An Efficient CPK-based Key Management Scheme in Wireless Sensor Networks

Jun ZHENG[1, a], Zhifu CHEN[1, b], Jinrong XUE[2, c] and Fan WU[1]

[1]Beijing Institute of Technology, Beijing, 100081, China

[2]National Computer System Engineering Research Institute of China, Beijing, 102209, China

[a]zhengjunbit@gmail.com, [b]chenzhifu2010@163.com, [c]xuejinrong_zdls@sina.com

Keywords: Elliptic Curve, CPK, Wireless Sensor Networks, Key Management, Collusion Attack.

Abstract. In recent years, the rapid development and wide usage of wireless sensor networks, makes it a research hotspot at home and abroad. Wireless sensor network is an open network, the sensor nodes can easily be captured or tampered. Ensuring network security is a prerequisite for communication. So, establishing session key between adjacent nodes is critical in key management scheme for wireless sensor networks. In order to save cost and control node size, nodes in sensor networks typically have characteristics of weak storage capacity, low communication bandwidth and limited computing ability. Therefore, ordinary key management mechanism cannot be applied to the sensor networks directly. In consideration of these features of sensor nodes, a CPK-based key management scheme for wireless sensor networks is proposed in this paper based on the complexity of solving elliptic curve discrete logarithm problem. Analysis shows that our scheme performs better than some existing scheme in security, storage efficiency, calculation cost and other aspects, so it can be applied to large-scale wireless sensor networks.

Introduction

Wireless Sensor Networks (WSN) is a distributed wireless network, including a large number of sensor nodes which monitor environmental data and send the collected data to the Management Center (MC) wirelessly. It has been identified as one of the most important research and widely used at present in transportation, environmental quality testing, health care, military battlefield, etc. WSN has the characteristics of large-scale deployment, vulnerable to be captured, node mobility, dynamic expansion, and so on. Security in WSN is crucial as it’s vulnerable to different types of deliberate attacks, such as the node could be physically destroyed, caught and tampered and so on. So, research on key management scheme in WSN is extremely important. But it’s not easy to design a perfect key management scheme in WSN because of the limited resources of each node, such as energy, memory, computing speed and bandwidth [1].

At present, many scholars have proposed many key management schemes for WSN, such as the classic E-G scheme based on random key pre-distribution which was presented by Eeschnaure and Gligor in 2002 [2]. In their scheme, a key ring consist of n different keys selected from a large key pool randomly, is assigned to each node. If the adjacent nodes have a same key, they can communicate each other, the scheme has the weakness of uncertain network connectivity and large storage cost. Chan and Perrig proposed the q-composite scheme [3] based on E-G scheme in 2003. In their scenario, two nodes must have q common keys, instead of only one to establish a secure connection. They improved the resiliency to node capture this way. Reference [4] proposed a new key distribution scheme on the basis of the combination of IBE (Identity Based Encryption) and Diffie-Hellman key-exchange protocol in 2007. Reference [5] proposed a hybrid design key management scheme based on the idea of combining design and identity based cryptography in 2012, which can guarantee the security of the network, but the storage cost is not ideal. In addition, many other scholars at home and abroad presented ECC-based scheme, tree-based scheme, binary t-order polynomial based scheme and other key management schemes [6,8], but they have not formed a perfect solution in all aspects yet. Thus, protocol ensuring security and saving resources at the same time, is the direction of research in WSN’s key management [9].

To solve these problems mentioned above, we propose a CPK-based efficient key management scheme in WSN. The rest of this paper is organized as follows: In Section 2 we introduce the principle of CPK in elliptic curve. In Section 3 we describe Our Scheme in detail. We analyze the safety and performance of our scheme compared with other solutions in Section 4. Finally, we summarize the achievements in Section 5.

CPK in Elliptic Curve

CPK (Combined Public Key Crypto-system) [10] is proposed by Professor Xianghao Nan, our expert in the field of information security. The design concept of CPK is that the corresponding elements in the seed matrix are combined to produce the user key according to the row and column coordinates generated by the hash value of user identifier. A large number of user keys can be generated this way when the seed matrix reaches an appropriate size.

