New Certificate Chain Discovery Methods for Trust Establishment in Ad Hoc Networks and Their Evaluation

Regular Paper

New Certificate Chain Discovery Methods for Trust Establishment in Ad Hoc Networks and Their Evaluation

Hisashi Mohri,[†] Ikuya Yasuda,[†,☆] Yoshiaki Takata[†,☆☆] and Hiroyuki Seki[†]

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, 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 modification 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 by numerical computation for large-size networks and by simulation on randomly generated unit disk graphs for moderate-size networks. The simulation results show that the communication cost of the proposed method is less than 10% of the 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[20]. Unfortunately, these characteristics prevent us from applying traditional security techniques to ad hoc networks. In particular, though Public-Key Infrastructure (PKI)[6),7)] is one of the most useful security techniques, ordinary PKI systems cannot be applied to ad hoc networks[22]. PKI is a security infrastructure in which we can authenticate a public key by using *digital certificates*[6),7)]. In public-key cryptosystems, we have to obtain other users' public keys to securely communicate with those users. In PKI systems as shown in Fig. 1, we can verify that pk_v is the public key of v by using pk_u.

One of the problems in adopting ordinary PKI systems in ad hoc networks is that 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 that administers the PKI system in the network is only one node. Some methods assuming certificate authorities by using threshold signatures have been proposed[21),22)]. However, these methods may cause the 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 Pretty Good Privacy (PGP)[23)] in which each node can issue certificates to others in a self-organizing manner. Some authors have proposed web-of-trust-type systems for ad-hoc networks where each node has a distributed repository instead of a centralized repository[3),8),13),14)]. However, these systems suffer from the common problem 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 node's public key is called a *certificate-chain*[4)]. 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* (See Section 2 for the definitions of source node, destination node, and certificate-chain). Even if the source node does not directly sign the public key, the source node is able to verify a public key by discovering a certificate-chain from the source node

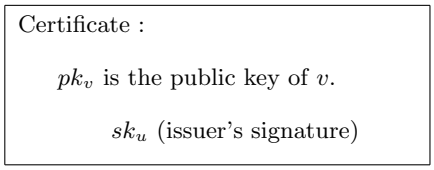
† Graduate School of Information Science, Nara Institute of Science and Technology
☆ Presently with Murata Manufacturing Company, Ltd.
☆☆ Presently with Kochi University of Technology

to the destination node because the trust relation represented by certificates is *transitive*. 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 a 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 discovery problem in ad hoc networks. The contributions of this paper are as follows.

- We give a formal model for PKI in ad hoc networks (section 4.1). A web-of-trust-type PKI system is defined as a weighted directed graph where an edge represents the direct trust relation guaranteed by a certificate between the nodes and the weight of the edge represents the number of hops in the lower (physical) layer.
- A new distributed algorithm for solving the certificate-chain discovery problem is proposed based on the model (section 4.2). We also propose a modification of the proposed method (section 6).
- Performance evaluations of the existing and proposed methods have been conducted under assumptions on packet/certificate sizes that are more realistic than the previous ones (sections 5, 6 and 7). The results have shown that the communication cost of the proposed algorithm is less than ten percent of that of the existing method.

After stating the certificate-chain discovery problem in section 2, we review related work in section 3. In section 4, we model a web-of-trust-type PKI system, show 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 a modification scheme of the basic scheme and compare the communication cost of these methods. For large-size networks, we derive formulae that approximately represent the communication costs of the existing and proposed methods and numerically evaluate the formulae in section 6. For moderate-size networks, communication costs are evaluated by computer simulation on a randomly generated unit disk graph as a routing graph and a random Hamilton graph as a trust model in section 7. In section 8, we discuss the robustness of the proposed method. Finally, we conclude this paper in section 9[☆].



v : A user
pk_v : The public key of u
u : The issuer (the signer of the certificate)
sk_u : The secret key of u (to sign)
pk_u : The public key of u (to verify)

Fig. 1 A certificate for v issued by u

2. Problem Statement

A certificate C is a data structure including a public key, ID of the owner of the public key, and the signature by a signer. For a certificate C, the owner of a public key is called the *user of* C and its signer is called the *issuer of* C (See Fig. 1). For simplicity, we write a certificate of which issuer and user are u and v respectively as $\langle u, v\rangle$.

We investigate a web-of-trust-type PKI system[8)~10),13),14)] where every node v has a repository and stores only the certificates of which v is the issuer or the user (see Fig. 1). Each node u stores certificates such that u is an issuer or a user of the certificates in our PKI system, and each node does not have a list of trusted nodes. This enables us to reduce communication cost for obtaining certificates to be

[☆] Sections 2, 4, 5, and 6 are partially based on our previous work[18)].

stored in a repository in advance. Moreover, this method reduces the cost of the certificate revocation phase. In this method, a certificate is only held by its issuer and user (the owner of the public key in the certificate). 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*). In this paper, we use "trust" as a trust relation between the issuer and the user of a certificate. That is, "an issuer trusts a user to be honest and to correctly authenticate the owner of a public key before signing it[15]." By a certificate-chain and the trust relation in a certificate, a source node can verify the public key of a destination node even if the source node does not issue the certificate in which user is the destination node.

