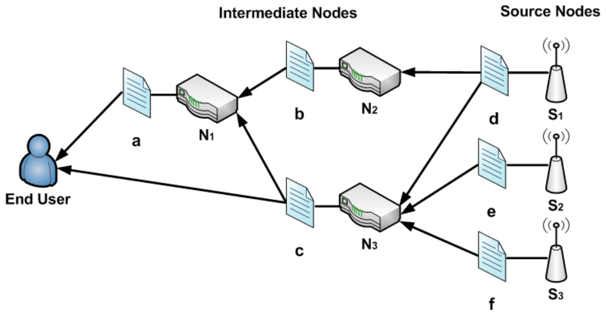


Fig. 1. An example network scenario for provenance-based information trust evaluation

III. Network Framework

A. Fundamental Framework

In this paper, we consider a network meant for detecting events and sharing information. There are three types of entities in the network: source nodes, intermediate nodes and end users. Source nodes generate new information about certain events which is then relayed by the intermediate nodes to the end users. End users receive the information and evaluate the trustworthiness of this information, and then make their decisions on further actions. One thing to be noted is, intermediate nodes could also be end users in their own perspectives and their further actions could be just to pass the information without modifications or to modify the information by infusing information received from many sources and then send it to other nodes.

To effectively evaluate the trustworthiness of information, we consider every piece of information as an information item. An information item is a statement that describes a certain event that happens in the network environment. Each information item consists of meta-data and payload. The meta-data contains the provenance of the information item, which in turn includes the information item's creation time, owner, location history, as well as the provenance of its input information item(s) in case it is generated by an intermediate node. The owner of an information item is the source node or intermediate node which published the information item. When an intermediate node N processes or merges some information items, we consider the information item it sends out as an entirely new information item owned by N. Fig. 1 depicts a simple network scenario that shows how information items are generated and routed to an end user node.

B. Centralized Reputation Distributed Trust

Based on how we defined "trust" and "reputation" of network nodes in Section I, we propose a framework named *Centralized Reputation Distributed Trust* (CRDT) to store, manage and update the trust and reputation values of various network nodes. As illustrated in Fig. 2, we will have a *Central Reputation Manager* to store and maintain a global reputation table which contains a global reputation value for every network node. However, at the same time, each node maintains a local trust value for all the other nodes that it has received information from. In this sense, reputation is a global value maintained by the Central Reputation Manager and trust is a local value calculated by each node by observing other information owners. A node M should always use the local trust value if possible instead of the global reputation evaluate the trust of any information items, in order to prevent itself being attacked by Conflicting Behavior Attackers. In addition, M can only give feedback to the local trust value, but not directly to the reputation value. In case M receives some information owned by a node N for the first time, it needs to request for the global reputation value $R(N)$, create a new local trust $T_M(N)$ and initialize it as $R(N)$.

The Central Reputation Manager periodically updates the reputation table by taking votes from nodes, and this can be done asynchronously. For example, the Central Reputation Manager wants to update the reputation values of N, it can just poll votes from the set of nodes that has a local trust value for N. We denote this set of nodes as $\mathbf{S_N}$. Every node in $\mathbf{S_N}$ will send a vote for N to the Central Reputation Manager. A vote sent by M for N contains two values, $T_M(N)$ and F_M^N, where F_M^N is a number maintained by M which is the number of times M has given feedback to N. We need this number in the votes because the local trust value $T_M(N)$ needs to be weighted based on F_M^N. The intuition behind this is that the more often a node M has given feedback to node N, the more M interacted with N and hence more accurate M's opinion on N should be, therefore we give more weight to $T_M(N)$.

The reputation voting process is the only part of our model that is subject to Bad Mouthing Attack. The attackers could take advantage of F_M^N by giving it a very high value and thus overwhelm the global reputation of the attacking targets with their own votes. Therefore, we set an upper limit for F_M^N as F_L. Moreover, all votes should also be weighted based on the reputation of the corresponding voters. Assuming the majority of the nodes are good, then the higher reputation the voter has, the lower probability that it is compromised and launch Bad Mouthing Attack, and thus the more we can trust its local trust values. By having this feature and setting a proper F_L value based on the context, Bad Mouthing Attack on the reputation voting can be effectively mitigated. Now we have the following equation to compute a new reputation:

$$R_{new}(N) = \frac{\sum_{M \in \mathbf{S_N}} (min\{F_M^N, F_L\} \cdot R(M) \cdot T_M(N))}{\sum_{M \in \mathbf{S_N}} min\{F_M^N, F_L\} \cdot R(M)} \quad (1)$$

$R_{new}(N)$ is not the final updated reputation. We introduce a factor α which falls in the range (0, 1] to incorporate the influence of past reputation on the updated reputation.

$$R_{updated}(N) = \alpha \cdot R_{new}(N) + (1 - \alpha) \cdot R_{old}(N) \quad (2)$$

where $R_{old}(N)$ is the reputation for node N currently in the reputation table and $R_{updated}(N)$ is the final updated reputation for N we are about to put in the reputation table to replace $R_{old}(N)$. By setting α higher, the observation made long time ago will carry less weight than that made recently, and thus it can be used as a countermeasure for On-off attacks.

