International Journal of Multimedia and Ubiquitous Engineering
Vol.8, No.6 (2013), pp.19-30
http://dx.doi.org/10.14257/ijmue.2013.8.6.03

Intrusion Tolerant-based Lightweight CA Model for Wireless Mesh Networks

Ping Guo[1], Jin Wang[1], JieZhong Zhu[2], YaPing Cheng[1] and Jeong-Uk Kim[3]

[1]School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
[2]Bing Jiang College, Nanjing University of Information Science & Technology, Nanjing 210044, China
[3]Department of Energy Grid, Sangmyung University, Seoul 110-743, Korea

Abstract

In a Wireless Mesh Network (WMN), we cannot assume the existence of a trusted certificate authority that is used in regular Public-Key Infrastructure (PKI). Hence a cooperative approach in which a cluster of some nodes can issue certificates to other nodes in a self-organizing manner is considered to be more suitable to the Ad-hoc nature of WMNs. In this paper, we propose a new trust establishment scheme providing an efficient and lightweight public key authentication without the presence of any trusted third party neither in bootstrapping network phase nor during network life time. Combined threshold mechanism and lightweight Certificate Authority idea, A lightweight and tolerate CA (LTCA) model has been put forwarded. LTCA is not only reformed to realize lightweight structure, but also equipped with intrusion tolerant ability which has been realized by dividing CA's private key into n shares and distributing them into n servers. Simulation results show that the threshold mechanism does not significantly increase the computational cost of communication, but the security of LTCA's private key is improved extremely.

Keywords*: wireless mesh networks, lightweight CA, intrusion tolerance*

1. Introduction

Wireless Mesh Network (WMN) is becoming a hot topic in the research of wireless networks. WMN is a special form of mobile AD HOC network, new broadband wireless network architecture, and an integration of wireless local area network (WLAN). The WMN technology can be acted as "the last mile" in military communications network, wireless metropolitan area networks, wireless sensor networks, and wireless LAN network. WMN integrates various existing wireless technologies, such as the IEEE 802.11 WLANs, IEEE 802.16 Broadband WMANs, the IEEE 802.15 WPANs, and even cellular phone network. Via WMN technology, the mobile user can connect to the Internet to enjoy the service at any time, from any location. Due to WMN has fixed and sufficient power backbone routers, the issue of mobility and energy consumption are less to consider. In order to allow WMN become an important extension of the wired network or even to replace part of the cable network, improving network communication capacity and communication quality, and providing safe access scheme are WMN research required to solve. The large WMN requires a large number of keys, how these keys are securely generated, updated and revoked is a very complex and difficult problem. It is very important to design a safe WMN authentication model that is responsible for the mobile nodes accessing, identification authentication and key

ISSN: 1975-0080 IJMUE
Copyright ⓒ 2013 SERSC

distribution. This paper focuses on authentication mechanism in WMN, which is the base of the secure routing, key management, and data security.

The paper is organized as follows. In Section 2 theories about intrusion tolerance and lightweight CA are reviewed. Section 3 introduces two important theories based our work. Section 4 describes the model in detail, and section 5 simulations are performed to test and analyze our model. The conclusion, acknowledgement and reference are in Section 6, 7, and 8 respectively.