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 the *certificate-chain discovery problem in ad hoc networks* (Definition 1). Assume that a node u wants to communicate with another node v. If u does not directly trust v, u has to find a certificate-chain from u to v. A *certificate-chain from u to v* is a sequence of certificates $\langle u_0, u_1\rangle, \langle u_1, u_2\rangle, \ldots, \langle u_{l-1}, u_l\rangle$ $(l \geq 1)$ such that $u = u_0$, $v = u_l$, and the user of the certificate $\langle u_{i-1}, u_i\rangle$ is the issuer of the next certificate $\langle u_i, u_{i+1}\rangle$. u_0 can verify the public key pk_{u_1} by the certificate $\langle u_0, u_1\rangle$. Also, u_0 can verify the public key pk_{u_2} by using the certificate $\langle u_1, u_2\rangle$ and the verified public key pk_{u_1}. By performing this verification repeatedly, u_0 can get the public key pk_{u_l}. We say u_0 and u_l are the *source node* and the *destination node* of the certificate-chain, respectively. The source node u can verify the public key pk_{u_1} based on a technique of the public-key cryptography (digital signature) and the trust relation ("u trusts u_1"). Moreover, u can verify the public key pk_v based on the certificate-chain and the trust relation "u_i trusts u_{i+1}"). If we do not assume the trust relation, the source node u has to consider another method to verify a certificate in which u is neither the issuer nor the user. If u cannot find a certificate-chain from u to v, u cannot verify the public key pk_v. Therefore, the trust relation is necessary to consider certificate-chain-discovery methods.

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 v has a repository and stores certificates in which v is the issuer or the user. 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[13),14)] that limit the issuing of certificates for simplifying the problem in Definition 1. However, such methods have the problem that a node cannot always issue certificates based on the trust relation among the nodes.

Kitada et al. proposed a public key management scheme for ad hoc networks[8)~10)]. Their proposed scheme is able to reduce the communication cost in the certificate revocation phase more than the method proposed by Capkun et al.[3)]. Kitada et al. also proposed the Ad hoc Simultaneous Nodes Search (ASNS) protocol to resolve the problem stated in Definition 1.

ASNS finds a certificate-chain as follows.

- The source node broadcasts a 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 search is proportional to the number of certificates.

However, ASNS has the following shortcomings. In distributed networks such as ad hoc networks, the protocol is completed, not when

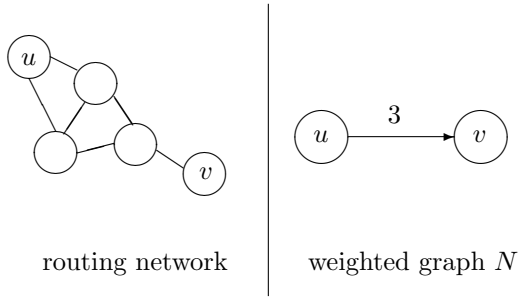


Fig. 2 Relation between routing network and weighted graph N

the destination node is discovered, but when all nodes in the network receive the packet. Thus, ASNS may have a heavy communication cost because of broadcasting packets with certificates.

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

A model of a web-of-trust-type PKI 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.

A node v in V represents a node in ad-hoc networks. An edge $u \rightarrow v$ in E represents a certificate from u to v. The weight $\phi(\langle u, v\rangle)$ of an edge $\langle u, v\rangle$ represents the number of hops from node u to node v (see Fig. 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. Thus, all nodes receive a search packet with a number of certificates whether or not a node needs the certificates. In this paper, we divide the certificate-chain discovery problem into the certificate searching phase and the certificate collecting phase, and propose a new algorithm for each phase.

4.2.1 Certificate Searching Phase

We assume that each node 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. The communication complexity of standard algorithms for constructing a spanning tree is $O(|E|)$, where $|E|$ is the number of elements of E[11].

Kitada et al. showed that the Kitada method can find a certificate-chain in a usual ad-hoc network with topology change by computer simulation[8]. This is because there are some certificate-chain from a given source node to a given destination node. Assume that a given source node u wants to find a certificate-chain to a given destination node v. Usually, there are more than one certificate-chains from u to v in a trust model. In the certificate searching phase, we construct a spanning tree on the trust model. When an intermediate node in a certificate-chain does not receive a search packet from u, another certificate-chain is found by the distributed algorithm constructing a spanning tree. Hence the probability that the distributed algorithm cannot find any certificate-chain from u to v in the trust model is low.

