Cluster-Based Trust Evaluation Scheme in an Ad Hoc Network

Seunghun Jin, Chanil Park, Daeseon Choi, Kyoil Chung, and Hyunsoo Yoon

ABSTRACT—This paper presents a new trust evaluation scheme in an ad hoc network. To overcome the limited information about unfamiliar nodes and to reduce the required memory space, we propose a cluster-based trust evaluation scheme, in which neighboring nodes form a cluster and select one node as a cluster head. The head issues a trust value certificate that can be referred to by its non-neighbor nodes. In this way, an evaluation of an unfamiliar node's trust can be done very efficiently and precisely. In this paper, we present a trust evaluation metric using this scheme and some operations for forming and managing a cluster. An analysis of the proposed scheme over some security problems is also presented.

Keywords—Ad hoc network, security, trust evaluation, clustering.

I. Introduction

A mobile ad hoc network (MANET) has particular characteristics. First, there is no infrastructure. Second, network topology is not fixed because of the mobility of nodes. Therefore, a security scheme for an ad hoc network should consider these characteristics. Many new types of threats are emerging, and they are hard to defend with conventional security methods [1]. To adapt to particular characteristics and defend against new threats, new security schemes based on trust evaluation are proposed [2]-[5]. In these schemes, each node evaluates the trust of the other nodes with which it communicates. The evaluation for the other node is performed by evaluating the node's own experience about the evaluated node. Based on the evaluated trust, security measures are taken, or security decisions are made.

In the previous schemes of trust evaluation [2], [3], each node has to evaluate the trust of every other node in the network. It is quite difficult for a node to evaluate accurately the trust of the nodes of which it has little interaction. Furthermore, these schemes require a great deal of memory space to store each of the other node's trust value. Such constraining of resources in an ad hoc network is a severe problem.

Hence, we propose a new trust evaluation scheme based on clustering, in which a cluster is built by neighboring nodes, and a cluster head elected by cluster members acts as a trust guarantor. In our scheme, the trust of a node is evaluated by combining the node's own experience and the information presented by the head of the cluster to which the evaluated node belongs. Even in a case when a node has no experience, it can evaluate other nodes based on the trust value provided by the cluster head. On the other hand, this means that the node does not have to store and manage experience data about all the other nodes in the network.

For an ad hoc network, a security scheme based on clustering was proposed [6]. In this research, clustering is used for certification of a public key. Therefore, before our approach, there has been no previous attempt to employ clustering for trust evaluation [7].

II. Cluster-Based Trust Evaluation Scheme

In our scheme, a cluster is first formed based on the trust values of the neighbor nodes. Then, a cluster head that has the highest trust value in the cluster issues a trust value certificate for cluster member nodes. Cluster forming is carried out as follows. An ad hoc node evaluates its neighbor nodes' trust

Manuscript received Jan. 03, 2005; revised May 09, 2005.

Seunghun Jin (phone: +82 42 860 1254, email: jinsh@etri.re.kr), Daeseon Choi (email : sunchoi@etri.re.kr), and Kyoil Chung (email: kyoil@etri.re.kr) are with Information Security Research Division, ETRI, Daejeon, Korea.

Chanil Park (email: chanil@camars.kaist.ac.kr) and Hyunsoo Yoon (email: hyoon@camars.kaist.ac.kr) are with EECS, KAIST, Daejeon, Korea.

values based on its experience. After calculating the trust values of its neighbor nodes, each node chooses one node that has the highest value as its trust guarantor. Then, the chosen node becomes the cluster head and the chooser becomes a member of the cluster. If the chosen node is already a member of another cluster, a node of the second highest trust value is chosen. In this way, a cluster is formed. The cluster head has the highest trust value among the cluster members. Figure 1 shows an evaluated trust value and chosen cluster head.