2. Related Work

In recent years, a number of authentication key management protocols are applied to WMN to meet special needs. In order to provide better fault tolerance, usually the authentication server is backed up, however this will cause each server contains all certification information and key, once a server is compromised, will divulge all certification information and key. To overcome such problems, in 1979, Shamir [1] and Blakeley [2] independently proposed the key concept of decentralized management mechanism to achieve the idea of threshold scheme, the key (*K*) is divided into *n* shares distributed to *n* authentication servers, which any *t* of the servers can recover *K*, but any server does not have all the key information, and cannot independently complete the authentication process. Threshold mechanism has two important features: (1)at most tolerate the *n-t* servers' error, because of needing *t* among *n* servers to corporately recover the authentication key; (2)at most tolerate *t-1* servers compromised, because less than *t* servers can not complete the authentication process. The reallocation of the sub-key is very critical in *(t,n)* threshold mechanism. In 1997, Desmedt and Jajodia [3] put forward a subkey redistribution scheme which is labeled as from *(t,n)* threshold to *(t', n')* threshold and does not require a collector(dealer). However, this scheme cannot tolerate failure. Wong et al [4] put forward VSR (Verifiable Secret Redistribution) scheme to improve the [5]. The VSR can choose t nodes from any of n nodes to compose of a collection of B, using Feldman VSS scheme [8] to detect subkey to discover whether there has been compromised node in B. Once it is found incorrect subkey, the agreement will be terminated and will choose other t node re-composed of a collection of B. Therefore, the worst-case scenario, the need $C_n^t - C_{n-t+1}^t$ times to restructure in order to complete the reallocation of the subkeys. Recently, Kim and Bahk put forward a fast FVSS (Fast VSS) redistribution scheme [6]. However, this scenario assumes that all the nodes in the new authentication group are honest, this assumption is not feasible, because the compromised nodes may still be in the new group. Literature [7] proposed a secure local authentication and complaints scheme (Secure Localized Authentication of the Billing scheme SLAB). Due to compromised Mesh routers continue to attack, a threshold-based digital signature mechanism [8] with voting strategy has been used to strengthen the system security level. Local voting mechanism, awarded to the Mesh client's certificate is issued by a group of neighbor's nodes Mesh routers, rather than the service provided by Mesh routers. Threshold digital signature technology, each neighbor node Mesh router needs to hold a part of the digital signature, to jointly produce a certificate to access the Mesh client, at least more than K neighbors Mesh routers need to send some of their own certificate. Although the above authentication scheme using a threshold scheme, makes the CA intrusion tolerant, but there are some disadvantages. These programs are based on traditional public key certificate CA mechanism, in which the heavy task of such a CA system assume the management, maintenance, certificate revocation, update, resources are relatively abundant in the wired network is also easy to become a system bottleneck. Considered the threshold program computational complexity, the need for more collaborative work between the authentication servers, no doubt the system load and complex further increase, which makes wireless environment, such as AD HOC shared key threshold mechanism is difficult to achieve the desired effect. WMN Unlike the wireless network, it is part of the infrastructure to support the wireless network, computing resources, storage capacity, and power supplies are usually abundant than other wireless networks, which is the material base for the deployment threshold mechanism in WMN.

3. Preliminaries

In this section, we discuss the intrusion tolerant technique and outline lightweight CA idea on which design of our scheme has been developed.

3.1. Intrusion Tolerance

An intrusion tolerance system [9] is that is, the notion of handling-react, counteract, recover, mask-a wide set of faults encompassing intentional and malicious faults, which may lead to failure of the system security properties if nothing is done to counter their effect on the system state. In short, instead of trying to prevent every single intrusion, these are allowed, but tolerated: the system has the means to trigger a series of mechanisms that prevent the intrusion from generating a system failure. In order to attack the system, attackers need to destroy more components or focus on only one during the short time, which is easier detected than some single attack. An intrusion tolerance system is an information system that still continuously provides stable services for the expecting users when facing attacks. An intrusion system is able to limit some attacks which can’t be detected by attack avoid and preventable methods. Some attacks probably can penetrate the outside protection, such as fire wall, authentication and encryption. An intrusion tolerance system should provide measures to promise the crucial applications to run correctly.

Usually, intrusion tolerance includes two important technologies [10].

One is tolerance Technology. Tolerance technology can recover a system from intrusion and attack, which including resources relocating and system redundant. Tolerant technology contains fault tolerance and intrusion tolerance. Among them fault tolerance focuses on error tolerance of software and hardware, but intrusion tolerance malicious attacks. Intrusion tolerance developed from fault tolerance, which added the research on dynamic attributes based on the research on static attributes. Usually we combine these two kinds of technologies to design a system, meanwhile, sort them according to the destructive results, not the reasons. This kind of system calls intrusion tolerance system.

The other is tolerance triggers. Intrusion detection system can become a trigger. Even the top class of IDS has too high miss alarming rate and too low scope of intrusion recognition. Triggers should have the highest ability of intrusion recognition and nearly zero rate of missing alarms in theoretic. The highest covering areas, the more fault can be detected. At the same time, errors should be found before spreading to all the system. For example, if tolerance system relies on redundancy and backup of the units, it must find out attacks and errors before the units crash.