By having the distributed local trust values, since a node would always use its local trust values for information trust evaluation, a malicious node N cannot specifically attack a node M and still make M think that it is trustworthy by maintaining a high reputation via other nodes' feedback. That is, the CRDT framework makes our model resilient to the Conflict Behavior Attack. If N wants M to trust it, the only way to do so is to send trustworthy information to M. Other nodes opinion on N will not affect M's opinion on N. Secondly, when two or more malicious nodes keep sending information and giving positive feedback to each other, they can only increase each other's local trust values. Other nodes do not actually care about and will not be affected by these nodes' local trust of each other, because they are keeping totally different copies of local trust values for these nodes. The global reputation value will not be affected much as well because of the upper limit K. This is actually a scenario of Collusion Attack which we will discuss more in Section V.

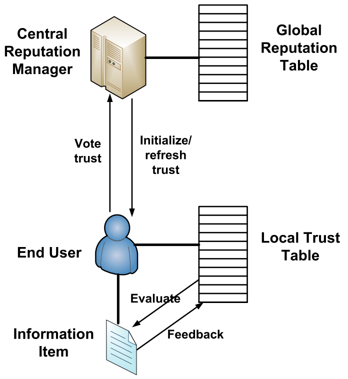


Fig. 2. Illustration of the Centralized Reputation Distributed Trust framework

By keeping the centralized reputations, we have a value that reflects all other nodes' opinion on a particular node N. Hence any node M can always request for N's reputation value from the Central Reputation Manager and use it as its initial local trust for N. This is particularly important when M receives some information owned by N for the first time without any prior interactions, or M's local trust value for N is corrupted and therefore needs to be refreshed, or M's local trust value for N is lost due to M's frequent mobility. In addition, by having a global reputation value, the central network administrator can have an overall picture of how each node behaves and thus can disconnect a node from the network when its reputation becomes lower than a certain threshold level.

C. Threat Model

In our network, every node has the ability to create any information and publish unlimited number of false information items to the network. Every nodes can give whatever feedback to the trust values in its own local trust table and provide whatever values during the reputation voting process. In addition, malicious nodes can always collaborate with each other and attack the network.

There are certain attack models we are not addressing in this paper. Therefore, we make some assumptions here:

1) Majority of the nodes are good nodes.
2) The provenance information is intact as being propagated in the network.
3) Every information item has a signature from its owner and no unauthorized data tampering happens in the network.
4) Information items are never dropped along the path by any malicious nodes or due to other reasons.
5) Network nodes cannot create fake IDs and thus Sybil Attack is not under our consideration.
6) Reputation and local trust tables are securely protected.

Assumption 2) and 3) can be ensured by adopting the data provenance architecture proposed in [19]. Sybil Attack happens most often in online reputation systems where users can create multiple accounts without any restrictions. It has been well studied and various defending techniques for Sybil Attack can be found in [20]–[22]. In most information sensitive sensor or tactical networks which we are concerning in this paper, nodes cannot freely create pseudonymous IDs, therefore we have Assumption 5).

IV. Primary Trust Model

Before considering Collusion Attack, we introduce a trust evaluation process to assign trust values to information items and adjust trust values of nodes in the network, which is an enhanced trust model of our previous work [18]. We refer to this evaluation process as the *Primary Trust Model*. In [18], we require every node to annotate a trust value (*Reported Trust*) on the meta-data of all the information it generates and sends to other nodes. However, this requirement introduces some communication overhead and computation complexity. The Reported Trust was required because we wanted to give the nodes the ability to publish information that is not fully trustworthy but still valuable. However, in most cases, network applications do not need such a feature and even if certain applications need it, the information payload itself can contain descriptions of the trustworthiness of itself. Therefore, we revised our trust evaluation process by removing the requirement for the Reported Trust in our new Primary Trust Model in order to eliminate the overhead and complexity associated with it. However, we do not compromise the accuracy and effectiveness of the trust evaluation. In this section, we provide an overview of our Primary Trust Model. Here we try to be brief in order to reduce the overlap with [18]. Our Primary Trust Model has the following three steps:

1) Initial information trust computation
2) Information trust adjustment
3) Owner's local trust feedback

A. Initial Information Trust Computation

First of all, let us consider the following question: when a node (end user) M receives an information item i whose owner is node N, how can M decide how much it can trust i, given that the user knows nothing that can help with its assessment except for the provenance of information item i and the local trust value of node N. It is intuitive to see that higher the local trust $T_M(N)$, higher probability that information item i is true, thus M should give a higher trust score to i. Therefore, we can directly assign the local trust of node N to information item i as follows:

$$T_i(i) = T_M(N) \tag{3}$$

We call $T_i(i)$ as the *initial trust* of information item i. This is the very first trust value we get for i solely based on i's provenance. In most cases, when a network event happens, multiple sources would generate information about it and send the information to the end users, so that a user node should be able to receive plenty of information items that describe the same event. Moreover, in tactical networks, normally an end user is able to pull information from multiple sources, in which case it can also receive multiple information items for one network event. In these situations, the end user nodes can adjust the initial trust by considering multiple information items that support or conflict with each other. In the next part of this section, we analyze how the adjustment can be done.

B. Information Trust Adjustment

1) Information Similarity: Different items about the same event in the network environment may be either supportive or conflicting. Similar information items are considered as supportive to each other, while conflicting information items compromise the trustworthiness of each other [16]. Therefore, we can adjust the information trust values based on their supports and conflicts to each other. For brevity, we suggest to use an existing clustering algorithm to group information items that describe same event into an *information collection*. Therefore, for each network event, an end user node can have collections of information items. For data similarity comparison, there has been lots of work done in the field of data mining [23], [24]. These techniques can be used to measure the similarity between any two information items within a collection.

