PRONET: Network Trust Assessment Based on Incomplete Provenance

Kannan Govindan*, Xinlei Wang*, Maifi Khan†, Gulustan Dogan§, Kai Zeng*
Gerald M Powell‡, Ted Brown§, Tarekh Abdelzaher†, Prasant Mohapatra*
*University of California Davis, Email: {gkannan, oscar, kzeng, prasant}@cs.ucdavis.edu,
†University of Illinois Urbana Champaign, Email: {maifi, zaher}@cs.uiuc.edu,
§City University of New York, Email: {gulustan, TBrown}@gc.cuny.edu
‡Army Research Lab, Email: gerald.m.powell@us.army.mi

***Abstract*—This paper presents a tool ProNet, that is used to obtain the network trust based on incomplete provenance. We consider a multihop scenario where a set of source nodes observe an event and disseminate their observations as an information item through a multihop path to the command center. Nodes are assumed to embed their provenance details on the information content. Received provenance may not be complete at the command center due to attackers dropping provenance or the unavailability of provenance. We design ProNet, a tool which is at the command center that acts on the received information item to determine the information trust, node-level trust and sequence-level trust. ProNet contains three steps. In the first step it reconstructs the complete provenance details of received information from the available provenance. In the second step it employs a data classification scheme to classify the data into a good and bad pool. In the third step it employs pattern mining on the reconstructed provenance of bad data pools to determine the frequently appearing node and node sequence. This frequent appearance will quantify the trust level of nodes and node sequence. Now an information quality/trust level of newly received information can be determined based on the occurrences of these node/sequence patterns on the provenance data. We provide a detailed analysis on false positive and false negatives.**

I. Introduction

In multihop information sharing networks it is often necessary to perform trust analysis on the nodes and also the information received to derive a wise decision out of the received information. Cryptography security services such as authentication and admission control alone cannot provide complete solutions for a trust analysis. Nodes can misbehave and may provide unreliable observations/results after passing through the initial cryptography check. Though they are authenticated nodes, the poor results may be due to single or combination of various factors such as misbehavior, faulty sensors and environmental factors. In these circumstances, trust management augmented with cryptography can provide a viable solution. One way to assess trust is by using the history of the information origin, widely known as provenance. In this paper we assume provenance as node ID, location and time of observation.

In ad-hoc networks, to be particular, tactical networks, if informers attach their provenance details on the information they provide later, they can query the command center and get rewarded accordingly based on the importance of the information they have provided [1]. In addition, the receiver can have more confidence on the received information if the provenance information is available. Hence provenance has become important entity in the modern day information sharing networks. It helps both the information providers and also the receiver. Complete provenance means the full identification details of a node that generated it and also processed the information. With complete provenance one can say without any ambiguity what nodes participated in the information gathering and processing. Often the literature on provenance-based trust assessment assumes whatever node processes the information will embed its provenance on the information [2], [3]. However, in a dynamic open networks often the full provenance collection is complex, and provenance may not be complete and accurate [4]. This is basically due to:

- Attackers: When the information passes through a multi-hop chain, some attackers may drop the previous nodes' provenance details to claim the origin of information as their own to get monetary reward if any. Attackers may refuse to attach their own provenance so that they can get away from the malicious node detection while at the same time create considerable damage in the decision making. Sometimes users may drop part of the large provenance data to avoid network congestion.
- Unavailability and unwillingness: In some cases provenance details will not be simply available. Possibly due to malfunctioning of the provenance information provider on the node. In some cases users are not willing to provide complete provenance for various confidentiality reasons; users may intentionally hide their part of the provenance.
- Loss of provenance: In some cases provenance data may be lost. Therefore instead of full ID, the received provenance may have partial ID.

In this paper we propose a strategy to assess the trustworthiness of the information based on available provenance. We do not back query the users for additional information as it can lead to confidentiality violations. We just work on the received information. We assume a mutli-hop network where every node observes the event and disseminates the information through the nodes until it reaches the command center node. The command center applies the proposed algorithm on the received information and determines the trust level of nodes and also the received information based on the available provenance details. The objective of this work is to identify a single node or pattern of less trustworthy nodes in the provenance chain and declare that particular information as a member of a bad set. We generate a rule to evaluate the trust of the information based on the history of appearance of one or sequence of nodes in either the good or bad data pools. Our proposed ProNet tool contains three steps: In the first step the complete provenance from the available limited provenance details is reconstructed. In the second step the information set is classified into two pools based on a classification technique. In the final step. pattern mining is applied to identify the sequence of nodes appearing on the bad information pool. This will help to identify the trustworthiness of future received information.

