Certificate Chain Discovery in Web of Trust for Ad Hoc Networks

Hisashi Mohri   Ikuya Yasuda   Yoshiaki Takata   Hiroyuki Seki
Graduate School of Information Science
Nara Institute of Science and Technology
Takayama 8916-5, Ikoma, Nara, 630-0192 Japan

Abstract

In an ad hoc network, we cannot assume a trusted certificate authority and a centralized repository that are used in ordinary Public-Key Infrastructure (PKI). Hence a PKI system of the web-of-trust type in which each node can issue certificates to others in a self-organizing manner has been studied. Although this system is useful for ad hoc networks whose topology can change, it has the problem that for authentication a node needs to find a certificate-chain to the destination node. In this paper, we formally model a web-of-trust-type PKI system, define the certificate-chain discovery problem, and propose a new distributed algorithm and its modifications that solve the problem. Furthermore, we propose a measure of communication cost, and according to the measure, we compare our algorithm with an existing method.

1. Introduction

An ad hoc wireless network is a formed network that can be de-formed on-the-fly without the need for any system administration[12]. Unfortunately, these characteristics prevent us from applying traditional security techniques to ad hoc networks. In particular, though PKI (Public-Key Infrastructure) [3, 7] is one of the most useful security techniques, ordinary PKI systems cannot be applied to ad hoc networks [14]. PKI is a security infrastructure in which we can authenticate a public key by using *digital certificates* [3, 7]. In public-key cryptosystems, we have to obtain other users' public keys to securely communicate with those users. In PKI systems using certificates as shown in Figure 1, we can verify that Pk_u is the public key of u by using Pk_{CA}.

One of the problems in adopting ordinary PKI systems is that in ad hoc networks we cannot assume a trusted certificate authority (to manage digital certificates) and a centralized repository (to store digital certificates securely) that are used in ordinary PKI systems. Moreover, we cannot assign the tasks of a trusted certificate authority or a centralized repository to any node in an ad hoc network. If we did so, because the node may move out of the network, the PKI system could not function. In particular, malicious nodes could easily attack the network to avoid the PKI system because a trusted third party to administer the PKI system in the network is only one node. Some methods assuming certificate authorities by using threshold signatures have been proposed [13, 14]. However, these methods may cause a problem that some nodes holding a fragment of the function of the certificate authority are attacked intensively. Hence, we focus on a *web-of-trust* type PKI system considered in PGP [15] in which each node can issue certificates to others in a self-organizing manner. Some authores have proposed web-of-trust-type systems for ad-hoc networks where each node has a distributed repository instead of a centralized repository [1, 4, 8, 9]. However, these systems suffer from a common problem such that a node must discover a certificate-chain if the node does not have enough certificates for key authentication.

A set of certificates for authenticating a nodes' public key is called a *certificate-chain* [2]. We call the node that authenticates a public key the *source node*, and the node whose public key will be verified by the source node the *destination node*. The source node can able to verify a public key even if the source node does not directly sign the public key by discovering a certificate-chain from the source node to the destination node because the trust relation represented by certificates the is *transitive*. At first, the source node trusts the nodes whose public keys are signed by the source node because the source node can verify the certificates using her public key. Next, the source node trusts nodes whose public keys are signed by the already trusted nodes because the source node can verify the certificates using the already trusted nodes' public keys. In ordinary PKI systems, we can find such certificate-chain from the set of certificates in a trusted repository. However, it is not trivial to discover a certificate-chain in distributed repositories.

In this paper, we investigate the certificate-chain dis-

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

Certificate :

Pk_u is the public key of u.

