A Rapid and Efficient Pre-deployment Key Scheme for Secure Data Transmissions in Sensor Networks Using Lagrange Interpolation Polynomial

Hua-Yi Lin[1], De-Jun Pan[2], Xin-Xiang Zhao[3] , and Zhi-Ren Qiu[4]

[1]Department of Information Management, Yung Ta Institute of Technology
[2]Department of Information Management, Tajen University
[3]Department of Information Management, I-Shou University
[4]Department of Information Management, Southern Taiwan University

calvan.lin@msa.hinet.net

Abstract

This study proposes a pre-deployment key management scheme that requires a few memory capacities and CPU computations to address secure data transmissions in Wireless Sensor Networks (*WSNs*). *The proposed scheme exploits threshold key management mechanisms by Lagrange Interpolation polynomial generating a key set for sensor nodes, and uses symmetric and irreversible cryptography schemes to encrypt transmitted data by the generated keys with Message Authentication Code* (*MAC*). *The sensor nodes merely have to aggregate and encrypt received data without complicated cryptography operations. The proposed approach can achieve rapid and efficient secure data transmissions with low communications, and is proper to be implemented on large-scale sensor networks.*

1. Introduction

The secure data transmission in WSN is an important issue. Due to the property of wireless communications, the data transmission is vulnerably exposed to a disclosed environment. So far, many related security research papers are proposed. The proposals can be classified into the following categories. They include (1) A single master key scheme, (2) Full pair-wise key scheme, (3) Random key pre-distribution schemes, and (4) Group-based key schemes.

In a single master key scheme, there is a common key (master key) shared by all nodes. This scheme is plain to secure data transmissions between sensor nodes, and simple to be carried out without large memories. However, the chief defect is as an adversary captures the master key of one node, the entire network is compromised.

Full pair-wise key scheme exploits a unique pair key between each two nodes to encrypt transmitted data [1] [2]. Thus every node requires pre-distributing and storing of n-1 keys. And therefore the total keys in WSNs are n (n-1)/2. In this system, each node needs to store a great quantity of keys for securing communications. Besides, increasing or decreasing nodes, the system needs to update pair-wise keys for all nodes. Although, this scheme provides a secure and resilient manner, the compromised node merely influences the secure link between other nodes connecting with it, and does not affect the security among un-compromised nodes. But, the huge memory and storage overhead are improper for large-scale sensor networks.

Random key pre-distribution schemes rely on the probability of a common shared key between two nodes [3]. Each node has a key ring randomly chosen k keys from the system

key pool *p*. If two nodes own a common key, a secure link exists between them. The larger size of a key ring increases the probability of sharing common keys and connectivities between two nodes. However, the larger size of a key ring raises the compromised effect on a path link shared between un-compromised nodes. Besides, a node probably has a same key shared with several nodes, and therefore can not recognize which one connecting with it. Additionally, larger storage capacities for a sensor node are required.

The group-based key scheme divides sensor nodes into several disjoint groups [4] [5]. Each sensor node in a group owns a common intra-group key to secure data communication with all its neighbors. The group-based key scheme outperforms the previous mentioned schemes in terms of performance, scalability, and storage overhead. However, the inter-group key management is complicated to be implemented.

The above mentioned schemes focus on key management issues in WSNs. However, no secure data transmission mechanisms are integrated into the key managements. In this paper, we propose a rapid and effect key management scheme for secure data transmissions. The proposed approach generates a private key for each node by inputting its identification into a Lagrange Interpolation polynomial. Subsequently, sensor node exploits hash functions (HMAC-160, RIPEMD-160) to generate MAC for the purpose of verifying the integrity of transited data [6]. Each node appends MAC to the forwarding data, then encrypts entire data using its private key, and sends the encrypted data to an upper node along a path to the base station. During the transmissions, each node only needs to do a few operations on hash functions, aggregations, and encryptions. This manner is plain to be implemented on sensor nodes with low memory capacity.