3.2. Lightweight CA

As reference [8, 11, 12] definition, a public key encryption without certificate management center is a 6-ary tuple $\Pi = (\varsigma_{CA}, \varsigma_U, \varepsilon_P, S_P, \varepsilon, D)$ defined as follows:

–CA Setup, ς_{CA}, is a probabilistic polynomial time (ppt) algorithm that takes as input *k*, the system's security parameter, and outputs the public/private keys pair*(pk_{CA}, sk_{CA})*.

–User Setup, ς_U, is also a ppt algorithm that takes as input *k*, the system's security parameter, and outputs the public/private keys pair *(pk_U, sk_U)*.

–Extract-Partial-Public-Key, ε_P, is also a ppt algorithm that takes the system's security parameter *k*, CA's private key sk_{CA} and the user's public key pk_U and identity ID_U as input and outputs P_U as partial public key.

–Set-Public-Key, S_P, is a deterministic algorithm that takes the system's security parameter *k*, CA's public key pk_{CA} and the user's public key pk_U and identity ID_U as input, and outputs*(pk_U, P_U)* as the user's extended public key if P_U is a valid partial public key.

–Encrypt, ε, is also a ppt algorithm that takes a plaintext $m \in \mathrm{M}$, pk_U, P_U and pk_{CA} as input and outputs a cipher text $c \in C$ or $\perp$ which means that pk_U is invalid.

–Decrypt, D, is a deterministic algorithm that takes a cipher text $c \in C$ and a private key sk_U as input and outputs the corresponding plaintext $m \in \mathrm{M}$ or $\perp$ which means that *c* is not a valid cipher text.

Among the above definition, the algorithms ς_{CA} and ε_P are executed by CA while the algorithms ς_U, S_P and D are executed by the user itself. Of course, the algorithm ε is executed by who wants to send cipher text to the user.

4. System Model Based on (t,n) Threshold Mechanism and Lightweight CA

In this section, we combine (t,n) threshold mechanism and lightweight CA idea to construct our LTCA(Lightweight and Tolerant Certificate Authority) model. Firstly, we give and demostrate LTCA's architecture.

4.1. LTCA's Architecture

LTCA composes of a group of sharing servers, control unit, management unit and a group of users/nodes, shown as figure 1 below.

4.1.1. Sharing Servers: S1, S2,…,Sn represents the sharing servers. No need to promise everyone of them security, but most of them are safe. They own CA's private key or a part of it, and every server runs a service program to manage many private keys and can provide services for many users.

4.1.2. User Units: User unit is the mobile nodes or roaming nodes/agents which want to use sharing servers, such as CA servers, user Ui. When a user connects with a sharing server, it should prove it has been authenticated. Then, it can use a private key di which is kept in the sharing servers to sign or encrypt a message M.

4.1.3. Management Units: Management units use to manage all the sharing servers, actually, they manage all the keys kept in the servers. When servers encounter attacks, management units can close or suspend some server, and indicate other servers to take proper measures to avoid the attacks. Management units can deliver orders to the sharing units, such as generating new keys, close, suspend, and refresh.

4.1.4. Control Units: Control units take charge of initialization of the sharing servers, allocation of the share of a private key, the system clock synchronization, collecting information of sharing servers and CA, analyzing those data, and telling what the sharing servers should do at a special time. For example, CA tells the control units that it needs NO1,2, and 5 servers to use the shares of a private key combined to sign a message. While, sharing servers NO1,2 and 5 will tell control units that they are now producing a digital signature. The detailed design of the control units will be described in next chapter.