Sk_{CA} (issuer's signature)

u	: A user
Pk_u	: The public key of u
CA	: The issuer (the signer of the certificate)
Sk_{CA}	: The secret key of *CA* (to sign the certificate)
Pk_{CA}	: The public key of *CA* (to verify the certificate)

Figure 1. A certificate for u issued by *CA*

covery problem in ad hoc networks. To make this problem clear, we formally define the certificate-chain discovery problem in section 2. We review an existing method as related work in section 3. In section 4, we model a web-of-trust-type PKI system as a weighted directed graph, explain that solving the problem can be reduced to finding a path between two nodes in the graph, and propose a new distributed algorithm for solving the problem. We divide the certificate-chain discovery into the *certificate searching phase* and the *certificate collecting phase*, and we propose a search method based on a distributed algorithm for constructing a spanning tree and a method for collecting all certificates in the discovered certificate-chain. The whole algorithm will be called the *basic scheme*. Furthermore, we propose a measure of communication cost, and according to the measure, we compare our method with the existing method in section 5. In section 6, we propose two modifications of the basic scheme and compare the communication cost of these methods. Finally, we conclude this paper in section 7.

2. Definition of the Problem

We investigate a web-of-trust-type PKI system where every node has a repository and stores certificates that the node signed or in which the public key of the node is certified by another node such as [4, 5, 6, 8, 9]. It is possible to reduce communication cost by limiting certificates in a repository. Moreover, this method reduces the cost of the certificate revocation phase. In this method, a certificate is only held by its issuer and usr (the owner of the public key in the certificate), so the certificate revocation phase is done by the issuer or the user. When the issuer (or the user) wants to revoke the certificate, she only has to send revocation information to the user (or the issuer) without heavy computation and communication (e.g., using a *certificate revocation list*).

However, there is a new problem of discovering a path of certificates based on the trust relationship in distributed repositories to verify whether a public key is correct. We call this problem *certificate-chain discovery problem in ad hoc networks*.

Definition 1 ***Certificate-chain discovery problem in ad hoc networks***
Assume that we are given a web-of-trust-type PKI system where every node has a repository and stores certificates that the node signed or in which the public key of the node was certified by another node. Also assume that we are given a source node and a destination node. Then, find a certificate-chain from the source node to the destination node and collect all certificates in the certificate-chain.

3. Related Work

Some authors have proposed PKI systems that limit the issuing of certificates for simplifying the problem in Definition 1 [9, 8]. However, such methods prevent nodes from issuing certificates based on the trust relation among the nodes. We should have a generalized method to solve the problem.

Kitada et al. considered the same problem described in Definition 1 and proposed the ASNS (Ad hoc Simultaneous Nodes Search) protocol [4, 5, 6]. ASNS finds a certificate-chain as follows.

- The source node broadcasts the search packet p to nodes that the source node directly trusts.
- If a node v receives a packet p, v modifies and sends p as follows.
  - The node v adds its own certificate to the packet p, rewrites the address of p to the nodes that v directly trusts, and broadcasts p to the nodes that v directly trusts.
  - If v directly trusts the destination node, v adds its own certificate to p, rewrites the address of p to the destination node, and sends p to the destination node.
  - If v is the destination node of p, v adds its own certificate to p and sends p to the source node.
- If a node receives more than one packet sent by an identical source node, the node processes only the first packet as above and discards all other packets.

Because each node processes only the first packet, the number of packets per one search is proportional to the number of certificates.

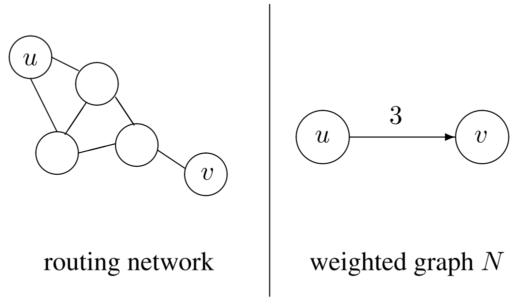

Figure 2. Relation between routing network and weighted graph N

However, ASNS has the following problem. In distributed networks such as ad hoc networks, the protocol is completed not when the destination node is discovered, but when all nodes in the network receive the packet. Thus, ASNS has heavy communication cost because of broadcasting packets with certificates. We need a new method with lower communication cost to solve the certificate-chain discovery problem.

4. Proposed Method

In this section, we formally define a web-of-trust-type PKI system to make the certificate-chain discovery problem clear. Based on this model, we divide the problem into two phases. Finally, we propose a new distributed algorithm to solve the problem.