4.2.2 Certificate Collecting Phase

When the certificate searching phase is completed, each node knows which node is the parent in the constructed tree. However, no nodes, including the source node, 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.

☆ Note that an edge $\langle u, v\rangle$ in a trust model does not mean that node u can directly communicate with node v.

- 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.

5.1 Preliminaries

5.1.1 Definition of the Communication Cost

In Kitada's work[8], communication cost is defined as the number of packets. This definition does not consider the size of a certificate and the number of certificates in a packet. This definition is not realistic because a packet that includes several heavy certificates is counted as "one packet". Thus, we define 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.

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

That is, we define the communication cost as the total message bits by taking the size of a certificate and the number of certificates in the packet into account.

5.1.2 Cost Tree

To estimate the communication cost of the two methods, we consider a *cost tree*. A cost tree is a balanced tree such that the root node is a source node, the number of nodes is $2|V|$ where $|V|$ is the number of nodes in a trust model, and the degree of a node is m where m is the average number of certificates issued by the node. This tree represents search packet flows from a given source node to all other nodes on the trust model.

In both of the two methods, a source node u broadcasts a search packet p to nodes that u directly trusts in the certificate searching phase. When a node receives more than one packet sent by an identical source node u, all the packets except the first one are discarded. Even after a destination node has been found, the other nodes do not know it and thus the distributed algorithm for constructing a spanning tree on the trust model does not halt until all the packets are discarded. Therefore the total number of nodes in the cost tree is equal to $2|V|$.

Though we assume a cost tree is balanced, the distributed algorithms in both methods may not construct a balanced tree because an ad-hoc network in the real world is asynchronous. If we assume an unbalanced cost tree instead, the height of the unbalanced tree is larger than a balanced tree with the same number of nodes. Thus, our method outperforms the Kitada method for an unbalanced tree more than a balanced tree because certificates are attached to search packets. Therefore we assume a balanced tree as the cost tree so that comparison of the two methods is not disadvantageous to the Kitada Method. The number of nodes $|V|$ and the number of edges $|E|$ (i.e., the number of certificates) in a trust model can be represented by using the height k of a cost tree and the average number m of the node degree in the cost tree as follows:

$$2|V| = \frac{m^k - 1}{m - 1},$$
$$|E| = m \times |V|.$$

5.2 Analysis of the Kitada Method

In this subsection, we examine the Kitada method. Though they analyzed their method[8], it was based on a communication cost that did 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 the 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 a chain is discovered. That is, the following relation holds between the length and the height:

$$\begin{aligned}&(\text{the length of a certificate chain in a spanning tree})\\ \leq\ &(\text{the height of the tree}).\end{aligned}$$

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.

5.2.1 Certificate Searching Phase

In this phase, the source node broadcasts a 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 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.

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

5.2.2 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:

$$C_1(k) = n(k \times Cert + Cert\_rpl). \quad (1)$$

5.3 Analysis of the Basic Scheme

5.3.1 Certificate Searching Phase

In the basic scheme, a source node constructs a spanning tree using any distributed algorithm, and no certificates are added to a packet. Therefore, the cost $S_2(k)$ is as follows:

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

5.3.2 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 the Two Methods

5.4.1 Complexity Analysis

We analyze the fraction of the 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 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.

5.4.2 Numerical Analysis

We also compare the costs by numerical analysis. We let $n = 4$, $m = 4$, and $Cert = 2050$. Kitada et al. estimated that the average number of hops (n) for realistic ad-hoc networks is at most around four. In the Kitada's paper, they empirically showed that a trust model is possibly not connected if $m < 4$. Hence we only consider $m \geq 4$. Note that since a trust model is given independently of the trust establishment algorithm, we should compare the proposed method with the Kitada method with an identical value for m. Because the ratio of the cost of the Kitada method to ours is in proportion to m as shown in 5.4.1, $m = 4$ is the most advantageous setting for the Kitada method. In the RSA public key cryptosystems, the size of a public key is 1024bits and the size of each cipher text (or signature) is more than 1024bits. Therefore the size of a certificate is more than 2048bits[☆]. Also, we let $Cert\_req = 100$ and $Cert\_rpl = 100$. We assume the size of each ID as 10bits. These 10bit length IDs can identify $2^{10} = 1024$ nodes, which is large enough for an ordinary ad-hoc network. Each search packet $Cert\_req$ includes a source node, destination node, and pairs of the next node on the trust model and the next hop on the routing graph[8]. The number of pairs of the next node and the next hop is not a constant and depends on the number of certificates issued by each node (the degree of a node in a trust model). According to Kitada et al.'s estimation that the average degree of nodes to construct a web-of-trust is four[8], we assume that

[☆] In the real world, a certificate includes ID of the issuer and owner of a public key, a timestamp of expiration date and so on, and the size of a certificate may become larger than 2050bits.

