Certificate Issuing Using Proxy and Threshold Signatures in Self-initialized Ad Hoc Network*

Jeonil Kang, DaeHun Nyang**, Abedelaziz Mohaisen, Young-Geun Choi, and KoonSoon Kim

Information Security Research Laboratory,
INHA University, Republic of Korea
dreamx@seclab.inha.ac.kr, nyang@inha.ac.kr
{asm, choizak, soony}@seclab.inha.ac.kr

Abstract. In ad hoc network, it is very crucial to issue certificates safely in the self-initialized scheme where the system authority exists only at the beginning of the network operation. In order to solve this problem, early studies have presented some suggestions by removing the system authority itself and using certificate chain, or by making nodes act as system authorities for issuing other nodes' certificates. In this paper, using proxy and threshold signatures, we introduce a certificate issuing scheme that can solve many problems in the previously proposed solutions. We demonstrate our scheme's performance through the simulation results. Also, we discuss the security value of our scheme's in various aspects.

1 Introduction

In ad hoc network, certificates of public keys have been a general solution for key distribution and authentication. However, single point of failure and bottleneck problems which are caused by single certificate authority makes it hard to apply certificate-based public key cryptography to ad hoc network. To solve these problems, many researches have been performed [1,2,3]. One of the solutions is self-initialized (a.k.a. self-organized) certificate management scheme in which each node generates its own public and private keys. Due to the individual generation of keys, the central node which has to issue the certificates for other nodes can distribute its burden to other nodes. For the authority itself, there are two types of configuration. In the first configuration, the authority never exists all through the operation of the network while in the second configuration it only exists at the beginning of the network operation. For the first configuration, it is required to provide an additional method to verify other certificates because these certificates are independently issued by different nodes and signed by different private keys. The researches of Capkun *et al.* [1] and Li *et al.*[2] are well-know in handling this configuration. For the second case where the system authority exists only at the beginning of the network, it does not require much cost for verifying certificates

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)(IITA-2006-C1090-0603-0028).

** Corresponding Author

O. Gervasi and M. Gavrilova (Eds.): ICCSA 2007, LNCS 4707, Part III, pp. 886–899, 2007.

because all certificates are singed by a single private key. However, the problem that a node can sign certificates after the beginning still remains. To solve this, the *diffusion wave* method as by Luo *et al.* [3] has been considered.

In the diffusion wave method, the node which gathers several partial certificates and makes a complete certificate gains the authority for issuing other partial certificates. Therefore, the population of nodes that have the authority increases gradually. However, the diffusion wave may have another problem named the *originality of authority* in which a misbehaving node may obtain a valid certificate legally. The originality of authority problem may also lead to another problem in which a misbehaving node moves to other place in the network and misuse the network resources or mislead other nodes using its own valid certificate. Also, from the complexity perspectives, the diffusion wave requires much network traffic because some single nodes should gather several partial certificates and find other nodes for one certificate.

In this paper, we consider the complexity of the diffusion wave based scheme. To reduce the former work's complexity represented in the communication overhead, we introduce a novel certificate issuing scheme based on the proxy and threshold signatures. In addition to the communication reduction, our scheme provides a breaking-through for the required time for the signing operation.

2 Related Works: Self-initialized Certificate Management

2.1 Self-initialization

In the self-initialization network scenario, a node in the network generates its private and public key pair. To use the self-generated key pair, a node must ensure that other nodes' private keys are not modified by a third entities. On other side, even though two nodes are in one hop neighborhood, they cannot be perfectly sure of that in application layer unless they have additional protocols. Generally, certificates are used for informing the public key which is used when a node confirms the integrity of the message signed by the corresponding private key. However, the certificates must be signed by a trusted private key.

2.2 Trusted Certificate Chain

If there are not any system authority in the network and the certificates are individually signed by different private keys, a node must have the way or simply the criteria upon which it trusts other public keys. Unfortunately, on the side of the node, nothing can be trusted but its own private and public key pair. However, once a certificate issuing is performed, the issuer himself trust the public key associated with the issued certificate. In addition, the issuer can trust other certificates that can be verified by the trusted public key. In that way, a node can make trusted certificate chain started from its own keys as rewards for issuing other nodes' certificates.