4.1 Web-of-Trust in Ad Hoc Networks

Definition 2 ***Trust Model of Web-of-Trust in Ad Hoc Networks***
A trust model of web-of-trust in ad hoc networks is a weighted directed graph $N = (V, E, \phi)$, where

- *V is a set of nodes,*
- *E is a set of directed edges, and*
- *ϕ is a weight function that maps each directed edge to a non-negative integer.*

An edge in a trust model does not mean that the end nodes of the edge can directly communicate with each other. We let the weight $\phi(\langle u, v\rangle)$ of an edge $\langle u, v\rangle$ be the number of hops from node u to node v (See Figure 2). For simplicity, we assume that the set V in a trust model equals the set of nodes in the corresponding ad hoc network.

4.2 Basic Scheme

As we described in Section 3, certificates are added to a search packet in the Kitada method. This method incurs much communication cost because all nodes receive a search packet with a number of certificates regardless of the fact that some of the nodes do not need the certificates. In this paper, we divide the certificate-chain discovery problem into the certificate searching phase and the certificate collecting phase to make this problem clear, and propose a new algorithm for each phase.

Certificate Searching Phase We assume that each node knows the number of hops to any other node by using a routing protocol in the lower physical layer and knows only edges adjacent to the node in N. The problem in this phase is to find a certificate-chain from a given source node to a given destination node. Note that we need not find all certificate-chains; finding one certificate-chain is sufficient for authentication.

To solve the problem in the certificate searching phase, we use a distributed algorithm for constructing a spanning tree where the root node is the source node. We can use any distributed algorithm for constructing a spanning tree in a directed graph. Communication complexity of standard algorithms for constructing a spanning tree is $O(|E|)$, where $|E|$ is the number of the elements of E [10].

Certificate Collecting Phase When the certificate searching phase is completed, each node knows which node is the parent in the constructed tree. However, all nodes including the source node do not know about the entire tree and the source node needs to obtain all certificates in a certificate-chain. We can reduce this problem to the problem of collecting all certificates in a path from the source node to the destination node in the tree because there must be such a path in the spanning tree. To solve the problem in this phase, we propose the following method.

- The destination node sends a packet to the parent node.
- Each intermediate node that received the packet adds its own certificate to the packet and sends it to its parent node.

This process is repeated until the packet reaches the source node. When this process is completed, the source node obtains all certificates in a certificate-chain.

5. Evaluation

In this section, we define the communication cost, analyze the cost of the basic scheme and the Kitada method, and compare the cost of the two methods. As a result, we show that the cost of the basic scheme is lower than Kitada's.

5.1 Definition of the communication cost

At first, we define communication cost. In [4], commnication cost is defined as the number of packets. This definition does not consider the size of a certificate and the number of the certificates in a packet. This definition is not realistic because a packet that includes many heavy certificates is counted as "one packet".

Thus, we give a more realistic communication cost as follows.

Definition 3 ***The communication cost***
Let e be any edge in the directed graph N. We define the communication cost as follows.

$$\sum_{\textit{edge } e} \{\textit{total bit size on } e \times \phi(e)\}.$$

That is, to consider the size of a certificate and the number of the certificates in a packet, we define the communication cost as the total message bits.

5.2 Analysis of the Kitada Method

In this subsection, we analyze the Kitada method. Though they analyzed their method in [4], it is based on a communication cost that does not consider the size of a packet. We first divide the method into two phases and analyze each of the two phases by using our definition of communication cost to compare the method and our proposed method.

We can consider ASNS to be a distributed algorithm constructing a spanning tree in which certificates are added to a search packet. The length of a certificate-chain is the number of edges from the source node to the destination node. On the other hand, the height of the spanning tree must be at least the length of the chain because the chain is also a path in the tree, and the height may be longer than the chain because the distributed algorithm does not halt even if the chain is discovered. That is, the following relation holds between the length and the height:

(the length of a certificate chain in a spanning tree)
$\leq$ (the height of the tree).

