# Cluster Key Management Scheme for Wireless Sensor Networks

Qisheng Zhao
Huaihai Institute of Technology
Lianyungang Cangwu Road #59, 222005
Jiangsu China
Zhaoqisheng@tom.com

Xiaoming Liu
Huaihai Institute of Technology
Lianyungang Cangwu Road #59, 222005
Jiangsu China
Liuxm029@gmail.com

*Abstract*—**The information security of wireless sensor networks is a hot issue in research. This paper examines combining the scheme of the asymmetrical public key system and the threshold key scheme, and proposes a dynamic key management scheme through the second level key matrix of authentication mechanism from the cluster to node. The sink makes use of the lightweight ECC algorithm to preset the public key matrix to the node clusters, then the cluster head conducts the key authentication exchange and updates the key management with threshold key schemes, with no need for a third authentication center. It reduces computing and communication costs, using the preset public key encrypting the data. The clusters adopt bidirectional authentication to promote communication security.**

***Keywords- security; cluster; authentication; matrix; threshold***

## I. INTRODUCTION

The Wireless sensor network (WSN) consists of spatially distributed autonomous and battery-powered sensors which are embedded in sensor devices, data processing devices, energy devices, storage devices and communication devices to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location (i.e., base station or sink). In recent years, wireless sensor networks have proliferated to a wide range of applications such as battlefield surveillance in military applications, industrial process automation (monitoring and controlling), meteorological areas, home appliances, and health applications [1], [2]. However, wireless sensor nodes have limitations in terms of processing, centralization, limited power capacity, self-sufficiency, multi-hop routing, dynamic topology, and number of nodes. Sensor networks often provide services in hostile environments, which make them targets for malicious attackers. The WSNs face many security issues such as data intrusion and topology destruction via Sinkhole, Sybil, Wormholes, Hello flood intrusion, nodes captured etc[2],[3]. It thus makes it very challenging to provide security in WSNs. Currently; the main ideas for resisting intrusion are to establish security routing and key management schemes.

## II.RELATED WORKS

There are two types of key management scheme in wireless sensor networks: distributed architecture and clustering architecture. In the distributed architecture, there are no fixed fundamental facilities, the energy and power of network nodes are in the same level, and nodes have the ability of sensing, signal processing and wireless communication. The nodes communicate with preset keys to establish the security channels. For the distributed architecture, the nodes divide into three categories: base station, cluster heads, ordinary nodes. The base station is in charge of distributing and updating keys as a distributing center, ordinary nodes just have an ID and some corresponding keys, and others belong to the clusters' mission [5]. As mentioned in past literatures, μTESLA adopts a sharing keys generating algorithm in full networks unless the key pool, and the real keys are stored in the base station; the full networks share the key generating algorithm. It also uses the key chains which are made by one-way hash functions. However, μTESLA creates a high demand for time synchronization, and high memory overhead caused by the delay of publishing keys[3]. Another multi-μTESLA introduces the multi-level key loop, but it can't tolerate packet loss [4]; MMμTESLA introduces threshold cryptogram, and separates the authentication key into key shadows distributing to multiple base stations. The sensor nodes make use of key shadows to reconstitute authentication keys and broadcast them. However, MMμTESLA multiple base stations may raise new security issues [6]. Sun proposes an improved key management by taking advantage of a one-way hash function to alleviate the influence of compromised sensors. The function does not affect the connection between neighboring sensor nodes, but it still cannot withstand the Dos attack to the ordinary nodes, or even derivative keys conflicts [5] ,[7]. The literature [8] introduces the symmetric key cryptographic algorithms. It is used at the link layer of WSNs, but if one node key is exposed, others will be open and the data of the full network will be vulnerable. The literature [9],[10],[11] proposes Q-composite probabilistic plans based on an E-G scheme. These schemes generate less communication traffic and calculated quantity, but need increased storage for the keys and have limited networks connectivity and security. Wen[14] proposes multi-polynomial functions, and a dynamic multicast key management scheme, but this is weak in dealing with multicast unreliability in clusters. Other researchers [15] make use of location-based group key allocation, and broadcast-based rekeying by using dynamic composition key schemes. Liu [12] and Liu [13] propose a novel (q,l) threshold secret sharing scheme in which the key is divided into shadows. This promotes security, but this kind of probabilistic network is low in persistence ration and energy consumption. He et al's [16] is different in that the slice keys are put into "virtual cluster head" (VCH) and reconstructed by a "physical cluster head". However, the proposal does not provide detail about VCH, and never discusses the communication during the VCHs, This paper proposes a security sharing and group dynamic clustering key agreement scheme through $^{(q,l)}$ threshold key authentication mechanism in the cluster management.