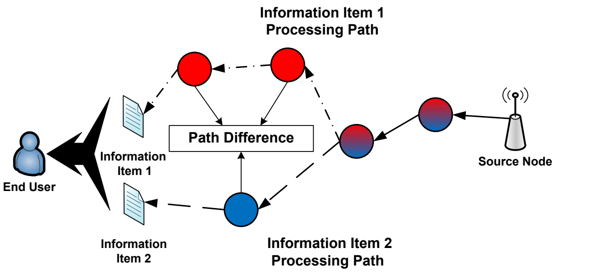


Fig. 3. Illustration of path difference in a multi-hop information flow

We assume any two information items i and i' within a collection have a similarity score of $S(i, i')$ which ranges from -1 to 1, where -1 means completely conflicting with each other and 1 means exactly the same. Now what we really care about is how to actually utilize the similarity scores to adjust the information trust. The first step of our adjustment process is to assign a *similarity factor* Δ_i to information item i which belongs to a collection C_i as follows:

$$\Delta_i = \frac{\sum_{i,i' \in C_i, i \neq i'} S(i,i')}{|C_i| - 1} \tag{4}$$

where $|C_i|$ is the number of information items in the collection C_i. Each information item in a collection will be assigned with a similarity factor with respect to the information collection. A negative similarity factor means there are more conflicts in the collection and a positive similarity factor means there are more supports.

2) Path Difference: If several independent nodes provide the same information about a particular event, such information is likely to be true. However, even when certain information items have high similarity with each other, if their provenance contains a large number of the same nodes, then there can be high correlation among these information items, and therefore they are not as supportive to each other. Thus, we need to take the provenance correlation between any two information items into consideration as well.

We only care about the *Processing Path* [18] of the information items because those nodes which only did "pass" action do not affect the information. In the rest of this paper, we will just use "path" to refer to the information processing path. We use *path difference* to measure the provenance correlation between two different information items. Illustration of path difference is shown in Fig. 3. For any two information items i and i', their path difference can be calculated as follows:

$$PD(i,i') = \frac{max\{|P_i|, |P_{i'}|\} - S\{P_i, P_{i'}\}}{max\{|P_i|, |P_{i'}|\}} \tag{5}$$

where $|P_i|$ and $|P_{i'}|$ are the numbers of nodes on the paths of information items i and i' respectively and $S\{P_i, P_{i'}\}$ is the number of common nodes on the two paths. For information item i, we can now assign a *path difference factor* Θ_i to account for the overall provenance correlation with other items in the same information collection as follows:

$$\Theta_i = \frac{\sum_{i,i' \in C_i, i \neq i'} PD(i,i')}{|C_i| - 1} \tag{6}$$

By assigning the path difference factor, if a malicious owner node sends out the same false information item multiple times in order to deceive an end user node, the end user node will only consider them as one item in the information collection.

3) Initial Trust Adjustment: By having the similarity factor and path difference factor, we can proceed to actually adjust the information trust. The amount of adjustment to be made on the initial trust of information item i is denoted by λ_i and given as follows:

$$\lambda_i = \Delta_i \cdot \Theta_i \cdot e^{-\frac{1}{|C_i|}} \cdot \omega \tag{7}$$

where ω is a user-defined parameter which determines the range of possible adjustment. The reason we introduced another term $e^{-\frac{1}{|C_i|}}$ is that the more items in i's collection, the more we should be convinced by the similarity factor, thus the more influence this adjustment factor should have. It increases negative exponentially with the collection size because as the collection size increases, this effect should become smaller. λ_i can be either positive or negative depending on whether most of other information items in the same collection are supportive or conflictive to i. Finally, we have the *adjusted trust* $T_a(i)$ as follows.

$$T_a(i) = T_i(i) + \lambda_i, \ 0 \leq T_a(i) \leq 1 \tag{8}$$

C. Owner's Local Trust Feedback

In addition to information trust evaluation, we also would like to dynamically update the local trust values of the information owners. Therefore, we have a trust feedback step to update the local trust of the information owners based on the difference between the initial information trust and the adjusted information trust. One important thing to be noted is, a node can maximally give one feedback based on one information item it receives.

Given an information item, if we have increased its trust during the above adjustment phase, the corresponding owner node's local trust should get credits, and the new local trust of the owner (denoted by $T'_M(N)$) will be:

$$T'_M(N) = T_M(N) + \rho \cdot \lambda_i(1 - T_M(N)) \tag{9}$$

Otherwise, when λ_i is negative, we penalize the owner's local trust, so that the new local trust of the owner will be:

$$T'_M(N) = T_M(N) + \rho \cdot \lambda_i \tag{10}$$

where ρ is the user defined parameter to control the weight of the current feedback. We use different formulas to update the local trust when feedback is positive and negative because we believe the reputation should be hard to build up, but easy to tear down [15]. The local trust value increases gradually to 1 when feedback is positive whereas it decreases linearly for the negative feedback. This is another effective countermeasure for the On-off Attack in our model, as it takes long-time interaction and consistent good behavior for a node to build up a good local trust in other nodes or the global reputation but only a few bad actions will ruin them.