The remainder of this paper is organized as follows. Section 2 presents the proposed polynomial-based key management scheme for sensor networks. Section 3 then describes the approach of secure data transformations. Subsequently, Section 4 presents analyses of the scheme. Meanwhile, Section 5 shows the simulation and analytical results for the proposed scheme. Finally, Section 6 suggests possible future research directions and conclusions.

2. Polynomial-based Key Management Scheme

In order to speed up data encryption of sensor nodes, this study exploits a symmetric cryptography scheme using a Lagrange Interpolation polynomial. During the initial phase, the base station generates a secret key *SK* and chooses a prime number *p* (*p* >= *SK*), where *SK* is the system master key. Subsequently, system randomly chooses a polynomial $h(x)$ of degree t-1, $h(x) = a_{t-1}X^{t-1} + \ldots + a_1X + SK \bmod p$, where *SK* and a_i (i = 1, 2,…, k-1) are less than random prime *p*. In fact, the polynomial implies the system master key *SK*. This study can derive the master key from $h(0) = SK \bmod p$. Additionally, every sensor node has a well-known unique identity number ID_i, the system generates each node's private key $K_i = h(ID_i) \bmod p$ for i = 1,2,…,n. Each pair (ID_i, K_i) has a corresponding coordinate on a two-D space as shown in figure 1. Sine $h(x)$ is a polynomial of degree t-1, only t or above t pairs can reveal `$h(x)$, where t is the threshold value. If the disclosed data are lower than t pairs, adversaries can not derive any information from $h(x)$. In order to protect *SK*, this study employs a secret sharing scheme to distribute *SK* into t sensor nodes. However, *SK* can be reconstructed by collecting a threshold number of t keys.

After system confirms each sensor has a private key, then the system abandons *SK* for secure issues. After the initial phase, the base station owns a key ring of private keys for all sensor nodes, sensor node i owns a private key K_i. Sensor node exploits its private key K_i to encrypt transmitted data from node-to-node and to the base station.

Once the system collapses or any t sensor nodes convict the base station of insecurities, the new system can reconstruct SK from acquiring above t private keys of sensor nodes. Since each node ID_i, (i = 1,2,…,n) holds a shadow (secret sharing key) $K_i = h(ID_i) \bmod p$ as shown in figure 2. Any cooperation with t sensor nodes, this study can reconstruct the master key SK using Lagrange Interpolation polynomial.

$$h(x) = \sum_{i=1}^{t} K_i \prod_{j=1, j\neq i}^{t} \frac{x - ID_j}{ID_i - ID_j} \bmod p$$

,where $SK = h(0)$. And then the system regains normality.

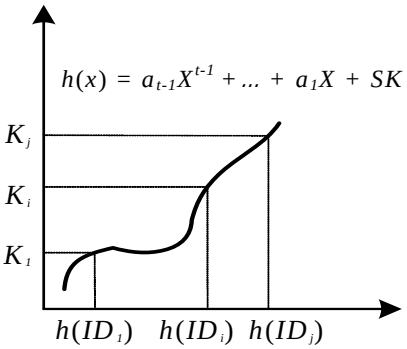

Figure 1. Each pair (ID_i, K_i) owns a corresponding coordinate on a two-D space

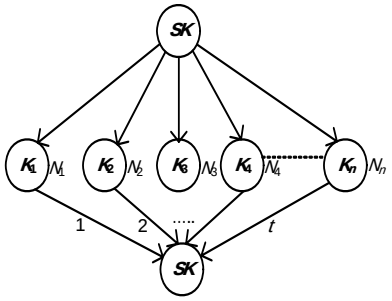

Figure 2. Using an (n, t) threshold cryptographic scheme to maintain and reconstruct SK

3. Secure data transmission scheme

In this section, this study presents the proposed secure data transmission scheme, which is well suited for sensor networks. This approach exploits the private key of a sensor node to encrypt transmitted data along a path to the base station. This method has the property of rapid computation, and low computing power. Besides, irreversible MAC functions can verify the integrity of transmitted data. The used notations in this system are as shown in table 1.