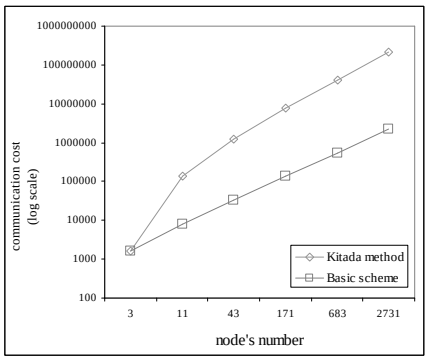


Fig. 3 Basic scheme vs. the Kitada method (searching cost)

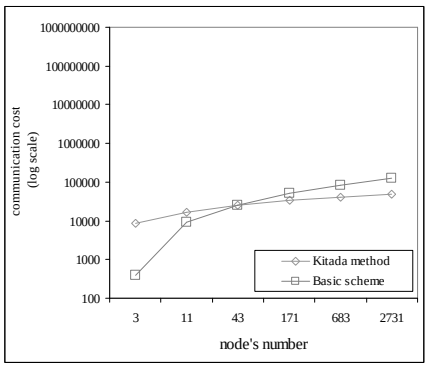


Fig. 4 Basic scheme vs. the Kitada method (collecting cost)

the average number of pairs of the next node and the next hop in a search packet is four. Therefore, the average size of a search packet is $10 + 10 + 4 \times (10 + 10) = 100$. On the other hand, each reply packet $Cert\_rpl$ may be less than 100bits. However, our method outperforms the Kitada method for less than 100bits more than it does for 100bits because $Cert$ divided by $Cert\_rpl$ is larger. Therefore, we assume that $Cert\_rpl$ is also 100bits so that the comparison of the two methods is not disadvantageous to the Kitada Method.

Fig. 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 Fig. 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 the basic scheme is higher than the Kitada method. Finally, in Fig. 5, we compare the total costs of the two methods. We can see that the basic scheme has a lower cost than the Kitada method.

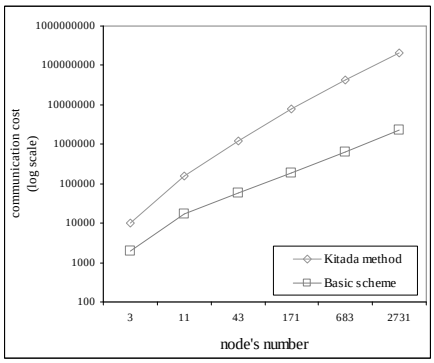


Fig. 5 Basic scheme vs. the Kitada method (total cost)

As a comparison in a more realistic environment, we also compared the proposed method and the Kitada method by computer simulation in section 7.

6. A Modification Scheme of the Proposed Method

The basic scheme has a disadvantage on the cost of the collecting phase. To reduce the cost, we revise the collecting phase of the basic scheme. 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 avoid this overhead, we modify the phase as follows:

- 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 no node knows whether the node itself is on the certificate-chain to the destination node).

After this modification, the cost $S_3(k)$ of the certificate searching phase is the same as $S_2(k)$, and the cost $C_3(k)$ of the collecting phase is as follows.

$$C_3(k) = (k-1) \times n \times (Cert + Cert\_rpl)/2)$$

We also investigated another modification in our previous work[18]. An idea of the other

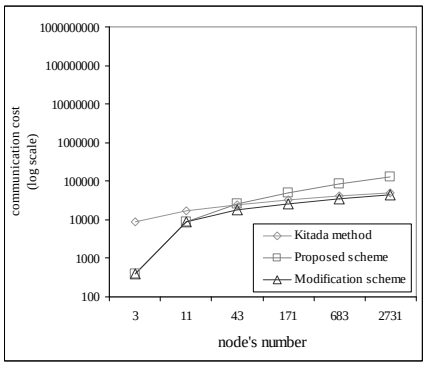


Fig. 6 Comparison of collecting costs

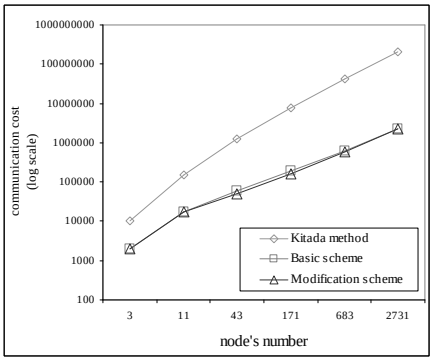


Fig. 7 Comparison of total costs

modification is to use a distributed algorithm for constructing a shortest path tree such as the distributed Bellman-Ford algorithm in the searching phase. However, the upper bound of the computation time for the distributed Bellman-Ford algorithm is $O(V \cdot E)$[11] and the one for the standard distributed Dijkstra algorithms is $O(V^2)$[17]. Because these costs are much higher than the costs of algorithms for spanning trees, the method using a distributed algorithm for constructing a shortest path tree require more total cost than our basic scheme and the Kitada method.