V. Anti-collusion Mechanism

A. Collusion Attack

Collusion Attack can severely impair the security and quality of information within a network. Collusion Attack is defined as a form of malicious behavior that involves two or more network entities collaborating with each other. Collusion attacks are indeed severe issues to trust model and can affect system performance. In this paper, we categorize the Collusion Attack into two different forms:

COLLUSION 1: *Two or more colluding nodes constantly send information to each other and then always give positive feedback regardless of what information they actually receive.*

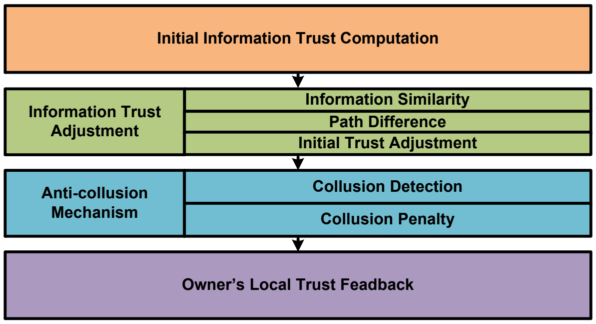


Fig. 4. Block diagram of the complete trust model

COLLUSION 2: *Two or more colluding nodes send similar false information items to the same end user through different processing paths so that the end user will think these information items are more trustworthy than they actually are. Therefore, colluding nodes can also control the feedback they get by adjusting the similarities of their information items.*

COLLUSION 1 is implicitly taken care by our CRDT framework. Even when the colluding nodes always give positive feedback to each other, they can only increase their own copies of local trust values for each other. When these local trust values are voted to the Central Reputation Manager, they will have very minimal influence on the reputation values if most of the nodes are "good". However, COLLUSION 2 is more complicated to deal with. CRDT does not help on overcoming this kind of Collusion Attack. Therefore, in the rest of the paper, we will no longer consider COLLUSION 1 and limit our focus only to COLLUSION 2.

Now, let us describe the colluding scenario more precisely by considering the network topology in Fig. 1. Suppose S_1, S_2 and S_3 are colluding nodes and they all send similar false information items to the end user. In order to attack the end user, these information items must be describing the same event and thus will be grouped in the same collection. The end user will now assign higher similarity factor to these information items than they should actually get. Besides, we consider this as an attack only when the intermediate nodes "pass" but do not process these information items, otherwise the intermediate nodes would become the owners when the information items reach the end user, in which case the information would have got evaluated and modified by these new owners.

We propose an *Anti-collusion Mechanism* (ACM) based on detecting information items that are possibly involved in collusions and consequently give penalty to the adjusted trust of these information items. As illustrated in Fig 4, our Anti-collusion Mechanism can be inserted between the second and third steps in our Primary Trust Model, so that the feedback given to the owner's local trust will take the collusion penalty into account as well. We will present the Anti-collusion Mechanism in the rest of this section.

B. Detecting Collusion Attack

Detection of Collusion Attack is not simple, especially when there are a large number of information items conflicting or supporting each other. To detect Collusion Attacks, we only consider the case that the end user receives number of information items in a collection, among which the number of true information items is more than the number of false information items. Otherwise, suppose an end user receives a collection of three information items within which two are similar false information items sent by colluding nodes, then the end user has no way to detect this Collusion Attack. In other words, the decision made by the system should agree with the majority opinion [17]. Therefore, we adopt the

Majority Rule described as follows.

Majority Rule: *Consider a collection of information items that describe a network event, the information item i with the highest $\Delta_i \cdot \Theta_i$ (product of similarity factor and path different factor) value is always considered as the most trustworthy item. We call this item the Reference Trustworthy Information Item and denote it as i_r.*

To detect a possible Collusion Attack, we introduce two threshold values, δ_r and δ_c, where δ_r is the maximum similarity value between a collusion item and reference information item i_r and δ_c is the minimum similarity between any two collusion items. Then, we formalize the procedure of detecting collusion information items as Algorithm 1.

Algorithm 1 Collusion Detection

```
$C \leftarrow$ collection of information items
$i_r \leftarrow$ Reference Trustworthy Information Item in $C$
$Col \leftarrow \emptyset$
for all information item $i \in C$ and $i \notin Col$ do
    if $S(i, i_r) \leqslant \delta_r$ then
        $Col_i \leftarrow \emptyset$
        Make $Col_i$ a subset of $Col$
        for all information item $j \in C$ and $j \neq i$ and $j \notin Col$ do
            if $S(j, i_r) \leqslant \delta_r$ and $S(j, i) \geqslant \delta_c$ then
                Add $j$ into $Col_i$
            end if
        end for
        if $Col_i \neq \emptyset$ then
            Add $i$ into $Col_i$
        else
            Remove $Col_i$ from $Col$
        end if
    end if
end for
return $Col$
```

C. Collusion Penalty

After detecting the collusion information items, the next step is to decrease the adjusted trust of these items before we give feedback to their owners. The degree of penalty we give should be proportional to the possibility of accurate detection instead of being a fixed amount [17]. In the following, we explain three rules we apply to adjust the level of collusion penalty based on the possibility of accurate detection.

Collusion Penalty Rule 1: *The larger number of information items detected in a colluding set (Col_i), the smaller penalty should be given to the trust of these information items.* Due to the randomness in the network, when we detect a colluding information set, it is possible that there are actually good nodes among the corresponding owners, but the information items they sent happen to be very similar to the other colluding items and thus are detected as colluding items too. The intuition behind this rule is that the more network nodes are detected as colluding, i.e., the larger detected colluding size, the more possible that the above kind of false positives exist in our detection result. Therefore, we give smaller penalty to a detected collusion that involves a larger number of nodes.

Collusion Penalty Rule 2: *The higher path difference an information item has within a detected colluding set, the higher penalty should be given to the trust of this information item.* The intuition behind this rule is that the more similar the processing paths of two information items, the more possible that these information items are similar even though the two owners are not colluding. Hence the collusion detection based on information similarity may not be sharp and accurate in this scenario. Therefore, we give smaller penalty to an information item with a smaller path difference factor θ within the detected colluding set. On the other hand, when the path difference is large, and still the information items are similar and detected as colluding, then our algorithm will have more confidence about the detection result. Therefore we give larger penalty.

