Jianting NING[1], Xinchun YIN[2]

College of Information Engineering, Yangzhou University, Yangzhou Jiangsu , China (1),
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing Jiangsu, China (2)

A Dynamic Self-healing Key Management Scheme For Wireless Sensor Networks Based on EBS

***Abstract**. A self-healing mechanism in key management is an important means for large-scale clustering wireless sensor networks that enable non-revoked nodes use their private information and the received broadcast messages to recover the lost session keys on their own. In this paper, we propose a dynamic self-healing key management scheme for large-scale clustering wireless sensor networks that is based on Exclusion Basis System (EBS). We use forward and backward key chains to form cluster session key chain for self-healing, take t-degree polynomial keys to replace the original keys used in EBS. The analysis shows that the proposed scheme has the properties of forward and backward secrecy and resisting to a collusion attack, which is suitable for resource-constrained wireless sensor networks*

***Streszczenie.** W artykule zaproponowano schemat samo-naprawialnego dynamicznego zarządzania kluczami w sieci czujników bezprzewodowych o dużej skali klasterów, oparty na metodzie wykluczeń EBS. Dzięki wykorzystaniu łańcuchów klucza w przód i wstecznych uzyskano samo-naprawialność, natomiast oryginalne klucze w EBS zastąpiono kluczami wielomianowymi stopnia t. Analiza działania wykazała skuteczność opisanego schematu w obronie przed atakami. (**Schemat samo-naprawialnego dynamicznego zarządzania kluczami, oparty na metodzie wykluczeń EBS, w zastosowaniu do sieci czujników bezprzewodowych**).*

Keywords: Wireless Sensor Networks; dynamic key management; self-healing; Exclusion Basis System (EBS);.
Słowa kluczowe: sieć czujników bezprzewodowych, dynamiczne zarządzanie kluczem, samo-naprawialność, metoda wykluczeń EBS.

Introduction

Recently, as one of the core technologies of the Internet of Things, wireless sensor networks (WSNs) is attracting more and more research interests because of its wide applications such as military operations, scientific explorations and so on. Among all security issues in WSNs, key management is a fundamental security issue for wireless sensor networks. Staddon et al. first proposed self-healing key distribution schemes with revocation capability in WSNs in 2002 [1]. Blundo et al. analyzed Staddon's schemes and showed that an adversary could though just broadcast messages to recover the group session key which proved that [1] is not safe [2]. Later on many self-healing key distribution schemes [3-9] based on [1] are proposed, Liu et al. in 2003 proposed a novel method by combining the personal secret key distribution scheme with the self-healing technique to improve the scheme in [1]. Dutta et al. proposed a self-healing group key distribution scheme based on one-way key chain [6].