Table 1. Variables and notations

ID_i	Identity of a sensor node.
K_i	Private key of node i.
$E_{Ki}(M)$	Encrypt plaintext message M using the private key of node i.
MAC (K_i, M):	Message Authentication Code of message M using a private key K_i.
$M_1 \| M_2$	Aggregate message M_1 and M_2.

In the proposed scheme, the processes of secure data transmissions are as shown in figure 3. In the beginning, when a sensor node N_3 receives data M_3, then N_3 exploits MAC functions to generate message authentication code MAC (K_3, M_3). Subsequently, N_3

concatenates plaintext message M_3 and MAC (K_3, M_3), then uses private key K_3 to encrypt (K_3, M_3|MAC (K_3, M_3)) as D_3, and forwards D_3 to the upper node N_2. As N_2 receives D_3, N_2 concatenates D_3 with its own plaintext message M_2. Subsequently, N_2 exploits MAC functions to generate message authentication code MAC (K_2, A_2). Then N_2 concatenates A_2 and MAC (K_2, A_2), and uses private key K_2 to encrypt (K_2, A_2|MAC (K_2, A_2)) as D_2. Subsequently, N_2 forwards D_2 to an upper node N_1. Following the same steps, node N_1 encrypts (K_1, A_1|MAC (K_1, A_1)) as D_1 and forwards D_1 to an upper node until to the base station.

When the base station receives D_1, it draws out the private key K_1 of N_1 from the key ring, then decrypts D_1, and compares drawn K_1 with received K_1 to verify the data source is accurately sent from N_1. Subsequently, base station verifies that the data A_1 from N_1 are not modified by the MAC code. According to the same way, the base station extracts D_2 from A_1, uses private key of N_2 from key ring to decrypt D_2, verifies the accuracy of K_2, and then confirms A_2 is not modified by K_2 and MAC code. Subsequently, the base station extracts D_3 from A_2, then decrypts D_3 using K_3, and then verifies that the message M_3 is not modified.

Since MAC (such as SHA-1, and RIPEMD-160) functions are rapid and plain to be operated, sensor nodes can encrypt message rapidly without heavy computing power. The computing load is allocated on the base station, including data decryption and verification.

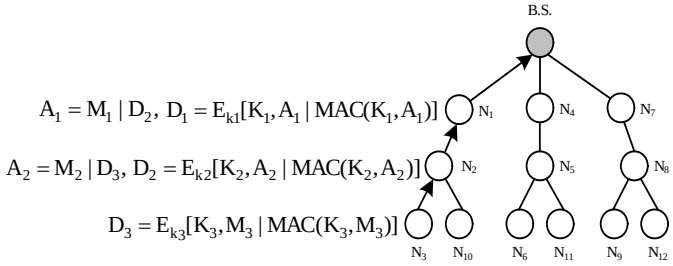

Figure 3. The processes of secure data transmissions

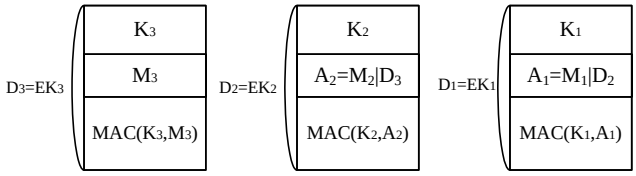

Figure 4. MAC of transmission data

4. Analyses of Proposed Schemes

(1)System security:

This study exploits a threshold scheme to protect the security of system master key *SK* [7]. If malicious attacks break below *t*-1 sensor nodes (depends on the degree of a chosen polynomial $h(x)$, higher degree is securer), then the system reveals nothing about *SK*. Meanwhile, the hostile attacks are unable to reconstruct the polynomial $h(x)$ and the private key of any sensor nodes, thus the system can retain security.