To find out a trusted certificate chain between two nodes A and B which both have their own chains, Capkun *et al.* suggested a method in which the resulting chain of concatenating two sub-chains beginning from an intersecting point of the two chains and

ending by the two nodes' corresponding certificates (i.e., certificate of A and certificate of B respectively) is used [1]. In this method, however, each node should have a number of certificates for making sub-chain. Also, even each node has that number of certificates, this does not ensure the existence of the trusted certificate chain.

In another method suggested by Li *et al.*, to establish the certificate chain, the routing path is used to forward the certificate request message using the resulting routing path if a node issues certificate for all one-hop neighbors. If a node sends the certificate request message, all nodes on backward route add their certificates in the response message sequentially. As a node, it is enough to store the neighbors' certificates only. However, this increases the total network overhead even under usual communication pattern in local area [4].

2.3 Certificate Signed by System Private Key

If all certificates are signed by the system authority's private key, all certificates are verifiable by the system authority's public key that all nodes know. However, it is hard to expect that the authority is always in the same coverage area because of the nodes' mobility. Also, if an attacker succeeds in cracking the authority, the whole system will collapse. Therefore, it is better that the system authority's private key should be fractionized and distributed to several nodes while no node possesses the whole private key. To obtain a complete certificate, a node should gather several partial certificates signed by partial private keys leading to that a few nodes that have a partial private key suffers from lots of service requests. Also, a node can be delayed from obtaining a complete certificate due to the deficiency of the nodes that have partial private keys.

Luo *et al.* introduce a method named the diffusion wave (illustrated in Fig. 1(b)) to reduce the time delay for obtaining a complete certificate. In their work, they enable the node that gains a complete certificate to have the authority for issuing a partial certificate. By doing so, the burden of a few nodes can be decentralized to many other nodes. However, this cannot reduce the overall network traffic. Also, an attacker can meet nodes which have a partial private key in the neighborhood and can try attacking them just after small movement. In addition, the diffusion wave should have strict rules for authenticating nodes. As time goes on, the originality of authority will be degraded so that we cannot expect those strict rules to apply to the nodes which newly join in the network.

3 Proposed Certificate Issuing Scheme

3.1 Preliminaries

Secret Sharing and Threshold Signature: A secret sharing is a cryptographic scheme in which a secret of an entity is distributed as partial secrets among other entities [5]. If many partial secrets over threshold are gathered, the secret can be recovered. Once the a secret is recovered, it is not secret anymore leading to that the secret owner (i.e. the corresponding entity) need to re-make and re-distribute the secret shares among other entities. However, the required operations upon secret recovery are resources' consuming ones. Therefore, for using the secret without recovery, threshold cryptography are suggested.

In the threshold signature scheme, we can obtain a signature of a document by gathering partial signatures over threshold value [6]. This is possible when the signer need to sign a document while revealing the overall secret is undesirable. In this paper, we use the threshold signature scheme for assigning the proxy signature keys to the proxy signers.

Proxy Signature: A proxy signature scheme consists of an original signer, proxy signers, and verifiers [7]. The original signer commits his warrant to sign to proxy signers. According to who can make the same signature as the original one, there are two types of proxy signature schemes which are *proxy-unprotected* and *proxy-protected*. In this paper, we use the proxy-protected signature scheme because the role of the proxy signers is not clear if we assume that the original signer can make proxy signature.

Bilinear Pairing: Consider a group $\mathbb{G}$ of problems for which it is hard to solve CDHP (Computational Diffie-Hellman Problem) but easy to solve DDHP (Decision Diffie-Hellman Problem). Those problems are called GDHP (Gap Diffie-Hellman Problem) [8], and the bilinear pairing is one of their solutions [9].

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be cyclic groups with order of a prime q. Let P be a generator of $\mathbb{G}_1$. If $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ is a multiplicative group, the bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ that satisfies the following properties on elliptic curves over finite field.