Collusion Penalty Rule 3: *The more number of times a detected information item's owner has been involved in a detected collusion for a recent short period of time, the higher penalty should be given to the trust of this information item.* The intuition behind this rule is that if a node is not a colluding node, then there is low probability that we detect it as a colluding node many times within a short period of time.

Based on the above three collusion penalty rules, we use the following equation to adjust the trust of the detected collusion information items:

$$T_a^*(i) = T_a(i) \cdot (1 - e^{-\frac{|Col_i|}{\Theta_i^{Col_i} \bullet K_{O_i}}}) \qquad (11)$$

where i is an information item that has been detected in a colluding set Col_i, $T_a^*(i)$ is the final adjusted trust value after imposing the collusion penalty, $|Col_i|$ is the size of the detected colluding set, $\Theta_i^{Col_i}$is the path difference factor within the detected colluding set Col_i which we can calculate by using Eq. (6), and K_{O_i} is the number of times that i's owner node has been detected as a colluding node for a recent time interval T_K which in turn is a parameter that can be set by the administrator.

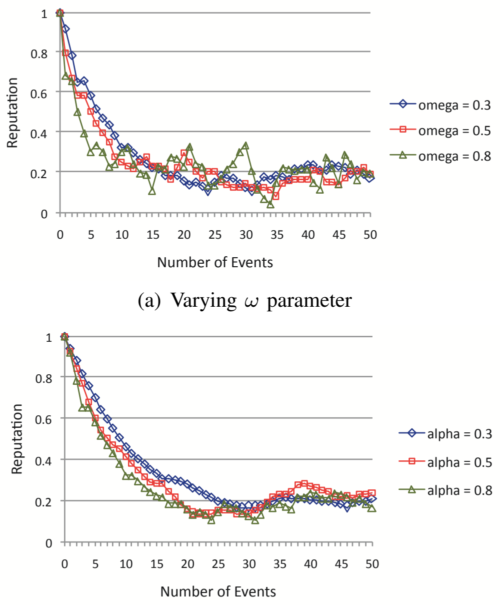
VI. Simulations and Analysis

In this section, we evaluate the effectiveness and efficiency of our Primary Trust Model and Anti-collusion Mechanism.

A. Experimental Settings

In our experiments, we conduct performance evaluation by generating a series of network events, with one event after another. For each network event, we have a certain number of source nodes in the network generate information items and send them to other nodes. Information trust evaluation and owner's trust feedback will be done after receiving all the information items for each network event. After each round of trust evaluation process, the central reputation manager takes votes from the network nodes to update its global reputation table. Therefore, the time in our experiments is in terms of number of network events instead of any actual time units like minutes or seconds, because our interest is to see how our trust model responds to malicious nodes with reference to the number events happen in the network. We assign each node a *default value*, which has a range of [0, 1]. This default value represents the probability that the node sends "good" information. That is, a node with default value of 0.8 sends 8 "good" information out of 10 on average. The default value is used as a reference of the ideal global reputation value that a node should get. We use it as a metric to evaluate how close the measured global reputation of a node actually gets from its default value.

Our simulations are divided into two settings. In the first setting, we focus on testing the effectiveness of our Primary Trust Model, i.e., convergence and accuracy of the trust model in identifying a malicious node. Here we are not considering Collusion Attack, so each malicious node works independently. The network is assumed to have N number of nodes and we vary N to find its impact on the effectiveness and also see the scalability of our trust model. Among these N nodes, we have 10% of the nodes as malicious nodes and set their default values to 0.2, that is, only 20% of the time the malicious nodes will send "good" information. For each network event, each of the N nodes will randomly pick $\frac{N}{2}$ other nodes as their information destination. In this setting, we



(a) Varying ω parameter

(b) Varying α parameter

Fig. 5. Global reputation of a particular compromised node versus number of events for different ω and α parameters

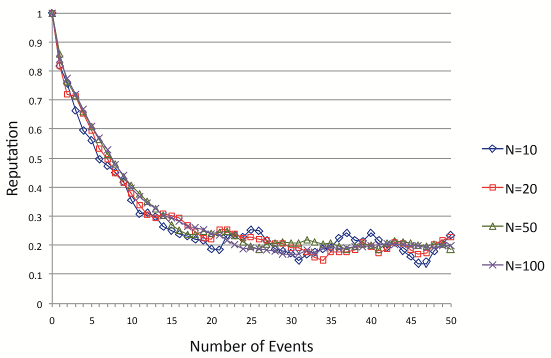


Fig. 6. Global reputation of a particular compromised node versus number of events for different number of nodes

vary the values of parameters α and ω to test their influence on the evaluation accuracy.

In the second setting, we focus on testing the effectiveness of our Anti-collusion Mechanism. We have 100 sender nodes and 1 receiver node. Among the 100 senders, we vary the number of colluding nodes C (less than 50% of all the nodes). All the colluding nodes have a default value of 0, which means they always send false information, but their information for each event will be the same in order to gain the highest similarity score. In this setting, we have $\alpha = 0.8$, $\omega = 0.3$, $\delta_r = 0.2$ and $\delta_c = 0.8$.