After forming a cluster, the cluster head plays the role of trust guarantor. The cluster head evaluates and guarantees the trust of the cluster member nodes. When a member node requests it, the cluster head issues the trust value certificate that contains the node's trust value. The member node uses the trust value certificate to show its trustworthiness when communicating with other nodes. Details of the trust evaluation metric and operation used in clustering and certification are explained in this section.

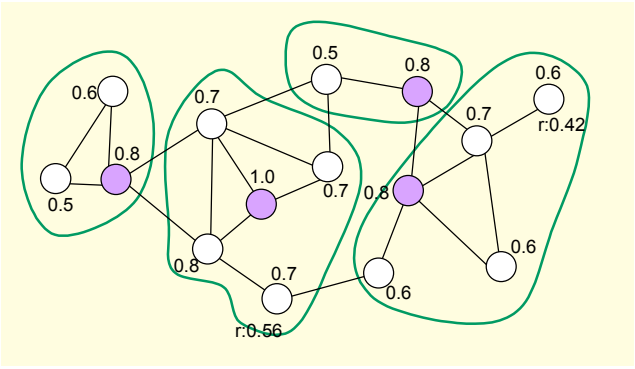

Fig. 1. A cluster forming in the proposed model. The numbers indicate the trust value of the nodes. The purple nodes are the cluster heads.

1. Trust Evaluation Metric

A node evaluates another node's trust value by combining the node's own experience data and the trust value provided by the head of the cluster to which the evaluated node belongs. Factors that can be used as the experience data are as follows.

- Communication data rate (V_c): the rate of successful communication with evaluated nodes. The value is between 0 and 1. The initial value is 1.
- Data delivery rate (V_d): the rate of successful packet delivery by the evaluated node. This rate measures how successful a node relays packets from a source to a destination as an intermediary. The value range and initial value are the same as the communication data rate.

A node needs to gather and store these experience data for all the other nodes. However, in our scheme, a node might delete the experience data for nodes with which it did not communicate for a certain specified period since these data might be obsolete in an ad hoc network that has a dynamic topology. In the case where there is no experience data about a node, the trust value can be obtained from a certificate issued by the head node. For a node that has old experience data, the trust value from a certificate is much more accurate. The experience data about neighbor nodes should not be deleted since they are used for cluster forming. Also, the head node does not delete the data about its member node while the cluster exists.

An experience metric combining these factors is defined as follows. The experience metric is an arithmetic means of factors. For the head node, the same metric is used:

$$V_E(i,j) = \sum_{v \in V} v \Big/ |V|$$

where $V_E(i,j)$ is the experience metric of node j evaluated by node i and $V = \{V_c, V_d\}$ is the set of experience factors.

A trust value metric that combines a node's own experience and trust value from the head is defined as follows.

$$V_T(i,j) = (V_E(i,j) \cdot ew + V_T(H,j) \cdot (1-ew)) \cdot V_B(j),$$

$$ew = ec / ec\_threshold$$

where $V_T(i,j)$ is the trust value of node j evaluated by node i, when i is H—it is a value evaluated by the head of the node's cluster; ew is the experience weight, which is fixed as 1 if it exceeds 1; ec is the experience count; $ec\_threshold$ is the threshold of experience count set by a node; and $V_B(j)$ is 0 if the node j is malicious or 1, otherwise.

The degree of whether a trust value can be evaluated based on a node's own experience or on a certificate from the head, or from both, is decided according to the experience weight. For example, only the value from a certificate is used for a node that the evaluator has no experience data on, or where the experience data is deleted due to the expiry of its validity. The node's own experience is the most accurate if the data are collected from a large experience. So, the trust value of a node for which the experience count exceeds a certain threshold is evaluated using only its own experience data. For other cases, the proportion of the experience data to a certificate from the head is in accordance to the experience weight.

For the head node, the same metric is applied except that $V_T(H)$ indicates the trust value given by the head node of a cluster to which the evaluated node had belong to just before. This case occurs when the node had moved in from another cluster. This means that when a head does not have enough

experience data about a member node, it relies on the trust value provided from the previous cluster. In the case where the node did not move in from another cluster and the head does not have any experience data about the node, the trust value calculated in the head is 1.