We assume that the length of a certificate chain equals the height of the tree, i.e., we estimate the communication cost based on the upper bound of the length. We also use this assumption in section 5.3.

Certificate Searching Phase In this phase, the source node broadcasts search packet to all nodes that the source node directly trusts. When a node receives the packet, the node adds its own certificate to the packet and broadcasts it to all nodes that the node directly trusts. A packet is transmitted until a node receives the same packet twice. Then, the communication cost in this phase $S_1(k)$ is given by the following equation, where n is the average number of hops, m is the average number of the degrees of nodes in N, $Cert$ is the size of a certificate, k is the height of the constructed spanning tree, and $Cert\_req$ is the packet size of a certificate search packet.

$$S_1(k) = n \sum_{i=1}^{k} \{Cert(i-1) + Cert\_req\} m^i.$$

Certificate Collecting Phase When the destination node receives a packet with certificates, the node adds its own certificate to the packet and sends it back to the source node. The destination node sends back k certificates to the source node because the number of certificates in this packet is equal to the length of a certificate-chain. Therefore, the cost $C_1(k)$ is given by the following equation, where $Cert\_rpl$ is the size of a replying packet:

$$C_1(k) = n(k \times Cert + Cert\_rpl).$$

5.3 Analysis of the Basic Scheme

Certificate Searching Phase In the basic scheme, a source node constructs a spanning tree using any distributed algorithm, and any certificates is not added to a packet. Then, the cost $S_2(k)$ is as follows:

$$S_2(k) = n \sum_{i=1}^{k} Cert\_req \times m^i.$$

Certificate Collecting Phase When the destination node receives a search packet, each node in the tree knows which node is the parent node. The destination node sends the packet to the parent node, and each intermediate node receiving the packet adds its own certificate and sends it to the parent node. Thus, the source node receives a packet with $(k-1)$ certificates. The cost $C_2(k)$ is as follows:

$$C_2(k) = n \sum_{i=1}^{k} \{Cert(i-1) + Cert\_rpl\}.$$

5.4 Comparison between Two Methods

Complexity Analysis We analyze the fraction of communication cost of the two methods. The total cost of the Kitada method is equal to $(S_1(k) + C_1(k))$ and the total cost

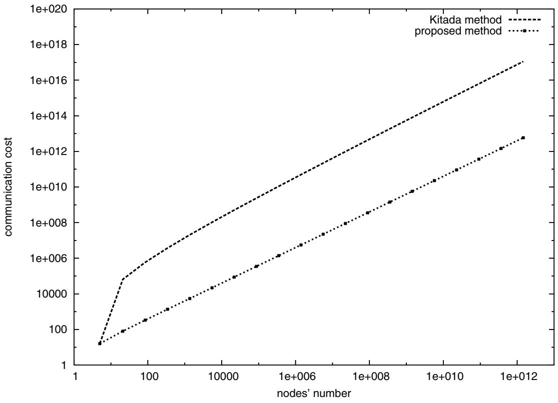

Figure 3. Basic scheme vs. the Kitada method (searching cost)

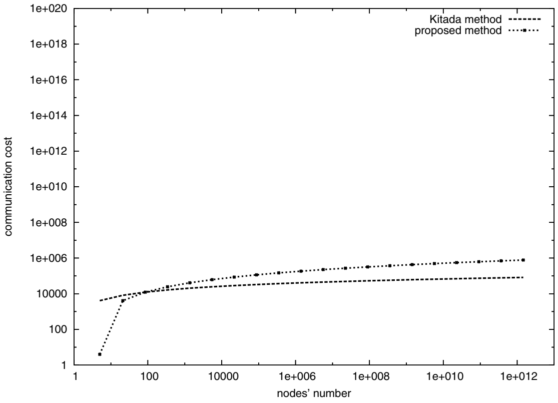

Figure 4. Basic scheme vs. the Kitada method (collecting cost)

of the basic scheme is equal to $(S_2(k) + C_2(k))$:

$$\frac{S_1(k) + C_1(k)}{S_2(k) + C_2(k)} = \frac{O(k \cdot Cert \cdot m^{k+2})}{O(m^{k+1})} = O(k \cdot Cert \cdot m).$$

This tells us that the cost of the Kitada method is $O(k \cdot Cert \cdot m)$ times the cost of the basic method.

