Distributed Key Management in Wireless Mesh Networks based on Verifiable Secret Sharing

Peng Xiao[1], Jingsha He[2] and Yingfang Fu[3]

[1]*College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China*
E-mail: xp1984@emails.bjut.edu.cn

[2]*School of Software Engineering, Beijing University of Technology, Beijing 100124, China*

[3]*Fantai Lingshi Technology (Beijing) Limited, Beijing 100044, China*

Abstract

Wireless mesh networks (WMNs) are wireless access networks based on IP technologies, which combine the advantages of WLANs and ad hoc networks and have thus become broadband access networks with high capacity, high speed and wide coverage. Due to the multi-hop characteristics of WMNs, security has become a critical issue in WMNs and a simple and effective distributed key management is essential for the development of secure WMNs. In this paper, we propose a distributed key management scheme in WMNs by combing several key technologies, such as ECC, (t, n) threshold cryptography, verifiable secret sharing, etc. Based on a zone-based hierarchical model, our proposed key management scheme is comprised of two parts: group key management in the backbone network and session key establishment in the zone networks. We also show the security and the effectiveness of the proposed scheme through analysis and simulations.

Keywords: *wireless mesh network; distributed key management; cryptography; cheater identification*

1. Introduction

Wireless mesh networks (WMNs) are a new technology for wireless networks that combine the advantages of ad hoc networks, wireless local area networks (WLANs), wireless personal area networks (WPANs) and wireless metropolitan area networks (WMANs) for the development of commercial wireless mobile networks. WMNs offer wireless broadband access using IP technologies and have quickly become an effective means for broadband network access with high capacity, high speed and wide coverage [1, 2].

In a WMN, data packets can be transported through one or more neighboring nodes to their final destinations via a so-called multi-hop network structure. Security has thus become more critical in WMNs than in other types of networks [3, 4, 5]. In a wired network, data is transmitted to its destination through electric wires, so disclosure of information can only happen when physical links are under attack. But in a wireless network, data is transmitted via open space and any node within the radio coverage area can receive radio signals in transmission. Moreover, in WMNs, the external environment can be more hazardous due to the lack of central administration. Simple and effective key management will be useful for the development of security in WMNs.

In this paper, we propose a distributed key management scheme based on several technologies, such as ad hoc network model, elliptic curve cryptography (ECC), (t, n) threshold cryptography, verifiable secret sharing, etc. The remainder of this paper is structured as follows. After discussing some related work in Section 2, we present a zone-based hierarchical WMN model in Section 3. Based on the proposed WMN model, our proposed distributed key management is carried out in two steps: group key management in the backbone network, which is presented in Section 4, and session key establishment in zone networks, which is presented in Section 5. We analyze the security properties of our proposed scheme in Section 6 and the performance in Section 7 through simulations. Finally, in Section 8, we conclude this paper in which we also discuss some future work.

2. Related Work

Hong, et al., proposed a key distribution scheme with self-healing property, which is optimal in terms of user memory storage and efficient in terms of communication complexity [6]. But the scheme requires that there exist a group manager for the entire network with a finite number of users, which makes it mainly useful in a wired network. The scheme also lacks verification on the correctness of the received keys.

IEEE P802.11s™/D1.01 provides efficient mesh security association (EMSA) [7] based on IEEE 802.11i standard in which the 802.1x scheme and four handshakes are used to implement access authentication and key establishment. However, it requires that some special wireless nodes called mesh key distributor (MKD) be present in WMNs to generate, distribute and store keys. The MKD breaks the equality of nodes in WMNs and, in addition, if MKD is compromised, the keys stored in it might be compromised.

Fu, et al., proposed a mutual authentication scheme in WMNs [8] based on (t, n) cryptography, but it cannot verify shared secret. Therefore, there is no way of detecting and identifying any dishonest node. As the result, an incorrect key can be constructed. For example, one can forge an invalid sub signature once the secret random integer b_r becomes known, which can be calculated if a valid signature is intercepted.

Duan et al. proposed an efficient location-based compromise-tolerant key management scheme for sensor networks based on sensor deployment and localization [9]. Dahshan, et al., proposed an ECC-based distributed key management scheme for mobile ad hoc networks based on (t, n) threshold cryptography [10]. But these schemes are only suitable for certain types of networks and fail to support cheater identification.

