Advanced Engineering Forum *Online: 2011-09-09*
ISSN: 2234-991X, Vol. 1, pp 295-299
doi:10.4028/www.scientific.net/AEF.1.295

A Hierarchical Key Management Scheme in Mobile Ad hoc Networks

Jun Wu [1, 2, a], Runhua Shi[1, 2], Hong Zhong[1, 2]

[1]Key Laboratory of Intelligent Computing & Signal Processing of Ministry of Education,

Anhui University, Hefei, 230039, China

[2]School of Computer Science and Technology, Anhui University,

Hefei, 230039, China

[a]wujun-ad@foxmail.com

Keywords: Key Management; Threshold Secret Sharing; Mobile Ad hoc networks; RSA Cryptosystem

Abstract. This paper proposes a hierarchical key management scheme in the mobile Ad hoc networks. In this scheme, there are two kinds of server nodes: the special server nodes and the ordinary server nodes, such that only when two kinds of server nodes collaborate can they provide a certificate service. In order to satisfy this special application, we design a new secret sharing scheme for splitting the system private key, in which it generates two different kinds of shares of the system private key: the special share and the ordinary share, where it needs at least one special share and t ordinary shares to recover the system private key, thus we call it $(1,t,n)$ threshold scheme. Furthermore, we present a distributed signature scheme for a user's certificate in the mobile Ad hoc networks based on this secret sharing.

Introduction

With the development of the mobile computing and wireless communication technology these years, Mobile Ad hoc networks (MANETs) had were widely used in military, emergency and civil occasions, but most of these applications required the highly security. Nowadays, many official documents required to be approved by the different departments hierarchically. For example, before the bank approved a lending proposition, the staff must have investigated and approved the economic conditions of the applicant first and then submitted the lending proposition for approval to the leaders hierarchically. In this paper, we mainly consider how to provide such services in MANETs.

To solve the problem above, we proposed a hierarchical key management scheme and extended its applications for the digital signature in MANETs. In such scheme, a group of server nodes, which collaborate to play the part of Certificate Authority (CA), are partitioned to two different kinds of server nodes: the special server node (SN) and the ordinary server node (ON). Furthermore, we correspondingly designed a threshold secret sharing scheme, in which it generated two kinds of shares: the special shares (Ss) and the ordinary shares (Os), and distributed them to SN and ON respectively. To recover the private key of the system, it needed one SN and t ONs at least. Here we called it $(1,t,n)$ threshold scheme, which not only had good practicability for the affairs such as the hierarchical approval, but also improved the security of the whole system compared with the traditional (t,n) threshold schemes.

Related Works

Because of the features of self organize and dynamic topology of MANETs, the centralized key management schemes in the traditional network are no longer fit for it. There are three main reasons [1-2]: firstly, there is a key management center (KMC) in the centralized key management schemes, which may induces the single point failure; secondly, MANETs have a limited bandwidth, and furthermore have certain restriction for the node computation, storage capacity and energy, so if

each node in the network applies for certificate service to KMC it will cause the network congestion and the resources of KMC will be exhausted; thirdly, the routing information of each node needs to renew frequently since MANETs' topology changes dynamic and wireless multi-hop communication itself has high bit error rate, which will lead to increase the delay of key management service ultimately.

To solve the first two problems above, it requires to distribute trust to the server node sets and to let all nodes perform key management jointly. As a result, two different methods, partially distributed key management scheme and fully distributed key management scheme, are proposed early or late. Zhou, Hass *et al*. [3] presented partially distributed key management scheme [4-6], which used (t, n) threshold secret sharing scheme [7] to distribute the services of the CA to a set of specialized server nodes. Each of these nodes is capable of generating a partially certificate using their shares of the system private key, but only by combining *t* such partial certificates can a valid certificate be obtained. It is obvious that it can decrease the delay of key management service when we accomplish certificate service between one-hop neighbor nodes instead of multi-hop communication. Later, Luo, Lu *et al*. [8] presented fully distributed key management scheme [9-10], which distributed the services of the CA to all the nodes. Any operations requiring the system private key can only be performed by a coalition of *t* or more nodes, the availability of the service is based on the assumption that every node will have a minimum of *t* one-hop neighbors. These schemes decrease the delay of service, but increase the risk to expose the system private key.

No matter partially, or fully, distributed key management scheme, there is a common problem that the service nodes are entirely peer, and they have the same ability to recover the secret message. Thus both of them aren't fit for providing a hierarchical key management and service in MANETs.

Proposed Scheme

Network Model. There are three kinds of nodes in our network model: special server nodes sv, ordinary server nodes $v_i (i = 1, 2, \cdots, n)$, and user nodes v_u. The scheme is divided into two phases: the secret sharing phase and the service phase. In the secret sharing phase, the off-line CA selects the relevant parameters, computes and distributes the shares of the system private key to the different server nodes. In the service phase, SN and ON provide certificate services for user nodes jointly. Notations used in the paper are defined as follows:

Table 1. The definition of all notations

pk	System public key
sk	System private key
d	Interference factor, special share of sk (Ss)
sk_i	Ordinary share of sk (Os)
sv	Special server node(SN)
v_i	Ordinary server node(ON)
${}_R Z^*_{\phi(n)}$	Select from $Z^*_{\phi(n)}$ randomly

The Secret Sharing Phase. In the secret sharing phase, the off-line CA selects the relevant parameters, calculates and distributes the shares of the system private key to all server nodes. Now, let us describe it in detail as follows:

Step1. According to the requirement of RSA cryptosystems, the off-line CA first generates the following parameters: sk, pk, d and $f(x)$.

[1] He first selects two large prime integers: p and q, and computes $N = pq$ and $\phi(N) = (p-1)(q-1)$. Please note that the values of p and q must satisfy the equations: $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are two large primes;

[2] Then he randomly selects a large odd integer sk as his private key such that $\gcd(\phi(N), sk) = 1$, where $1 < sk < \phi(N)$;

[3] He furthermore computes the public key $pk \equiv sk^{-1} (\mathrm{mod}\, \phi(N))$;

[4] In addition, he randomly selects a integer d, where $1 < d < \phi(N)$;

[5] Finally, he chooses a degree $(t-1)$ polynomial $f(x)$:

$$f(x) = d + sk + a_1 x + a_2 x^2 + \cdots + \mathrm{a}_{t-1} x^{t-1} (\mathrm{mod}\, \phi(N)), \quad (a_i \in {}_R Z^*_{\phi(N)}) \tag{1}$$

Step2. Given from the above polynomial, the off-line CA computes $sk_i = f(i)$ $(i = 1, 2, \cdots, n)$ and sends it to the ordinary server node v_i as his share, respectively. Furthermore, he privately sends the integer d to the special server node sv as his share. In addition, he selects a primitive root g in Z_N, computes: $v_d = g^d$, $v_s = g^{sk}$ and $v_j = g^{a_j}$ $(j = 1, 2, \cdots, t-1)$, and announces them over the public channel. Finally, he deletes all secret messages.

Step3. After receiving the respective share, sv checks its validity by the following verified Eq.1, v_i verifies the Eq.2. If one fails, they cancel this protocol and restarts; or else they complete the secret sharing phase successfully.

$$g^d = v_d \tag{2}$$

$$g^{sk_i} = verify(i) \tag{3}$$

where $verify(x) = v_s v_d \prod_{j=1}^{t-1} v_j^{x^j} \bmod N$.

The Service Phase. In the service phase, if a user node v_u wants to apply for a certificate, the request information, called CREQ, will be broadcasted. After receiving CREQ, one SN and t ON collaborate to sign a certificate for the user node v_u. Without loss of generality, let v_i $(i = 1, 2 \cdots, t)$ be t members of all ON that want to collaboratively generate the certificate for the user node v_u with the help of the special server node sv.

Step1. After receiving the request information of the user node v_u for the certificate, the ordinary node v_i authenticates the legality of the user, and furthermore provides the certificate services for the legal user as follows $(i = 1, 2 \cdots, t)$:

[1] The ON v_i computes the ordinary sub-key of the signature key: $gs_i = sk_i l_i(0)$, where $l_i(0) = \prod_{j=1, j \neq i}^{t} j/(j-i)$ $(i = 1, 2 \cdots, t)$.

[2] The ON v_i $(i = 1, 2 \cdots, t)$ signs the certificate $Cert_u$ using his sub-key gs_i, and then generates an ordinary partial signed certificate $Cert_u^{gs_i}$ and send it to v_u. Here, $Cert_u = id_u \,||\, pk_u \,||\, T$, id_u represents the ID of v_u, pk_u denotes the public key of v_u, and T is the expiry date of the certificate.

Step2. After the SN sv receives the request information CREQ for certificate, He first authenticates the legality of v_u. If v_u isn't legal, sv will discard CREQ, otherwise, sv will sign $Cert_u$ using his sharing sub-key d, and generate a special partial signed certificate $Cert_u^d$, and send it to v_u.

Step3. After receiving t ordinary partial signed certificate $Cert_u^{gs_i}$ $(i = 1, 2 \cdots, t)$ and a special partial signed certificate $Cert_u^d$, v_u computes the certificate signed with sk : $Cert_u^{sk} = \prod_{i=1}^{t} Cert_u^{gs_i} \Big/ Cert_u^d \bmod N$

Step4. The user v_u authenticates its validity of $Cert_u^{sk}$ using the public pk. If $Cert_u = (Cert_u^{sk})^{pk} (\bmod N)$, v_u will accept it. Otherwise, the user v_u will broadcast a complaint and another request for certificate again.

Scheme Analysis

Now we first prove that the present scheme is correct and secure, and then give a performance analysis.

Theorem 1. The user node v_u will get a correct certificate $Cert_u^{sk}$ signed by sk if two kinds of server nodes honestly execute the protocols.