Background

Randomly select key seeds KF_0, KB_0, one-way hash function $H(\cdot)$, $H_B(\cdot)$, d numbers $\delta_1,\delta_2,...,\delta_d$. Then we can get the corresponding forward key chain $\{KF_1,KF_2,...,KF_d\}$ for d sessions through $KF_j = H(KF_{j-1},\delta_j) = ... = H^{j-2}(KF_1,\delta_2) = H^{j-1}(KF_0,\delta_1)$ ($1 \le j \le d$). For session 1, the forward key is $KF_1 = H(KF_0,\delta_1)$, the backward key seed is $KB_1^0 = H(KB_0,\delta_1)$, and the backward key chain is $\{KB_1^0\}$. For session 2, the forward key is $KF_2 = H(KF_1,\delta_2)$, the backward key seed is $KB_2^0 = H(KB_1,\delta_2)$, and the backward key chain is $\{KB_2^0,KB_2^1\}$. For session j ($1 \le j \le d$), the forward key is $KF_j = H(KF_{j-1},\delta_j)$, the backward key seed is $KB_j^0 = H(KB_{j-1},\delta_j)$, then the backward key chain is $\{KB_j^0,KB_j^1,...,KB_j^{j-1}\}$, $j=1,2,...,d$, where $KB_j = KB_j^{j-1} = H_B(KB_j^{j-2}) = ... = H_B^{j-1}(KB_j^0)$. Let n, k and m be positive integers, such that k>1,n>m. An Exclusion Basis System of dimension (n, k, m), denoted by EBS(n,k,m), is a collection Γ of subsets of $[1,n] = \{1,2,...,\mathrm{n}\}$ such that for every integer $t \in [1,n]$ the following two properties hold: (1) t is in at most k subsets of Γ, and (2) There are exactly m subsets, say A1, A2,…, Am, in Γ such that $\bigcup_{i=1}^{m} A_i = \{1,2,\ldots,\mathrm{n}\} - \{t\}$. (That is, each element t is excluded by a union of exactly m subsets in Γ .) We take the EBS (n, k, m) described above as a wireless sensor network dynamic key management method, n is the number of nodes, k is the number of administrative keys and m is the number of rekeying messages. A set of (k+m) administrative keys is used to support a set of n nodes, and each node is assigned a distinct combination of k keys. A node can be simply admitted to the group by assigning one of the unused set of k keys out of the total of C(k+m,k), i.e., $(\mathrm{k}+\mathrm{m})!/(k!m!)$, distinct combinations. At system initialization, the system select key seeds KF_0, KB_0, on-way hash functions $H(\cdot)$, $H_B(\cdot)$ and $H_1(\cdot)$, and d numbers $\delta_1,\delta_2,...\delta_d$ randomly. Then system randomly chooses numbers $\alpha_1,\alpha_2,...,\alpha_d \in F_q$ for each session, then generate the forward key chain $\{KF_1,KF_2,...,KF_d\}$ and the corresponding d backward key chains $\{KB_j^0,KB_j^1,...,KB_j^{j-1}\}$ ($1 \le j \le d$) as described above. After strict registration and authentication, each sensor node N_a within cluster will be assigned its unique Id_a , forward hash function $H(\cdot)$ and one-way hash function $H_1(\cdot)$ which used for updating. And each node will be allocated a key-buffer of length L (kb[L],…,kb[1]), and two key-slots. Cluster head node chooses two polynomials of degree t at random $f(x) = \sum_0^t f_i x^i$ and $s(x) = \sum_0^t s_i x_i$, which is based on $(t+1,n)$ threshold secret sharing technique[16]. Then cluster head node randomly generates a number of k+m polynomial administrative keys $f_n(x) = \sum_{i=0}^t f_i x^i$, $n = 1,2,...$ $,k+m$, and distributes unused set of k polynomial keys out of the total of C(k+m,k) to each sensor node within cluster