3. Network Model

We assume a zone-based hierarchical network model for WMNs in this paper, which is shown in Figure 1 in which dashed and solid lines indicate wireless and wired links, respectively [11]. The network consists of one backbone network, one or more local area networks called zones and some scattered wired or wireless terminals.

In the backbone network, the mesh routers form an infrastructure with the properties of self-configuring, self-healing and self-organizing and there are at least two backbone routers to connect the network to the Internet. All backbone routers share an exclusive database of authorized certificates in an offline Certificate Authority (CA) that is supported by Internet Service Providers (ISPs) or network carriers. The CA connects to the network only when a malicious attacker is detected or when it is time for key update. The WMN backbone can be built with different radio technologies.

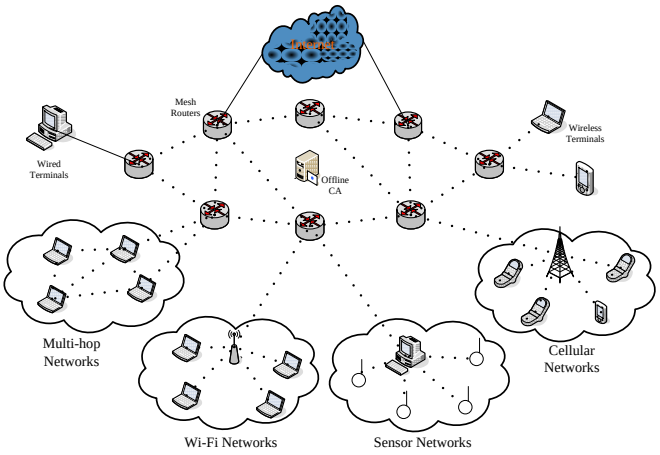

Figure 1. The Network Model

Each zone is connected to the backbone network through its border mesh routers called gateways to enable the integration of existing wireless networks, such as multi-hop networks, Wi-Fi networks, sensor networks, cellular networks, etc. In each zone, there is at least one mobile node called Access Point (AP) that connects to the backbone through Mesh Access Points (MAPs) in multi-hop networks and microwave towers in cellular networks. These APs may use different radio technologies, which require that the backbone border routers support the various radio technologies. There is also a database that stores user information, such as user ID, zone ID, authorized key, etc., in the zone. User terminals can roam from one zone to another or hand off from one AP to another in the same or different zones.

Terminals with Ethernet interfaces can also be connected to mesh routers via Ethernet links. For terminals with the same radio technologies as the mesh routers, they can directly communicate with mesh routers. If different radio technologies are used, terminals must communicate with a zone's AP which has Ethernet connections to mesh routers. Especially, mesh terminals can access the network through mesh routers as well as directly meshing with other mesh terminals to provide routing capabilities for improved connectivity and coverage.

We assume that terminals in a zone network communicate with each other within a relatively shorter range and those in a backbone network communicate with each other with a relatively longer range and that the cost of communication through the backbone network is higher than that in the zone networks.

4. Backbone Key Management

Since there is no CA or administrator center online in the backbone network, *n* selected mesh routers with higher performance can form a virtual CA to manage keys using a (*t*, *n*) threshold cryptographic method [12].

4.1. Certificates

Before accessing the network, all mesh nodes need to acquire a legal certificate from the offline CA, which is issued by Internet Service Providers (ISPs) or network carriers. In this paper, we adopt ECC for generating the certificates, for it can offer the same level of security

with smaller key sizes and faster computation time, which leads to lower power consumption than other public-key cryptographic algorithms such as the RSA.

The cryptography is built on a suitably chosen elliptic curve E defined over a finite field F_q of characteristic p and a base point $P \in E(F_q)$. According to [13], some domain parameters can be defined as follows:

(1) a field size q, where q is a prime power (in practice, either $q = p$, an odd prime, or $q = 2^m$).

(2) an indication *FR* (field representation) of the representation used for the elements of F_q.

(3) two field elements a and b in F_q that define the equation of the elliptic curve E over F_q (e.g., $y^2 = x^3 + ax + b$ in the case $p > 3$ and $y^2 + xy = x^3 + ax + b$ in the case $p = 2$).

(4) a finite point $P = (x_P, y_P)$ of prime order in $E(F_q)$, and $P \neq O$ where O denotes the point at infinity.

(5) the order n of the point P with $nP = O$ and $n > 2^{160}$ as commonly recommended.

(6) a cofactor $h = \#E(F_q)/n$ where $\#E(F_q)$ denotes the number of F_q -rational points on E.