Numerical Analysis We also compare the costs by numerical analysis. We let $n = 4$, $m = 4$, and $Cert = 1024$. In [4], they estimate that the average of hops (n) and the average of degree (m) to construct a web-of-trust are four. The size of a certificate $(Cert)$ is determined by the bits of RSA keys. Also, we let $Cert\_req = 1$ and $Cert\_rpl = 1$ for simplicity because, in the complexity analysis, we showed that the fraction of the total cost between the two methods does not depend on the size of a certificate requesting packet and certificate replying packet.

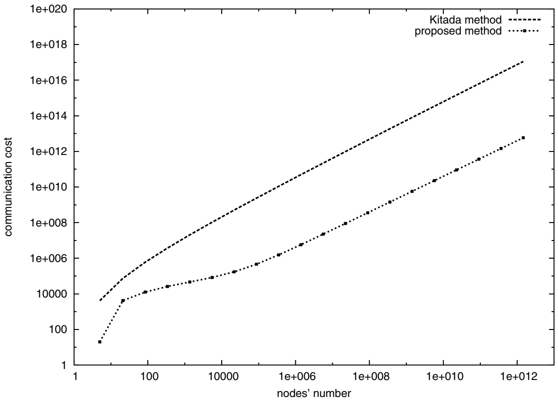

Figure 5. Basic scheme vs. the Kitada method (total cost)

Figure 3 shows the costs of the searching phase. The cost of the basic scheme is lower than the cost of the Kitada method. This is because the searching phase in the basic scheme broadcasts search packets without adding certificates. In Figure 4, we show the graph of the cost of the collecting phase. In the collecting phase of the Kitada method, the destination node sends packets to the source node. On the other hand, we have to collect certificates from the destination node to the source node while sending back the packet along with the certificate-chain in the proposed method. Thus, the cost of basic scheme is higher than the Kitada method. Finally, in Figure 5, we compare the total costs of the two methods. We can see that the basic scheme has lower cost than the Kitada method.

6. Modifications of Proposed Method

The basic scheme has a disadvantage on the cost of the collecting phase. To reduce the cost, we investigate two modifications of the basic scheme.

6.1 Modification 1

The first modification is to use a distributed algorithm for constructing the shortest path tree in the searching phase. In the basic scheme, we use a distributed algorithm for constructing a spanning tree but the certificate-chain found by the algorithm does not always have the minimum cost (the sum of the weights of the edges in the chain, i.e., the total sum of the hops in the chain). Constructing a shortest path tree, we can discover a chain with the minimum cost by which we can collect certificates with a lower cost.