according to a pre-generated random EBS matrix. Meanwhile, the cluster head node will broadcast $B_{N_a} = \{L \mid KF_0 \mid KB_{d-L-1} \mid s_d(a) \mid ... \mid s_{L+3}(a) \mid \delta_1 \mid ... \mid \delta_L\}$ .Sensor node N_a receive and decrypt B_{N_a} to get L, KF_0 , KB_{d-L-1} , $S_d(a)$, ..., $S_{L+3}(a)$, $\delta_1, \delta_2, ..., \delta_L$. Then N_a calculates $CK_j = KF_j + KB_{d-j+1}$ ($1 \le j \le L+2$), stores $\{CK_{L+2}, ..., CK_3\}$ and $\{CK_2, CK_1\}$ in the key-buffer and key-slots, respectively. CK_1 is used for the present cluster session key, Assume that $N = \{N_1, N_2, ..., N_p\}$ is the set of all active sensor nodes for j-th session, where p is the number of active user in session j. Let $T = \{t_1, t_2, ..., t_p\}$ be the set of all active users'secret values in j-th session. In session j, cluster head node generates a masking key sequence $\{G_j^1, G_j^2, ..., G_j^{j-1}, G_j^j\}$ where $G_j^i = KB_{d-j+1}^i \oplus \alpha_i$ ($j = 1,2,..., d;\ i = 1,2,..., j$), then broadcasts the following message:
$B_j = \{Z_j^i(x) = A_j^i(x) G_j^i + s_i(x)\} \cup \{E_{KB_{d-j+1}^0}(\delta_1)$, $E_{KB_{d-j+1}^1}(\delta_2)$, ..., $E_{KB_{d-j+1}^{j-1}}(\delta_j)\}$, $i = 1,2,..., j$, where S_j^i is a randomly number to mask G_j^i , $S_j^i / T_j \notin T = \{t_1, t_2, ..., t_p\}$, and $A_j^i(x) = 1 - (S_j^i x - T_j) \prod_{n=1}^{p} x - t_n$. When $N_i \subset N = \{N_1, N_2, ..., N_p\}$ receives the *j*-th broadcast message B_j , N_i can evaluate $A_j^i(x) = 1$ by using its secret value t_i . For any revoked user, however, the $A_j^i(x)$ is a random value. Assume Suppose that sensor node N_a joins in the cluster in session *i*, and not revoked in session *j* ($1 \le i \le j$), then it can recover the cluster session key CK_j from the broadcast message B_j as follows: 1) N_a computes $G_j^i = Z_j^i(t_d) - s_i(t_d)$ where $A_j^i(t_d) = 1$; 2) N_a evaluates $KB_{d-j+1}^i = G_j^i \oplus \alpha_i$; 3) N_a computes all the future $\{KB_{d-j+1}^i, KB_{d-j+1}^{i+1}, ..., KB_{d-j+1}^{j-1}\}$ through the one-way hash function $H(\cdot)$, then get $KB_{d-j+1} = KB_{d-j+1}^{j-1} = H^{j-i}(KB_{d-j+1}^{i-1})$. Meanwhile, N_a calculates the forward key $KF_j = H^{j-1}(KF_0)$ by using the preloaded key seed KF_0 and one-way hash function $H(\cdot)$. Thus, gets the cluster session key $CK_j = KF_j + KB_{d-j+1}$ for j-th session; 4) N_a can decrypt $\{E_{KB_{d-j+1}^{i-1}}(\delta_i), E_{KB_{d-j+1}^{i}}(\delta_{i+1})$,..., $E_{KB_{d-j+1}^{j-1}}(\delta_j)\}$ by using the corresponding keys $\{KB_{d-j+1}^{i}, KB_{d-j+1}^{i+1}, ..., KB_{d-j+1}^{j-1}\}$, thus getting the corresponding self-healing keys $\{\delta_i, \delta_{i+1}, ..., \delta_j\}$. If N_a has already obtained $KB_{d-j'+1}$ from $B_{j'}$, he can recover all the session keys KB_{d-l+1} ($j' < l < j$) with $KB_{d-j'+1}$ and the self-healing keys $\{\delta_i, \delta_{i+1}, ..., \delta_j\}$ .The cluster head node broadcasts update packets $B_j = \{f'(x)\} \cup \{Z_j(x) = A_j(x) G_j + s'(x)\} \cup \{E_{KB_{d-j+1}^0}(\delta_1)$, $E_{KB_{d-j+1}^1}(\delta_2), ..., E_{KB_{d-j+1}^{j-1}}(\delta_j)\}$, where $f'(x) = \sum_{i=0}^{t} f_i' x^i$, $s'(x) = \sum_{i=0}^{t} s_i' x^i$, $f_i' = H_1(f_i)$, $s_i' = H_1(s_i)$. Each node receives and decrypts the packet, calculates $f'(x)$ to replace the original $f(x)$, thus completing the administrative key update. Also, each node can decrypt the packet and calculate to get $CK_j = KF_j + KB_{d-j+1}$ like described above. And CK_j will be put in the key-buffer and switches the active-key. According to EBS, we need to update all k administrative keys $E_i^{p_j}$, $j = 1,2,\ldots,k$, $p_j \in \{1,2,\ldots,k+m\}$ that N_i owns. So the cluster head node broadcasts m data packets $E_i^{q_S}(E_i^{p_1}(f_{p_1}'), E_i^{p_2}(f_{p_2}')), \ldots, E_i^{p_k}(f_{p_k}')$, $s = 1,2,\ldots,m$, $\{q_1, q_2, \ldots\ldots, q_m\} \in \{1,2,\ldots\ldots,k+m\} - \{p_1, p_2, \ldots\ldots, p_k\}$ within the cluster, f_{p_i}' is the new administrative key, that is generated by using the one-way hash function $H_1(\cdot)$. We build our quantitative analysis of the proposed scheme's performance according to steady-state distributions of 2-dimensional Markov chain in [17]. Let p_L = Pr{ B_j is lost}, p_F = Pr{ B_j authentication fails | B_j is received} and p_S = 1- p_L - p_F . Also, let $p(i, j)$ denote the steady-state probability of state (i, j) , and $p_\varepsilon(k)$ the probability that there were exactly k empty slots. Then, we can get $p_\varepsilon(k) = (p_L + p_F)^k \cdot p(0,0)$, $k = 0,...,d$, $p(0,0) = \frac{1 - (p_L + p_F)}{1 - (p_L + p_F)^{d+1}}$. Assume that $E(N^H)$ is the expected number of hash computations per updating, when there are k empty slots in key-buffer, we have $E(N^H) = \sum_{k=0}^{d-1} (k+1)(1 - p_L)(p_L + p_F)^k p(0,0) + (d+1) \cdot (p_L + p_F)^d \cdot p(0,0)$. The overhead of communication between the cluster head node and the sensor nodes is mainly about two parts: the cost of init key C_{init} and the cost of update key C_{update} . We take n is the ratio of the communication cost of init key C_{init} to that of update key C_{update} , so we can get the expected communication cost $E_{comm} = n \cdot [1/d + (p_L + \quad p_F)^d \cdot p(0,0)] + \quad \sum_{t=0}^{d-1} (p_L + p_F)^t p(0,0)$.