Let $E(F_{\mathrm{p}}):(y^2 = x^3 + ax + b) \bmod q$ be an elliptic curve over a finite field F_{p}, where $4a^3 + 27b^2 \neq 0$. Point $G(x, y)$ in the elliptic curve is selected as the origin, q is the order of the Abel group formed by the generator G. CPK in elliptic curve is based on Elliptic curve discrete logarithm problem (ECDLP) which refers to it is very difficult find an integer d satisfy dG=Q given another point Q and generator G in the elliptic curve. Select numbers from large prime field randomly to be the elements of the private seed matrix $\mathrm{Seed}_{sk}(m \times n)$. We obtain the public seed matrix $\mathrm{Seed}_{pk}(m \times n)$ when the private seed matrix element mapped to a point in the elliptic curve. $pk_{ij} = sk_{ij}G$ and sk_{ij} form a pair of public and private key.

$$Seed_{sk} = \begin{pmatrix} sk_{11} & sk_{12} & \cdots & sk_{1n} \\ sk_{21} & sk_{22} & \cdots & sk_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ sk_{m1} & sk_{m2} & \cdots & sk_{mn} \end{pmatrix} \quad Seed_{pk} = \begin{pmatrix} sk_{11}G & sk_{12}G & \cdots & sk_{1n}G \\ sk_{21}G & sk_{22}G & \cdots & sk_{2n}G \\ \vdots & \vdots & \ddots & \vdots \\ sk_{m1}G & sk_{m2}G & \cdots & sk_{mn}G \end{pmatrix}$$

The combination of any number of public and private key pair can constitute a new key pair. According to this principle, if $a_1, a_2, \cdots, a_m$ and $b_1, b_2, \cdots, b_n$ are the row-coordinates and column-coordinates generated by the entity identifier and the mapping function, the entity's key pair is obtained in the following ways:

$$SK = (sk_{(a1,b1)} + sk_{(a2,b2)} + \cdots + sk_{(am,bn)}) \bmod q \qquad PK = sk_{(a1,b1)}G + sk_{(a2,b2)}G + \cdots + sk_{(am,bn)}G = SK \cdot G \tag{1}$$

The Proposed Scheme

We use a distributed network architecture in our scheme due to its advantages of short communication path and less forwarding times. Sensor nodes in the hierarchical WSN are classified into three: base station (BS), cluster head nodes (CH) and ordinary sensor nodes (SN). CH node with more adequate resource is responsible for data collection and the key management in the cluster, and it’s more difficult to be captured. The BS with plenty of energy and powerful computing capability which is responsible for the network initialization and communicate with both MC (Management Center) and CH nodes is always credible. The network structure is shown in the Fig.1.

Parameters including the elliptic curve $E(F_p):(y^2 = x^3 + ax + b) \bmod q$, node identifier, hash functions $H_{1 \cdots n} : ID_{N_i} \to \{0,1\}^c$, and seed matrix are generated by OKS (Offline Key Server) before deployment. Function $E(k) = (k \bmod q) \cdot G$ shows a mapping from a prime number k to a point in the elliptic curve. Elements $\{s_u, s_{u+1}, \cdots, s_v\}$, $(1 \leq u \leq v \leq n)$ are selected from the set of large prime numbers less than p $S = \{s_1, s_2, \cdots, s_n\}$ to consist the private seed matrix Seed_{sk} by each CH node. Element $s_j (1 \leq j \leq n)$ selected from the rest of set S is assigned to a CH node CH$_{\mathrm{j}}$ or ordinary node N$_{\text{i}}$ to be their identifier. And then, we can get the private key of a node SK_{N_i} according to elliptic curve's CPK and compute the public identifier of the node $ID_{N_i} = E(SK_{N_i}) = (SK_{N_i} \bmod q)\cdot G$.

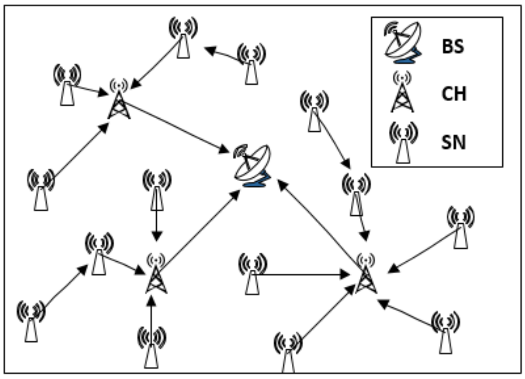

Fig.1 Network architecture

Key Negotiation between Base Station and Cluster Heads. Due to the relatively abundant resources and strong processing capacity of BS and the CH nodes, we can use asymmetric encryption scheme during the certification and key negotiation. Before the network deployment, node identifier $\mathrm{ID}_{\mathrm{N}_1}$, private key SK_{N1}, public seed matrix Seed_{pk} and other parameters are written into node's memory. A node can calculate another node's public key easily according to its identifier and the matrix. The BS and a CH node establish a session key as follows:

a) CH → BS: $\{\{\{T_{CH} \| p_{CH}\cdot \mathrm{G} \| ID_{CH} \| r\}_{PK_{BS}}\}_{SK_{CH}} \| ID_{CH}\}$ CH node generates a message containing a large prime number p_{CH}, ID_{CH}, timestamp T_{CH}, point $p_{CH}\cdot \mathrm{G}$, and a fresh random number r .Then encrypt the message with PK_{BS}, and sent the message to the BS through the network after signing it.

b) BS → CH: $\{\{\{T_{BS} \| p_{BS}\cdot \mathrm{G} \| ID_{BS} \| (r+1)\}_{PK_{CH}}\}_{SK_{BS}} \| ID_{BS}\}$ BS can parse ID_{CH} from the received message, and obtain the public key of CH PK_{CH} readily according to the formulas of CPK and the seed matrix. After verifying the legality of the CH node, BS decrypt the message using PK_{CH} and SK_{BS} successively to get the crucial p_{CH}. Then BS generates a message similar to the received one containing the fresh number *r+1* not *r*. Then encrypt the message with PK_{BS}, and sent the encrypted and signed message to the CH node.

c) CH → BS: $\{\{\{T_{CH} \| (r+2) \| "ok"\}_{PK_{BS}}\}_{SK_{CH}} \| ID_{CH}\}$ If the CH node can decrypt the data with its private key and the BS's public key, and get the prime number p_{BS} successfully, it means that the BS and the CH node complete the identity authentication. Then, all of them can calculate their shared session key $K_{CH\Leftrightarrow BS} = (pk_{(h_1,1)} + pk_{(h_2,2)} + \cdots + pk_{(vh,n)}) \bmod q$ by calculating $P = ((p_{CH} + p_{BS}) \bmod q)\cdot \mathrm{G}$ and the seed matrix, where $h_1, h_2, \cdots, h_n$ are row indexes of the public key matrix corresponding to *n* mapped value of *P*.

Key Negotiation between Sensor Nodes. We use symmetric key encryption between ordinary nodes because of their weak ability of computing and limited memory. Adjacent nodes in the same cluster can be certified each other according to ID_{Ni} of the other node, and then calculate their shared keys. CH node is responsible for broadcasting their member nodes' ID regularly in public. Suppose node A and B are two common nodes inside cluster CH, A, B establish a shared key with the following procedures:

a) Calculating and Sending MSG: Node A calculate $MSG_A = ((SK_A + ID_{CH}) \bmod q)\cdot G$ using its private key SK_A and send it to node B. And node B send $MSG_B = ((SK_B + ID_{CH}) \bmod q)\cdot G$ to node A in the same way.

b) Verifying: After receiving MSG_B sent by node B and the identifier of B obtained from broadcasting data, node A can verify that whether the equation $ID_B+(ID_{CH} \bmod q)\cdot G{==}\mathrm{MSG_B}$ is correct or not, and node B verify whether $ID_B+(ID_{CH} \bmod q)\cdot G{==}\mathrm{MSG_B}$ is correct at the same time.

c) Getting a shared key: If node A and B are both verified in the step 2, the two nodes can negotiate a shared key. Node A calculate $k_a = E(SK_A) + ID_B + (ID_{CH} \bmod q)\cdot G = E(SK_A) + MSG_B$ and the different hash values of k_a, marked as $H_{1\cdots n}(k_a)$ ($H_1(k_a) = v_1$, $H_2(k_a) = v_2$, $\cdots$, $H_n(k_a) = v_n$). $H_i(k_a) = v_i$ is corresponding to the v_i -th element of the i-th column of the seed matrix, that's to say $pk_{(\mathrm{vi},i)}$. Node A generates a shared key $Key_A = (pk_{(v1,1)} + pk_{(v2,2)} + \cdots + pk_{(vn,n)}) \bmod q$ according to the CPK principle. Similarly, node B can calculate $k_b = E(SK_B) + MSG_A$, and calculate $Key_B = (pk_{(u1,1)} + pk_{(u2,2)} + \cdots + pk_{(un,n)}) \bmod q$ finally. Node A and node B get a final shared key $Key_{AB} = Key_A = Key_B$, due to $k_a = k_b = ((SK_A + SK_B + ID_{CH}) \bmod q)\cdot G$