Given a valid set of domain parameters (*q, FR, a, b, P, n, h*), an entity A's private key is a random integer $\omega_A \in_R [1, n-1]$ while its public key is the point $W_A = \omega_A P$.

4.2. Group Key Establishment

Based on the (*t, n*) threshold secret sharing, the secret group key SK is partitioned into *n* pieces $SK_1, ..., SK_n$ and are assigned to *n* selected routers. Then, any *t* out of the *n* routers can reconstruct *SK* while any *m* out of the *n* routers cannot reconstruct SK if $m<t$.

A traditional (*t, n*) threshold secret sharing method is comprised of three parts: system parameters, secret distribution algorithm and secret reconstruction algorithm [14].

(1) System parameters: the private key SK is the secret to be shared and is defined in a finite field $GF(p)$ where p is a prime greater than SK and n; $d_1, d_2, ..., d_n$ are different integers defined in $GF(p)$ that are represented as the public identification of *n* participants.

(2) Secret distribution algorithm: a (*t-1*)-degree polynomial $f(x) = a_{t-1}x^{t-1} + ... + a_1 x + a_0 \mod p$ in the finite field $GF(p)$ is chosen, where $a_{t-1}, ..., a_1$ are random integers and $a_0 = SK$; then the key pieces $SK_i = f(d_i) \mod p$ are calculated and delivered to each responding participant through secure channels.

(3) Secret reconstruction algorithm: t coordinates (d_1, SK_1), (d_2, SK_2), ..., (d_t, SK_t) can be acquired through participants' cooperation. According to Lagrange interpolation polynomial, $SK = f(0) = \sum_{i=1}^{t} SK_i \prod_{j \neq i, j=1}^{t} \frac{d_j}{d_j - d_i} \mod p$.

When a new participant receives at least *t* key pieces, it can reconstruct the secret key.

However, there are several disadvantages when the above method is used in WMNs:

(1) Since a dishonest participant may deliver an incorrect key piece or there may be something wrong happened in data transmission, the correct secret key *SK* may not be reconstructed.

(2) A malicious attacker among the *n* participants may deliberately deliver a faked key piece to others and, at the same time, receive all the correct key pieces from others so that only it can reconstruct the correct secret key *SK* while those who receive a faked key piece cannot.

(3) In a wireless mobile environment, an attacker can break through one key piece holder in a limited amount of time and then proceed to attacking another key piece holder. So in a not very long time, it may break through *t* key holders, acquire *t* key pieces and calculate the shared secret key *SK* .

4.3. Cheater Detection

How to detect whether an incorrect key piece is received or not? Let us consider the equation in the matrix form: $D \bullet A = S$ in which

$$D = \begin{bmatrix} d_1^{t-1} & ... & d_1 & 1 \\ d_2^{t-1} & ... & d_2 & 1 \\ ... & ... & ... & ... \\ d_t^{t-1} & ... & d_t & 1 \end{bmatrix}, A = \begin{bmatrix} a_{t-1} \\ a_{t-2} \\ ... \\ a_0 \end{bmatrix}, S = \begin{bmatrix} SK_1 \\ SK_2 \\ ... \\ SK_n \end{bmatrix} \text{ and } \overline{D} = \begin{bmatrix} d_1^{t-1} & ... & d_1 & 1 & SK_1 \\ d_2^{t-1} & ... & d_2 & 1 & SK_2 \\ ... & ... & ... & ... & ... \\ d_t^{t-1} & ... & d_t & 1 & SK_t \end{bmatrix}.$$ Then

the substance of the Lagrange interpolation polynomial has a unique feasible solution *A* of the matrix equation when *D* and *S* known. A necessary and sufficient condition is that the rank of the augmented matrix $\overline{D}$ and the rank of *D* are the same and equal to *t*, denoted as $R(\overline{D}) = R(D) = t$. If there exists an incorrect key piece (d_i, SK_i) and the condition $R(\overline{D}) = R(D) = t$ is satisfied, then a wrong secret key *SK* will be reconstructed.