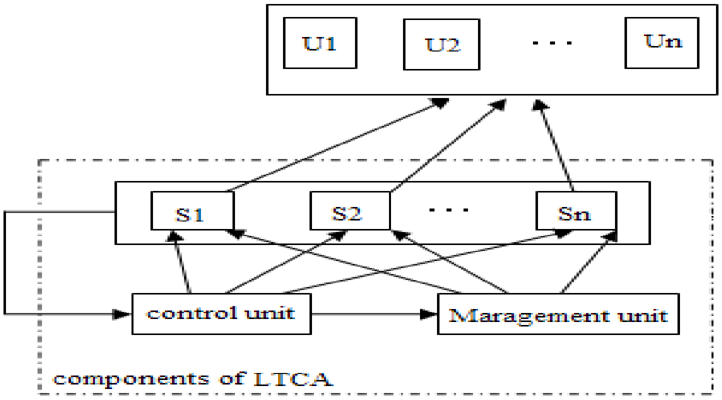


Figure 1. LTCA architecture

4.2. LTCA (Lightweight and Tolerant Certificate Authority) Working Process

In this section, we demonstrate how LTCA work, including the whole system model, key generation and signature generation and verification.

4.2.1. LTCA Working Process: Shown as the Figure 2, the digital signature system was composed of a group of servers S={s_1, s_2, …, s_n}, a group of users U={u_1, u_2, …, u_k}, a timing refreshing server(TFS), a key distributed server(KDS) and a public board server(PBS).

On the stage of initialization, KDS calculated all the shares of the key and distributed the shares to every server; during the signature process, a timing refreshment server updated all the servers periodically, at the same time, key distributing function f(Key) reassigned the shares and promised the new shares combined together not to change the older key; after refreshment, all the old shares of the servers were eliminated; PBS used to preserved all the public parameters, and could be accessed, but not be changed; servers and users had authorized connection.

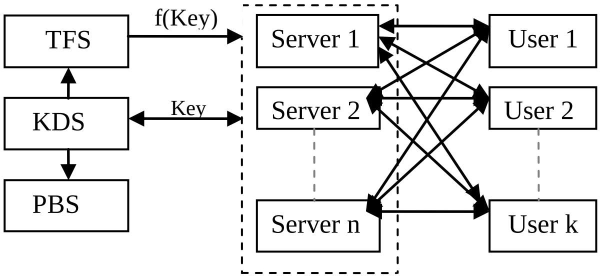


Figure 2. System working process

4.2.2. Key Generation: KDS finished works as follows:

Step1: Parameters preparing. Elliptic curve system $T=(p,F,a,b,G,n,h)$ [13]. P was a prime, F was a finite field, G was a basic point on the elliptic curve, n was a prime and was a rank of the G, and h was a unilateral function, such as SHA-1.

Step2: KDS chose a random number *d* and $d \in [1, n-1]$.

Step3: Calculating Q=*d*G. If Q was an infinite point or Q was equal to G, then returned to step 2, otherwise Q was a public key.

Step4: According to the sharing scheme of t-out-of-n[14][15][16], for example, Table1 was the 3-out-of-4 sharing scheme, the key *d* was partitioned into many different combinations, and every combination corresponded with *t* different servers. Parameter *t* was the value of threshold, which meant finishing signature at least needed *t* servers, and parameter *n* was the total number of all the servers. It can be configured with the following method: KDS selected *t* servers from a group of *n* servers $(s_1, s_2, \ldots, s_t (t \le n))$; partitioned key $d_i (i=1,2,...,n, d=\sum_{i=1}^{t} d_i)$ d_i was bound with the *ID* of a group of keys; the entire d_i was sent to n servers $(s_1, s_2, ..., s_n)$ securely.

Step5: In the term of $d_i (i=1,2,...,n)$, KDS calculated $Q_i = d_i G$. Q_i was bound with user ID_{ui} and key ID_{keyi} , and published $\{ID_{ui}, ID_{keyi}, Q_i\}$ to PBS.

Table 1. Key sharing scheme of 3-out-of-4

$IDkey_i$	Shadow server 1	Shadow server 2	Shadow server 3	Shadow server 4
1	d_1	d_2	d_3	d_3
2	d_1	d_2	d_4	d_4
3	d_1	d_1	d_3	d_4
4	d_2	d_2	d_3	d_4

Combination one: $d＝d_1＋d_2＋d_3$
Combination two: $d＝d_1＋d_2＋d_4$
Combination three: $d＝d_1＋d_3＋d_4$
Combination four: $d＝d_2＋d_3＋d_4$

4.2.3. Signature Generation: The signature request from users:

Step1: Users U_i ($1 \le i \le t$) selected a combination, correspondingly, a group of shadow servers.

