Computing and Informatics, Vol. 30, 2011, 1167–1180

MODIFIED AUTONOMOUS KEY MANAGEMENT SCHEME WITH REDUCED COMMUNICATION/COMPUTATION COSTS IN MANET

Chu-Hsing Lin

Department of Computer Science, Tunghai University
181, Section 3, Taichung Port Road
Taichung 40704, Taiwan
e-mail: chlin@thu.edu.tw

Chen-Yu Lee, Deng-Jyi Chen

Department of Computer Science, National Chiao Tung University
1001 Ta-Hsueh Road
HsinChu, 30050, Taiwan
e-mail: chenyu@cs.nctu.edu.tw, djchen@csie.nctu.edu.tw

Abstract. The growing applications of mobile ad hoc networks (MANETs) made related security issues much more important. B. Zhu et al. proposed management scheme using Shamir's secret sharing scheme to construct ar tonomous Key Management (AKM) hierarchy structure. However, Shamir's s sharing in AKM to control key hierarchy incurs high message transmission This paper modifies the secret sharing scheme and applies it to AKM to r communication and computation costs.

Keywords: Mobile ad hoc network, key management, autonomous key ma ment

1 INTRODUCTION

Key management within a *Mobile Ad hoc Network (MANET)* is a security issue that cannot be ignored. Many researchers have dedicated themselves to this field since 1999. Some schemes are suitable for a limited number of nodes and are inefficient, insecure, or unreliable when the nodes increase [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Nodes may join the MANET and leave later normally. Thus, the key management scheme in MANET must be dynamic. The main challenge of MANET is that each node handles the joining or leaving of nodes with the limited resources, such as CPU computation, storage, and the power consumption [13]. The mobility of a MANET increases its unreliability and limits the bandwidth of wireless environment due to frequent topology changes.

B. Zhu et al. proposed a key management scheme [14] using the secret sharing method [15, 16, 17, 18] to construct an AKM hierarchy structure with flexibility and adaptivity. This scheme needs no central party to control the key structure, and each node cooperates to create virtual nodes in building the key hierarchy. The method proposed in [19] dynamic group key management schemes with forward secrecy and backward secrecy based on elliptic curve cryptosystem (ECC) [20], forming a self-certified public key cryptosystem [21].

However, a message of 2 048 bits would make computing or calculating AKM communication difficult. Thus, this study modifies the design of each operation in the AKM scheme. Section 2 briefly introduces Shamir's secret sharing scheme and AKM key management. Section 3 describes the modified AKM, which reduces the share size with the same security properties. Section 4 discusses the performance improvement compared with the original AKM. Results indicate improved performance of communication and a computation cost reduction to $\frac{1}{t}$ of the original AKM.

2 RELATED WORKS

2.1 Shamir's Secret Sharing Scheme

Let t, n be positive integers, $t \leq n$. Shamir proposed a (t, n)-threshold scheme in 1979 [15]. His scheme is a method of sharing a key K among a set of n participants in such a way that any t participants can compute the value of key K, but no group of $t-1$ participants can do so.

2.1.1 The Shamir (t, n)-Threshold Scheme in Z_p

D (the dealer) chooses n distinct, nonzero elements of Z_p, denoted x_i, $1 \leq i \leq n$, where $p > n$ is a large prime. D gives the values x_i to P_i, and each value x_i is public.

2.1.2 Share Distribution

1. Suppose D wants to share a key $K \in Z_p$. D secretly chooses (independently and randomly) $t-1$ elements of Z_p, $a_1, \ldots, a_{t-1}$.
2. For $1 \leq i \leq n$, D computes $y_i = a(x_i)$, where

$$a(x) = K + \sum_{j=1}^{t-1} a_j x^j \bmod p.$$

   Thus

$$y_i = a(x_i) = K + \sum_{j=1}^{t-1} a_j x^j \bmod p.$$

3. For $1 \leq i \leq n$, D gives the share y_i to P_i.

2.1.3 Proactive Security

It is difficult to compromise the secret key K under (t, n)-threshold scheme unless the adversary collects at least t shares. In practice, since each share exists in a machine, the risk of the secret key being compromised depends on the security of machine. For security concerns, it is necessary to update each share for a period of time. A proactive threshold scheme allows users to refresh shares without disclosing the secret key.

1. Let

$$y_i = a(x_i) = K + \sum_{j=1}^{t-1} a_j (x_i^j) \bmod p$$

   be the original share of key K for P_i.
2. The dealer D then computes

$$y_i' = a(x_i') = \sum_{j=1}^{t-1} a_j (x_i'^j) \bmod p.$$

3. For $1 \leq i \leq n$, D gives the share y_i' to P_i.
4. For $1 \leq i \leq n$, P_i computes $(y_i + y_i')$ as a new share.