6.1 Numerical Analysis

We compare the above methods by numerical analysis. We let $n = 4$, $m = 4$, $Cert = 2050$, and $Cert\_req = Cert\_rpl = 100$.

The cost of the modification scheme is the same as the cost of the basic scheme and is lower than the Kitada method (see Fig. 3). Fig. 6 shows a graph of the cost of the collecting phase. The modification method also has a lower cost than the Kitada method in this phase.

We compare the total costs of all the above methods (Fig. 7). From this comparison, we obtain the following result:

$$\begin{aligned} & S_1(k) + C_1(k) \quad \text{(the Kitada method)} \\ > & S_2(k) + C_2(k) \quad \text{(basic scheme)} \\ > & S_3(k) + C_3(k) \quad \text{(modification scheme).} \end{aligned}$$

Note that for (1) and (2), $C_1(k) \leq C_3(k)$ if and only if $k \geq \frac{Cert}{Cert\_rpl} + 2$. We assumed that $Cert = 2050$ and $Cert\_rpl = 100$ and hence $C_1(k) < C_3(k)$ if $k > 22$. Fig. 6 conforms to this result. The number of nodes in a network is already more than 100,000 when $k = 9$ and is out of range in Fig. 6.

7. Simulation Results

The numerical analysis in Section 6 showed that the modification scheme requires the least communication cost among the existing and proposed methods. In this section, we compare the modification scheme and the Kitada method more precisely for moderate-size networks. We describe the simulation scenarios, and then we show the average weight in the trust model and the communication costs of the two methods. All simulations were performed with a simulator implemented in Java with a java library of graph algorithms and optimization[12].

7.1 Simulation Scenario

We use the following simulation scenarios.

- The number of nodes $|V|$ are 20, 40, 60, 80, and 100.
- The number of certificate $|E|$ is $4 \times |V|$.
- The power range of each node is 100.
- The simulation regions are as follows.
  - 100×100 ($20 \leq |V| \leq 100$),
  - 200×200 ($20 \leq |V| \leq 100$),
  - 300×300 ($20 \leq |V| \leq 100$),
  - 400×400 ($40 \leq |V| \leq 100$),
  - 500×500 ($60 \leq |V| \leq 100$),
  - 600×600 ($n = 100$).
- The routing graph is a *unit disc graph*. Formally, a unit disc graph is the intersection graph of a set of unit diameter closed disks in the plane[2]. Generally, a unit disc graph is not always connected, which means that an ad hoc network itself is not formed. Because we focus on the cost of the two methods in this section, we used connected unit disc graphs.
- The trust model is a *random Hamilton graph* where each weight of an edge $\langle u, v \rangle$ is defined as the shortest path length from i to j in the unit disc graph (see Definition