Value V_B in the metric is a value broadcasted by a head node for identifying a node as a malicious node or an intruder. The node that receives this value adds this node name to its blacklist. Some methods to detect a malicious node are described in [3].

2. Operations

In this section, we explain the details of operations for clustering and certification. In these operations, the bootstrapping security model [8] is used to protect the integrity of the transmitted trust value.

A. Hello

Nodes broadcast a hello message that contains a cluster head searching message. If there are several cluster heads in the neighborhood, the node should select the one that has the largest cluster members as its cluster head, and it becomes a member of the selected cluster. If there is no cluster head in the neighborhood, the node elects a new cluster head among its neighbor nodes.

B. Cluster Construction

Each node recommends one of its neighbor nodes as a cluster head, and the recommender becomes a member of the cluster. In this manner, the one cluster head and its recommenders (a cluster head can be recommended from many other neighbor nodes) form a one-hop range cluster. When a node recommends a cluster head, it sends to the cluster head a recommend message that includes recommendation certificates (R_Certificate). These certificates are used to authenticate whether the cluster head has many cluster members that trust the head. The following are a cluster head recommend message and recommendation certificates.

> Recommend Message:
> M(node1's id, node2's id, node2's trust value, R_Certificates, "recommend message");
>
> Recommendation Certificate(R_Certificate):
> {node1's id, node2's id, create time, validation, "Recommend", node1's PUB_KEY, signature(node's id, node2's id, create time, validation, "Recommend")}

In this message, node1 indicates the recommender and node2 indicates that the node is recommended as a cluster head. In the recommendation certificate, *validation* represents the period of validity. After the validation period expires, the cluster head has to request new certificates from the cluster member nodes.

C. Trust Certificate

After a cluster has formed, a node requests a trust value certificate to its cluster head. The head issues the node's trust value certificate and sends it with the head's own recommendation certificates that verify that the cluster head is trustworthy. The trust value certificate is defined below.

> Trust value certificate(T_Certificate):
> {cluster head's id, node1'id, node1's trust value, cluster head's Pub key, create time, validation, signature(cluster head's id, node1's id, node1's trust value, create time)}

D. Node Join

A join operation is executed in two situations. One is when a node first enters into the network. Another is the case when a node moves from one cluster to another cluster. In the first case, the cluster head has to evaluate the node's trust value from scratch because the node does not have a trust value certificate or recommend certificates. In the second case, an incoming node gives to the new cluster head the trust value certificate and recommendation certificates that are received from the previous cluster head. Using these certificates, the new cluster head authenticates and establishes an initial trust value of the new member node without its own experience. The join operation is carried out as follows. First, a node broadcasts a hello message. Any cluster head that receives the message sends a respond message to the node. The respond message of the head contains the number of member nodes. Second, after receiving the response message from the cluster head, the joining node sends to the cluster head the join message shown below.

> Join Message:
> M(node1's id, cluster head's id, previous cluster head's id, T_Certificate, R_Certificates, "join message")

If there are more than two cluster heads in the neighborhood, the joining node selects one cluster head that has more cluster members than the other cluster heads. Third, after the cluster head receives a join message, it evaluates the trust value of the joining node by referring to the previous trust value certified by the previous cluster head. If the joining node is not able to find any cluster head in a one-hop range, the node extends the search area to a two-hop range. If there is no cluster head in a two-hop range, then the joining node has to reconstruct the cluster with the neighbor nodes.

E. Node Leave

A node that moves from one cluster to another node sends a leave message to its cluster head. After receiving this message, the cluster head deletes its data about the node.

III. Analysis

Our trust evaluation scheme can be applied to many security problems such as secure routing and new types of attacks in a mobile ad hoc network MANET.