2.2 Autonomous Key Management (AKM)

Autonomous key management (AKM) for a mobile ad hoc network (MANET) with a large number of nodes is based on a hierarchical structure to provide flexibility and adaptivity. Every leaf node in the logical tree structure is a real ad hoc device, and the other nodes are virtual nodes. The root node holds the global secret key, and AKM distributes key shares to its children recursively from the root down to the leaves using Shamir's secret sharing scheme.

Every node except the AKM root node must store its own public key pair and its parent node secret share. The secret share each virtual branch node holds is as the secret key, and the public key can be generated using any asymmetric cryptographic scheme, such as RSA. Additionally, every real node has its PKI key pair before joining AKM.

A tree with node A as its root is called region A. AKM includes seven node-based/region-based operations from node joining, region partitioning, to node leaving. AKM runs dynamically with continuous node joining/leaves. Section 3 describes these details.

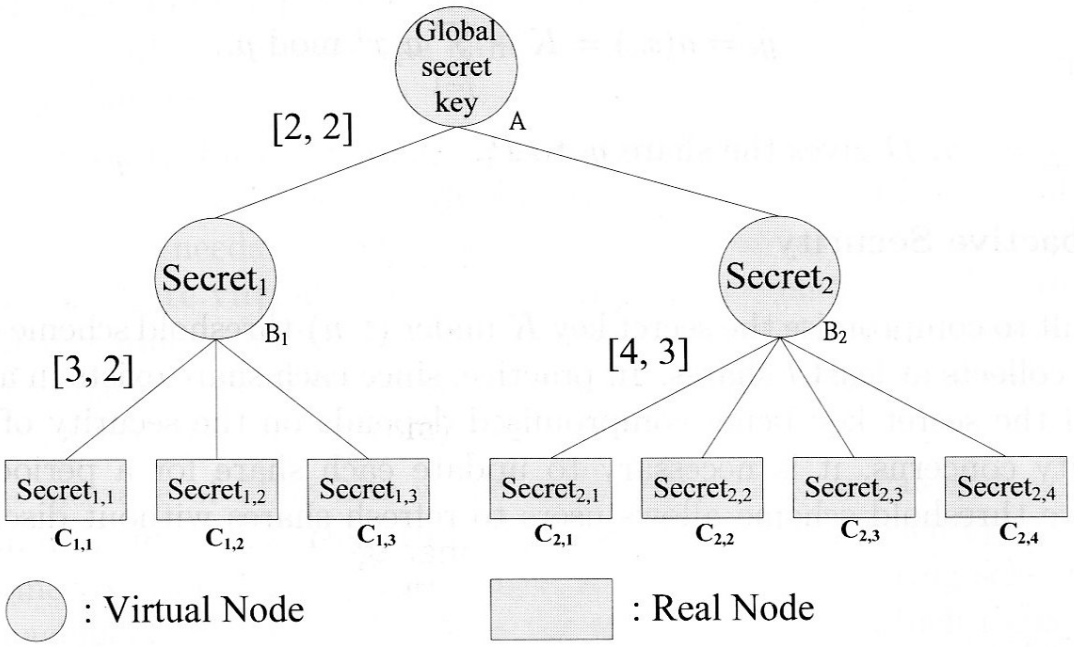


Fig. 1. Example of AKM

3 MODIFIED AKM

This section modifies the secret sharing of AKM. AKM runs dynamically in seven node-based/region-based operations. The seven operations are update, join, leave, merge, partition, expansion, and contraction.

These operations are designed based on the following rules:

1. All leaves in the hierarchy of AKM are real nodes. Each real node i has its own secret key SK_i, and $PK_i = g^{SK_i} \bmod p$, where g is a random generator.
2. The non-leaf nodes are virtual nodes, and their secret keys are generated directly/indirectly from real nodes through some region-based operations.
3. A tree with node A as root is called $Region_A$. For example, region A has virtual nodes B_1, B_2, and real nodes $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, $C_{2,1}$, $C_{2,2}$, $C_{2,3}$, and $C_{2,4}$. The number of the nodes that know the secret of region is *Overall Region Size (ORS)*.

4. The *Regional Trust Coefficient (RTC)* is the ratio of the threshold to ORS, and indicates how secure the region is. The AKM sets a *Global Trust Coefficient (GTC)* as a lower bound of all the RTC. Figure 1 shows an example, in which the ORS is 4 and RTC is 0.75 of the region B_2. The GTC of region A would be 0.2.

3.1 Function Update

Function update prevents any intruders from compromising the secret, and the AKM updates keys periodically. First, the region with (n, t)-threshold must select t nodes and each node is indicated as node $i \in 1, \ldots, t$.