- **Bilinear:** $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- **Non-degenerate:** there exists $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$.
- **Computable:** there exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

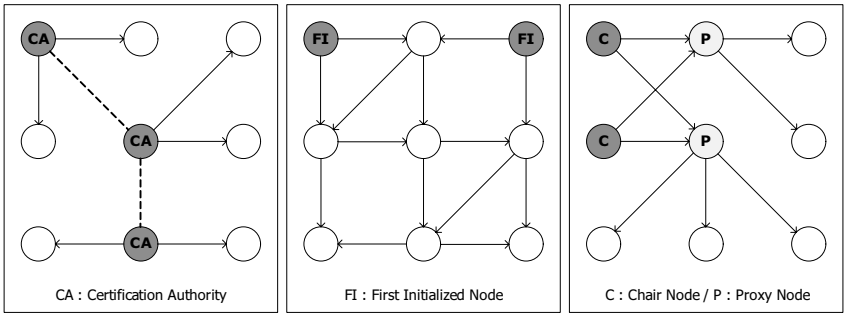
Using those properties, we can easily verify whether $xyP = zP$ or not when we have P, xP, yP and zP on the elliptic curves.

3.2 Basic Protocol

Our proposed scheme is based on the signature schemes and its relative methods described in [10] and [11]. We assume that a system authority exists at the beginning of the network operation. Also, all nodes in the network generate their public and private key pair in self-initialization. The following is the procedure of our scheme:

1) By using secret sharing scheme, the authority shares its partial private key with the first initialized nodes which are called chair nodes. Also, the authority directly assigns proxy signature keys to delegation nodes which are called proxy nodes.
2) By using threshold signature scheme, the chair nodes which are assigned partial private key from the system authority can assign proxy signature key to a proxy node without the system authority.
3) A proxy node which has proxy signature key can issue proxy certificates for normal nodes which should receive their proxy certificates from the proxy nodes only.

A simple illustration of our scheme versus other schemes (i.e. multiple CA and diffusion wave) is in Fig. 1(c).



(a) Multiple CA (b) Diffusion Wave (c) Our Scheme

Fig. 1. Certificate Issuing Schemes

Common Parameters: Consider PK_o and SK_o such that $PK_o = SK_oP \in \mathbb{G}_1$ are the system public and private key pair (P is a generator of group $\mathbb{G}_1$). Also, consider PK_p and SK_p such that $PK_p = SK_pP \in \mathbb{G}_1$ are proxy node's public and private key pair. Then, $(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, P, \mathbf{H}_1, \mathbf{H}_2)$ are common parameters where $\mathbf{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $\mathbf{H}_2 : \{0,1\}^* \rightarrow \mathbb{G}_2$ are two different hash functions.

Proxy Signature Key: The system authority generates a $(t-1)$ degree polynomial $f(x) = SK_o + a_1x + \cdots + a_{t-1}x^{t-1}$ with random coefficients $a_1, a_2, \cdots, a_{t-1} \in_R \mathbb{Z}_q^*$ and the private key SK_o. For all chair nodes $i \in I$ where $I = \{1, 2, \cdots, n\}$, the authority distributes $x_i = f(i)$. If a node collects all elements of a subset $S \subset I$ such that $|S| \geq t$, it can recover the secret key SK_o by using Lagrange interpolation in which $SK_o = \sum_{i \in S} L_i x_i$ where $L_i = \prod_{j \in S} -x_j/(x_i - x_j)$.

The proxy signature key can be assigned for the proxy node by two ways. In the first way, it is assigned by the system authority directly while in the other way it is assigned by the chair nodes.

- From **the system authority:** the system authority makes a warrant w and computes its hash value $\mathbf{H}_2(w)$. The authority then assigns w and $SK_{o_w} = SK_o\mathbf{H}_2(w)$ to a proxy node.
- From **the chair nodes:** the chair node i makes a warrant w for a proxy node and computes $x_i\mathbf{H}_2(w)$. The chair node i then sends the result to the proxy node. The proxy node can make the proxy signature key such that $SK_{o_w} = SK_o\mathbf{H}_2(w) = \sum_{i \in S} L_i x_i \mathbf{H}_2(w)$ if it gathers the partial signatures over t. In this case, the warrant w must not include specific information of the chair nodes for using threshold signature scheme.