Processing of New Nodes and Compromised Nodes. The OKS assign a unique identifier which will be broadcasted by the BS and the CH nodes in the whole network to the new node. Other nodes certificate each other by the identifier and the message sent by other node, so as to establish a secure connection. After a node is captured, the corresponding CH node generates a revocation message that contains the encrypted node identifier to the BS, and then BS broadcast the message in the WSN, indicating that the node exited the network. Besides, regular key update at fixed time intervals is required for the security of WSN. During key update process, only a column of the seed matrix need to be updated every time in order to reduce network traffic.

Performance Analysis

Security Analysis. The key generation is based on CPK cryptosystem in elliptic curve, the public seed matrix and node identifier are the only open parameters. If an attacker tries to imitate the legitimate nodes and establish a secure connection with other nodes in the network, he must get the private key seed matrix first, but it's more difficult than solving the elliptic curve discrete logarithm problem which is widely considered to be very difficult at present. So, our scheme has strong security and it's not easy to be attacked.

In our Scheme, timestamp and fresh random number can prevent our scheme from *Replay Attack* during key negotiation. Key update when nodes join or exit the network can ensure the forward and backward security. And our scheme is based on CPK which can resist *Collusion Attack*, it's proved in [11] in detail. If an attacker want to implement *Collusion Attack,* he must capture $m \times n$ nodes. In the actual attack, it's very difficult to capture so many nodes and solve the set of linear equations. Besides, there may have been a rekeying in WSN when those nodes were captured. Therefore, *Collusion Attack* cannot be implemented in our Scheme.

Computational Cost. The process of key generation in our scheme contains calculating map value according to node identifier and calculating the shared key in accordance with the combination of corresponding elements in seed matrix. There are mainly two operations, HASH and point-addition in elliptic curve. But, the computational cost of the two operations is much smaller than bilinear operation and exponential operation. Point addition operation in elliptic curve means that given two points P and Q both in elliptic curve *E,* calculating another point R in *E* and R= P + Q, it's actually just a polynomial operation.

In our scheme, the HASH operation is used to select the matrix elements based on node identifier. Three addition and point-addition operations are corresponding to the three processes as follows: calculating of authentication information, verifying the identity of the other node and calculating the session key. The computational complexity compared with other typical key management schemes is shown in Table 1. Obviously, our scheme has great advantages in terms of computing performance compared with the schemes in [4] and [5].

Table.1 Comparison of computational cost

scheme	Addition	Multiplication	HASH	Bilinear operation	Exponential operation	Addition on elliptic curve
The scheme in [4]	-	1	3	2	1	-
The scheme in [5]	-	2	1	1	-	2
Our scheme	3	0	1	0	0	3

Storage Performance. Our scheme with ECC-based key has the advantage of shorter key length under the premise of ensuring the same security. For example, 160 bit ECC key can provide the same level of security compared with 1024 bit RSA key [12]. Assuming that the total number of nodes in WSN is *k*, and the seed matrix in the scheme is a matrix with *m* rows and *n* columns, each column represents a level of combination, then m^n pair-keys can be generated theoretically [13]. The WSN network is divided into *t* clusters, each cluster contains k/*t* nodes, and each node stores *N* matrix elements. Then, the relationship between *N* and *k* is $\sqrt{N}^{\sqrt{N}} = k/t$, if we take $t = \sqrt{k}$, then $k = N^{\frac{\sqrt{N}}{4}}$, the relation between the number of the matrix elements and the size of the network is shown in Fig.2. As the network size grows, the total amount of storage required grows slowly. That is, the scheme with a small amount of storage space can support large-scale networks.