Each node i generates update share $S_{i,j}(1 \leq j \leq n)$ of key 0. The node i selects random numbers $x_j(1 \leq j \leq n)$ and $r_d(1 \leq d \leq i-1)$ to compute coefficients $a_d = (r_d|0)(1 \leq d \leq t-1)$. $S_{i,j} = a(x_j = \sum_{r=0}^{t-1} a_r(x_j)^r \pmod p$, for $1 \leq j \leq n$. Node i then distributes $S_{i,j}$ to node $j \in 1, \ldots, n$. When node j receives the update shares distributed from other t nodes in the region, it computes a new share

$$S'_j = S_j + \sum_{i=1}^{t} S_{i,j} \pmod p. \tag{1}$$

The previous section describes how AKM can manage its secret sharing hierarchical structure using seven region-based functions. These operations cover all possible region changes from node joining to leaving. The key update frequency in MANET is adjustable depending on the application environment. If the frequency is high, the MANET would be secure enough against adversaries, but would result in lower performance and heavy power consumption. On the contrary, if the frequency is low, the communication between nodes in MANET suffers from key inconsistency after many nodes join and leave continuously.

3.2 Function Join

Function Join is used when a node i wants to join a (t, n)-threshold region. The node sends a request to node $j \in 1, \ldots, t$ in the region. Upon receiving the request, node j checks its *certificate revoking list (CRL)* first. If node j accepts the request, it computes a partial share S'_j of node i:

$$S'_j = S_j l_j(i) + \Delta_j \pmod q \tag{2}$$

where

$$l_j(i) = \prod_{r=1, r\neq j}^{t} \frac{ID_i - ID_r}{ID_j - ID_r} \pmod q, \quad \Delta_j = \sum_{r=1, r\neq j}^{t} \sigma(j-r) \cdot S_{j,r} \tag{3}$$

that $S_{j,r}$ is a number which pairs of nodes $(j,r) \in 1 \leq j \leq t, 1 \leq r \leq t$, and

$$\sigma(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ 0, & \text{otherwise} \end{cases}$$

After receiving all partial shares, node i generates its secret share S_i:

$$S_i = \sum_{j=1}^{t} S'_j = \sum_{j=1}^{t} S_j l_j(ID_i) + \sum_{j=1}^{t} \Delta_j \pmod{q}. \tag{4}$$

3.3 Function Leave

Function Leave is used when a node leaves a region. Any node j removes the certificate of node i from its key management records when receiving a leave request from node i or detecting the node leaves. The share key of node j does not change until the AKM updates key periodically.

3.4 Function Merge

Function Merge is used when the number of nodes in a region is below the threshold. The region is simply divided into many parts and they join to the other sibling regions respectively. As in Algorithm 1, AKM performs Function Merge on region S_i and merges its nodes $S_{i,1}$ to $S_{i,r}$ into regions S_j and S_k as $S_{j,(n+1)}, \ldots, S_{j,(n+p)}$ and $S_{k,(n+1)}, \ldots, S_{k,(n+q)}$.

Algorithm 1 Merge

Require: The merged region S_i which contains nodes $S_{i,1}, \cdots, S_{i,r}$, and the destination t regions $S_{D_0}, S_{D_1}, \cdots, S_{D_{t-1}}$.

Ensure: Region $S_{D_0}, S_{D_1}, \cdots, S_{D_{t-1}}$.

1: Separate S_i into t parts: $[S_{i,1}, \cdots, S_{i,\lceil \frac{r}{t} \rceil}], [S_{i,\lceil \frac{r}{t} \rceil+1}, \cdots, S_{i,2\lceil \frac{r}{t} \rceil}]$, $\cdots, [S_{i,(t-2)\lceil \frac{r}{t} \rceil+1}, \cdots, S_{i,(t-1)\lceil \frac{r}{t} \rceil}], [S_{i,(t-1)\lceil \frac{r}{t} \rceil+1}, \cdots, S_{i,r}]$.
2: **for** $u = 0$ to $t-2$ **do**
3: **for** $v = 1$ to $\lceil \frac{r}{t} \rceil$ **do**
4: *Join* $S_{i,u\lceil \frac{r}{t} \rceil+v}$ into S_{D_u}
5: **end for**
6: **end for**
7: **for** $v = 1$ to $r - t\lfloor \frac{r}{t} \rfloor$ **do**
8: *Join* $S_{i,(t-1)\lceil \frac{r}{t} \rceil+v}$ into $S_{D_{t-1}}$
9: **end for**