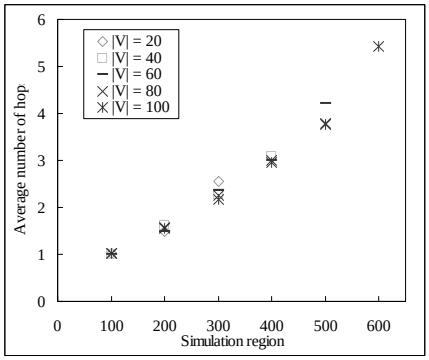


Fig. 8 average number of hops in a unit disc graph

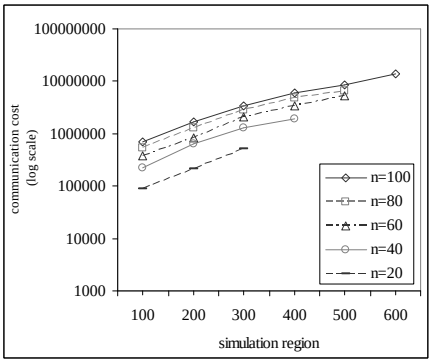


Fig. 10 Average searching cost of the Kitada method

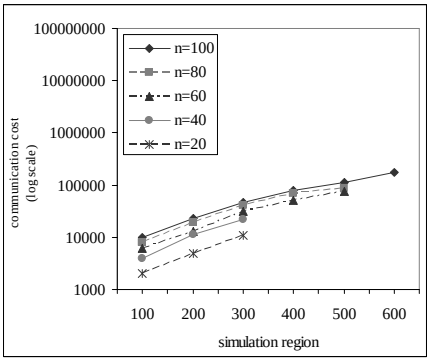


Fig. 9 Average searching cost of the Modification scheme

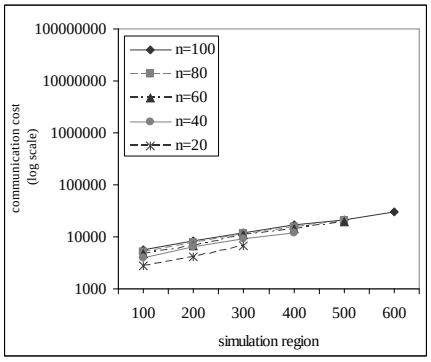


Fig. 11 Average collecting cost of the Modification scheme

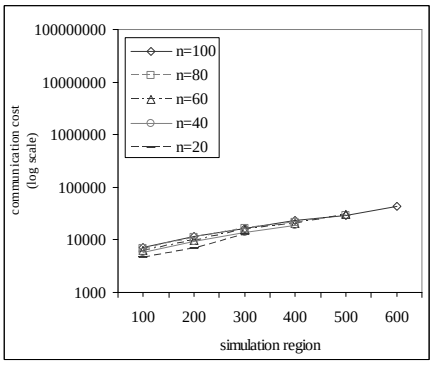


Fig. 12 Average collecting cost of the Kitada method

2). Kitada et al. assume a trust model as a strongly connected graph in simulation scenario, so we use a random Hamilton graph as an example of such graphs.

- The packet size is 100bits,
- The certificate size is 2050bits.

7.2 Results

7.2.1 Average Number of Hops

We show the average number of hops in a unit disc graph in Fig. 8. We evaluate the number of hops by computing the average of the shortest hops between arbitrary pairs of distinct nodes in a unit disc graph. When the simulation region is 500×500, the number of hops nearly equals four, which matches the number of hops ($m = 4$) adopted in the numerical analysis (see sections 5.4.2 and 6.1).

7.2.2 Comparison of the two methods

We show the communication costs of the modification scheme and the Kitada method by simulations in Fig. 9–13. The modification scheme outperforms the Kitada method also in the simulation results. Below we give detailed comparisons.

Fig. 9 and 10 show the costs of the searching phase. The cost of the modification scheme is much lower than the cost of the Kitada method. In the searching phase of both methods, a source node constructs a spanning tree. For simplicity, we use a shortest path tree as the constructed spanning tree. A constructed spanning tree is not always a shortest path tree because of node failure or non-uniform communication delay. However, the probability of not constructing a shortest path tree is not high. Furthermore, whether we use a shortest path

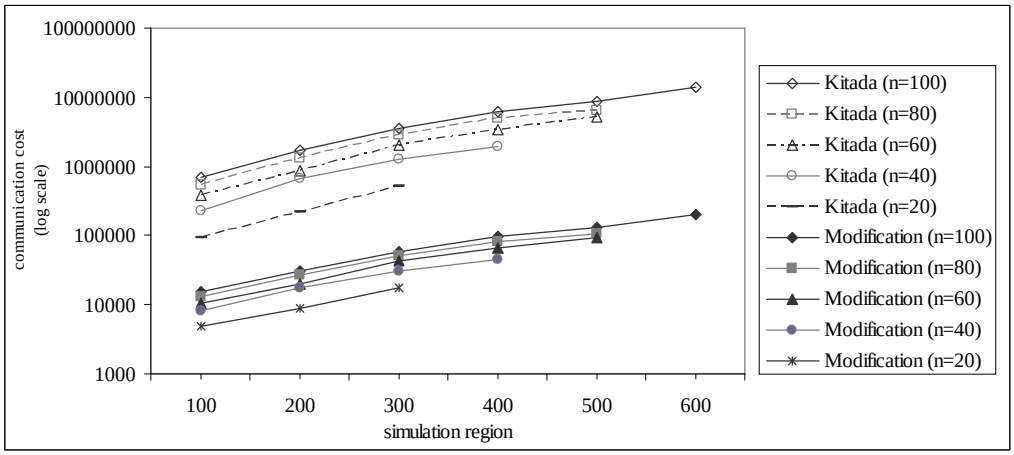


Fig. 13 Modification scheme vs. the Kitada method (total cost)

tree or not does not largely affect the comparison result of the two methods. Fig. 11 and 12 show the costs of the collection phase. Also for this phase, we evaluate the costs by using a shortest path tree. We show the total cost of the two methods in Fig. 13. We evaluate the average of the total cost as the sum of the average searching cost and the average collecting cost. The simulation results showed that the total cost of the modification scheme is less than 10% of the cost of the Kitada method.

8. Security Consideration

In this section, we discuss the robustness of our method to some known attacks.