In both of the two settings, we set the initial global reputation of all the malicious nodes to be 1.0 so that when the receiver(s) receive information from a malicious node for the first time, it will set its local trust of the malicious node to be 1.0. In this way, we can test how fast our Primary Trust Model and Anti-collusion Mechanism respond to the nodes which suddenly become compromised. We set the feedback weight parameter ρ to be 0.5.

B. *Effectiveness of Trust Evaluation*

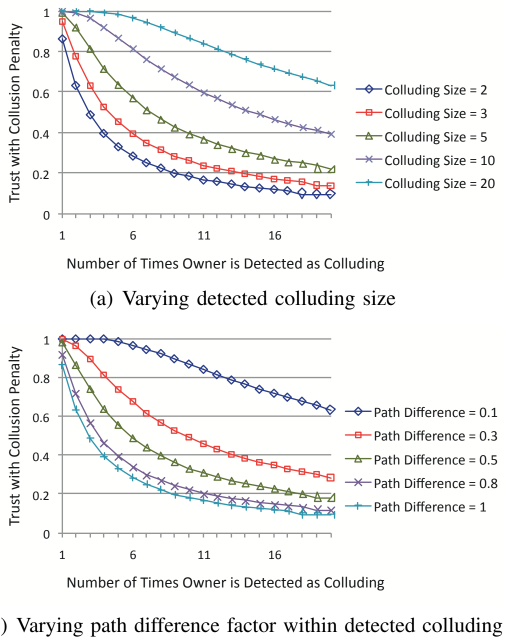
In order to test the effectiveness of our trust model, instead of looking at any particular local trust values, we test how the global reputation values of the malicious nodes change with the number of events occur in the network. Although we have multiple malicious nodes in our first experimental setting, each malicious node is working independently, therefore we only look at one malicious node. The rest of the malicious nodes in the network will get similar results.

First of all, we want to test how the user-defined ω and α parameters affect the trust evaluation. Fig. 5(a) shows the results when we have 10 nodes in total and fixed α value of 0.8 while varying the ω as 0.3, 0.5 and 0.8. According to Eq. (10), the malicious node's local trust should get a linear decrease for each negative feedback. That's why we observe that all the three curves drop very fast at the beginning until 10 to 15 events in the network. After that, all three curves become relatively stabilized with a mean value close to 0.2 since we assigned a default value of 0.2 to the malicious node. These results actually show that our trust model can accurately evaluate the reputation of network nodes after the initial catch-up period. By comparing the three curves, another observation is, with larger ω value, the curve drops faster, which is expected because with larger ω, the similarity factor and path difference factor for the latest network event have higher influence on the adjusted information trust which in turn reflects on the resulting reputation. However, when the ω is larger, the curve fluctuates more. This is also expected because of the random phenomenon of malicious nodes sending "good" or "bad" information. With larger ω, recent events have higher influence on the resulting reputation, and thus the randomness gets amplified. Fig. 5(b) shows the results when we have 10 nodes in total and fixed ω value of 0.3 while varying the α values as 0.3, 0.5 and 0.8. Again, we observe that all the three curves drop steeply at the beginning and then become relatively stabilized with a mean value close to 0.2. Similar to Fig. 5(a), when α value is larger, the curve drops faster, but fluctuates more than the case when α is smaller. Similar to ω, the higher α we set, the higher influence the recent events will have on the resulting reputation. Therefore, we can explain this observation by the same reason above.

Fig. 6 shows how the total number of nodes N affects the evaluation results. Here we set ω = 0.3 and α = 0.8 and vary the values of N. The four curves correspond to cases when N = 10, 20, 50 and 100 respectively. Since we still have all the nodes randomly choose $\frac{N}{2}$ number of nodes as their information destination, the average number of information items received for every event by each node remains the same. In addition, since we have the same percentage of malicious nodes for different N and we have fixed ω and α values, we can observe that the three curves drop at almost the same speed and finally reach about the same level. However, more number of nodes means more voters available for each round of voting process. Thus, when there are more number of nodes in the network, we can see the corresponding curve is more smooth and accurate. Hence, we can claim that our trust model is scalable. In fact, by having more nodes in the system, we get more accurate and stable evaluations.

C. *Collusion Handling*

Before we show the results of our second experiment setting, we want to analyze the collusion penalty rules in Eq. (11). Fig. 7 shows how various (a) detected colluding sizes and (b) path difference factors affect the penalty level as the number of times that a particular information owner has been detected as colluding in the past time interval T_K increases. Assume the information trust is always 1 without collusion penalty, the curves in this figure give us an idea about how much penalty is given to the information trust. When the detected colluding size is larger, there is higher probability that we have false positives in the detection results. As shown in Fig. 7(a), the penalty given is smaller in this case. Therefore,



(a) Varying detected colluding size

(b) Varying path difference factor within detected colluding set

Fig. 7. Impact of penalty rules on the trust of a colluding information item

if good nodes are detected as colluding nodes, they will not get much penalty. But as the number of times that a particular node has been detected as colluding increases, we have higher confidence that this node is indeed a colluding node, therefore it gets higher penalty based on Eq. (11). Similarly, when two or more detected colluding information items have smaller path difference, there is higher probability of false positives, and therefore penalty given to these information items is smaller. Again, the penalty gets increased as the number of times that the corresponding owner node has been detected as colluding increases. Hence, although our collusion detection algorithm is not 100% accurate, the Anti-collusion Mechanism can adjust the penalty level adaptively according to the confidence about the detection results.