Suppose that R_j is the set of the nodes revoked in and before session j. For the broadcast message $B_j = \{Z_j^i(x) = A_j^i(x) G_j^i + s_i(x)\} \cup \{E_{KB_{d-j+1}^0}(\delta_1), E_{KB_{d-j+1}^1}, \cdots, E_{KB_{d-j+1}^{j-1}}(\delta_j)\}$ as mentioned above, and a node needs to obtain the corresponding self-healing key δ_j which is randomly chosen and an active node's secret to get the *j*-th cluster session key $CK_j = KF_j + KB_{d-j+1}$. The corresponding self-healing key δ_j is encrypted by the corresponding backward session key KB_{d-j+1}^{j-1}, which $KB_{d-j+1}^{j-1} = G_j^{j-1} \oplus \alpha_{j-1}$. But for any revoked node $N_R \in R_j$, cannot calculate the masking keys G_j^{j-1}, because $A_j^i(x)$ is a random value for N_R . Moreover, for any revoked node $N_R \in R_j$ do not have α_{j-1} . Therefore, the nodes in R_j cannot get the values of the self-healing keys to obtain the future group session keys. Meanwhile, $f(x)$, $S(x)$ are t-degree polynomial based on $(t+1,n)$ threshold secret sharing technique which need *t + 1* points to recover. The above analysis shows that the proposed scheme is forward secure. Suppose that U_{j+1} is the set of the nodes which join in session *j + 1*. For users in U_{j+1} can only get the current backward session key KB_{d-j} to compute CK_{j+1} which is the last key in the cluster key chain from the broadcast B_{j+1}. Thus users in U_{j+1} can only get the current self-healing key δ_{j+1}. Since one-way hash function is irreversible, it is computationally infeasible for any user in U_{j+1} use KB_{d-j} and δ_{j+1} to compute the previous cluster key CK_j. Meanwhile, $f(x)$, $S(x)$ are t-degree polynomial based on $(t+1,n)$ threshold secret sharing technique which need *t + 1* points on the polynomial $f(x)$ to recover. $f^{'}(x)$ is the result of $f(x)$ after one-way hash operation, since one-way hash function is irreversible, so it is hard to recover the previous polynomial $f(x)$. So nodes in U_{j+1} can not get any information about the previous cluster session key or administrative key. The above analysis shows that the proposed scheme is backward secure. Suppose that R_j is the set of the nodes revoked in and before session j_1+1, and U_{j_2} is the set of the nodes who join from session j_2. Assume that $N_R \in R_j$ colludes with $N_{j_2} \in U_{j_2}$, they need the self-healing keys between δ_{j_1} and δ_{j_2} to recover $KB_{d-j+1} = KB_{d-j+1}^{j-1}$ which could be used to compute CK_j ($j_1 < j < j_2$). For the equation $Z_{j_2}^{j_1}(x) = A_{j_2}^{j_1}(x) G_{j_2}^{j_1} + s_{j_1}(x)$ from B_{j_2}, U_{j_2} needs the value of $s_{j_1}(x)$ to obtain $G_{j_2}^{j_1}$. And N_{j_2} can get $K_{d-j_2+1}^{j_1-1}$ with N_R 's secret α_{j_1}. Thus, N_R and N_{j_2} can get $\{KB_{d-j_2+1}^{j_1-1},...,KB_{d-j_2+1}^{j_2-1}\}$ and $\{\delta_{j_1},...,\delta_{j_2}\}$. But they cannot obtain $s_{j_1}(x)$ unless they can recover the secret polynomial $s(x)$ which based on $(t+1,n)$ threshold secret sharing technique. Therefore, they cannot recover the self-healing keys between δ_{j_1} and δ_{j_2}. Thus they cannot get the backward session keys $KB_{d-j+1} = KB_{d-j+1}^{j-1}$ to calculate CK_j without the corresponding self-healing keys. Therefore, the proposed scheme can resist to the collusion attack. Fig.1 gives the relationship between the number of captured nodes and the fraction of keys compromised.