Sybil attack In a distributed network without a trusted third party maintaining identities (ID), a malicious node can have not only a legitimate ID but also one or more counterfeit IDs. This attack is called the Sybil attack[5]. If a node succeeds in the Sybil attack, it can improperly raise its ranking in a reputation system[16] by voting itself using counterfeit IDs, for example.
We consider two variations of the Sybil attack. One is that a malicious node obtains one or more counterfeit IDs but it has only one (legitimate) public key (and its corresponding secret key). The other is that a malicious node obtains both of counterfeit IDs and public keys. The former case is easy to treat. Remember that a certificate binds the ID and the public key of a user. Hence, if we find two certificates that bind different IDs with the same public key, we know that the user of the public key is malicious. For the latter case, it is generally impossible to find out whether or not the Sybil attack occurs since no node can always know the physical relation between a public key and its owner. However, our PKI system can find a certificate-chain correctly as long as every node (including the attacker of the Sybil attack) faithfully relays packets. The public-key exchange and the creation of a certificate are usually done through a side channel (e.g., over an infrared channel at the time of a physical encounter) before constructing an ad-hoc network[3]. Therefore the threat of the latter type of Sybil attack is not serious.

Man-in-the-middle attack When nodes A and B exchange their public keys pk_A and pk_B via a key exchange protocol, a malicious intermediate node M may be able to give A pk_M instead of pk_B and give B pk_M instead of pk_A. If this attack succeeds, M can eavesdrop on communications between A and B. However, in ordinary PKI systems, a certificate $\langle u, v \rangle$ is issued if u trusts v to be honest and has correctly authenticated v as the owner of its public key before signing it[15]. Moreover, in web-of-trust-type PKI systems, the source node can verify the public key of the destination node by a certificate-chain. Therefore, web-of-trust-type PKI systems including our method can prevent the man-in-the middle attack.
Also, when the public key exchange between A and B is completed, this attack would be foiled with public keys because the man-in-the-middle would not have the private key to be able to decrypt messages

encrypted under A's public key[19].

Denial of service attack Denial of service (DoS) attacks disable routing protocols of ad-hoc networks, and some authors have investigated efficient DoS attacks for ad-hoc networks[1]. The main aim of our method is to get the public key of another node securely and to prevent the eavesdroppings of malicious nodes, so the resilience against DoS attacks is out of the scope of our research. However, we do not assume a perfectly reliable routing on ad-hoc networks, i.e., the proposed method works even if a routing is unreliable.

No single security system can prevent all kinds of attacks. Generally, we combine several security techniques against these attacks. The main aim of our method is to provide a secure *End-to-End encryption*[19]. In the End-to-End encryption, the sender of a message and its receiver should have a shared key or the sender should have the public key of the receiver. Because a message is encrypted by a shared key between the sender and the receiver, even an intermediate host cannot decrypt it. This encryption scheme does not depend on routing protocols, so we usually can use some key management and exchange protocol on routing protocols. On the other hand, there is *Link encryption*[19] such that the sender and the receiver and each intermediate node should share a key. Because every block of a packet, not only the data part but also the header part, is encrypted in the Link encryption, each intermediate node has to decrypt the packet to check the header part and encrypt it before sending it to next node. In the encryption, each intermediate node can read not only the header part but also the data part because each intermediate node has the shared key and can decrypt the packet. We investigate a secure End-to-End encryption scheme in ad-hoc networks, where we should use keys and key management schemes distinct from those in the Link encryption for security. Though the Link encryption is also important, we do not consider it in this paper.

9. 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 a modification for solving the problem. Furthermore, we proposed a measure for the communication cost, and according to the measure, we compared our algorithms with the Kitada method by numerical analysis. The result of the numerical analysis is summarized in Table 1. To evaluate the performance of the modification scheme in a more realistic environment, we evaluated it and the Kitada method by simulation. The simulation results showed that the modification scheme requires a lower cost than the Kitada method.

Table 1 The proposed schema vs. the Kitada method

Method Name	Search	Collect	Total
Basic	✓		✓
Modification	✓	✓†	✓

Note : "✓" means that the cost is lower than the Kitada method.
† $C_1(k) > C_3(k)$ if and only if $k < \frac{Cert}{Cert\_rpl} + 2$.

Unfortunately, existing routing protocols for ad hoc networks are unable to catch up with frequent link changes[20]. These protocols minimize the effect of the dynamic change of the topology caused by nodes' mobility by reducing time, communication, and round complexity. The proposed methods also address node mobility by reducing such complexities as existing routing protocols do.

Our future work will include 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.[3] and Kitada et al.[8].

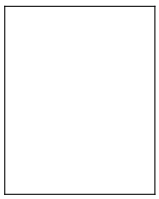
Acknowledgments The authors would like to thank Dr. Toshimitsu Masuzawa, Professor at Osaka University, for helpful comments about distributed algorithms. The authors also would like to thank Dr. Keiichi Yasumoto, Associate Professor at Nara Institute of Science and Technology, for helpful comments about ad-hoc Networks.

References

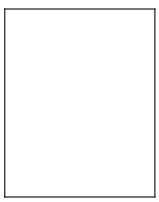
1) I.Aad, J.P.Hubaux, and E.W.Knightly, "Denial of Service Resilience in Ad Hoc Networks," ACM Annual International Conference on Mobile Computing and Networking (MobiCom), pp.202–215 (2002).
2) H. Breu and D. G. Kirkpatrick, "Unit Disk Graph Recognition is NP-hard," Computational Geometry: Theory and Applications, **9**, pp.3–24 (1993).
3) 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, pp.52–64 (2003).
4) 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, **4**, 9, pp.285–322 (2001).
5) J.R.Douceur, "The Sybil Attack," The 1st International Workshop on Peer-to-Peer Systems (IPTPS), LNCS 2429, pp.251–260 (2002).
6) 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).
7) L. M. Kornfelder, "Toward a Practical Public-Key Cryptosystem," bachelor's thesis, Dept. Electrical Eng., Massachusetts Inst.of Technology, Cambridge (2005).
8) 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, pp.1571-1583, (2005).
9) 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).
10) 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), pp.375-381 (2005).
11) N. A. Lynch, "Distributed Algorithms," Morgan Kaufmann Publishers, San Francisco, California (1996).
12) H. T. Lau, "A Java Library of Graph Algorithms and Optimization," Chapman & Hall/CRC, Boca Raton, Florida (2006).
13) X. Li, S. Gordon, and Jill Slay, "On Demand Public Key Management for Wireless Ad Hoc Networks," Australian Telecommunication Networks and Applications Conference (ATNAC), pp.36–43 (2004).
14) R.Li, J.Li, H.Kameda, and P.Liu, "Localized Public-Key Management for Mobile Ad Hoc Networks," IEEE Global Telecommunications Conference (Globecom), pp.1284–1289 (2004).
15) U. Maurer, "Modelling a Public-Key Infrastructure," European Symposium on Research in Computer Security (ESORICS), LNCS 1146, pp.325–350 (1996).
16) N. Mezzetti, "A Socially Inspired Reputation Model," European PKIWorkshop: Research and Applications (EuroPKI), LNCS 3093, pp.191–204 (2004).
17) 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, pp.21–32 (1994).
18) H. Mohri, I. Yasuda, Y. Takata, and H. Seki, "Certificate Chain Discovery in Web of Trust for Ad Hoc Networks," IEEE International Symposium on Ubisafe Computing (UbiSafe), to appear (2007).
19) C. P. Pfleeger and S. L. Pfleeger, "Security in Computing," Prentice Hall, Upper Saddle River, New Jersey (2006).
20) C.K.Toh, "Ad-Hoc Mobile Wireless Networks: Protocols and Systems," Prentice Hall, Upper Saddle River, New Jersey (2001).
21) 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), pp.52–61 (2004).
22) L. Zhou and Z. J. Hass, "Securing Ad Hoc Networks," IEEE Network, **13**, 6, pp.24–30, (1999).
23) P. Zimmermann, "The Official PGP User's Guide," MIT Press, Cambridge, Massachusetts (1995).

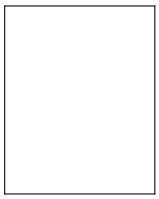
(Received April 2, 2007)
(Accepted October 2, 2007)



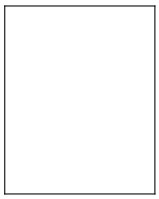
Hisashi Mohri received a B.E. degree from Nara University of Education in 2003 and received an M.E. degree from Nara Institute of Science and Technology in 2005. In 2005, he entered the Ph.D. course of Nara Institute of Science and Technology. His research interest includes security in ubiquitous computing environments.



Ikuya Yasuda received a B.S. degree from Nara Women's University in 2005 and received an M.E. degree from Nara Institute of Science and Technology in 2007. She joined the Murata Manufacturing Company, Ltd, in 2007. Her research topic in the master's course was security techniques for ad-hoc networks.



Yoshiaki Takata received a D.E. degree in information and computer sciences from Osaka University in 1997. He was an Assistant Professor at Nara Institute of Science and Technology from 1997 to 2007. In 2007, he joined the faculty of Kochi University of Technology. His research interest includes formal specification and verification of software systems.



Hiroyuki Seki received a D.E. degree from Osaka University in 1987. He was with Osaka University as an Assistant Professor in 1990-1992 and an Associate Professor in 1992-1994. In 1994, he joined the faculty of Nara Institute of Science and Technology, where he has been a Professor since 1996. His current research includes formal language theory and formal approach to safe and secure system design.