In the initial stage of our key establishment, a public key piece (d_0, SK_0) generated by the offline CA is broadcasted in the whole network. After *t* key pieces are collected, a new matrix equation $D' \bullet A' = S'$ is established, where

$$D' = \begin{bmatrix} d_1^{t-1} & ... & d_1 & 1 \\ d_2^{t-1} & ... & d_2 & 1 \\ ... & ... & ... & ... \\ d_t^{t-1} & ... & d_t & 1 \\ d_0^{t-1} & ... & d_0 & 1 \end{bmatrix}, A' = \begin{bmatrix} a_{t-1} \\ a_{t-2} \\ ... \\ a_0 \end{bmatrix}, S' = \begin{bmatrix} SK_1 \\ SK_2 \\ ... \\ SK_n \\ SK_0 \end{bmatrix} \text{ and } \overline{D}' = \begin{bmatrix} d_1^{t-1} & ... & d_1 & 1 & SK_1 \\ d_2^{t-1} & ... & d_2 & 1 & SK_2 \\ ... & ... & ... & ... & ... \\ d_t^{t-1} & ... & d_t & 1 & SK_t \\ d_0^{t-1} & ... & d_0 & 1 & SK_0 \end{bmatrix}.$$

At the moment, we can deduce that $R(\overline{D}) = R(D) \le t+1$. So,

(1) If $R(\overline{D}) = R(D) < t$, there are infinite number of solutions for the equation. Therefore, more key pieces have to be collected to get the unique solution until $R(\overline{D}) = R(D) = t$.

(2) If $R(\overline{D}) = R(D) = t$, there is an unique solution for the equation. So, a correct secret

key SK will be reconstructed.

If $R(\overline{D}) = R(D) = t+1$, there is no feasible solution for the equation. So, there exists at least one incorrect key piece from the participants. Consequently, a cheater is detected.

4.4. Cheater Identification

If a cheater is detected in the network, it should be identified. Thus, in our scheme, all key pieces must be delivered with the owners' digital signature.

When a new participant acquires the key piece from an existing participant, the latter must deliver its own key piece along with its digital signature. After the new participant has collected t key pieces and detected a cheater in the network as described in the last section, it will broadcast a request to arouse the offline CA in the network. And then which it will deliver all the collected key pieces with their digital signatures to the CA. The offline CA can verify which key pieces are faked through the pre-selected secret (*t-1*)-degree polynomial $f(x)$ and which participants are dishonest through their digital signatures and registered certificates.

(1) If there are truly faked key pieces that cannot satisfy $f(x)$, those who generated them are the cheaters, which cannot denied due to their digital signatures. In this case, the group key must be updated and the offline CA will revoke the current group key SK and start a new key establishment.

(2) If there is no faked one, which means all the key pieces satisfy $f(x)$, the new participant who provides them is a cheater. In this case, the group key doesn't need to be updated.

Once being identified, the cheater must be dealt with as follows.

(1) With a strict security policy, the cheater will be disassociated from the network and its certificate will be revoked by the offline CA, which is broadcast through the entire network.

(2) With a weaker security policy, the cheater will be recorded by the offline CA and its credit level will be reduced. After losing its credit many times, which reaches a preset threshold, the cheater will be disassociated with its certificate revoked.

4.5. Key Update

Following are several conditions in which the group key needs to be updated.

(1) A new mesh router connects to the backbone network.

(2) An existing mesh router leaves the backbone network.

(3) A cheater is detected in the network as described in the last section.

(4) To prevent a mobile attacker from breaking *t* key pieces, every key piece should be updated within a defined cycle *T*. Only after at least *t* key pieces are obtained in the same cycle, can the secret be reconstructed.

Key update involves the following four steps:

(1) The offline CA is aroused in the network.

(2) The CA constructs a new group key SK' and selects a new polynomial $f'(x)$. The new key pieces (d_i, SK_i') are calculated and delivered to n selected mesh routers. Then, the CA disconnects itself from the network and remains offline.

(3) A mesh router requests (t-1) key pieces from other mesh routers.

(4) After t key pieces are collected, the mesh router reconstructs the new group key SK', during which cheater detection and identification can be carried out as described.

4.6. Summarized Algorithms

Here we give out summarized algorithms to describe our distributed key management, including key pieces distribution algorithm and group key reconstruction algorithm. The first algorithm is operated by the offline CA, while the second algorithm is operated by a new participant.