## III.THE CLUSTERING KEY SCHEME

### A. The Model Architecture

In the hierarchical wireless sensor network, the distribution is based on clusters, and one cluster has a cluster head and

© 2014. The authors - Published by Atlantis Press

multi-cluster members, in which the cluster head has a special enhanced ability in communication, computing, memory, power etc. Cluster members are just ordinary nodes which are allowed communicating only with a cluster head or between each other to decrease energy consumption. As the cluster head has more energy, it can undertake higher demands of computing and communications, and between clusters, it's easier to adopt the public key scheme to make authentication with a base station. Another way is that, inside the cluster, the members all communicate with the cluster head and make authentication with it, so we adopt the threshold key model $(q,l)$ to carry out ordinary node security. The network architecture model and algorithm are as follows.

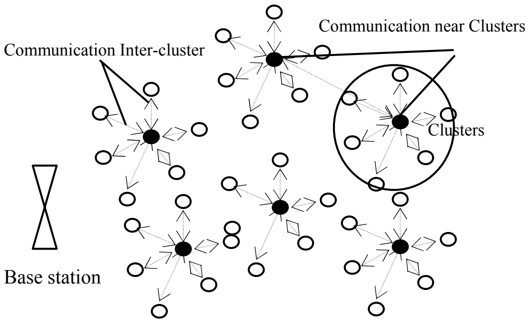

Figure 1 the Clustering Key Model Architecture

### B. Key Generation and Distribution

Before network nodes deployment, the key management center (KMC) selects a hash function. The elliptic curve and related parameters generates the public/private key matrix, and distributes the above information to the nodes. Each node has its own id, keys and other related parameters.

### C. 3.2.1 The keys between the Base station and Cluster head

The authentication system using the combine public key does not use third CA support, and thus realizes authentication sooner, and it's easier to get the number of pair keys from the combination of few key divisor and to resolve the issue in large scale certification.

First select an elliptic curve of $E_p(a,b)$, $G=(x_G,y_G)$ as the cardinal point, the times points of cardinal point $G$ compose the subgroup $S$ of $E_p(a,b)$, and in the subgroup $S$, all the elements are the times points $kG$ of cardinal point $G$ ,so that is :

$$S=\{G,2G,3G......nG\}=\{(x_1,y_1),(x_2,y_2),..(x_n,y_n)\} \tag{1}$$

For the given elliptic curve $T=(a,b,G,n,p)$, establish the private key matrix of SSK and public key matrix of PSK as follows:

$$SSK=\begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & \cdots & r_{1,j} \\ r_{2,1} & r_{2,2} & \cdots & \cdots & r_{2,j} \\ & & \cdots & \cdots & \\ r_{i,1} & r_{i,2} & \cdots & \cdots & r_{i,j} \end{bmatrix} \tag{2}$$

$$PSK=\begin{bmatrix} (x_{1,1},y_{1,1}) & (x_{1,2},y_{1,2}) & \cdots & \cdots & (x_{1,j},y_{1,j}) \\ (x_{2,1},y_{2,1}) & (x_{2,1},y_{2,1}) & \cdots & \cdots & (x_{2,j},y_{2,j}) \\ & & \cdots & \cdots & \\ (x_{i,1},y_{i,1}) & (x_{i,2},y_{i,2}) & \cdots & \cdots & (x_{i,j},y_{i,j}) \end{bmatrix} \tag{3}$$

In that, $r_{i,j}$ is the time value of $(x_{i,j},y_{i,j})$ to the cardinal point $G$ ,

$(x_{i,j},y_{i,j})=r_{i,j}*G$ , in which, $1\le r_{i,j}\le(n-1)$

We assume that the private key matrix of SSK and public key matrix of PSK is $s*t$ . Choose at random one from the line of $S$ , compose and combine at most of $S^t$ keys.