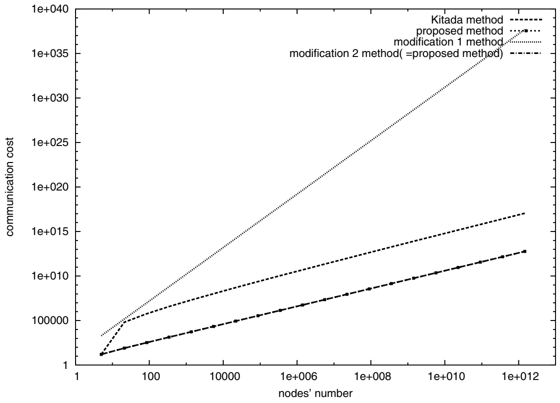

Figure 6. Comparison of searching costs

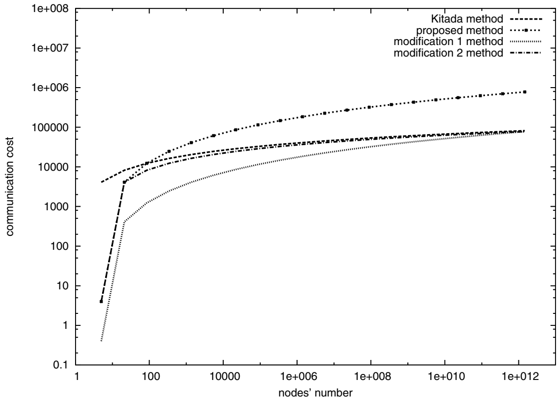

Figure 7. Comparison of collecting costs

However, the upper bound of the computation time for the distributed Bellman-Ford algorithm is $O(V \cdot E)$ [10] and the one for the standard distributed Dijkstra algorithms is $O(V^2)$ [11]. These costs are much higher than the algorithms for spanning trees. Consequently, the cost $S_3(k)$ for the searching phase of the modified method is higher than $S_2(k)$ while the cost $C_3(k)$ for the collecting phase of the modified method is lower than $C_2(k)$.

6.2 Modification 2

The other modification is to revise only the collecting phase. In the basic scheme, the source node obtains all certificates in a certificate-chain by making each intermediate node add its certificate to the replying packet. This scheme requires extra cost because the packet from the destination node runs through the whole chain, expanded with the added certificates. To solve the problem, we modify the phase as follows:

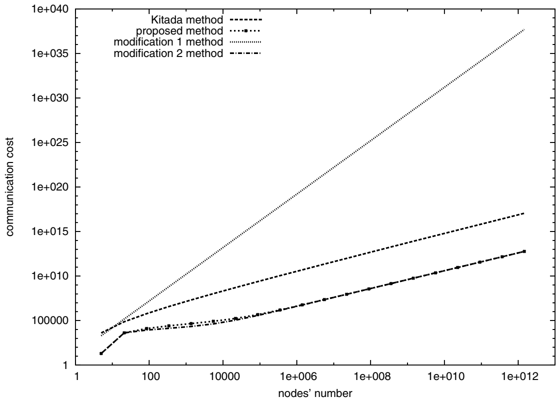

Figure 8. Comparison of total costs

- The destination node sends a packet to the parent node in the certificate-chain.
- Each intermediate node receiving the packet sends its certificate directly to the source node and also sends the packet to the parent node (because each node does not know whether the node itself is on the certificate-chain to the destination node).

After this modification, the cost $S_4(k)$ of certificate searching phase is the same as $S_2(k)$, and the cost $C_4(k)$ of the collecting phase is as follows.

$$C_4(k) = k \times n \times (Cert + Cert\_rpl).$$

6.3 Numerical Analysis

We compare the above methods by numerical analysis. We let $n = 4$, $m = 4$, $Cert = 1024$, and $Cert\_req = Cert\_rpl = 1$.

Figure 6 shows the costs of the searching phase. Note that we use the distributed Bellman-Ford algorithm for constructing the shortest path tree in modification 1. The cost of modification 2 is the same as the cost of the basic scheme and is lower than the Kitada method. On the other hand, the cost of modification 1 is higher than the Kitada method. Figure 7 shows a graph of the cost of the collecting phase. We assume that the weight of the shortest path from the source node to the destination node is one tenth of the weight of the path in a spanning tree. Both modifications have lower cost than the Kitada method.

We compare the total costs of all the above methods (Figure 8). From this comparison, we obtain the following result:

$$\begin{array}{rll} & S_3(k) + C_3(k) & \text{(modification 1)} \\ > & S_1(k) + C_1(k) & \text{(the Kitada method)} \end{array}$$

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

Table 1. Our schemes vs. the Kitada method

Method Name	Search	Collect	Total
Basic scheme	√		√
Modification 1	√		
Modification 2	√	√	√

Note : "√" means that the cost is lower than the Kitada method.

$$\begin{aligned} &> \quad S_2(k) + C_2(k) \qquad \text{(basic scheme)} \\ &> \quad S_4(k) + C_4(k) \qquad \text{(modification 2).} \end{aligned}$$

From these figures, we can see that modification 2 requires a lower cost than the Kitada method in both phases. Clearly, modification 2 requires a lower cost in total because the total cost is the sum of the cost of the two phases. On the other hand, the total cost of modification 1 is higher than the Kitada method, because the cost of the distributed Bellman-Ford algorithm is much higher and thus it blots out the advantage in the collecting phase.