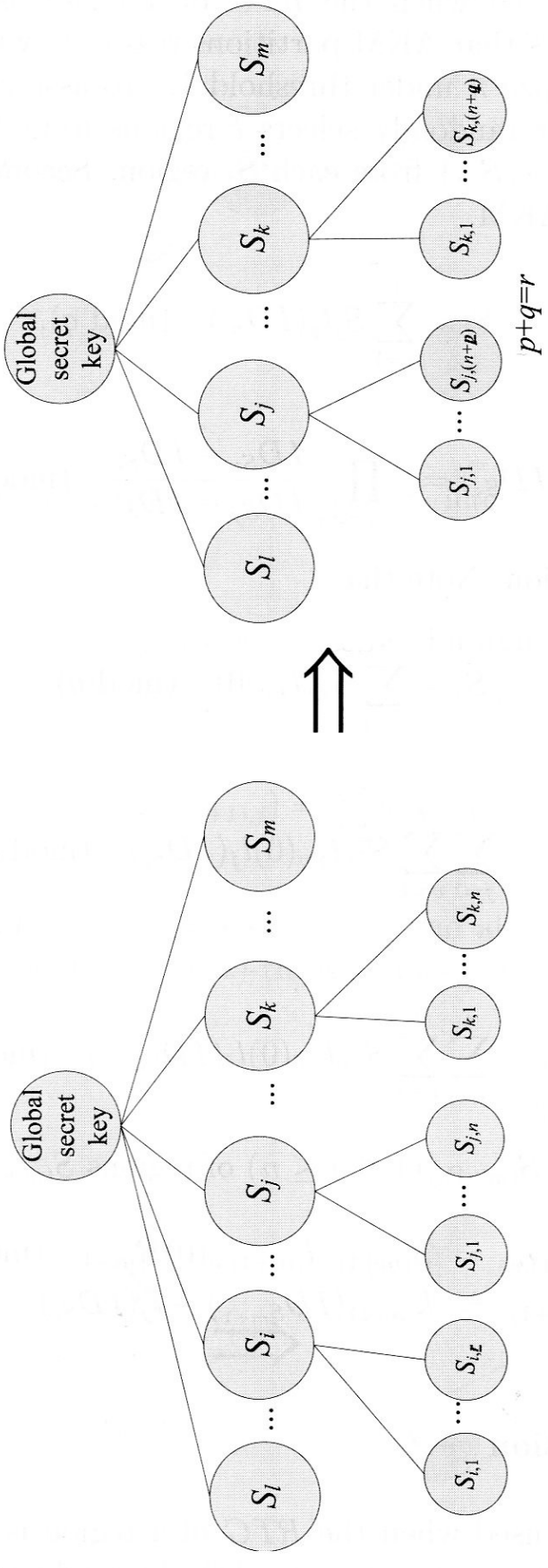


Fig. 2. Function Merge – merges S_i into S_j and S_k

3.5 Function Partition

Function Partition is used when the RTC of a region is under the GTC. For example, Figure 3 shows that AKM partitions region S_i with $2n$ nodes into S_i and $S_{(m+1)}$ with the same size n under threshold k. To assign the secret share to the nodes in $S_{(m+1)}$, it first randomly selects t regions from S_1 to S_m and randomly chooses t nodes $\{S_{j,1}, \cdots, S_{j,t}\}$ from each S_j region. Second, it creates a new node $S_{(m+1)}$, and joins into AKM.

Note that

$$S_i = \sum_{j=1}^{t} S_j l_j(ID_{S_i}) \pmod{q}, \tag{5}$$

where

$$l_j(ID_{S_j}) = \prod_{r=1, r\neq j}^{t} \frac{ID_{S_i} - ID_{S_r}}{ID_{S_j} - ID_{S_r}} \pmod{q} \tag{6}$$

by Lagrange interpolation. Note that

$$S_j = \sum_{v=1}^{t} S_{j,v} l_{j,v}(0) \pmod{q}. \tag{7}$$

Thus

$$S_i = \sum_{j=1}^{t} \sum_{v=1}^{t} S_{j,v} l_{j,v}(0) l_j(ID_{S_i}) \pmod{q}. \tag{8}$$

We also can get

$$S_{m+1} = \sum_{j=1}^{t} \sum_{v=1}^{t} S_{j,v} l_{j,v}(0) l_j(ID_{S_{m+i}}) \pmod{q}. \tag{9}$$