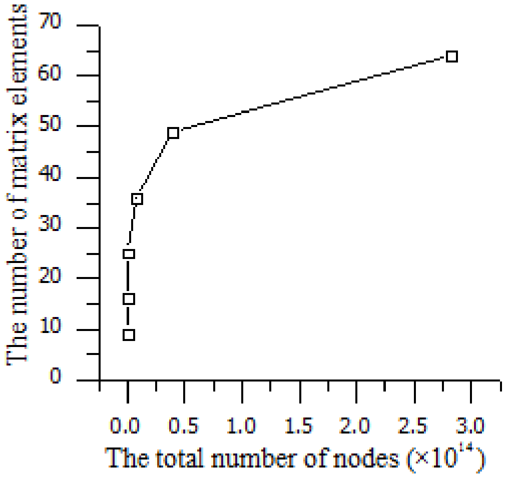

Fig.2 Relation between the number of the matrix elements and the size of the network

In our scheme, each node need to save its identifier and the seed matrix, the storage performance compared with the scheme [4] and the scheme [5] is shown in Table 2. Thus, we can conclude that our scheme has an absolute advantage in storage usage.

Table.2 Comparison of storage performance

scheme	Storage Cost
The scheme in [4]	IBE parameter π and k keys between adjacent nodes
The scheme in [5]	A large number of parameters $\{k(s_{i,j},m)\}$,P,n, $k_{BS-si,sj}$, $E(F_p)$ etc. and $\sqrt{k}$ keys in each node
Our scheme	a seed matrix and a node identifier

Connectivity and Communication Cost. Each pair of nodes in the communication range can authenticate each other, negotiate session key and establish a secure connection in our scheme. So our scheme has great advantage in connectivity compared with the classic E-G scheme [2] which is based on random key pre-distribution. If two nodes want to complete authentication and shared key generation, the only thing they should do is to exchange their identifier and a point coordinates in elliptic curve. So, the communication cost in our scheme is very small.

Summary

In this paper, we proposed an efficient key management scheme based on CPK in WSN, combining the security of asymmetric encryption with the efficiency of symmetric encryption. Analysis shows that our scheme has great advantages in storage performance, computational cost, and communication cost compared with other schemes, so it can be applied to large-scale wireless sensor networks. The further research of this paper is trying to reduce the resource consumption of node and improve our scheme.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China under Grant No.61272511.

References

[1] Junqi Zhang, Vijay Varadharajan. *Wireless sensor network key management survey and taxonomy*. Journal of Network and Computer Applications.2010.33 (2010)63–75

[2] ESCHNAURE L, GLIGOR V D.*A key-management scheme for distributed sensor networks*. Proc of the 9th ACM conference computer and communication security .New York, 2002: 41-47.

[3] Chan H, et al.*Random key predistribution schemes for sensor networks*. IEEE symposium on Research in Security and Privacy. New York: IEEE publishing, 2003 .197-213.

[4] Geng Yang, Jiangtao Wang. *A Key Establish Scheme for WSN Based on IBE and Diffie-Hellman Algorithms*. ACTA ELECTRONICA SINICA, 2007, 35(1):180-184.

[5] Minqing Zhang, Wenhua Fu, Xuguang Wu. *Key management scheme based on composite design and identity encryption for clustered wireless sensor networks*. Journal of Computer Applications, 2012, 05:1392-1396.

[6] CHENG Hong-bing, YANG Geng. *Secrets Shared Scheme for Wireless Sensor Networks Based on Complementary Tree Protocol.* Journal of Nanjing University of Posts and Telecommunications (Natural Science), 2007, 06:32-37.

[7] Jianwei Jiang. *Research on key Management Scheme for WSN based on ECC*. Liaoning Technical University, 2011.

[8] DENG Ya-ping，YANG Jia. *Bivariate polynomials key management scheme for WSN.* Journal of Computer Applications, 2010, S1:112-116.

[9] Shen Zhao, Jun Zheng. *Efficient key management Scheme in Wireless Sensor Networks.* Source: 2014 International Conference on Information Technology and Computer Applications (ITCA), p1132-1136, 2015

[10] Xianghao Nan, Huaping Chen. Zhong Chen, Yifa Li. *CPK-Cryptosystem Standard (V3.0).*Computer Security, 2009, 11:1-2.

[11] Yuhan Zhuang. *Research and Implementation of Key Management Center for Combined Public Key Cryptosystem.* South China University of Technology, 2010.

[12] FengYu Lei. *Research on Key Management in Wireless Sensor Networks*. Huazhong University of Science and Technology. 2010

[13] Le Tan, Jingjing Li, Dongyang Long. *Key Management Scheme for Mobile Ad Hoc Network Based On Combined Public Key*. Computer Engineering, 2009, 10:132-134+138