Proof. Without loss of generality, we assume the partial signed certificates received by v_u are $Cert_u^{gs_i}$ $(i = 1, 2 \cdots, t)$ and $Cert_u^d$, respectively, then v_u can compute $Cert_u^{sk}$ as follows:

$$\begin{aligned} Cert_u^{sk} &= \prod_{i=1}^{t} Cert_u^{gs_i} \Big/ Cert_u^d \bmod N \\ &= \prod_{i=1}^{t} Cert_u^{s_i l_i(0)} \Big/ Cert_u^d \bmod N \\ &= Cert_u^{d+sk} \Big/ Cert_u^d \bmod N \\ &= Cert_u^{sk} \bmod N \end{aligned} \tag{4}$$

Theorem 2. No matter how many ONs there are, they can't recover the system private key sk.

Proof. According to threshold secret scheme of Shamir [7], we know that t or more than ONs can recover the secret information $d + sk$, but less than t ONs can't obtain any information about $d + sk$. Since $d, sk \in_R z_{\phi(N)}^*$, we can't obtain any information about the private key sk without knowing d.

Theorem 3. The ordinary share sk_i $(i = 1, 2, \cdots, t)$, the special share d, and the system private key sk are not obtained and revealed only from the partial signed certificate at the service phrase.

Proof. Based on the difficulty of Discrete Logarithm problem, it is obvious that there are the following results:

(1) After v_u received the partial signed certificate $Cert_u^{sk_i l_i(0)}$ and $Cert_u^d$, he can't obtain sk_i or d in a polynomial time, that is, it is the computational security.

(2) After v_u obtained the certificate $Cert_u^{sk}$, similarly he can't compute sk from $Cert_u^{sk}$ in a polynomial time, that is, it is the computational security.

Give from Theorem 1, 2 and 3, we can see that our scheme is correct and secure. Further more, our scheme combines the advantages of partially distributed key management scheme and fully distributed key management. In our scheme, there are two kinds of shares of the system private key: Ss and Os, which are distributed to SN and ON separately. In order to provide the certificate services for the user nodes, it needs one SN and at least t ONs to collaborate to play the role of CA. On the one hand, there always exist some nodes with higher performance of computing capability,

memory space and wireless transmission capability in MANETs, which can be selected as SNs. The appointment of SNs makes our scheme have the same security as partially distributed key management; On the other hand, given from Theorem 2, no matter how many ONs there are, they can't recover the system private key. Thus we can select lots of nodes as ONs, in such a way that there are at least t ONs in neighbors of each user node, that is, the certificate services can be accomplished among one-hop neighbor nodes. That is, our scheme has the same availability as fully distributed key management scheme. In addition, our scheme has the same communication complexity and computational complexity as almost all the existing distributed key management scheme based on RSA cryptography.

Conclusions

In this paper, we have proposed a hierarchical key management scheme for MANETs, designed a new secret sharing scheme for splitting the system private key, and presented a distributed digital signature scheme based on this secret sharing scheme. Our scheme has advantages of both partially distributed scheme and fully distributed scheme, such as security and availability. Thus our scheme is very fit to provide secure services to some affairs such as hierarchical approval in MANETs.

Acknowledgements. This work was supported by the Natural Science Foundation of Anhui Province (No. 11040606M141), Research Program of Anhui Province Education Department (No. KJ2010A009) and the 211 Project of Anhui University.

References

[1] C.L. Du, M.Z. Hu, H.L. Zhang, New group key management framework for mobile ad hoc network based on identity authentication in elliptic curve field, Journal on Communications. 28 (2007)53-59.

[2] D.M. Yang, D.J. Mu, Z. Xu, Novel key management and authentication scheme for ad hoc space networks, Journal on Communications. 27(2006)104-107.

[3] L. Zhou, Z.J. Hass, Securing Ad Hoc Networks, IEEE Networks Special Issue on Network Security. 113(1999)24-30.

[4] H. Luo, P. Zerfos, J .Kong, *et al*. Self-Securing Ad Hoc Wireless Networks, Proceedings of the 7th IEEE Symposium on Computers and Communications ,(2002)567-574.

[5] K. Fokin, Key Management in Ad Hoc Networks, 2008, http://www.computer.org/

[6] Y.C. Zhang, Y.D. Wang, *et al*. Distributed Key Management for Ad Hoc Network, J. Wuhan Univ. (Nat. Sci. Ed.),55(2009) 85-88.

[7] A. Shamir, How to Share a Secret, Communications of ACM. 22(1979) 612-613.

[8] H. Luo, S. Lu, Ubiquitous and Robust Authentication Services for Ad hoc Wireless network, Dept. of Computer Science, UCLA, TR-20030, (2000).

[9] J.Kong, P. Zerfos, H. Luo, *et al*. Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks, Proceedings of the IEEE 9th International Conference on Network Protocols, (2001)251-260.

[10] J. Kong, P. Zerfos , H. Luo, *et al*. Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks, ACM Transactions on Networking. 12(2004)1049-1063.

[11] William Stallings, *et al*. Cryptography and Network Security-Principles and Practices, Fourth Edition, Beijing, Publishing House of Electronics Industry, 2009.