To generate each share $S_{(m+1),j} (1 \leq j \leq n)$ of regions $S_{(m+1)}$, $S'_{(m+1),v}$, where

$$S'_{(m+1),v} = S_{(m+1),v} l_{(m+1),v}(0) R_{m+1} \pmod{q}, \tag{10}$$

$$R_{(m+1)} = l_{(m+1)}(ID_{S_{m+1}}) - l_j(ID_{S_i}). \tag{11}$$

3.6 Function Expansion

Function Expansion is used when the RTC of a region is under the GTC. AKM must perform expansion operation to extend the hierarchy when the RTCs are under or equal to GTC in all the AKM regions. The function ensures that all the RTCs of regions are not lower than GTC when nodes increase continuously. Figure 4 shows that AKM extends region S_i from one level to two levels with the same threshold. It selects t nodes in region S_i, and executes function join to create a new node

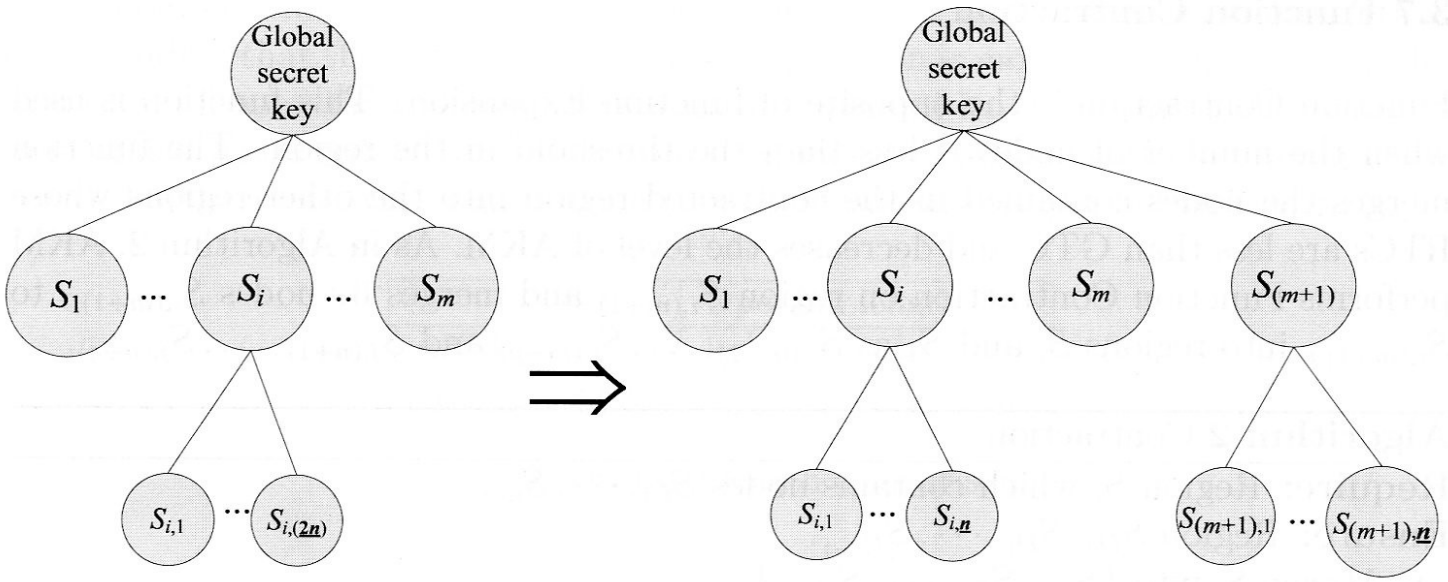


Fig. 3. Function Partition – partition of S_i into S_j and S_{m+1}

$S_{i,(n+1)}$. It then moves $S_{i,1}, \ldots, S_{i,m}$ to be $S_{i,(n+1)}$'s children, $S_{i,(n+1),1}, \ldots, S_{i,(n+1),m}$ with shares $S_{i,(n+1),j}, 1 \leq j \leq m$, that

$$S_{i,(n+1),j} = a(ID_{i,(n+1),j}) = \sum_{r=1}^{t} a_r x^r \pmod{q} \tag{12}$$

where $a_r = r_r | s_r (1 \leq r \leq t)$, $S_{i,(n+1)} = s_t s_{t-1} \ldots s_1$, and all r_rs are the same used in region S_i. Region $S_{i,(n+1)}$ continues (n, t)-threshold as in region S_i.