Step2: Message M, U_i produced a random positive integer k_i ($1 \le k_i \le n-1$), calculating $R_i ＝ k_i G$, and then broadcasting R_i to PBS.

Step3: U_i sent its signature request <Request, IDu_i, M, $IDkey_i$, k_i, R_i> to the shadow servers selected in the step 1 and simultaneous TFS was unlocked and entered into a timing status.

The response by the group of shadow servers: when the selected shadow server's s_r ($1 \leq r \leq t$) received the request from u_i, first authorized u_i, and then executed the actions as follows:

Step1: Calculating $r = R_{ix}$ (mod n), (R_{ix} was the value of X coordinates of point R_i ） $s_i = d_i r + k_i h(M)$ (mod n).

Step2: s_r sent the responding message <Response, r, s_i> to u_i.

4.2.4. Signature Recombination at the End of the User:

Step1: Parameters preparing. Elliptic curve T=(p, F, a, b, G, n, h), the user's public key Q and the signature (r, s).

Step2: User U_i checked r was the valid point of the elliptic curve, and satisfied 1<s, r<n-1.

Step3: User U_i calculated $e = h(M)$, $w = e^{-1}$ and $V = swG\text{-}rwQ$.

Step4: v_x was the value of point V's X coordinate's MOD n. If $v_x = r$, then the signature was received, otherwise rejected.

5. Discussion

In this section, we discuss LTCA's characteristics, security and simulation results.

5.1. LTCA Model Characteristics

There are two important features of LTCA, as its abbreviation, lightweight and tolerance.

5.1.1. Lightweight CA: In order to solve the problems of heavy work, centralized control and maybe a network bottleneck in the traditional certificate-based public key system, we combine lightweight CA idea to reduce not only the components but the functions of our model, which takes charge of issuing slavery public key and is nothing to do with the generation of private key and master public key for an user. Hence, our model (LTCA) doesn't need to manage certificate, update and revoke the master public keys. It is only a half-trusted third party, but in the traditional certificate-based public key system, CA requires high security based on the assumption that CA is completely trusted.

5.1.2. Tolerant CA: We introduce intrusion tolerance idea into our model and implement by *(t,n)* threshold mechanism that LTCA's private key has been divided into *n* shares and distributed to *n* different servers, among which only *t* of them could reconstruct LTCA's private key. This mechanism is very important for LTCA to defense DoS (Denial of Services) attack triggered by the malicious or unauthorized third parties.

5.2. Security Analysis

5.2.1. Capturing Less Than *t* Authentication Servers: The security of the scheme is based on the elliptic curve discrete logarithm of the difficult solution and the unconditional security in Shamir *(t,n)* threshold mechanism that any *t-1* or less than *t-1* sub-keys can't reconstruction LTCA's private key.

5.2.2. DoS Attack: The authentication servers LTCA respectively hold the shares of the periodically refresh, and make the new child share still share the same secret in the refresh. Each server's old share was destroyed, making the attacker get used to the cycle of child share, but current and future cycle attack without any help the attacker only in a cycle of breach authentication server number more than threshold value, and access to their current all child key, can truly break the system. Otherwise, due to the attackers in each cycle owning the child keys has not reached threshold value, each cycle can get LTCA private key to effectively address the private key to LTCA the DoS attack

5.3. Experiment and Simulation

Firstly, we implement a prototype according our LTCA model, and then we adopt OPNET [17] to simulate our LTCA to test its performance.

5.3.1. LTCA Compares with Single CA Structure: According to our model, a (3,5) prototype system has been implemented (C++ programmed, authentication server configuration Intel DuoCore 1.99GHz CPU, 2GB RAM). In order to analyze our model's performance, a single CA as lightweight idea as our model has been implemented. ECC (160bits) algorithm and RSA (1024bits) algorithm have been run on them respectively. The results of authenticating one user are showed in Figure 3. Simulation results show that LTCA's initialization and communication costs are more than single CA, which it is within an accepted scope; and RSA algorithm cost expenses more than ECC algorithm.