The proxy node should check if $\hat{e}(SK_{o_w}, P) = \hat{e}(\mathbf{H}_2(w), PK_o)$. Only when this is satisfied, the proxy node makes proxy signature key $SK_w = SK_{o_w} + SK_p\mathbf{H}_2(w) = (SK_o + SK_p)\mathbf{H}_2(w)$. If DLP (Discrete Logarithm Problem) is hard to solve, then SK_o is hard to be revealed during those process.

Certificate Issuing Through the Proxy Node: A node's certificate that includes its self-initialized public key can be issued by a proxy node only. A proxy node generates a valid proxy certificate $\langle \Lambda, c, U, w \rangle$ for the unsigned certificate Λ where $c = \mathbf{H}_1(\Lambda \| \hat{e}(P,P)^r)$, $r \in_R \mathbb{Z}_q^*$, and $U = cSK_w + rP$. In addition, r, $\hat{e}(P,P)^r$, and rP can be computed in the idle time so that a proxy node needs only one hash computation, one addition, and one multiplication operations on the elliptic curves in real time for signing a certificate.

Certificate Verifying: A node needs to verify other nodes' certificates to communicate with them trustfully. To verify a certificate $\langle \Lambda, c, U, w \rangle$, a node can be sure of the validation of a certificate by checking if equality in Equation (1) is satisfied.

$$c = \mathbf{H}_1(\Lambda \| \hat{e}(U,P)\hat{e}(\mathbf{H}_2(w), PK_o + PK_p)^{-c}) \tag{1}$$

3.3 Other Certificate Management Issues

Certificate Revocation: If the chair or other proxy nodes find a misbehaving node that is harmful to the system or the other nodes, they may exchange that information through the CRL (Certificate Revocation List). There is the information about issuer (e.g. the proxy node) of a revoked certificate in the CRL so that other nodes can know what proxy node mass-produces the certificates for the misbehaving nodes. Therefore, the discipline measure for that proxy node may be desired.

Delegation Deprivation: If some chair nodes want to deprive the authority of a proxy node, they should firstly make a report signed by system private key to notify that report. To do so, they should cooperate with other chair nodes to gather partial reports. Through this cooperation, threshold signature can be used. The report to notify the deprivation has to flood into network for all nodes. If a node receives the report that has the same issuer information with its certificate's issuer, the node must revoke its certificate and try to obtain a new certificate from other proxy nodes.

Also, the chair nodes should assign new proxy nodes when the number of proxy nodes decreases below a specific number.

Certificate Renewal: If the valid date of a certificate is expired or the corresponding deprivation report is arrived, a node can obtain a new certificate from the proxy nodes. If necessary, a node can obtain several certificates in anticipation.

Selection of a New Chair Node: The number of the nodes that have the partial system private keys must be over the threshold value t. Otherwise, no more nodes can be proxy nodes. So that, the chair and proxy nodes cannot maintain their interoperability. Therefore, a new chair node need to be selected though extremely strict authentication rules and a new partial private key should be assigned to that chair node.

4 Simulation

4.1 Simulation Environment

In oder to analyze the performance of our scheme, we built our simulation on the OMNeT++. The simulation parameters are shown in Table 1.

Table 1. Simulation Parameters

Parameters		Explain
Playground Size		$300\times300\text{m}^2$ $600\times600\text{m}^2$
Data Link		IEEE 802.11
Max Interference		74.142m
Routing		AODV-UU
Transport		UDP
Mobility	Type	random way point
	Moving Speed	up to 10 m/s
Num. of Nodes	Diffusion Wave	50 (first initialized nodes: 15)
	Our Scheme	50 (chair nodes: 10, proxy nodes: 5)
Message Length		4096 bits
Schedule Frequency		1 time per $5 \sim 10$ sec
Simulation Time		up to 43,200 sec

Playground Size and Coverage: Let N_t be the total number of nodes, P_{size} be the size of playground, and C_{size} be the coverage area of a node. The average number of neighbor nodes N_n can be obtained as in Equation (2).

$$N_n = \frac{N_t}{P_{size}} \times C_{size} = \frac{N_t}{P_{size}} \times \pi r^2 \tag{2}$$

where r is the maximum interference distance of a node. Note that, N_t/P_{size} refers to the node density on the whole area of playground.