Uniqueness of the proposed approach:

1) We provide an approach to evaluate the trustworthiness even when the complete provenance is unavailable. We provide a balance between the false positive analysis and the granularity of the provenance available.
2) Instead of just the node-level trust we also concentrate on sequence-level trust analysis. Sometimes the node individually may be good but the association of node (sequence) with other nodes may be malicious or less trustworthy. We identity those patterns.
3) We work at the information level (application layer) rather than the packet level (network layer) and provide detailed performance analysis.

II. Related work

An agent-based approach to manage the trustworthiness of information in a dynamic information sharing environment is presented in [5]. Here a provenance graph of a derived information is used for the trust assessments. Information trust assessment based on path and information similarity is proposed in [6]. The idea is that when the information item is received from totally disjoint paths and the information contents are similar, then it is highly likely that the information is trustworthy and also nodes which processed the information are trustworthy. A data provenance trust model which estimates the level of trustworthiness of both information and information providers is presented in [7]. Four aspects that affect the trustworthiness of the data have been taken into account to build such a trust model, which are (a) data similarity, (b) path similarity, (c) data conflict and (d) data deduction. However, this model is vulnerable to collusion attacks [2]. Majority rule based technique to detect the malicious colluding parties is proposed in [2], [8]. Our approach considerably differs from all of the above in the sense that we handle incomplete provenance.

III. Fundamental framework

A. System model

We assume a single command center, which is an information processing unit and set of nodes deployed for event monitoring. Every node which is close to the event makes observations about the event and sends the observation as information item to the command center through multiple intermediate nodes as shown in Fig 1. Information either passes through the intermediate nodes without change or get processed and passed on. The processing can be either fusion or aggregation. Here we consider information in the form of information items. Each information item has one or multiple owners. Each information item consists of information metadata and an information payload as shown in Fig. 2. The metadata contains the provenance of the information item provided by the user. For instance, in Fig. 1 the information i is outcome of actions of node D, and node A and also the combinatorial interactive trust of node A and D. The provenance in turn includes the information item's creation time, owner, location history, as well as the provenance of the all other intermediate nodes which performed operations on the information item. The receiver collects all the information. Before we proceed further let us make the following formal definitions:

DEFINITION 1: **Trustworthiness of Information Items (Trust).** The trustworthiness of an information item i, denoted as $T(i)$, is the probability of i being true.

DEFINITION 2: **Trust of Nodes.** The trust of a node N, is the probability that N sends correct information.

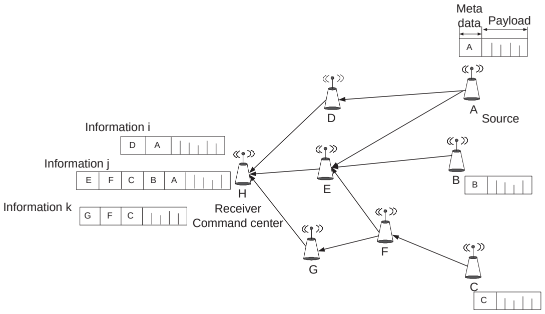


Fig. 1. Illustration of information dissemination

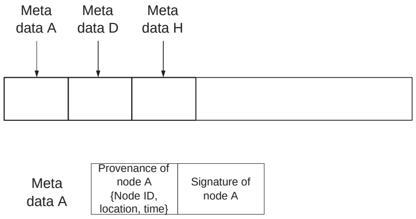


Fig. 2. Information packet

DEFINITION 3: **Provenance.** The provenance of the information contains two factors: the ownership of an information, and the log of the tasks applied on the information by authorized entities. The three most basic entities of a provenance graph are Agents, Processes and Artifacts. In a multi-hop sensor or tactical network, we can call these entities as: *Node ID*, *Actions*, *Attributes*. We consider the two most basic types of action, "to pass" or "to process", as the Root Action. Attributes contains time of operation and location of node.