Algorithm 1: Key Pieces Distribution

Input: t, n, p
Output: (d_i, s_i) for all i

```
while in the same cylce do
    for i = 1 to t − 1 do
        a_i=Random.Generate();
    end
    d_0=Random.Generate();
    SK_0=∑_0^{t−1} a_i ∗ (d_0)^i mod p;
    boardcasts (d_0, SK_0) in the whole network;
    for j = 1 to n do
        requsets participant j's authentication message;
        if j is authencated then
            d_j=TPM.Generate();
            s_j=∑_0^{t−1} a_i ∗ (d_j)^i mod p;
            sends (d_j, s_j) to user j;
        end
    end
end
if the cycle ends then
    starts a new key pieces distribution process;
end
```

Figure 2. Key Pieces Distribution Algorithm

Algorithm 2: Group Key Reconstruction

Input: t, n, p
Output: SK

```
counter=0;
requests key pieces from the network;
while a response from a key holder j is received do
    gets (d_j, s_j) and its digital signature SIG_j();
    counter++;
    if counter == t then
        break;
    end
end
construts the augmented matrix D̄ and computes its rank Rank(D̄);
if Rank(D̄) < t then
    requests another key piece;
end
if Rank(D̄) == t then
    SK = Σ_1^t(SK_i) Π_1^t (d_j/(d_j − d_i)) mod p where i ≠ j;
end
if Rank(D̄) > t then
    received incorrect key pieces;
    arouses the offline CA and sends it all (d_j, s_j, SIG_j());
    starts a new group key reconstruction process;
end
```

Figure 3. Group Key Reconstruction Algorithm

5. Zone Key Management

Before a terminal node accesses the zone network, mutual authentication must be performed between the terminal node and its corresponding mesh AP. Therefore, they need to acquire a valid temporary session key through key agreement.

5.1. Key ECC Agreement

Two entities A and B can complete key agreement with their key pair *(ω, W)* in which $W_A = \omega_A P$ as described in Section 4.1:

(1) A selects $r_A \in_R [1, n-1]$, computes point $R_A = r_A P$ and sends R_A to B.

(2) B selects $r_B \in_R [1, n-1]$, computes point $R_B = r_B P$ and sends R_B to A;

(3) A checks to see whether R_B is not the same as O, R_B satisfies the equation of E and x_B, y_B are elements in the F_q. If the validation fails, A terminates the protocol. Otherwise, A computes $s_A = (r_A + R_A \omega_A) \bmod n$ and $K = hs_A(R_B + R_B W_B)$. If $K = O$, A also terminates the protocol.

(4) B does the same validation and if it fails, B terminates. Otherwise, B computes $s_B = (r_B + R_B \omega_B) \bmod n$ and $K = hs_B(R_A + R_A W_A)$. If $K = O$, B terminates.

(5) The session key is the point K.

We can see that

$$K = hs_A(R_B + R_B W_B) = hs_B(R_A + R_A W_A) = h(r_A r_B + r_A \omega_B R_B + r_B \omega_A R_A + \omega_A \omega_B R_A R_B)P .$$

5.2. Session Key Establishment

A terminal node and its AP complete their session key establishment in the following two steps:

(1) They finish mutual authentication in the way of EAP-TLS with their certificates issued by the offline CA. When they can both get authenticated, key agreement will proceed. Otherwise, the process terminates.

(2) They exchange their selected random integers and calculate the shared session key.

They can now communication using the session key. For each new session, a new key agreement will be carried out.

6. Security Analyses

The security properties of our proposed distributed key management in WMNs are listed as follows.

(1) Zone-based hierarchical network topology can be easily expanded and integrated with current network infrastructure.

(2) Distributed key management can improve security, robustness and trustworthiness of WMNs compared to EMSA.

(3) Verifiable secret sharing includes cheater detection and identification, which can isolate the dishonest participants from the network compared to Fu's scheme.

(4) Mutual authentication can help guarantee the identities of network nodes.

(5) Periodic key update increases the difficulty of deciphering the current group key.

(6) Perfect forward secrecy: when the private keys of all participants are deciphered, the previous session keys are still secure. Only if t key pieces are collected in a limited cycle T, can the group key in this cycle be compromised.

(7) Known key security: even if attackers know all previous session keys, the protocol can still guarantee that the current session key is secure, for previous group keys have no relationship with the current session key.

7. Simulation Results

We have also performed some simulation using OPNET 14.5 to verify the effectiveness of our scheme with respect to the security properties. The simulation scenario assumes an area of $1000 \times 1000m^2$ in which all the mobile nodes use a protocol that is based on IEEE 802.11b.