(2)Accuracy and integrity:

This study employs MAC function to verify the accuracy and integrity of data. During the period of delivery, the sender node appends MAC to the transmitted data. Once the receiver obtains the data, it uses a one way hash function (HMAC-160, RIPEMD-160) to verify the integrity of the received data. Since MAC is irreversible, given a random number *y*, there is no way to figure out *x*, such thus $H(x) = y$. Besides, when $a \neq b$, then $H(a) \neq H(b)$. Therefore, if malicious nodes modify messages, the receiver detects the alteration immediately.

(3) Fault tolerance:

In this scheme, sensor nodes own shared keys of *SK*. Once the base station collapses on condition that the system acquires *t* shared key of *SK* from sensor nodes, and then the system can reconstruct *SK*. Subsequently, this system rebuilds the key ring and private key of sensor nodes immediately and the system regains normality. This scheme is plain and efficient for achieving the fault tolerance of entire system.

(4) Storage and Computing Power:

Sensor nodes own a small amount of storage and lower computing power, thus the asymmetric key schemes (such as RSA, PKI) are improper. In the proposed scheme, each sensor node only needs to keep an individual private key and a hash function (such as HMAC-160, RIPEMD-160 etc.). Thus, small storage requirements are ideally suited for WSNs. In addition, a sensor node only has to perform a hash operation and encrypts the message by a symmetric key. The requirement of computing power is quite low.

5. Evaluation of Computing

The complexity of computing is allocated on the base station. During the initial phase, the base station computes a private key for each sensor node using a polynomial $h(x)$ of degree t-1, and the time complexity depends on the degree of $h(x)$. Higher exponent needs more computing power to perform exponential operations. This paper evaluates the following operations as shown in figure 5:

(1)Private key operations: Each sensor node needs a private key to encrypt transmitted data. Initially, base station calculates *n* private key on polynomial $h(x)$ for *n* nodes in the system. (2)Encryptions: Each sensor node encrypts the transmitted data depending on the amount of messages. (3)Decryptions: Only the base station has to perform decryptions. Once the base station receives data, then decrypts the outside layer of the encrypted message, verifies MAC, until the source message is confirmed. (4)Concatenate operations: Sensor nodes perform aggregating operations and forwarding data to upper nodes. (5)MAC compute: Each sensor node creates a MAC as data pass through it. The base station validates each MAC of descendent nodes. (6)Key and MAC comparisons: The base station receives data, and then validates the integrity of MACs as well as accuracies of private keys. (7)Different scheme comparisons: Figure 6 indicates the number of keys in different schemes. The proposed Lagrange Interpolation polynomial-based approach has lower number of keys in each node or the system. Figure 7 presents the key usage in the system. This scheme needs a few memory capacities to store keys as the number of nodes increase. Figure 8 indicates the key usage in each node. This approach only needs quite few memory capacities to store its private key. (8)System recovery: Once the base station collapses, the system collects *t* shares (private keys of sensor nodes), and then can reconstruct *SK*. Detailed processes of recovering *SK* are introduced in section 2. This study respectively exploits polynomials of degree 5, 10, 15, and 20 to renew *SK* under 1-100 sensor nodes conditions. This study assumes that sensor nodes are deployed on a reachable radio range, and randomly adds new nodes on a bound

region. The simulation result, as shown in figure 9, indicates that the successful ratio of reconstructing *SK* declines as the sensor nodes and the polynomial degree increase. However, the system still maintains a high probability of renewing *SK*.