7. Conclusion

In this paper, we modeled web-of-trust-type PKI systems, formally defined the certificate-chain discovery problem, and proposed a new distributed algorithm as well as two modifications for solving the problem. Furthermore, we proposed a measure of communication cost, and according to the measure, we compared our algorithms with the Kitada method. As a result, we showed that our basic scheme and modification 2 achieve a lower communication cost. In particular, in both phases and the total of them, modification 2 requires lower cost than the Kitada method. The result is summarized as Table 1.

Unfortunately, existing routing protocols for ad hoc networks are unable to catch up with frequent link changes [12]. These protocols minimize the effect of dynamic change of the topology caused by nodes' mobility by reducing time, communication, and round complexity. Our proposed methods also address node mobility by reducing such complexities as existing routing protocols do.

Our future work will include a simulation-based comparison between the proposed method and the Kitada method. We will also try to adopt other distributed algorithms for constructing the shortest path tree such as the distributed Dijkstra algorithm instead of the distributed Bellman-Ford algorithm in modification 1. Because the upper bound of the computation time of the Dijkstra algorithm is lower than the Bellman-Ford, this change will reduce the cost of modification 1. Furthermore, we will investigate the modeling of web-of-trust-type PKI systems for ad hoc networks more deeply to construct a new trust model combining the models of Capkun et al. [1] and Kitada et al. [4].

References

[1] S. Capkun, L. Buttyan, and J. P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(2):52–64, 2003.

[2] D. E. Clarke, J.-E. Elien, C. M. Ellison, M. Fredette, A. Morcos, and R. L. Rivest. Certificate chain discovery in spki/sdsi. *Journal of Computer Security*, 9(4):285–322, 2001.

[3] R. Housley, W. Polk, W. Ford, and D. Solo. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 3280, 2002.

[4] Y. Kitada, Y. Arakawa, K. Takemori, A. Watanabe, and I. Sasase. On demand distributed public key management using routing information for wirelss ad hoc netwoks. *IEICE Transactions on Information and Systems*, J88-D1(10):1571–1583, October 2005.

[5] Y. Kitada, A. Watanabe, K. Takemori, and I. Sasase. On demand distributed public key management for wireless ad hoc networks. *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)*, 2005.

[6] Y. Kitada, A. Watanabe, K. Takemori, and I. Sasase. On demand distributed public key management without considering routing tables for wireless ad hoc networks. *Asia Pacific Symposium on Information Technology (APSITT)*, pages 375–381, 2005.

[7] L. M. Kornfelder. *Toward a Practical Public-Key Cryptosystem*. bachelor's thesis, Dept. Electrical Eng., Massachusetts Inst. of Technology, Cambridge, 2005.

[8] R. Li, J. Li, H. Kameda, and P. Liu. Localized public-key management for mobile ad hoc networks. *IEEE Global Telecommunications Conference (Globecom)*, pages 1284–1289, 2004.

[9] X. Li, S. Gordon, and J. Slay. On demand public key management for wireless ad hoc networks. *Australian Telecommunication Networks and Applications Conference (ATNAC)*, pages 36–43, 2004.

[10] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, San Francisco, California, 1996.

[11] K. Miura, T. Masuzawa, and N. Tokura. A distributed shortest paths algorithm with distance-dependent message complexities. *IEICE Transactions on Information and Systems*, J77-D1(1):21–32, January 1994.

[12] C. K. Toh. *Ad-Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall, Upper Saddle River, New Jersey, 2001.

[13] S. Yu and R. Kravets. Composite key management for ad hoc networks. *IEEE Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services (Mobiquitous)*, pages 52–61, 2004.

[14] L. Zhou and Z. J. Hass. Securing ad hoc network. *IEEE Network*, 13(6):24–30, 1999.

[15] P. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, Massachusetts, 1995.

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE