Reputation based trust management using TCG in Mobile Ad-Hoc networks (RTA)

Segla Kpodjedo, Samuel Pierre, École Polytechnique, Montréal, Canada. {segla, samuel.pierre}@polymtl.ca
Makan Pourzandi, Software Research, Ericsson Canada, Montréal, Canada. makan.pourzandi@ericsson.com

Abstract—The Mobile Ad-Hoc Networks (MANET) are more and more important due to their increasing use. At the same time, the Trusted Computing Group (TCG) approach in using TPM based hardware root of trust is increasingly used in mobile devices providing a trustable source of knowledge about software composition of devices.

In this paper, we develop a new approach to evaluate trust among peers in an Ad Hoc network, based on the reputation of their software composition.


I. INTRODUCTION

With increasing use of Mobile Ad-Hoc Networks (MANET) the trust between peers has become important.

Our work concerns sensor networks, where the sensor behavior is determined by its software composition and configuration. An example of such networks is Vehicle-To-Vehicle (V2V) communications. In V2V networks, vehicles/cars use radio devices to inform other peers about road conditions or possible accidents ahead [1, 2, 3].

The current reputation models have been developed to evaluate human users’ behaviors. They are then not adapted to sensor networks where there is no human being involved, peers are highly mobile (there are high chances that a node would never meet the same peer twice in its life span) and traditional identification based on network address is very difficult as nodes can easily change network address. In this paper, we propose a new trust management schema in Ad-Hoc networks based on the reputation of peer’s software composition.

II. BASIC CONCEPTS OF RTA

A. Context

We consider an Ad-Hoc environment with large number (>1000s) of mobile nodes/sensors (e.g. V2V networks). These nodes do not frequently change their software composition.

We assume that TCG software stack is used to securely report integrity measurements for all software components of the node, and that the integrity of measurement software can be guaranteed [4].

In addition, we assume that majority of mobile nodes do not expose any malicious behavior.

B. Software composition to predict node behavior

We base our trust management on software composition reputation. We argue that in many mobile environments (in particular an Ad-Hoc sensor and V2V environments), there is no human intervention in the outcome of a transaction/information exchange. In these environments, the mobile nodes (sensors/cars) automatically exchange information without any explicit human intervention. The behavior of a sensor is then determined by its software composition and configuration, i.e. different sensors with the same software composition and configuration will expose the same behavioral pattern.

C. Reputation based on software composition

Our approach is then based on evaluating the probability of a peer exposing malicious behavioral based on its software composition. The probability of malicious behavior of the software composition is a function of probability of malicious behavior of its different software components. Based on our assumption that the majority of the devices have a valid software composition and do not expose malicious behavior, this probabilistic approach will detect, after some learning time, malicious software. This approach then base trust on real nature of the mobile node, i.e. can detect a previously trusted peer which has been recently contaminated by malicious software.

D. Reliable reporting of device software composition

TCG can use integrity measurements to validate/authenticate the peers. But the current TCG approach has a major scalability problem. Even though the integrity measurement of different software components can be securely reported to peers, this information does not show the trust on the software. Actually, the real trust comes from the fact that the peer/administrator can associate the integrity measurements to some trust value; the integrity measurement per se does not guarantee any trust.

Furthermore, in our context of mobile sensors, it is impossible to receive from a central management unit all the integrity measurements of different software components running in different devices and their associated trust value. This central management unit scenario could be valid in an enterprise environment. Though in Ad Hoc networks as for cars or sensors, there is no central management point.
In this case, because of the diversity of brands, the randomness of patches and software updates (which could depend on geographical situation, random moves of mobile nodes and their proximity to updating points etc), the life cycle of devices (where devices could sometimes run for 10 to 20 years) the diversity of software combination is so high that it is impossible to make a judgment based on knowing all valid integrity measurements for software. In RTA, we evaluate trust on peers without the need for a central management unit to associate trust with different software components. Furthermore, RTA evaluates trust on peers never met before based on their software composition.

E. Evaluation of software composition

RTA assumes that upon initialization the device has a software composition without any malicious software component. Therefore, the device trusts peers with the same software composition (i.e. initial trust value).

Each device evaluates its interactions with other devices and stores the result associating them to the software composition of the peer device.

Each device also interrogates other trusted devices for their interaction results. The device keeps the statistics on the occurrence of security problems with different software and completes them with the results from trusted peers. Therefore, each device over time calculates the probability of malicious behavior associated with each software component.

At its start time, RTA does not have enough reputation information to reliably evaluate software compositions. But, as RTA accumulates information by its own transactions, and input from other trusted peers, the accuracy of predictions augments.

F. RTA Components

Figure 1 introduces the internal architecture of a node in RTA framework. The Discovery component discovers different nodes. The Reporter component is on charge of receiving reported reputations information. Interactions between nodes need to be approved by Reputation component (which checks the software configuration of peer nodes). The Evaluation engine evaluates the trust on the peer node and the results of different interactions are stored in the Reputations Database component.

III. RESULTS

We simulated our algorithm by implementing nodes as agents. Each agent evolves in a virtual world interacting with other agents and has some random software composition. The software base covers 100 software components, with 50 malware software components and a total of 1800 versions of files. We used a simple average function to estimate the probability of exposing malicious behavior. We simulate 100 mobile nodes, with 15% of nodes being infected. After 1000 transactions, RTA had only 5% false negative and 4% of false positive. After 2000 transactions, all the malicious configurations are detected. Our simulation proves that RTA algorithm is efficient, depending on the number of transactions.

IV. CONCLUSIONS

In this paper, we presented a new approach to trust evaluation in Ad-Hoc networks based on reputation of software components of mobile nodes. Based on the reliable and trustworthy integrity measurement reporting mechanisms introduced by TCG, we store reputation information for different software components of mobile nodes. We then use a trust function to evaluate the trust in the peer based on software components in the peer software composition.

The main contribution of this work is to expose a mechanism to extend the usefulness of TCG integrity measurement mechanisms from a centrally managed enterprise environment to an open environment like Ad-Hoc networks where the peers are highly mobile and a reputation based on simple peer identity is useless.

The results of our simulation showed that even though we used a simple probabilistic approach, RTA demonstrated great efficiency at detecting malicious software. We expect that these results can be greatly improved in the future by using more sophisticated probabilistic approaches, for example Beta distribution.

Even though, this work has for primary objective Ad-Hoc sensor networks, we believe that it can be extended further to other networks which humans do not intervene in the node behavior.

REFERENCES