Figure 4 shows the success rate of key distribution and reconstruction when n varies from *1* to *16* and $t=2, 4, 8$. We can see that if $n<t$, key reconstruction cannot be implemented and that when the threshold t increases, the success rate gets lower because of radio collision and transmission delay. Figure 5 shows the length of time for the identification of a cheater when there is a malicious node in the network that provides an incorrect key piece to others. As the

threshold *t* increases, more key pieces need to be verified and more time is needed for cheater identification.

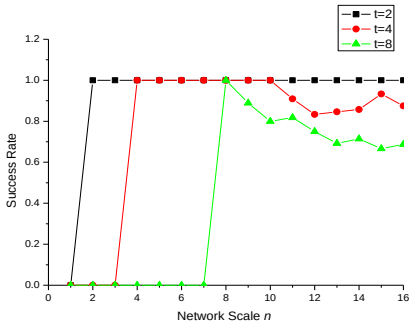

Figure 4. Success Rate of Group Establishment

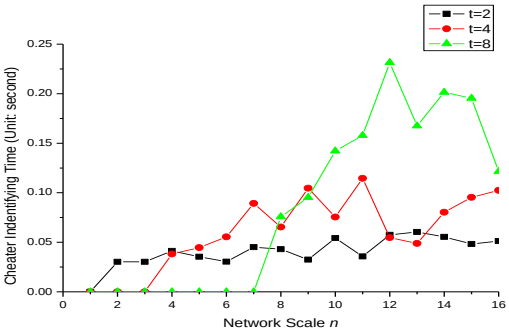

Figure 5. Time for Cheater Key Identification

Figure 6 shows the success rate on session key establishment when the scale of the zone network *n* varies from *1* to *16* and all the *n* zone members connect to the same zone AP at the same time. Due to collision of radio signals, some members cannot complete the establishment immediately and will retry for a few times. And the retry times is set as *1*, *2*, and *3* in three curves respectively. Figure 7 shows the average time for session key establishment in the zone network.

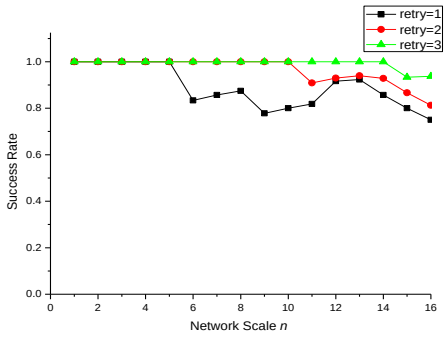

Figure 6. Success Rate of Session Key Establishment

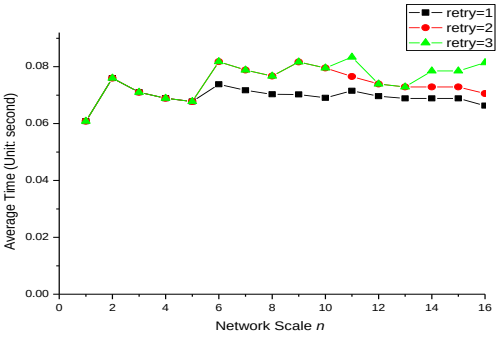

Figure 7. Average Time of Session Key Establishment

We also compare our scheme with Shamir's [14]. Figure 8 shows the success rate for group key establishment with the scale of the zone network n varying from 1 to 16 and the threshold t=4 while Figure 9 shows the average time for group key establishment in the zone network, which is defined as the total amount of time for group key establishment divided by the total number of network members. We can see from the comparison that the success rates of the two schemes are very close while the average time in our scheme is a little longer. This shows that our proposed scheme can maintain the success rate at the cost of some additional computation in order to deal with the security problems.