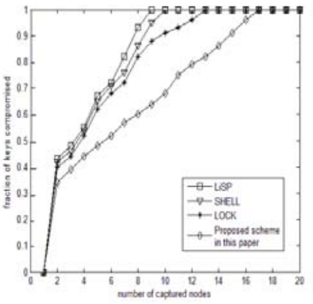

Fig. 1. Relationship between the number of captured nodes and the fraction of keys compromised(m=5)

Conclusion

We propose a dynamic key management scheme for wireless sensor networks with the property of self-healing. We take t-degree polynomial keys to replace the original keys used in EBS, use forward and backward key chains and broadcast polynomial key to achieve self-healing, forward and backward secrecy and resisting to a collusion attack. Meanwhile, this scheme has a small calculation and communication overhead, which is efficient and secure for resource-constrained wireless sensor networks.

Acknowledgments

The authors wish to thank the helpful comments and suggestions from my teachers and colleagues in Yangzhou University. This work was supported in part by National High Technology Research and Development Program of China (863 Program) (2007AA0124487), National Natural Science Foundation of China (60473012), and Colleges and Universities of Jiangsu Province Plans to Graduate Research and Innovation (CXLX11_1008).

REFERENCES

[1] Staddon, J., Miner, S., Franklin, M., Balfanz, D., Malkin, M., Dean, D, Self-healing Key Distribution with Revocation. In: Proc. of IEEE Symposium on Security and Privacy, pp. 241–257 (2002)

[2] Blundo, C., D'Arco, P., Listo, M., A Flaw in a Self-Healing Key Distribution Scheme. In: Proc. of Information Theory Workshop, Paris, pp. 163–166 (2003)

[3] Liu, D., Ning, P., Sun, K., Efficient Self-healing Key Distribution with Revocation Capability. In: Proc. of the 10th ACM CCS 2003, pp. 27–31 (2003)

[4] Saez, G.: On Threshold Self-healing Key Distribution Schemes, In: Smart, N.P.(ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 340–354. Springer,Heidelberg (2005)

[5] Zou, X.K., Dai, Y.S., A Robust and Stateless Self-Healing Group Key Management Scheme. In: International Conference on Communication Technology, ICCT 2006,vol. 28, pp. 455–459 (2006)

[6] Dutta, R., Chang, E., Mukhopadhyay, S., Efficient Self-healing Key Distribution with Revocation for Wireless Sensor Networks Using One Way Hash Chains. In:Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 385–400. Springer,Heidelberg (2007)

***Authors**: Jianting Ning, master student at Yangzhou University, Jiangsu , China, mainly research interests include Information Security, cryptography and security policies in wireless sensor networks., E-mail: jelly408385909@163.coml; Xinchun Yin, Ph.D., professor, doctoral supervisor at Yangzhou University, Jiangsu , China, mainly research interests include Information Security, High Performance Computing and Software Quality Assurance., E-mail: xcyin@yzu.edu.cn l.*