B. Attack model

We assume that every node should sign the metadata with its private key when it attaches the metadata. When another node receives the information it can read the metadata and find out which node has sent the information from the node ID which is in the metadata. Now the other node can verify the integrity of the metadata by verifying the signature using the corresponding public key of the previous node. Here the node cannot alter the metadata of the previous node because if it alters the metadata it has to sign the metadata with its private key. Hence integrity of the provenance can be ensured. In addition, the nodes can never forge or fake provenance. Due to the cryptography if any other nodes change the provenance in an unauthorized way, or if any nodes provide a fake ID within their own provenance, it can be detected by the recipient.

After considering this, we list the following attack models:

- The attacker can drop completely the metadata of all the previous users or some selective previous users as shown in Fig. 3. For instance, the node last in the provenance chain can completely drop all the provenance information in the chain and can place its own metadata to claim the full credits for the source of information.
- The attacker can refuse to attach its own provenance after performing operations on the information payload. So that the attackers can get away from detection at the same time it can cause damage in the decision making.

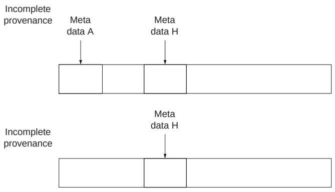
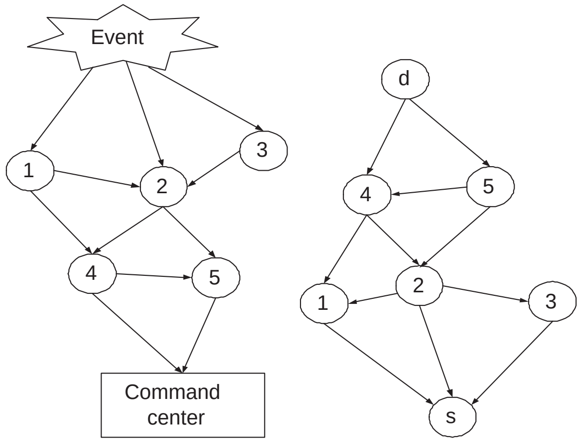


Fig. 3. Illustration of incomplete provenance



(a) Original graph

(b) Directed bottom up graph

Fig. 4. Pictorial representation of the source to destination connected graph

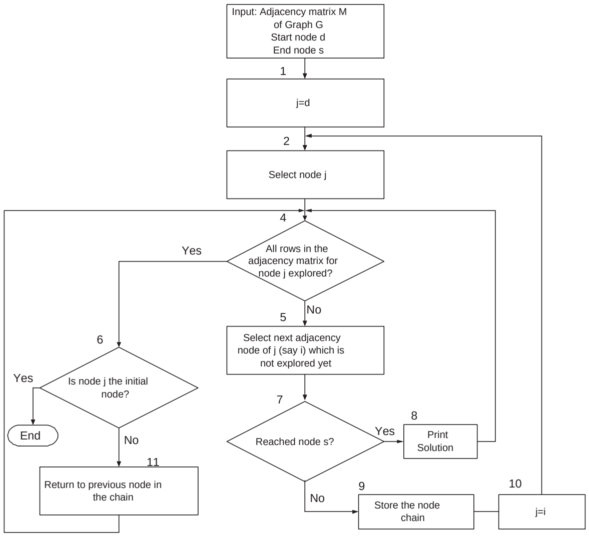

Input: Adjacency matrix M of Graph G Start node d End node s
1
j=d
2
Select node j
4
All rows in the adjacency matrix for node j explored?
Yes
No
5
Select next adjacency node of j (say i) which is not explored yet
6
Is node j the initial node?
Yes
End
No
11
Return to previous node in the chain
7
Reached node s?
Yes
8
Print Solution
No
9
Store the node chain
10
j=i

Fig. 5. Flow chart to determine All possible paths

IV. ProNet Implementation Details

The steps to be followed in our approach are explained below:

A. Provenance reconstruction

We profile the nodes' location and corresponding MAC address at the command center when we receive the information items. We first reconstruct the full provenance for nodes for which only partial provenance details are available. We assume there are enough messages that include the full provenance that this is possible. Then we reconstruct the complete provenance chain if any of the nodes provenance details are completely missing.

Reconstruction of partial provenance information:

1) If MAC address is partially available we look at the profile for the corresponding match with location, time and narrow down to particular address.
2) If location is missing, we see in the repository if the location of that particular node is previously reported. If not, we see who is the neighbor in the provenance chain. We look at the time of the report from the previous neighbor and next hop neighbor and then determine approximate location of the node by combining the neighbors location and time information.
3) If time is missing, we can apply filtering based on previously reported time and neighbor nodes' time.

If provenance of a particular node is not completely recoverable after these three steps then we ignore the partial details and declare that one node's provenance is completely missing and proceed to the next section where we do further investigation.

Reconstruction of provenance chain:

We follow two approaches for provenance chain reconstruction.

- Based on Path set constructions (PC).
- Based on Profiling Approach (PA).

TABLE I
Adjacency matrix

Node	Reachable set
d	4, 5
4	1, 2
5	4, 2
1	s
2	1, s, 3
3	s

1) Provenance reconstruction based on Path set Construction (PC): In this approach the command center first constructs all possible paths as follows:

We assume that the command center knows the topology of the network nodes, which is a directed acyclic graph (DAG). The topology can be learned through the location profiling of all the nodes in the initial stage. Initially we assume that the nodes behave genuinely and over the time they act selfishly and drop provenance packets. We don't assume any other prior information like the underlined routing protocol or the information traffic pattern, etc. The nodes are allowed to choose their own routing decision while disseminating the information packet.

In the first step the command center determines the event location by correlating the several provenance information paths. Now the command center follows the bottom-up reachbility set construction from the command center to the event location using the network topology information. This is a one time construction. The reachbility set is constructed based on [9]. The difference here is that we have DAG and hence the complexity will be half as compared to [9]. The modified algorithm is given in Fig 5.

We construct all possible paths from source to destination using the algorithm given in Fig 5. A general exhaustive search approach for all possible path construction would require an algorithm of $O(N^2)$ to calculate the all possible paths for

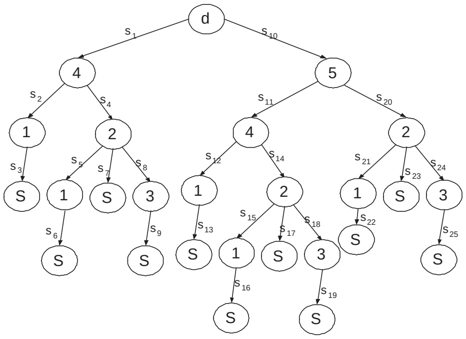


Fig. 6. Pictorial representation of the source to destination connected graph

a graph with N nodes. To reduce the computational cost, we exploit the fact that the tree-branch set of a particular node is a subset of the tree-branch of its parent node. The computational complexity here is $\frac{N \times K \times L}{2}$ where N is the number of nodes in the network, K is maximum number of adjacency and L is the depth of the graph.

We illustrate the path construction process using Fig. 4. a. In the original graph Fig. 4. a we replace the source with the node 's' and the destination with node 'd'. We replace all out going edges with the incoming edges and vice versa and construct the bottom-up graph as shown in Fig. 4. b. Then we construct the adjacency matrix as shown in Table. I by reading the bottom up graph. Here in the Table. I the index in the first column corresponds to a parent node and rest of the index corresponds to nodes reached by direct outgoing edges from a parent node.

This table will be used as input in the algorithm given in Fig. 5. The output of this algorithm are the all available paths from the source to the destination as shown in Fig. 6. Destination node 'd' is picked up as the first node in algorithm given in Fig. 5. One of the adjacency nodes of the destination in the bottom-up graph will be picked up next and then the process will continue until a source node is found. Once the source node is found, again the one level immediate parent will be picked up and one of the left out adjacency node will be explored until source node is reached. This way going backward and forward will eventually provide us all possible paths. Now the outgoing edges replaced with incoming edges will give all possible paths as given in Fig. 6. Here in Fig. 6 the number s_x on particular edge denotes that particular node is reached after running Algorithm for x steps. Now the final all possible paths will be obtained by converting all incoming edges in Fig 6 to outgoing edges.

Now the complete provenance of all the possible paths are given in Table. II.

TABLE II
ALL POSSIBLE PATHS FROM SOURCE TO DESTINATION

Path	Nodes
1	s, 1, 4, d
2	s, 1, 2, 4, d
3	s, 2, 4, d
…	…

However, if we don't know the node topology then we can use the profiling based approach as explained below.

2) Provenance reconstruction based on profiling approach (PA): In this model we store the provenance of a node over time. We assume in the initial stage during the deployment, every node behaves well and attaches the true provenance values. There is no attack or provenance drops in the initial stage. In addition there is no provenance loss and also the users are fully cooperating in embedding their provenance details. Hence the provenance is fully available in the learning phase. We then profile all the possible paths and create a profile by storing the provenance information. We call it as *profile database*. We keep on improving the profile database over the time. Now after the learning period, the network nodes will start to behave their own way and there could be provenance drops. The missing provenance details in the received information can be determined by comparing the received information metadata against the provenance chains of the profile database as explained in the following section.

3) String search to narrow down the full provenance chain: From Section IV-A1 and Section IV-A2 we can determine all complete possible provenance chains. The question is using the incomplete provenance available in the received information metadata, how to narrow down one candidate provenance out of all possible paths. For this, we do exhaustive string matching in the complete provenance pattern using Algorithm 1. We define two main decision factors that will help us to identify the complete provenance chain:

- Total length of common subsequences
- Order of common subsequences

We first start to look for the available incomplete provenance pattern among the set of complete provenance chain sets ($\Phi(n)$) constructed using either PA or PC methods. From this set we are able to narrow down to subset ($\Phi_1(n)$) of complete provenance chain that contains the available incomplete provenance pattern. From this subset we again narrow down to subset ($\Phi_2(n)$) that has the available incomplete provenance in the same particular sequence. Now we classify all the nodes in the ($\Phi_2(n)$) as possible candidate for the full provenance. This procedure is explained in Algorithm 1 where the sequence $X[.]$ is compared against the all possible path set $A[.]$. Though this approach may generate several possible paths, our trust assignment in Section IV-C itself, based on accumulating several instance reports and then applying frequent sequence mining, the final trust assignment will be optimal and reduce the false alarms. After the provenance reconstruction, the next step is to classify the information based on the payload as follows.

Algorithm 1 Algorithm to find the provenance sequence

```
$N \leftarrow size(X[.])$, $M \leftarrow size(A[.])$, $C \leftarrow 0$
for all $i$ such that $0 \leq i \leq N$ do
    for all $j$ such that $n \leq j \leq M$ do
        if $X[i] == A[j]$ then
            $C++$
            $n \leftarrow i$
        end if
    end for
end for
if $C == N$ then
    Add A[.] into candidate set
end if=0
```

B. Information classifications

Path difference and information similarity-based approaches are used to classify the information into bad and good pools [6], [8]. Each information item will be assigned an information similarity factor (ISF_I) and a path difference (PDF_I) factor with respect to the information collection they belong to based on [6], [8]. Then our classification result will be determined by the product of these two factors. A positive $ISF_I \cdot PDF_I$ means there is more support in the collection, so that we can put the information item into the “good” pool, and a negative $ISF_I \cdot PDF_I$ means there are more conflicts so we put it into the “bad” pool. The idea is if the two information items are similar and coming from entirely dissimilar paths then there is a high probability that these two information items belong to good pool.

C. *Pattern mining*

Once the information is classified into the two different pools, we apply frequency pattern mining into the divided information-sets to determine the frequently appearing nodes and node sequences.

Since there can be many nodes observing an event and many nodes in the pools, checking all the possible combinations in each information pool can be tedious. The algorithm also will not scale well if we search the entire pool for all possible combinations. Therefore, we propose to exploit the apriori property to narrow down the search space to enhance the scalability of the algorithm [10], [11]. The main idea of the algorithm is as follows: At the first step, we estimate the probability of occurrence of each individual feature. By feature we mean the appearance of single node or nodes pattern in a given pool. If a feature occurs lower than a predefined threshold (α) number of times, we remove that feature from further consideration as it has very limited discriminative power and is statistically insignificant. After that first step, we retain features that have high support. Let us assume that the set is S_1. At the next step, we first generate the set of the candidate features by taking the Cartesian product $S_1 X S_1$ where ‘X’ represents the Cartesian product. $S_1 X S_1$ represents all possible combinations of features occurring as pair. This Cartesian product is an algorithmic way to determine frequently appearing node pairs in the pool. Next, we estimate the probability of each item in $S_1 X S_1$ and remove the items that has support lower than α and generate a set S_2. Next, we generate the candidate patterns of length 3 as $S_2 X S_1$. We continue this process until we find all the combination of features or get the empty set.

This way we can find out the single node or pattern of nodes which are frequently appearing in the bad data pool. We name these nodes/patterns as bad signatures. We repeat this exercise for the good data pool and find out the frequently appearing patterns in the good data pool. Now we can generate a rule based on this frequent appearance. In the future received messages, if we see these frequently appearing bad/good node or node patterns, we can classify those data as bad/good data. We constantly keep update this data set and change the rule dynamically. If a particular node or node patterns appearing in both bad and good pool frequently then those nodes or node patterns will be assigned with unknown trust (score 0.5).

Trust calculation: The received information can be classified as trustworthy or untrustworthy as follows

- If all the node and node patterns in the newly received information provenance found frequently appearing in the good pool earlier then that particular information will be assigned high trust (score 1).
- In the newly received information provenance at least one node/node patterns was earlier found appearing only in the bad pool then that particular information will be assigned with low trust (score 0).
- If some of the nodes trust score is unknown and rest of the node/node sequence trust is high and if that particular information has high $ISF_I \cdot PDF_I$ then it will be assigned high trust. If it has low $ISF_I \cdot PDF_I$ then it will be assigned the score of 0.

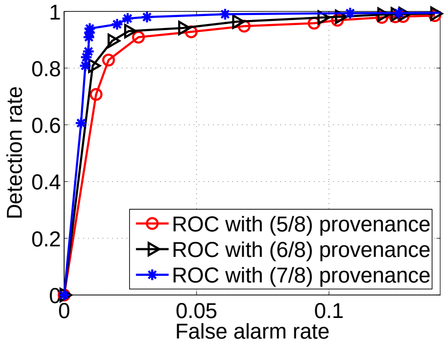


Fig. 7. ROC performance of path construction approach

Now these knowledge will be used to further fine tune the information classifications of Section. IV-B.

V. Results and Analysis

We have conducted extensive simulations. We assume 100 nodes distributed uniformly random over the space of 1000×1000 and centralised command center. For PC method, using the network topology we construct all possible paths and record them in the command center. Now we determined the source to destination shortest path and in the path we have dropped 1, 2, 3 node’s provenance information. We have compared the received incomplete provenance against each path in the record that has been stored previously and determine the path which is closely matching as per Algorithm 1. That particular path will be picked up as candidate path. The receiver operative characteristics (ROC) performance of the PC method is shown in Fig. 7. In ROC plot detection means correctly identifying the provenance and false alarm means wrongly identifying other nodes as rightly missing provenance. For the PA method we select a random source and a fixed command center destination and determine a path between them and record the path (the intermediate nodes). We repeat this 100 times and store all these paths in the record database and use it as a profile model. Now in the new path we intentionally drop 1, 2 and 3 number of node’s provenance in the chain and try to look for the missing pattern in the profile. We determine the detection rate and false positive rate this way and plot the ROC plot as shown in Fig 8. The ROC performance with the PC method is much better than the PA method. It is because in PA method there are chances that we miss to profile some of the paths. In addition, the training overhead in profiling based approach is high. We get reasonably good results only after storing 100 paths. In addition we can see that both methods generate more false alarms. This is because there are chances that we pick up more than one path as candidate paths. However, the detection rate is mostly high. To measure the trust level of nodes, we assume a node pattern of length 2 is malicious with trust factor 0.2. That means 80% of the time the information which contains the particular node pattern is untrustworthy. We determine the trust of the information which contains this particular node pattern in the chain. However that particular node/node pattern is completely dropped in the provenance chain of the information. Our provenance reconstruction was able to capture the missing provenance and also the trust of this particular information. The threshold value (α) in the pattern mining is chosen as 2% of the total information collection size. We have plotted ROC performance of PC and PA methods in Fig. 9. Here we compare the information against the

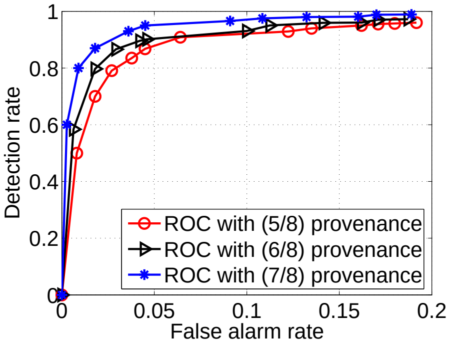


Fig. 8. ROC performance of PA approach

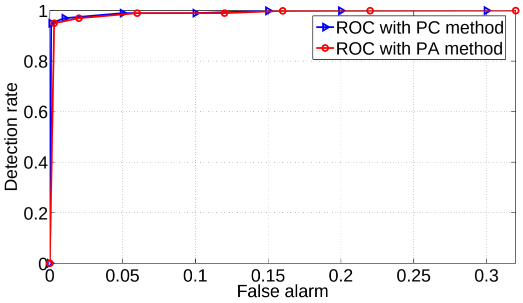


Fig. 9. ROC performance of PC method in information trust assessment

ground truth which we assume is available for the performance evaluation purpose. When we identify the information is trust worthy (trust score 1) and that particular information is close enough with ground truth (i.e., high ISF_I) then we assume that detection is achieved. If we declare the information is trustworthy and the information is totally opposite compared to ground truth (i.e., negative ISF_I) then we assume it is false negative. If we identify the information is untrustworthy (trust score 0) however it has high ISF_I with respect to the ground truth then we assume it is a false alarm. The false negative performance of our approach with PC method is shown in Fig 10. We can see that the false negative is almost 0. Similar performance was achieved for PA method also.

VI. Conclusion

We have proposed a scheme to determine the node and sequence level-trust based on the available incomplete provenance information. We have provided an analysis to give a balance between available provenance and the accuracy in the trust establishment. The analysis shows that one can get good performance even when part of provenance chain is missing. Performance with the complete topological information is better than that of the profiling based approach. Provenance based trust analysis is relatively new area and there are several practical questions remain to be answered. The provenance overhead analysis is an issue to be addressed. In some cases, the provenance could be many orders larger than the size of the information product. For example, in some cases provenance may include pictures or video to support a report. Distribution of this provenance via a multi-hop network to the pertinent authorities in a timely manner can be a challenge. In addition storage and timely retrieval of provenance add additional complexity. We hope that the near future research in this direction may answer some of the questions.

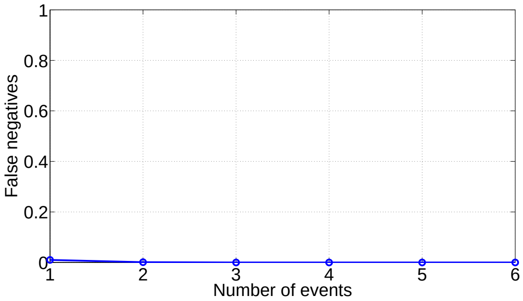


Fig. 10. False negatives versus number of events with PC method

Limitations: This approach has a few limitations. In the case of profiling based approach the initial training phase is crucial. A larger training phase would result in better performance. Our proposed trust method is more conservative and achieves better detection rates. At the same time it generates false alarms. The false alarm rate increases when the amount of available provenance information decreases.

Acknowledgment

This research 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] S. Xu, R. Sandhu, and E. Bertino, "TIUPAM: A Framework for Trustworthiness-centric Assured Information Sharing," in *The 3rd IFIP International Conference on Trust Management (TM'09)*, pp. 164–175, 2009.

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

[3] W. Zhou, E. Cronin, and B. T. Loo, "Provenance-aware secure networks," in *Proceedings of the International Conference on Data Engineering Workshops (ICDE)*, 2008.

[4] U. Braun , S. Garfinkel , D. A. Holl , K. Muniswamy-reddy , M. I. Seltzer, "Issues in automatic provenance collection," in *Proc. IPAW06*, 2006.

[5] B. Yu, S. Kallurkar, G. Vaidyanathan, and D. Steiner, "Managing the pedigree and quality of information in dynamic information sharing environments," in *The 6th ACM international joint conference on Autonomous agents and multiagent systems AAMAS '07*, pp. 1–3, 2007.

[6] X. Wang, K. Govindan and P. Mohapatra, "Provenance based information trustworthiness evaluation in multi-hop networks," in *IEEE Global Communication Conference, Globecom-10*, 2010.

[7] 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, 2008.

[8] X. Wang, K. Govindan and P. Mohapatra, "Collusion-resilient Quality of Information Evaluation Based on Information Provenance," in *IEEE SECON-11*, 2011.

[9] M. Migliore, V. Martorana, and F. Sciortino, "An Algorithm to Find All Paths between Two Nodes in a Graph," vol. 87, pp. 231–236, 1990.

[10] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the Twentieth International Conference on Very Large Data Bases (VLDB94)*, pp. 487–499, 1994.

[11] M. M. H. Khan, H. K. Le, H. Ahmadi, T. F. Abdelzaher, and J. Han, "Dustminer: troubleshooting interactive complexity bugs in sensor networks," in *SenSys 08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pp. 99–112, 2008.