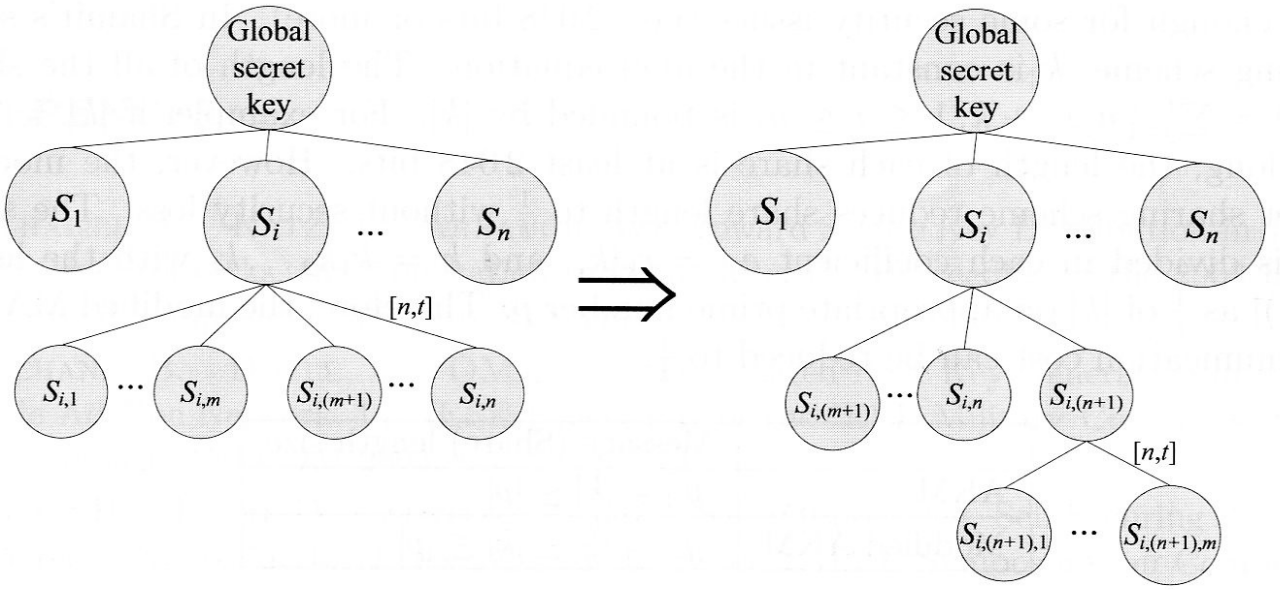


Fig. 4. Function Expansion

3.7 Function Contraction

Function Contraction is the opposite of function Expansion. This function is used when the number of nodes is less than the threshold in the region. The function merges the nodes contained in the contracted region into the other regions whose RTCs are less than GTC and decreases the level of AKM. As in Algorithm 2, AKM performs Function Contraction on region $S_{i,(m+1)}$ and merges its nodes $S_{i,(m+1),1}$ to $S_{i,(m+1),r}$ into regions S_i and S_j as $S_{i,(m+1)}, \ldots, S_{i,(m+p)}$ and $S_{j,(n+1)}, \ldots, S_{j,(n+q)}$.

Algorithm 2 Contraction

Require: Region S_i which contains nodes $S_{i,1}, \cdots, S_{i,r}$.
Ensure: Region $S_{D_0}, S_{D_1}, \cdots, S_{D_{t-1}}$.
1: *Merge* S_i into $\{S_{D_0}, S_{D_1}, \cdots, S_{D_{t-1}}\}$
2: **if** $S_i \notin \{S_{D_0}, S_{D_1}, \cdots, S_{D_{t-1}}$ **then**
3: *Delete* S_i
4: **end if**

The seven-region-based operations on MANET of modified AKM handle key management. The scheme needs a *trusted authority (TA)* to start up, neither any central authorities to compute and distribute shares.

4 PERFORMANCE ANALYSIS

This section discusses the performance improvement of the proposed method in terms of communication cost and computation cost. The modified AKM inherits the AKM structure, and transmissions between each node are (update) shares. Thus, the single message discussion must be transmitted with significant information.

The length of secret key k, protected by the secret sharing scheme, must be long enough for some security issues (i.e., 2 048 bits or more). In Shamir's secret sharing scheme, k is constant in the $a(x)$ equation. The length of all the shares $a(x_i) = \sum_{j=1}^{t-1} a_j x^j + k, 1 \leq i \leq n$, is bounded by $|k|$. For example, if $|k| = 2\,048$ bits long, the length of each share is at least 2 048 bits. However, the modified secret sharing scheme reduces share length to $\frac{1}{t}$ without security loss. The secret key is divided in each coefficient $a_j = r_k|k_j$, and $k = k_1 k_2 \ldots k_t$ with the length $|a(x_i)|$ as $\frac{1}{t}$ of $|k|$ on appropriate prime number p. Therefore, the modified MANET communication cost can be reduced to $\frac{1}{t}$.