If the playground size P_{size} is $300\times300\text{m}^2$ and the maximum distance for transmission r is about 74m, a node can directly communicate with 9.55 nodes on average. In $600\times600\text{m}^2$ playground, however, a node can directly communicate with only about 2.38 nodes on average.

Mobility and Communication Chances: In our simulation, the node's moving speed increases from 1 m/s to 10 m/s. Even though we assume that one node has a fixed position and another node directly passes through the surrounding of that node, the maximum connection time is 148 seconds in 1 m/s and 14.8 seconds in 10 m/s.

$$\text{Mean Distance} = 2r \times \frac{2}{\pi} \int_0^{\frac{\pi}{2}} \sin\theta d\theta = \frac{4r}{\pi} \tag{3}$$

Therefore, the distance for passing the circumference of a node is about 94.2m on average. So that, the link between two nodes is kept for 94.2 seconds in 1 m/s and 9.4

seconds in 10 m/s. Since one message is sprang up every 7.5 seconds on average, a node has 12.5 chances of communication in 1 m/s and 1.3 chances in 10 m/s.

Transport Protocol: In our simulation, we use UDP (User Datagram Protocol) as a transport protocol instead of TCP (Transmission Control Protocol) even though our scheme needs high integrity. Our selection is based on that the congestion control mechanism of TCP unexpectedly makes lots of burden if the network channel is not error-free like the case of wireless communication environment[12,13]. The integrity of datagram can be guaranteed by re-transmission if the error is detected in the application layer. In spite of the fact that there are several suggestions for TCP implementation in wireless communication environment, we could not be sure of correctness of these suggestions and feasibility due to the un-integrity with our simulator.

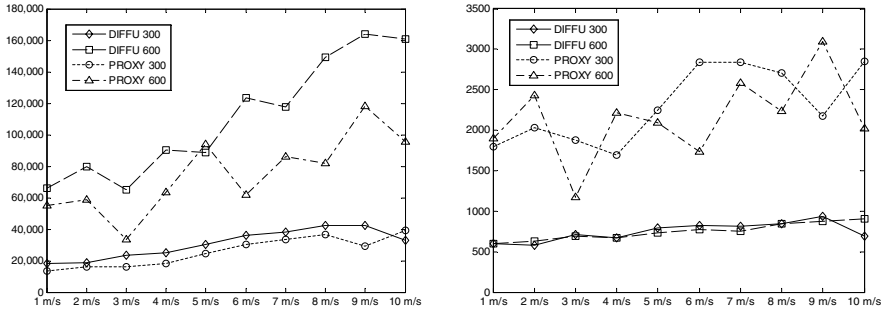
Message Frequency and Operation Time: Basically, the time for encryption and decryption on elliptic curves over finite filed is known to take hundreds of milliseconds on typical embedded devices including those with ARM-based micro controller [14]. In addition, we used the random delay up to 0.2 seconds for avoiding packet loose caused by radio interference in our simulation. Therefore, our assumption of $5\sim10$ seconds as the message schedule interval in the application layer makes the required time for encryption and decryption operations negligible. So that, those time parameters are expected not to make any congestion in the application layer.

4.2 Simulation Results: Network Performance

From the simulation results, as shown in Fig. 2(a), the total amount of generated packets in our scheme is much less than that of the diffusion wave. In both cases, however, the total amount of packets gradually increases by nodes' speed. On the contrary, we can see that the amount of unicast packets in our scheme is more than the diffusion wave in Fig. 2(b). The reason of more unicast packets is that our scheme in the simulation uses a number of unicast packets to inform proxy nodes' identifiers. Those informing packets seem to be significantly larger than partial signature gathering packets.

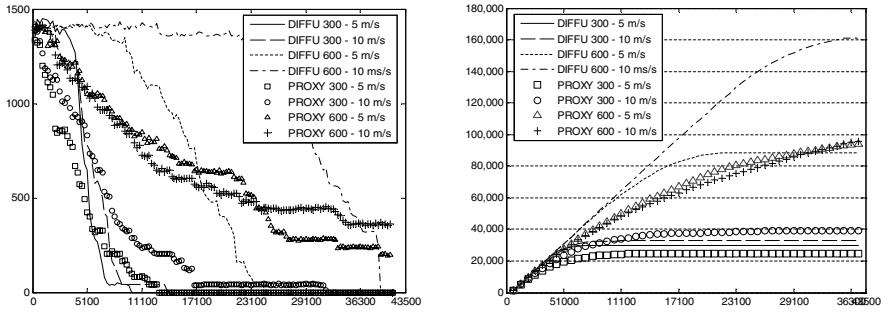
The number of stacked packets sent from UDP are shown in Fig. 3. In an early stage, the number of packets in our scheme is rapidly decreased because the proxy nodes can issue certificates for some nodes nearby them. On the contrary, the number of packets in the diffusion wave is kept in the similar level at the point of that time, but it is decreased at an accelerating pace after certain time because the nodes that can issue partial certificates for other nodes are increased. However, the diffusion wave does not work in $600\times600\text{m}^2$ playground environment by the moving speed of 10 m/s. The overall number of packets in our scheme decreases linearly by the lapse of time. The reason of why there is some long intervals in which the number of packets is kept in the same level is due to that a few number of nodes suffer from its bad environment caused by the random movement for searching the proxy nodes.

The increase of the number of nodes that have a complete certificate also reflects the difference between the diffusion wave and our scheme as shown in Fig. 4. The shape of Fig. 4 is very similar with Fig. 3(a) because the nodes that obtain a complete certificate



(a) total packets (b) unicast packets

Fig. 2. The total amount of packets sent from UDP according to the increment of moving speed



(a) 300 seconds stacked (b) totally stacked

Fig. 3. The number of stacked packets send from UDP by the lapse of time

do not make the broadcast messages any more. As shown in Fig. 2(b), the number of the unicast messages is very small as compared with the number of the broadcast messages. Therefore, the main reasons that the diffusion wave method suffers from lots of network traffic is not only due to transmitting several partial certificates, but also for finding the nodes that can issue partial certificates.

4.3 Simulation Results: Node Performance

The diffusion wave requires more decryption even though the proxy nodes should sign as much as the number of normal nodes in our scheme. In our simulation, one node in the diffusion wave signs about 3 times while one proxy node signs about 7 times on average. However, the overall number of times for signing certificates in the diffusion wave is about 5 times greater than in our scheme. If the threshold is lager than our simulation threshold parameter, the burden will be larger. This situation is shown in Fig. 5.

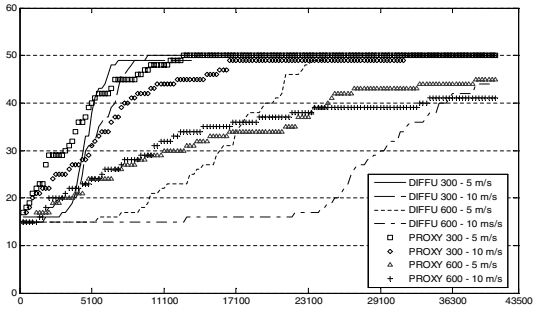


Fig. 4. The number of nodes which have a complete certificate by the lapse of time

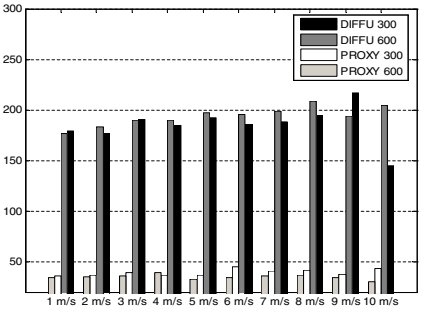


Fig. 5. The number of times for signing certificates in each scenario

5 Protocol Evaluation

5.1 Flexibility

Our proposed scheme is similar to multiple CA model (illustrated in Fig. 1(a)) in that both have several certificate issuers. However, the proxy nodes in our scheme are transfered certificate issuing competence from one authority (i.e. system authority) on contrary with multiple CA model which has several authorities. A node in multiple CA model should request its CA the validations of other nodes' certificates. So that, the CAs in multiple CA model should prepare the certificate chain to other CAs. However, in our scheme, there is not need for such request due to the warrant from system authority though a node also should know other proxy nodes' public keys. The validations of proxy nodes' public keys are guaranteed by the warrant signed by system private key.