The user's public/private key is the matrix abscissa value according to mapping sequence of the user identity. Respectively select the factor of corresponding position in the public/private key matrix generated after the combination operation. Selection of the mapping function ensures that a different user identity has a different mapping value. Assuming the mapping sequence is $(a_1,a_2,...,a_n)$ , then respectively selects and combines the public key matrix of PSK and private key matrix of SSK to the public key PK and private key SK:

$$PK=(x_{a_1,1}y_{a_1,1})+(x_{a_2,2}y_{a_2,2})+......+(x_{a_t,t}y_{a_t,t}),$$

$$SK=x_{a_1,1}+x_{a_2,2}+......x_{a_t,t}\bmod(n) \tag{4}$$

( $PK$ , $SK$ ) composes the pair of the private and public key.

### D. Key defining:

The identify key is generated by the identified entity from the key matrix.

The random key in the composite key is composed of the first order identify key with the random sequence defined by the system. The random sequences eliminate the linear relationship with the private key variables, and the keys are all defined by the key management center.

Updating keys are defined by the nodes themselves and compounded with the first-order composite keys, to form second-order composite keys, which provide personal privacy protections, and allow the individual to define the signature. The updating key is used as the update for keys. The updating keys of PSK and SSK are $UPK_A$ and $USK_A$ .

The first order composite key is composed of the identity key and the random key. The KDC generates a pair of first-order random keys for the nodes: $rsk_A^{'}$ , $RPK_A^{'}$ . The first-order composite private $cpk_A^{'}$ is compounded from the identity private key and the random private key:

$$cpk_A^{'}=(isk_A+rsk_A)\bmod n \tag{5}$$

### E. The key of the second-order composite

Except identity key (I), random key(R) needs to set up the updating key (U) as defined by nodes by themselves. Node A defines the pair of updating key as $UPK_A$ , $usk_A$ . The updating key belongs to the node used and changes each time.

The second-order composite private key $csk_A^{''}$ is comprised of the first-order composite private key $csk_A^{'}$ and the updating key $usk_A$ coming from the signature party:

$$csk_A^{''} = (csk_A^{'} + usk_A) \bmod n \tag{6}$$

The second random public key is composed of the first-order random key and updating key, calculated by the signature party:

$$RPK_A^{''} = (RPK_A^{'} + UPK_A) \bmod n \tag{7}$$

The second-order composite public key is compound of the identity public key and second-order random public key (provided by signature party), calculated by

$$CPK_A^{''} = IPK_A + RPK_A^{''} \tag{8}$$

### F. The Threshold Key Scheme in Cluster Management

Before network nodes deployment, the key management center (KMC) selects a hash function, the elliptic curve and related parameters, and generates the public/private key matrix. It then distributes the above information to the nodes. Each node has its own id, keys, functions and other related parameters.

Suppose the open information in the threshold key scheme covers: one big prime number $r > n\beta$ is the primitive element from finite field of $GF(r)$ , $m = pq$ $p$ and $q$ is different prime numbers and confidential.

Assume in the cluster member, there are n nodes and like $P_1$ , $P_2$ ,… $P_n$ , sharing $k \in GF(\mathrm{r})$ of the key information generated from the cluster head. The key and authentication slices are distributed in the algorithm as following:

(1) At first select n-1 times polynomials function of $f(x) = b_{t-1}x^{t-1} + \ldots + b_1 x + k \bmod r$ secretively, in which $b_1, b_2, b_3, \ldots, \mathrm{b}_{t-1}$ and $f(x)$ all belong to $GF(\mathrm{r})$ . Randomly select $e_1$ , $e_2$ as prime as $\varphi(r)$ relatively, and expose the $e_1$ , $e_2$ and $\varphi(r)$ as an Euler function.

(2) Calculate:

$d_1 = (e_1 - 1) \bmod \varphi(m)$ And $d_2 = (e_2 - 1) \bmod \varphi(m)$ ;

From i =1,2,..i, make progression as follow: compute $s_i = f(\beta^i)$ , $w_i = s_i^{e_2 d_1} \bmod m$ and distribute $s_i$ and $w_i$ to the clustering members as key and authentication slices.