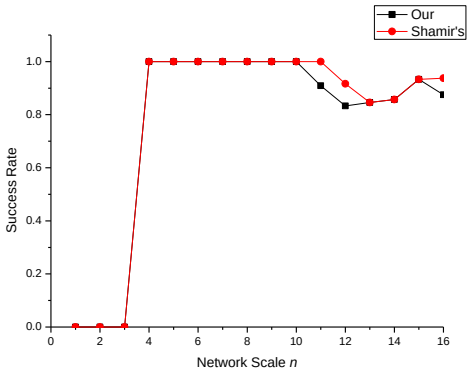

Figure 8. Success Rate Comparison

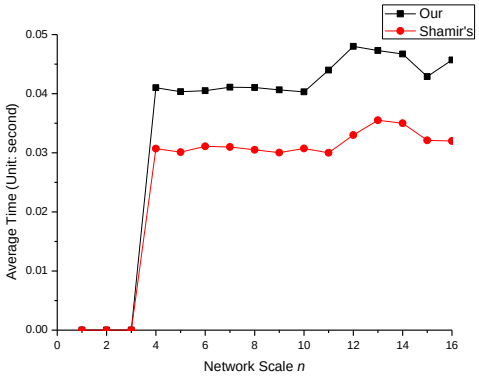

Figure 9. Average Time Comparison

8. Conclusions

In this paper, we presented a distributed key management scheme for the security of WMNs. The scheme is based on several technologies, such as ad hoc network model, ECC, (*t*, *n*) threshold cryptography, verifiable secret sharing, etc. In the future, we will consider distributed key management in handoff and roaming scenarios for WMNs to further improve our protocol.

Acknowledgements

This work was supported by Beijing Natural Science Foundation under grant number 4122009, and National Natural Science Foundation of China under grant number 61272500.

References

[1] P. Mohapatra, L. Jian and G. Chao, "QoS in Mobile Ad Hoc Networks", IEEE Wireless Communications, vol. 10, **(2003)**, pp. 44-52.
[2] I. F. Akyildiz, X. Wang and W. Wang, "Wireless Mesh Networks: a Survey", Computer Networks, vol. 47, **(2005)**, pp. 445-487.
[3] T. Gamer, L. Volker and M. Zitterbart, "Differentiated Security in Wireless Mesh Networks", Security and Communication Networks, vol. 4, **(2011)**, pp. 257-266.
[4] H. Redwan and K. Kim, "Survey of Security Requirements, Attacks and Network Integration in Wireless Mesh Networks", 2008 New Technologies, Mobility and Security Conference and Workshops, IEEE Computer Society, NW Washington, DC USA, **(2008)**, pp. 3-9.
[5] M. Cesana1, A. Boukerche and A. Zomaya, "Security for QoS Assured Wireless and Mobile Networks", Security and Communication Networks, vol. 4, **(2011)**, pp. 239-241.
[6] D. Hong and J. S. Kang, "An Efficient Key Distribution Scheme with Self-healing Property", IEEE Communications Letters, vol. 9, **(2005)**, pp. 759-761.
[7] 802.11 Working Group of the IEEE 802 Committee, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE P802.11s™/D1.01, **(2007)**, pp. 1-780.
[8] Y. Fu, J. He, L. Luan, R. Wang and G. Li, "A Zone-based Distributed Key Management Scheme for Wireless Mesh Networks", 32nd Annual IEEE International Computer Software and Applications Conference, Turku, Finland, **(2008)**, pp. 68-71.
[9] M. J. Duan and J. Xu, "An Efficient Location-based Compromise-tolerant Key Management Scheme for Sensor Networks", Information Processing Letters, vol. 111, **(2011)**, pp. 503-507.
[10] H. Dahshan and J. Irvine, "An Elliptic Curve Distributed Key Management for Mobile Ad Hoc Network", 2010 IEEE 71st Vehicular Technology Conference, Taipei, Taiwan, **(2010)**, pp. 1-5.
[11] P. Xiao, J. He and Y. Fu, "A Secure Mutual Authentication Protocol for Roaming in Wireless Mesh Networks", Journal of Networks, vol. 7, **(2012)**, pp. 267-274.

[12] Y. Fu, J. He, R. Wang and G. Li, "Mutual Authentication in Wireless Mesh Networks", 2008 International Conference on Communications, Beijing, China, **(2008)**, pp. 2606-2610.
[13] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An Efficient Protocol for Authenticated Key Agreement", Designs, Codes and Cryptography, vol. 28, **(2003)**, pp. 119-134.
[14] A. Shamir, "How to Share a Secret", Communications of the ACM, vol. 22, **(1979)**, pp. 612-613.

Authors

Peng Xiao is currently a Ph.D. candidate in the College of Computer Science and Technology at Beijing University of Technology. His research interests include network security, trusted authentication in WMNs and Ad Hoc networks.

Jingsha He is currently a professor of the School of Software Engineering at Beijing University of Technology. His research interests include network security and wireless communication technologies.

Yingfang Fu is currently a researcher in Fantai Lingshi Technology (Beijing) Limited, Beijing 100044, China. Her research interests include network security and trusted computing in WMNs.