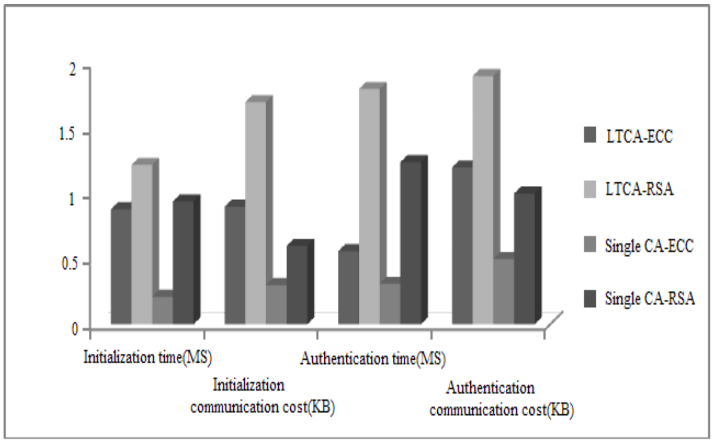


Figure 3. Performance comparison of authenticating one user under the different authentication mechanism with different algorithms

5.3.2. LTCA Comparing with ID-RSA CA Structure: ID-RSA scheme is the identity of the constructed by RSA algorithm based on single CA certification system, using the LTCA scheme (3, 5) to construct threshold mechanism with ECC algorithm, the simulation results are shown in Figure 4.

LTCA (3, 5) threshold mechanism and single CA authentication system based on ID-RSA[18] provide service for nodes (LTCA generated slavery public key process, while ID-RSA generated node's certificate process) respectively. When the number of nodes reaches

100, LTCA takes about 150ms more than the ID-RSA's time. However, LTCA scheme has the capability of intrusion tolerance, while the ID-RSA is a single CA authentication scheme. The time cost is worth and within an acceptable range**.**

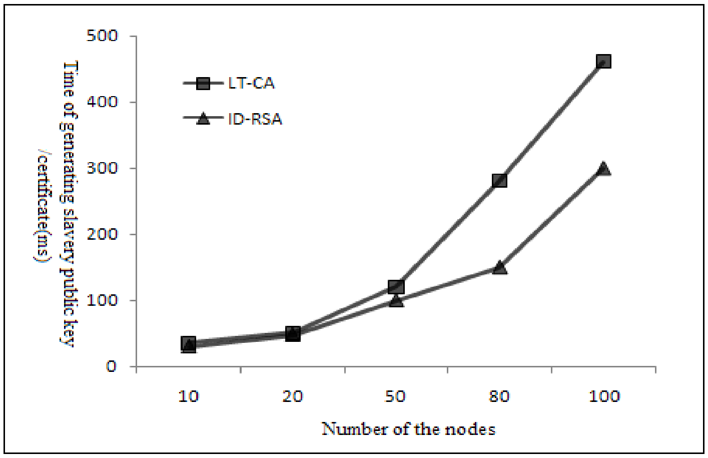


Figure 4. Performance comparison of LTCA generating slavery public key from CA with ID-RSA algorithm generating certificate

5.3.3. Comparison of Different (t,n) Mechanism with Different Algorithm: We evaluate the performance of the LTCA via simulations in OPNET. The simulation environment is set up with the parameters listed in Table 2.

Table 2. Simulation parameters setting

Parameter	Value	Parameter	Value
MAC protocol	MAC-802.11	Packet	512Bytes
Simulation time	600 Sec	Computer	Intel DuoDore1.99GHz CPU
Routing protocol	AODV[19]	RAM	2GB
Nodes size	500m*500m	ECC	160bits
Node moving speed	10m/s	RSA	1024bits
Data rates	2Mbits/s		

The results of signature time, verification time and authentication time under the different value *t* in *(t,n)* threshold mechanism are showed in Figure 4. ECC algorithm in our LTCA model is more efficient than RSA, but as far as verification concerned, RSA is more efficient than ECC in LTCA. Regarding the total time of authentication, ECC is efficient under the same threshold value. For instance, the total time of ECC authentication is 68% of total time of RSA authentication in LTCA (3,10); the ratio between them becomes 70% in LTCA(9,10).