(3) when needed, the sharing key, as long as $n$ participants in the arbitrary, just $t$ co-operator can restore the key $k$ ; assume there are cluster member $P_1$ , $P_2$ ,… $P_t$ co-operators, using the $t$ key slices we can get $t$ interpolation points $(\beta^1, s_1)$ , $(\beta^2, s_2)$ ,…, $(\beta^t, s_t)$ , and then using Lagrange interpolation can refactor the polynomial $f(x)$ of $t-1$ times, then $s_t = f(\beta^t)$ , at last could calculate $k = f(0)$ , and the calculating formula of key k as follow:

$$k = \sum_{i=1}^{t} s_i \prod_{j=1,\, j \neq i}^{t} \frac{-\beta^j}{\beta^i - \beta^j} \bmod r \tag{9}$$

### G. The Security of Key Management

To promote the difficulty of cracking keys, put them into the polynomial: for the node $i$ , randomly choose the $S_i$ as the portion of the key S and $a_{i,j}(j \in 1,2,\ldots,n-1)$ , then construct the threshold polynomial $f_i(x)$ of $(t,n)$ and so:

$$f(x) = S + a_1 x + a_2 x + \ldots + a_{n-1} x^{n-1} \bmod r \tag{10}$$

$$f_i(x) = S_i + a_{i,1} x + a_{i,2} x + \ldots + a_{i,n-1} x^{n-1} \bmod r \tag{11}$$

Node $i$ calculates key portions according to the above formulas, and sends them to the node $j$ with security channel; node $j$ collects the threshold polynomials from the $t$ nodes from the cluster member, computing $f_i(x)$ to get the main key $S$ :

$$\begin{aligned} & f_1(j) + f_2(j) + \ldots + f_t(j) = S_1 + a_{1,1}x + a_{1,2}x^2 + \ldots + a_{1,n-1}x^{n-1} \\ & + \ldots + a_{2,n-1}x^{n-1} + \ldots + S_t + a_{t,1}x + a_{t,2}x^2 + \ldots + a_{t,n-1}x^{n-1} \\ & = (S_1 + S_2 + \ldots + S_i) + (a_{1,1} + a_{2,1} + \ldots + a_{t,1})x \\ & + \ldots + (a_{1,n-1} + a_{2,n-1} + \ldots + a_{t,n-1})x^{n-1} \\ & = S + a_1 x + a_2 x^2 + \ldots + a_{n-1}x^{n-1} \bmod x = f(x) \end{aligned} \tag{12}$$

## IV. Security analyses

When needed to recover the sharing key $k$ , there will be some inner deceivers in the $t$ nodes participants from the cluster members. The deceivers use the fault fragment to prevent the regular recovering of sharing key $k$ information, and the outer deceivers would try to join the key recovering as well as getting the information key $k$ . The thresholds keys can take advantage of authentication fragments effectively to detect inner and outer deceivers. Assume that the participants of $P_1$ , $P_2$ ,… $P_t$ share the key $k$ information and $P_i (1 \leq i \leq t)$ , if $W_i^{e_1} \equiv S_i^{e_2} \bmod m$ , then $P_i$ is the legal participator bringing forth the real fragments inside the cluster; otherwise, they are deceivers.

To verify, if the $P_i$ is the legal participant with real fragment， they must satisfy the formula

$$W_i^{e_1} \equiv S_i^{e_1 e_2 d_1} \equiv S_i^{e_2} \bmod m \tag{13}$$

If not, it means that the node shows troubled fragments. If it happens to lose a package in key recovery, then it will not satisfy this formula, and can let the node resend the fragment again in order to recover the correct key. If some node is attacked by a deceiver and produces troubled fragments, then it must degrade the credit level, change the node, or replace the polynomials.

For the recovery of the sharing keys, if only in $n$ participants, just random $t$ co-operators can restore the key $k$ ; assume there are cluster member $P_1$ , $P_2$ ,… $P_t$ co-operators, using the kept $t$ key slices we can get $t$ interpolation points $(\beta^1, s_1)$, $(\beta^2, s_2)$ ,…, $(\beta^t, s_t)$, and then using Lagrange interpolation can refactor the polynomial $f(x)$ of $t-1$ times, then $s_t = f(\beta^t)$ , at last could calculate $k = f(0)$ , and the calculating formula of key k is as follows:

$$k = \sum_{i=1}^{t} s_i \prod_{j=1, j\neq i}^{t} \frac{-\beta^j}{\beta^i - \beta^j} \bmod r \qquad (14)$$

When the node are captured, according to the information of the captured node, we can directly or indirectly calculate probability $F$ which means the non-captured nodes; the less of $F$ the stronger the anti-captured node. The resistance to capture for the nodes is the important index to measure the safety of key management schemes. In the key sharing model $(q, l)$, we discuss the nodes and cluster in terms of anti-captured ability. An ordinary node as the cluster member just have the key sharing with the cluster and the function to generate the key between the members, so it strongly resists capture and has less value, but with the cluster it has many keys with the base station and distributed functions. The enemy can read the nodes' memory and compute most keys. When there are $x$ nodes which were captured, the enemy can calculate the probability $p(x)$ of the random $k_i$ :

$$p(x) = \begin{cases} 0 & 0\le x\le q \\ 1-\sum_{i=0}^{q-1} \frac{C_l^i C_{n-l}^{x-i}}{C_n^x} & q\le x\le n \end{cases} \qquad (15)$$