1. Secure Routing

Our trust evaluation scheme can be applied to secure a routing problem in an ad hoc network. In an ad hoc network, a node sends and receives a routing request packet and a routing reply packet to find a path from itself (source node) to any destination node. When any node receives a route request packet, it appends its own address to the route record in the route request packet and rebroadcasts the packet [9].

With our scheme, the node that receives a routing request packet appends the T_Certificate, the R_Certificates, and the node's own address. After receiving the route reply packet, the source node checks the trust value of the intermediate nodes. If there is any node that has a low trust value, then the source node discards the routing path and rebroadcasts the routing request packet to find another path.

2. Other Security Problems

We further evaluated the efficiency of the proposed model by analyzing it over several attacks in an ad hoc network.

- Message forgery attack: In our model, we used an asymmetric cryptographic method proposed by Rakesh [5]. By attaching the signature and the public key of the sender in the message, the receiver node can detect if the message was forged. The receiver node also can detect that the public key is really the public key of the sender.
- Black hole attack: In this attack, a malicious node advertises itself as the shortest path to other nodes and drops all packets which come on it. The trust guarantor of the malicious node will set a low trust value for the malicious node. Therefore, the node that wants to send a packet will discard the routing path that goes through the malicious node.
- Selfishness: In an ad hoc network, it is possible that a node doesn't route a packet from the other nodes or simply drops some packets to save their power or other energy. We call these nodes selfish nodes. In our model, a selfish node cannot have a high trust value because of the data delivery rate. By not providing packet forwarding for low trusted nodes, a network can encourage cooperation and reduce selfishness.

IV. Conclusion

In this paper, we proposed a new trust evaluation scheme in ad hoc networks. By referring to the trust value certified by the cluster head, which is the most trusted node in the cluster to which the evaluated node belongs, a node can evaluate the trust value of the unfamiliar node in spite of the absence of direct experience about the node. This means that a node can securely communicate with another node that it has never contacted before. Also, a node does not have to store experience data about all the other nodes. With security analysis, we have shown that our scheme can be applied to many security problems in an ad hoc network.

References

[1] H. Deng, Wei. Li, and Dharma P. Agrawal, "Routing Security in Wireless Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 40, no. 10, Oct. 2002, pp.70-75.

[2] Z. Yan, P. Zhang, and Teemupekka Virtanen, *Trust Evaluation Based Security Solution in Ad Hoc Networks*, Technical Report, Nokia Research Center, Helsinki, Finland, Oct. 2003.

[3] Asad Amir Pirzada and Chris McDonald, "Establishing Trust In Pure Ad-hoc Networks," *Proc. Australasian Computer Science Conf.,* Jan. 2004, pp.47-54.

[4] S.P. Marsh, *Formalizing Trust as a Computational Concept*, Ph.D. Thesis, Dept. of Mathematics and Computer Science, Univ. of Stirling, 1994.

[5] Mikko Sarela and Maarit Hietalahti, *Security Topics and Mobility Management in Hierarchical Ad Hoc Networks: A Literature Survey*, Interim Report of Project Samoyed, Helsinki Univ. of Technology, Apr. 2004.

[6] Marc Bechler, Hans-Joachim Hof, Daniel Kraft, Frank Pahlke, and Lars Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks," *Proc. IEEE INFOCOM*, vol.4, Mar. 2004, pp.2393-2403.

[7] Chan-Il Park, Y. H.Lee, H.Yoon, D.S. Choi, and S.H. Jin, "Cluster-Based Trust Evaluation in Ad Hoc Networks," *Proc. Int'l Conf. Advanced Communication Technology*, 4C-03, Feb. 2005.

[8] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proc. Int'l Symp. Parallel Architectures, Algorithms, and Networks*, 1999, pp. 310-315.

[9] Rakesh Babu Bobba, Laurent Eschenauer, Virgil Gligor, and William Arbaugh, *Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks*, Technical Report, TR2002-44, Univ. of Maryland, May 2002.