5.2 Network Performance

From the shown simulation results, it is clear that the diffusion wave takes much time for obtaining first authentication (i.e. a complete certificate). However, the diffusion wave also has the positive effect such that it shortens the time to take a complete certificate with increasing speed. Instead of that effect, many nodes make lots of network traffic for

finding the nodes which are available in service and for requesting the several partial certificates. In addition, the diffusion wave provide stable and equal chances for all nodes to take their certificate in a good network environment. However, it is very hard to obtain a complete certificate in bad network environment.

The proxy signature method requires less traffic for issuing the certificates than the diffusion wave but there is a large difference of difficulty to obtain certificate for each node. Also, if the system authority at the beginning of the network operation initialize a few chair nodes only but any proxy nodes, the network will not work because of the rarity of the proxy nodes. Also, proxy nodes can exist after the chair nodes distribute the proxy signature keys for the proxy nodes. Therefore, even though we did not simulate that case, it is expected that the system authority must initialize not only the chair nodes but also the proxy nodes.

5.3 Security Consideration

There are many direct or indirect security issues about certificate management in ad hoc network, but we consider two direct security issues that the diffusion wave holds but our scheme mitigates.

Random Attack on the Network: Let τ be the threshold for recovering the secret, N_s be the number of nodes that have the secret, and N_a be the number of nodes that the attacker can attack. The attacker cannot obtain any information from neighbor nodes by attacking directly.

If the attacker cannot analyze the network traffic so that he cannot attack the nodes selectively(or simple, if the nodes behave very carefully for hiding their roles from the attackers), the attacker can try to attack only randomly selected nodes. Therefore, the expected number of nodes that have the secret is only $(N_s N_a)/N_t$ when N_a number of nodes are randomly selected from N_t nodes. Since the secret can be recovered only if $(N_s N_a)/N_t \geq \tau$, the attacker needs to attack N_a number of nodes shown in Equation (4) to obtain the system secret on average.

$$N_a \geq \frac{N_t \tau}{N_s} \tag{4}$$

Based on that, in the diffusion wave, the attacker can recover the system secret if the attacker successfully attacks only τ nodes because $N_s \simeq N_t$.

In our scheme, the probability p that an attacker can succeed in performing such an attack is shown in Equation (5) because all nodes selected by the attack should have the secret. Fig. 6 shows those probabilities.

$$\Pr[p] = \frac{N_s}{n} \frac{(N_s - 1)}{(N_t - 1)} \frac{(N_s - 2)}{(N_t - 2)} \cdots \frac{(N_s - \tau + 1)}{(N_t - \tau + 1)} = \frac{N_s!(N_t - \tau)!}{N_t!(N_s - \tau)!} = \frac{{}_{N_s}P_\tau}{{}_{N_t}P_\tau} \tag{5}$$

Originality of the Authority: The authentication and the certificate issuing are totally two different operations. However, we can consider that the certificate issuing is a part of the authentication if the authentication is a condition for obtaining the certificate.

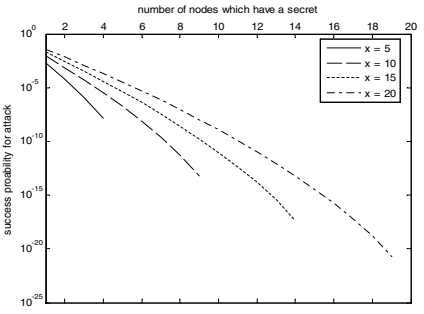
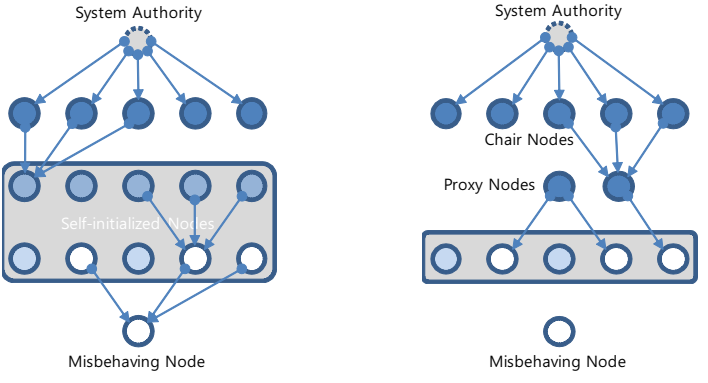


Fig. 6. The success probability for attack with the threshold and the number of nodes that have the secret



(a) diffusion wave (b) proxy signature

Fig. 7. The problem of the originality of the authority

Therefore, we can state that the proxy nodes have a part of the authority for authentication. In that point of view, the discussion about the originality of the authority is required.