In the second experimental setting, we compare the effectiveness of our trust model with and without the Anti-collusion Mechanism. We also test how the number of colluding nodes affects the evaluation accuracy. We have total 100 information senders, and we set the number of colluding nodes C = 20, 30 and 40 in order to keep the majority of them good nodes. Assuming that the initial reputation of all nodes are 1, we focus on how the final adjusted trust of information items sent by a colluding node changes with the number of events, and what the final adjusted trust value is when it reaches a steady level. From the simulation results shown in Fig 8, we can see that our trust model can decrease adjusted trust of information items received from a colluding node as the number of events in the network increases. Comparing the curves with or without the Anti-collusion Mechanism, we observe that the more number of colluding nodes we have, the slower the curve drops. This can be explained by the reason that the more number of colluding nodes, the higher similarity factor the collusion information items can get. Thus, the local trust values of the colluding nodes decrease slower. Comparing any two curves with the same C value, we can see that when we are implementing the Anti-collusion Mechanism, the curve drops much faster. More importantly, all the colluding nodes are always sending false information which should get a trust value of 0. However, without the Anti-collusion Mechanism, when the adjusted trust of information items received from a colluding node becomes stabilized, the receiver gets a value of about 0.25, 0.38 and 0.45 corresponding to C = 20, 30 and 40 respectively. We can see that incase of Collusion Attack, the accuracy of our algorithm can be increased using the proposed Anti-collusion Mechanism, in which case all three curves drop to a level very close to 0.

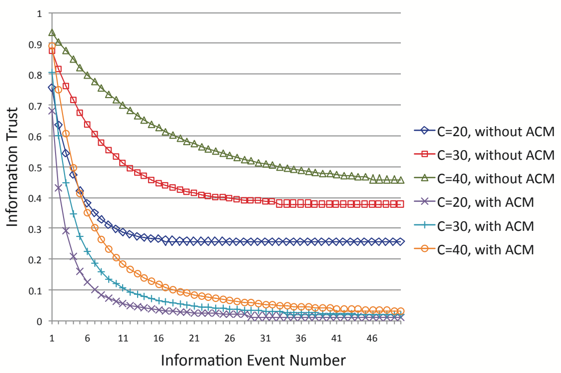


Fig. 8. Final adjusted trust versus event number for information from a colluding node

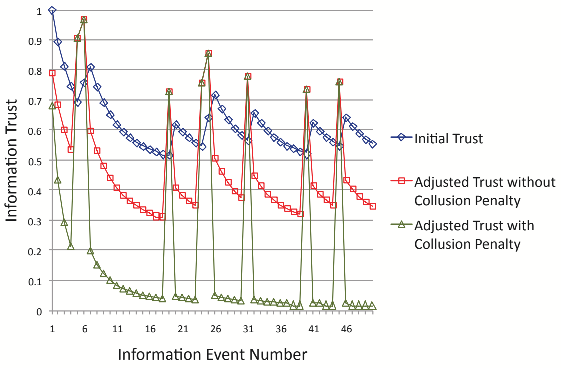


Fig. 9. Comparison of information trust for which the owner is a colluding node with default value of 0.2

Next, for the same experimental setting, we set $C = 20$ and let one malicious node (the "subject" node) have a default value of 0.2, which means there is 20% probability that the subject node's information item is "good" and otherwise it will be a "bad" colluding item. Again, we let the subject node have an initial trust of 1.0. We want to examine and compare the initial trust and adjusted trust (with and without collusion penalty) of the information items owned by the subject node as the number of events increases. The result is shown in Fig. 9. We can see that among the 50 network events, 8 events cause 6 spikes on the curves with 2 spikes each includes two events. These are the events for which the subject node sends "good" information because of its default value of 0.2. We observe that our trust model can capture these "good" information items successfully and make correct adjustments on the information trust, i.e., decrease the trust for "bad" information items but increase the trust for "good" information items. Also, for each "bad" information item, collusion penalty is given as they are also detected as colluding items. However, for "good" information items, no collusion penalty is applied. Moreover, it is worth noting that the initial trust always increases after "good" information items. This is expected because a "good" information item increases the local trust of the subject node and thus the following information item gets a higher initial trust based on Eq. (3).

VII. Limitations and Future Work

The most apparent limitation of our scheme is the communication overhead associated with the provenance meta-data, because each information item needs to have its own provenance meta-data. The size of the provenance meta-data will be minimal when information items only go through very small number of hops. However, it will increase dramatically if information items goes through many hops. Study has show that sensor networks can be modeled as a small-world network [25], [26], i.e., no matter how large the network is, there is a relatively short path between any two nodes. In real life sensor or tactical network scenarios, it is very unlikely that data go through too many hops before reaching its final destination. Hence, we believe that our scheme will not have tremendous overhead in real life applications. In our upcoming research, we will carry out an in-depth study of the communication overhead of our scheme through analysis and simulations. Moreover, the amount of communication overhead also depends on the level of provenance details carried in the information meta-data. We will investigate the trade-off between the effectiveness of information quality evaluation and the level of provenance details. We will also look into the possibility to have incomplete provenance in the meta-data in order to reduce the communication overhead as well as for privacy and confidentiality concerns.

Furthermore, our collusion detection is based on the Majority Rule which assumes that the number of malicious nodes is less than the number of honest nodes. This is reasonable for a large network but it may not hold in a small local neighborhood. Therefore, our Anti-collusion Mechanism requires the end user nodes to gather information from as many paths as possible to effectively reduce the probability that the number of colluding information items exceeds the number of good information items for a certain network event.

Though we have evaluated our scheme by extensive analysis and simulations in this paper, as our future work, we will also try to assess the performance of our scheme by applying it in a real distributed and dynamic environment.