When $x < q$ , we consider that $k_i$ is secure in the view of information theory; and when $x \geq q$ , the $k_i$ will be exposed to capture, but the node owing $k_i$ might be not captured; $F(x)$ stands for the resistance to capture, $m$ is the number of ordinary nodes, $n$ is the total clusters.

$$F(x) = p(x)\square\frac{m}{m-\frac{mx}{n}} = p(x)\square\frac{n}{n-x} \qquad (16)$$

In which for the model $(q, l)$, when $q$ and $l$ are defined and decided, $F(x)$ will increase as the x goes up; the decided $x$ , the more of $\{l-q\}$ the less of $F(x)$ , the more security of the networking.

## V.Summary

This paper proposes a step by step key authentication scheme not under the KMC control, and combines the scheme of the asymmetrical public key system and threshold key scheme. It proposes a dynamic key management scheme through the second level key matrix authentication mechanism from the cluster to node. This reduces the computing and communication costs, and as the clusters adopt bidirectional authentication, promotes communication security.

## Acknowledgment

This research was supported by China National Foundation NO.61103017, Lianyungang Science and Technology Project No.CG1215 and Lianyungang Social Development Project No.SH1212.

## References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey;" Computer Networks, pp.393-433,Elsevier, 2002.

[2] Pamo B, Perrig A, Gligor V. Distributed detection of node replication attacks in sensor networks. In Proc. of the IEEE Symp. On Security and Privacy. IEEE Computer Society, 2005.49-63.

[3] A. Perrig, R. Szewczyk. et al. SPINS: Security protocols for sensor networks. Wireless Networks, 2002, 8(5):521~534.

[4] D.G. Liu, P. Ning. Multi-Level μTESLA: A Broadcast Authentication System for Distributed Sensor Networks, ACM (TECS), 2004, 3(4):800-836.

[5] H. Chan, A. Perrig, D. Song. Key distribution techniques for sensor networks. Wireless Networks, 2004, 6(2): 277~303.

[6] Shen Yu Long, Pei Qing qin. MMμTESLA: Broadcast authentication protocol for multiple-base-station sensor networks [J]. Chinese Journal of Computer.2007.4:539-547.

[7] Sun Qian. A Novel Key Pre-distribution for Wireless Sensor Networks. Physics Procedia 25 (2012) 2183 – 2189.

[8] L.L. Si, Zhigang Ji. The Application of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks. Physics Procedia 25 ( 2012 ) 552-559.

[9] Du W.L., Deng J., et al. A key management scheme for wireless sensor networks using deployment knowledge[C]. The 23th IEEE Computer and Communications Proceedings 2004.586-597.

[10] Levi A, Tasc S E. et al. Simple extensible and flexible reandom key predistribution schemes for wireless sensor networks using reusable key pools [J]. Journal of Interlligent Manufacturing, 2009.21(5):625-645.

[11] Jaworski J, Ren M, and Rybarczyk K. Random key redistribution for wi reless sensor networks using deployment knowledge [J]. Computing, 20 09, 85(1-2): 57-76.

[12] Liu wei, Luo rong. A Lightweight Key Establishment Protocol for Wireless Sensor Networks [J]. Journal of Electronics&Information Technology. 2010(32):869-874.

[13] Liu Y.N.,Wang J.et al. Threshold key sharing Model in wireless sensor networks[J]. Journal of Electronics Information Technology. 2011(33):1913-1919.

[14] Wen T., Zhang Y. et al. Dynamic group key management scheme for homogeneous wireless sensor networks [J]. Journal on Communications, 2012(33):164-172.

[15] Paek K, Song U, Kim H, Kim J. Energy-efficient key-management (EEKM) protocol for large-scale distributed sensor networks. Journal of Information Science and Engineering 2008; 24:1837–58.

[16] Xiaobing He, Michael Niedermeier Dynamic key management in wireless sensor networks: A survey. Journal of Network and Computer Applications 2013 (36) 611–622.