Let us refer to nodes initialized by the system authority at the beginning of the network operation as the $1^{\mathbf{st}}$ generation and those authenticated by the nodes of $1^{\mathbf{st}}$ generation as the $2^{\mathbf{nd}}$ generation and so on. If the nodes in certain generation are authenticated by the nodes of previous generation, the originality of authority will degrade increasingly as the generation raises. That is, the misbehaving nodes can join the network legally because of low originality of authority even though the system authority initially does not want them to join in the network. The diffusion wave scheme (shown in Fig. 7(a)) does not consider this problem.

However, in our scheme (an illustration is shown in Fig. 7(b)), the nodes that can authenticate other nodes are only the proxy nodes authenticated by the system authority

or the chair nodes of 1^{st} generation. Therefore, the number of the generation does not exceed 3 in our scheme. So that, it is expected that the problem caused by the originality of the authority rarely happens.

6 Conclusion

In this paper, we introduced the certificate issuing scheme using proxy and threshold signatures for self-initialized ad hoc network. In our introduced scheme, chair nodes that can distribute partial proxy keys for proxy nodes are authenticated by the system authority. In addition, proxy nodes that can issue certificates for other nodes are authenticated and initialized by the system authority or the chair nodes. As it is demonstrated in the simulation results, our scheme has many advantages over the diffusion wave which is a main scheme for the certificate issuing in the self-initialized ad hoc network. If, however, the system authority does not authenticate and initialize any proxy nodes, it is expected that more time is required until there are enough number of the proxy nodes for well-operating network. Therefore, it is recommended that the system authority authenticate the proxy nodes with the chair nodes to save the time.

References

1. Capkun, S., Buttyán, L., Hubaux, J.-P.: Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing 2, 52–64 (2003)
2. Li, R., Li, J., Hisao, K., Liu, P.: Localized public-key management for mobile ad hoc networks. In: IEEE Global Telecommunications Conference, pp. 1639–1646 (2004)
3. Luo, H., Kong, J., Zerfos, P., Lu, S., Zhang, L.: Ursa: ubiquitous and robust access control for mobile ad hoc networks. IEEE/ACM Transactions on Networking 12, 1049–1063 (2004)
4. Li, J., Blake, C., Couto, D.S.J.D., Lee, H.I., Morris, R.: Capacity of ad hoc wireless networks. In: ACM Conference on Mobile Computing and Networking, pp. 61–69. ACM Press, New York (2001)
5. Shamir, A.: How to share a secret. Communications of ACM 22, 612–613 (1979)
6. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, Heidelberg (1990)
7. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: ACM Conference on Computer and Communications Security, pp. 48–57. ACM Press, New York (1996)
8. Okamoto, T., Pointcheval, D.: The gap-problems: A new class of problems for the security of cryptographic schemes. In: Kim, K.-c. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001)
9. Menezes, A., Okamoto, T., Vanstone, S.A.: Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory 39(5), 1639–1646 (1993)
10. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
11. Zhang, F., Safavi-NainiTatsuaki, R., Lin, C.Y.: New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairings. Cryptology ePrint Archive (2003)

12. Lakshman, T.V., Madhow, U.: The performance of tcp/ip for networks with high bandwidth-delay products and random loss. IEEE/ACM Trans. Netw. 5, 336–350 (1997)
13. Lefevre, F., Vivier, G.: Understanding tcps behavior over wireless links. In: IEEE Symposium on Communications and Vehicular Technology, SCVT-200, pp. 123–130. IEEE Computer Society Press, Los Alamitos (2000)
14. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing elliptic curve cryptography and rsa on 8-bit cpus. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 119–132. Springer, Heidelberg (2004)