VIII. Conclusion

We have proposed an information trust computation strategy based on information provenance. Our model can capture the trustworthiness of information flowing in the network as well as dynamically adjust the local trust and global reputation of the network nodes. In addition to considering common attack types for reputation systems like Bad-mouthing Attack, On-off Attack and Conflict Behavior Attack, we have focused on Collusion Attack and proposed a mechanism to detect and penalize the colluding information items. From our simulation results, we can observe that the information and node trust values obtained closely follow our expectation, and our Anti-collusion Mechanism also makes the evaluation more accurate when there are collusions. Our approach is generic and does not get influenced by mobility or any other network dynamics as long as the transmitted information meta-data contains proper provenance details.

Acknowledgment

This work was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official polices, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy right notation here on.

References

[1] L. Gomez, A. Laube, and A. Sorniotti, "Trustworthiness assessment of wireless sensor data for business applications," in *AINA '09: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications*, pp. 355–362, 2009.

[2] Y. Sun, Z. Han, W. Yu, and K. Liu, "Attacks on trust evaluation in distributed networks," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, pp. 1461–1466, 2006.

[3] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys*, vol. 42, no. 1, pp. 1–31, 2009.

[4] A. Boukerch, L. Xu, and K. EL-Khatib, "Trust-based security for wireless ad hoc and sensor networks," *Computer Communications*, no. 30, pp. 2413–2427, 2007.

[5] M. J. Probst and S. K. Kasera, "Statistical trust establishment in wireless sensor networks," in *Proceedings of the 13th International Conference on Parallel and Distributed Systems*, pp. 1–8, 2007.

[6] C. Papageorgiou, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, "Dynamic trust establishment in emergency ad hoc networks," in *Proceedings of the 2009 International Conference On Communications And Mobile Computing*, pp. 26–30, 2009.

[7] L. Capra and M. Musolesi, "Autonomic trust prediction for pervasive systems," in *20th International Conference on Advanced Information Networking and Applications, 2006. AINA 2006*, vol. 2, 2006.

[8] F. Skopik, D. Schall, and S. Dustdar, "Start trusting strangers? Bootstrapping and prediction of trust," in *WISE '09: The 10th International Conference on Web Information Systems Engineering*, pp. 275–289, Springer, 2009.

[9] S. Reidt and S. Wolthusen, "Efficient distribution of trust authority functions in tactical networks," in *Proceedings of the 2007 Information Assurance and Security Workshop*, pp. 84–91, 2007.

[10] H. Chen, H. Wu, X. Cao, and C. Gao, "Trust propagation and aggregation in wireless sensor networks," in *FCST '07: Japan-China Joint Workshop on Frontier of Computer Science and Technology*, pp. 13–20, 2007.

[11] Y. Bachrach, A. Parnes, A. Procaccia, and J. Rosenschein, "Gossip-based aggregation of trust in decentralized reputation systems," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 2, pp. 153–172, 2009.

[12] P. Buneman, S. Khanna, and T. Wang-Chiew, "Why and where: A characterization of data provenance," in *ICDT '01: The 8th International Conference on Database Theory*, pp. 316–330, Springer, 2001.

[13] Y. L. Simmhan, B. Plale, D. Gannon, and S. Marru, "Performance evaluation of the Karma provenance framework for scientific workflows," in *Lecture Notes in Computer Science: Provenance and Annotation of Data*, vol. 4145, pp. 222–236, Springer Berlin / Heidelberg, 2006.

[14] P. Buneman and W. Tan, "Provenance in databases," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 1171–1173, 2007.

[15] B. Yu, S. Kallurkar, and R. Flo, "A Demspter-Shafer approach to provenance-aware trust assessment," in *CTS '08: International Symposium on Collaborative Technologies and Systems*, pp. 383–390, May 2008.

[16] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, "An approach to evaluate data trustworthiness based on data provenance," in *SDM '08: Proceedings of the 5th VLDB workshop on Secure Data Management*, pp. 82–98, Springer-Verlag, 2008.

[17] C. Dai, H.-S. Lim, E. Bertino, and Y.-S. Moon, "Assessing the trustworthiness of location data based on provenance," in *GIS '09: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 276–285, 2009.

[18] X. Wang, K. Govindan, and P. Mohapatra, "Provenance-based information trustworthiness evaluation in multi-hop networks," in *GLOBECOM '10: IEEE Global Telecommunications Conference*, 2010.

[19] A. Moitra, B. Barnett, A. Crapo, and S. Dill, "Data provenance architecture to support information assurance in a multi-level secure environment," in *MILCOM '09: IEEE Military Communications Conference*, pp. 1–7, 2009.

[20] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, pp. 259–268, ACM, 2004.

[21] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 267–278, ACM, 2006.

[22] B. Levine, C. Shields, and N. Margolin, "A survey of solutions to the sybil attack," *University of Massachusetts Amherst, Amherst, MA*, 2006.

[23] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *VLDB '98: Proceedings of the 24th International Conference on Very Large Data Bases*, pp. 194–205, 1998.

[24] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *In Proceedings of 2008 SIAM Data Mining Conference*, 2008.

[25] G. Sharma and R. Mazumdar, "Hybrid sensor networks: a small world," in *MobiHoc '05: The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, vol. 4, pp. 366–377, 2005.

[26] S. Chinnappen-Rimer and G. Hancke, "Modelling a wireless sensor network as a small world network," in *WNIS '09: International Conference on Wireless Networks and Information Systems*, pp. 7–10, IEEE, 2009.