Operations	Private Key	Encryption	Decryption	Concatenate	Compute MAC	Key Comparison	MAC Comparison
Base Station	n	0	Concatenate times	0	Amount of descendents	Amount of descendents	Amount of descendents
Sensor Node	1	1/per data	0	1/per data	1	0	0

Figure 5. Evaluation of computing operations

Scheme	Single master key	Full pair-wise key	Random key predistribuition	Group-based key	Lagrange Interpolation polynomial
Communication key	Single key	Pair keys	Common keys Path keys	Direct keys Path keys	Private key
Key updates	No	No	No	No	No
The number of keys per node	1	*n*-1	*m* (key ring size)	2M (Memory size)	1
The number of keys for the system	1	*n* (*n*-1)/2	*P* (key pool size)	2Mmn (m nodes, n groups)	*n*

Figure 6. Evaluate the number of kevs

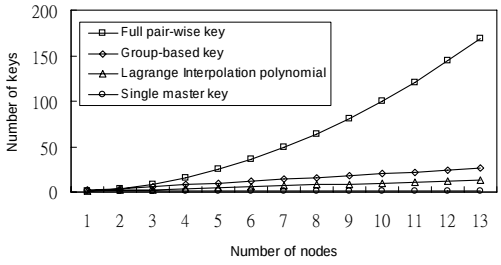

Figure 7. Key usage of the entire system

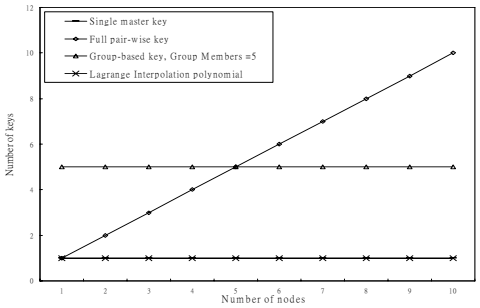

Figure 8. Key usage for each node

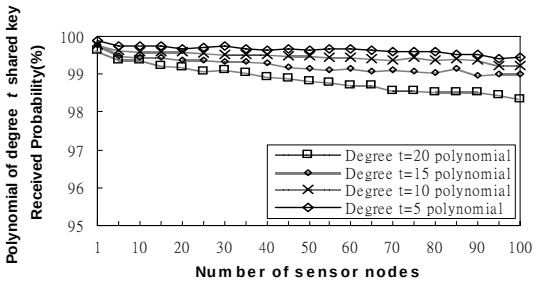

Figure 9. Successful ratio of reconstructing *SK*

6. Conclusions

This paper proposes a rapid and efficient polynomial-based key management scheme to secure data transformations. Moreover, a threshold scheme is introduced to achieve system fault tolerance, thus the system can endure *t*-1 insecure sensor nodes. Besides, this approach employs a symmetric key scheme to speed up en/decrypting operations and save computing power, therefore each sensor only needs a small space to store its private key, and low computing powers are required. Additionally, since hash functions are plain and swift to be performed, this study exploits MACs to ensure the integrity of aggregated data along the

passing through nodes. The complex operations of decrypting encrypted data are allocated on the powerful base station. The proposed scheme is well suited for large-scale sensor nodes.

7. References

[1] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks”, ACM Transactions on Information and System Securit, May 2005, vol 8, issue 2, pp. 228-258.
[2] D. Liu and P. Ning, “Establishing pair-wise key establishments for static sensor networks“, in Proceedings of 10th ACM Conference on Computer and Communications Securit, Oct. 2003, pp. 52-61.
[3] H. Chan, A. Perrig, and D. song, ” Random key predistribution schemes for sensor network”, in Proceedings of IEEE Symposium on Research in Security and Privacy, May 2003, pp. 197-213.
[4] D. Liu, P. Ning, and W. Du, “Group-based key pre-distribution in wireless sensor networks”, in Proceedings of ACM Workshop on Wireless Security, Sep. 2005.
[5] L. Zhou, J. Ni and C. V. Ravishankar, ”Efficient key establishment for group-based wireless sensor deployments,” in Proceedings of the 4th ACM Workshop on Wireless security, Sep. 2005, pp. 1-10.
[6] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks”, in Proceedings of the 9th ACM conference on Computer and communications security, Nov. 2002, pp. 41–47.
[7] F. Delgosha and F. Fekri, “Threshold key-establishment in distributed sensor networks Using a multivariate scheme”, IEEE Inforcom, Apr. 2006.