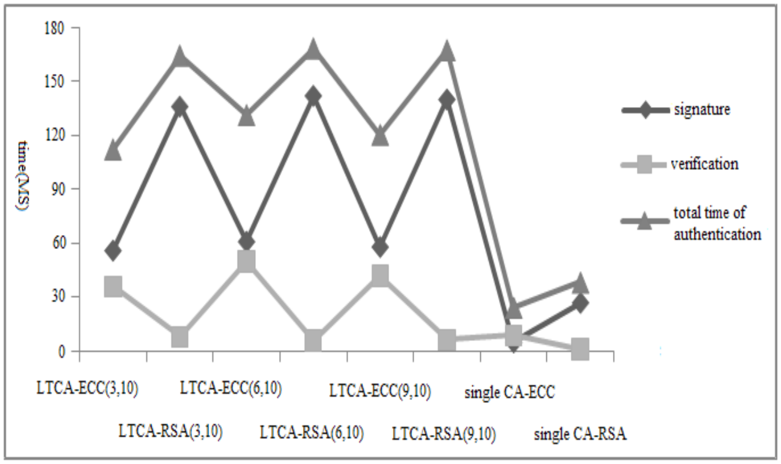


Figure 5. Performance comparison with different (t,n) mechanism with different algorithm in LTCA

6. Conclusion

Authentication in wireless networks is very important, which usually is viewed as the first defense of the network. In this paper, LTCA model based on lightweight CA idea and intrusion tolerance is constructed. Since users or nodes need to be authenticated when they access WMN, LTCA plays an important role. Our model improves the security of CA. Though LTCA adopts threshold mechanism, the lightweight method makes it efficient in computation and communication which is showed by the simulation results. As the article's limitation, we are not able to interpret all the work in detail, and we hope to improve and implement our model in the future.

Acknowledgements

This work was funded in part by the state ministry of science and technology innovation fund under NSTIF grants 10C26216205256. It was also supported by the Industrial Strategic Technology Development Program (10041740) funded by the Ministry of Knowledge Economy (MKE) Korea, and by the Natural Science Foundation of Jiangsu Province (No. BK2012461). Professor Jeong-Uk Kim is the corresponding author.

References

[1] A. Shamir, "How to Share a Secret", Communications of the ACM, ACM, vol. 22, no. 11, **(1979)**.

[2] G. R. Blakley, "Safeguarding Cryptographic Keys", Proceedings of the National Computer Conference, **(1979)** June 4-7, New York City, USA.

[3] Y. Desmed and S. Jajodia, "Redistributing Secret Shares to New Access Structures and Its Application[EB/OL]", http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.2968, **(1997)**.

[4] T. M. Wong, C. X. Wang and J. M. Wing, "Verifiable Secret Redistribution for Archive Systems", Proceedings of the 1st International IEEE Security in Storage Workshop, **(2002)** December 11, Greenbelt, Maryland, USA.

[5] P. Feldman, "A Practical Scheme for Non-interactive Verifiable Secret Sharing", Proceedings of the 28th IEEE Annual Symposium on Foundations of Computer Science, **(2011)** October 12-14, Los Angeles, CA, USA.

Copyright ⓒ 2013 SERSC

[6] J. Kim and S. Bahk, "Design of Certification Authority Using Secret Redistribution and Multicast Routing in Wireless Mesh Networks", Computer Networks, Elsevier, vol. 53, no. 1, **(2009)**.

[7] H. J. Zhu, X. D. Lin, R. X. Lu, H. Pinhan and X. M. Shen, "SLAB: Secure Localized Authentication and Billing Scheme for Wireless Mesh Networks", IEEE Transaction on Wireless Communications, IEEE, vol. 7, no. 10, **(2008)**.

[8] Z. F. Cao, H. J. Zhu and R. X. Lu, "Provably Secure Robust Threshold Partial Blind Signature", Science in China Series F: Information Sciences, China Science Press, vol. 49, no. 5, **(2006)**.