	Message (Share) length size
AKM	$\lvert y_i\rvert = \lvert k\rvert \leq \lvert p\rvert$
Modified AKM	$\lvert y_i\rvert = \frac{\lvert k\rvert}{t} \leq \lvert k\rvert \leq \lvert p\rvert$

Table 1. Message length comparison

Computation cost on the MANET environment is a very important issue. Certain mobile ad-hoc devices have restricted power, and cannot support jobs requiring

heavy computation cost. The proposed improvement also influences computation cost. Finding that the critical mathematical operation is module multiplication (/division) in all operations is easy, depending on operand length. Almost all operands in modified AKM reduce, resulting from each modified AKM share as $\frac{1}{t}$ faster than AKM. Furthermore, the computation cost of all operations can be reduced to $\frac{1}{t}$.

	Operand length size
AKM	$\lvert y_i\rvert = \lvert k\rvert \leq \lvert p\rvert$
Modified AKM	$\lvert y_i\rvert = \frac{\lvert k\rvert}{t} \leq \lvert k\rvert \leq \lvert p\rvert$

Table 2. Operand length comparison

5 CONCLUSION

The security of mobile ad hoc networks influences their applications. To achieve the sufficient security, autonomous key management for large number nodes is an important issue. This paper proposes the modified AKM to reduce the communication cost and computation cost to $\frac{1}{t}$ of the original cost without security loss. Results show that the modified AKM is more practical because it can handle huge numbers of dynamic nodes in a MANET while meeting sufficient security requirements. The proposed methodology can be applied to all the schemes based on a cryptographic threshold scheme to truncate message size without endangering security.

Further research on this topic will attempt to simplify the computation complexity of some AKM operations for the workability of ad hoc devices. Furthermore, we will apply the proposed concept on to a vehicular ad hoc network (VANET) because the environment of VANET is more dynamic and the topology changes faster, resulting in narrower bandwith.

REFERENCES

[1] Khalili, A.—Katz, J.—Arbaugh, W.: Toward Secure Key Distribution in Truly Ad Hoc Networks. In: Applications and the Internet, Proceeding of the 2003 International Symposium, 2003, pp. 342–346.

[2] Lehane, B.—Doyle, L.—OMahony, D.: Shared RSA Key Generation in a Mobile Ad Hoc Network. In: IEEE Proceeding of the IEEE Military Communications Conference, MILCOM 2003, Vol. 2, pp. 814–819.

[3] Luo, H.—Kong, J.—Zerfos, P.—Lu, S.—Zhang, L.: Self-Securing Ad Hoc Wireless Networks. In: Proceeding of the Seventh IEEE Symposium on Computers and Communications, ISCC02, 2002, pp. 567–574.

[4] 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, Vol. 12, 2004, No. 6, pp. 1049–1063.

[5] Kong, J.—Zerfos, P.—Luo, H.—Lu, S.—Zhang, L.: Providing Robust and Ubiquitous Security Support for Mobile Ad Hoc Networks. In: Network Protocols, Proceeding of the IEEE 9th International Conference, ICNP01, 2001, pp. 251–260.

[6] Zhou, L.—Haas, Z. J.: Securing Ad Hoc Networks. IEEE Network on Network Security, Vol. 13, 1999, No. 6, pp. 24–30.

[7] Capkuny, S.—Buttyán, L.—Hubaux, J. P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. Technical Report 2002/34, EPFL/IC, 2002.

[8] Omar, M.—Challal, Y.—Bouabdallah, A.: Reliable and Fully Distributed Trust Model for Mobile Ad Hoc Networks. Computers & Security, Vol. 28, 2009, pp. 199–214.

[9] Park, Y.—Park, Y.—Moon, S.: ID-based Private Key Update Protocol with Anonymity for Mobile Ad-Hoc Networks. In: Proceedings of 2010 International Conference of Computational Science and its Applications, 2010.

[10] Hamouid, K.—Adi, K.: Secure and Robust Threshold Key Management (SRKM) Scheme for Ad Hoc Networks. Security and communication networks, Vol. 3, 2010, pp. 517–534.

[11] Li, L.—Liu, R. S.: Securing Cluster-Based Ad Hoc Networks with Distributed Authorities. IEEE Transactions on Wireless Communications, Vol. 9, 2010, No. 10, pp. 3072–3081.

[12] Saravanan, D.—Rajalakshmi, D.—Maheswari, D.: DYCRASEN: A Dynamic Cryptographic Asymmetric Key Management for Sensor Network using Hash Function. International Journal of Computer Applications, Vol. 18, 2011, No. 8, pp. 1–3.

[13] Yang, H.—Luo, H.—Ye, F.—Lu, S.—Zhang, L.: Security in Mobile Ad Hoc Networks Challenges and Solutions. IEEE Wireless Communications, Vol. 11, 2004, No. 1, pp. 38–47.

[14] Zhu, B.—Bao, F.—Deng, R. H.—Kankanhalli, M. S.—Wang, G.: Efficient and Robust Key Management for Large Mobile Ad Hoc Networks. Computer networks, Vol. 48, 2005, pp. 657–682.

[15] Shamir, A.: How to Share a Secret. Communications of the ACM, Vol. 22, 1979, No. 11, pp. 612–613.

[16] Desmedt, Y.: Threshold Cryptography. European Transactions on Telecommunications, Vol. 5, 1944, No. 4, pp. 449–457.

[17] Desmedt, Y.—Frankel, Y.: Threshold Cryptosystems. In: Proceedings of Advances in Cryptology – CRYPTO '89, LNCS 0435, 1990, pp. 307–315.

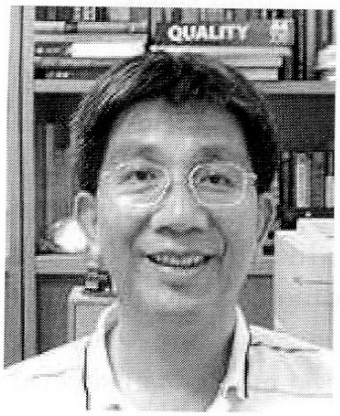
[18] Gennaro, R.—Jarecki, S.—Krawczyk, H.—Rabin, T.: Secure Distributed Key Generation for Discrete-Log Based Cryptosystem. In: Theory and Application of Cryptographic Techniques, Proceedings of Eurocrypt '99, LNCS 1592, 1999, pp. 295–310.

[19] Tsaur, W. J.—Pai, H. T.: Dynamic Key Management Schemes for Secure Group Communication Based on Hierarchical Clustering in Mobile Ad Hoc Networks. In: Frontiers of High Performance Computing and Networking, in: Proceedings of International Workshops, ISPA 2007 Workshops, LNCS 4743, 2007, pp. 475–484.

[20] Lauter, K.: The Advantages of Elliptic Curve Cryptography for Wireless Security. IEEE Wireless Communications, Vol. 11, 2004, No. 1, pp. 62–67.

[21] GIRAULT, M: Self-Certified Public Keys. In: Theory and Application of Cryptographic Techniques, Proceedings of Eurocrypt '91, LNCS 547, 1991, pp. 490–497.

[22] LIN, C. H.—LEE, C. Y.: Modified Autonomous Key Management Scheme with Reduced Communication/Computation Costs in MANET. In Intelligent, Mobile and Internet Services in Ubiquitous Computing. In: Proceedings of the 4th International Workshop IMIS 2010, Krakow 2010, pp. 818–821.



Chu-Hsing LIN received his Ph. D. degree in Computer Sciences from National Tsing Hua University, Taiwan, 1991. Since then, he has been a faculty member at the Computer Science Department of Tunghai University. He was the Director of the Computer Center from 1995 to 1999, and the Chair of the CS Department from 2004 to 2007. He has also been one of the Board Directors of the Chinese Information Security Association (CCISA) from 2001 till now. He has published over 50 papers in academic journals and international conferences. He has received over twenty project grants from government departments and private companies in recent years. In 2006 and 2008, he was awarded the Outstanding Instructor Award of Master and Ph.D Thesis, repectively, by the IICM (Institute of Information and Computing Machinery). His current research interests include multimedia information security, wireless ad hoc networks, and embedded systems applications.

Chen-Yu LEE received his M. Sc. degree in Computer Science and Information Engineering from Tunghai University, Taiwan in 2000. He is currently working towards his Ph. D. degree in Computer Science from National Chiao Tung University, Taiwan. His research interests include cryptography, information security, and DRM systems.

Deng-Jyi Chen received a B. Sc. degree in Computer Science from Missouri State University (cape Girardeau), U.S.A., and M. Sc. and Ph. D. degree in Computer Science from the University of Texas (Arlington), U.S.A. in 1983, 1985, 1988, respectively. He is now a Professor in the Computer Science and Information Engineering Department of National Chiao Tung University, Hsinchu, Taiwan. Before joining the faculty of National Chiao Tung University, he was with National Cheng Kung University, Tainan, Taiwan. He has published more than 130 related papers in the area of software engineering (software reuse, object-oriented systems, visual requirement representation), multimedia application systems (visual authoring tools), e-learning and e-testing systems, performance and reliability modeling, and evaluation of distributed systems, computer networks. Some of his research results have been transferred to industrial sectors and used in product design. He has been a chief project leader of more than 10 commercial products. Some of these products are widely used around the world. He has received both research awards and teaching awards from various organizations in Taiwan and serves as a committee member in several academic and industrial organizations.