[9] M. Boneh, "Building intrusion tolerant applications", Proceedings of the 16th USENIX Security Symposium in Washington, **(2010)** August 12-15, Boston, MA, USA.

[10] J. W. Jing and Y. Z. Tian, "Intrusion Tolerance Technology on the Internet", Journal of Graduate College of China Science Institute, University of China Science Institute Press, vol. 18, **(2010)**.

[11] Z. C. Chai, Z. F. Cao and R. X. Lu, "Threshold Password Authentication against Guessing Attacks in Ad hoc networks", Ad Hoc Networks, Elsevier, vol. 5, no. 7, **(2007)**.

[12] X. L. Dong, L. H. Wang and Z. F. Cao, "EP2DF: an Efficient Privacy-preserving Date-forwarding Scheme for Service-oriented Vehicular Ad Hoc Networks", IEEE Transactions on Vehicular Technology, IEEE, vol. 60, no. 2, **(2011)**.

[13] Q. L. Xu, "Elliptic curves cryptography", Computer research and development, China Science Press, vol. 36, **(1999)**.

[14] F. Desmedt, "Threshold cryptosystems", Advances in Cryptology Crypto′89, **(1989)** August 20-24, Santa Barbara, California, USA.

[15] X. D. Qing and Y. W. Xing, "The design and implementation of a digital signature system based on elliptic curves", Computer engineering and application, China Westsouth engineering Press, vol. 28, **(2003)**.

[16] X. F. Zhang and Z. G. Qing, "The analysis of an encrypting cryptosystem based on elliptic curves", Journal of electronic science university, Chian Electronic Science University Press, vol. 30, no. 6, **(2005)**.

[17] Y. Sun and C. Meng, "Handout of OPNET Communication Simulation", National defense science and technology press, China, **(2005)**.

[18] T. Eissa, S. A. Razak and M. D. Ngadi, "Towards providing a new lightweight authentication and encryption scheme for MANET", Wireless Network, vol. 17, **(2011)**.

[19] C. Perkins, R. E. Belding and S. Das, "Ad hoc On-Demand Distance Vector(AODV) Routing", RFC3651, **(2003)**.

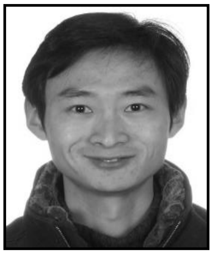
Authors

Ping Guo

She obtained her B.S and M.S degree in the Computer Software and Theory from LanZhou University, China in 1997 and 2005, respectively. She received Ph.D degree in Nanjing University of Science and Technology(NUST) in 2012. She is a lecturer in the College of Computer & Software, Nanjing University of Information Science & Technology(NUIST) from 2005 till now. Her research interests focus on wireless network security, multiple-hop wireless networks authentication and key management.

Jin Wang

He received the B.S. and M.S. degree in the Electronical Engineering from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree from the Computer Engineering Department of Kyung Hee University Korea in 2010. Now, he is a professor in the Computer and Software Institute, Nanjing University of Information Science and technology. His research interests mainly include routing method and algorithm design, performance evaluation and optimization for wireless ad hoc and sensor networks. He is a member of the IEEE and ACM.



Jiezhong Zhu

He obtained his Masters in Computer Engineering from HoHai University, China in 2004. Now, he is an associated professor in Bing Jiang College, Nanjing University of Information Science & Technology(NUIST). His research interests include wireless communication network, information security and cloud computing.

YaPing Cheng

She obtained her Masters in System Analysis and Data Integration from University of Information Science & Technology (NUIST), China in 2006. Now, she is an associated professor in the College of Computer & Software, Nanjing University of Information Science & Technology (NUIST). Her research interests include image processing, information security and digital watermarking.

Jeong-Uk Kim

He obtained his B.S. degree in Control and Instrumentation Engineering from Seoul National University in 1987, M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology in 1989, and 1993, respectively. He is a professor in SangMyung University in Seoul. His research interests include smart grid demand response, building automation system, and renewable energy.

Copyright ⓒ 2013 SERSC