

High Effect Secure Data Transmission Mechanisms in Wireless Sensor Networks Using ID-Based Key Management Scheme

Hua-Yi Lin

Department of Information Management, China University of Technology, Taiwan, R.O.C.

calvan.lin@msa.hinet.net

doi: 10.4156/jcit.vol4.issue1.lin

Abstract

This study presents an ID-based key management scheme that exploits an implied public key to address secure data transmissions for reducing required memory capacities and CPU computations in Wireless Sensor Networks (WSNs). In the proposed scheme, each sensor node only has to store a secret key for encrypting and aggregating data during transmissions. Furthermore, the system supports direct and aggregate data transmission modes that provide rapid and secure transmission methods without complex decryption and large bandwidth consumption for sensor nodes. This approach is efficient enough to be implemented on large-scale WSNs.

Keywords

WSN, sensor node, ID-based, secret-key, aggregate.

1. Introduction

Secure data transmissions in WSNs are an important issue. The properties of wireless communications mean that data transmissions are vulnerable to a disclosed environment. Many related security research papers have been presented. The proposals can be classified into the following categories: (1) single master key, (2) full pair-wise key, (3) random key pre-distribution scheme, and (4) group-based key.

In a single master key scheme, all nodes share a common key (master key). This method easily secures data transmissions between sensor nodes, and is simple to implement without a large memory requirement. However, the entire network is compromised if an adversary captures the master key of one node.

A full pair-wise key scheme adopts a unique pair key between each two nodes to encrypt transmitted data [1] [2]. Thus, each node needs pre-distributing and storing $n-1$ keys, and each WSN has $n(n-1)/2$ keys. In this system, each node needs to store a large number of keys for securing communications. Moreover, the system needs to update pair-wise keys for all nodes

when increasing or decreasing the number of nodes. Although, this scheme is secure and resilient, a compromised node merely affects the secure link between the other nodes connecting with it, and does not affect the security among un-compromised nodes. However, the huge memory and storage overhead make it inappropriate for large-scale sensor networks. Additionally, node addition and deletion complicate system re-keying.

Random key pre-distribution scheme relies on a probability of a common shared key between two nodes [3]. Each node has a key ring of k randomly chosen keys from the system key pool p . If two nodes have a common key, then a secure link exists between them. The larger key ring size increases the probability of sharing common keys and connectivities between two nodes, increasing the storage capacity required for each sensor node. However, the key ring size raises the compromised effect on a path link shared between un-compromised nodes. Moreover, a node probably shares a key with several nodes, and therefore cannot recognize which node connecting with it.

The group-based key method divides sensor nodes into several disjoint groups. Each sensor node has a common intra-group key to secure data communications with its neighbors [4] [5]. The group-based key method outperforms other methods in terms of performance, scalability, and storage overhead [6]. However, the inter-group key management is complex to implement.

The above methods concentrate on key managements in WSNs. However, these systems lack secure data transmission mechanisms, which should be integrated into key managements. This study proposes a system that gives each node a unique and public identity such as *MAC* and *IP*, for non-repudiation after transmitting data. Each node acquires a private secret-key from the base station before deployment, and the system then adopts the identity (*ID*) and secret key as the encryption and decryption keys, respectively. Enhancement of the efficient data transmissions, the proposed system provides two transmission modes, named direct and aggregate data transmissions. Moreover, to ensure that the transmitted data are

accurate and complete [6], the proposed scheme employs a hash function such as HMAC-160 and RIPEMD-160 to generate a Hash Message Authentication Code (HMAC) for verifying the integrity of transmitted data. On the transmission path, each node aggregates HMAC in the transmitted data; encrypts transmitted data using the identity (*ID*) of the next node, and then transmits the encrypted results to the next node until the base station receives the transmitted data. Subsequently, each node merely has to perform a few hash operations, data aggregations, and data encryptions. The proposed method is suitable for implementation on sensor nodes with low memory capacities and CPU computations.

The remainder of this paper is organized as follows. Section 2 presents the proposed ID-based key management scheme. Section 3 describes the approach of secure data transformations. Subsequently, Section 4 presents security 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. ID-based key management scheme

This study attempts to improve the efficiency of operations and avoiding a large number of storage keys [7] [8] by adopting an ID-based scheme for secure data transmission. The keys are generated according to the following procedure:

Initial phase: The base station takes a confidential parameter x as input, and outputs a system master-key P and a system parameter y . After that, the base station keeps x and P , and the parameter y is well-known.

Secret-key phase : SH_k takes node ID_i , y and P as inputs, and then outputs the private secret key SK_i of node ID_i .

$$SK_i = SH_k(ID_i, y, P)$$

, where SH is a one-way hash function such as HMAC or SHA.

Data encryption phase : Sensor node j takes a plain message M , y and the identity ID_i , and then outputs a corresponding ciphertext C .

$$C = EK_{ID_i, y}(M)$$

Data decryption phase : Receiver takes ciphertext C , y and the secret key SK_i of node j as inputs, and then outputs the corresponding plaintext M .

$$M = DE_{sk_i, y}(C)$$

The base station ensures that each sensor node obtains its own secret key, for security issues, the system abandons secret key of each node. Thus, the

attacker invades the base station, and the system reveals nothing about secret keys. Moreover, the system prevents forged *ID* attacks by requiring users to be verified by the base station, and then acquire secret keys for data transmissions.

The sensor node then adopts the secret key for secure data transmissions, and the base station exploits the secret key for data decryptions.

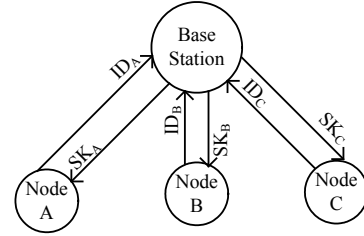


Figure 1. ID-based secret-key acquiring steps

3. Secure data transmission scheme

In this study, sensor nodes employ a secret key to encrypt transmitted data, but do not decrypt transmitted data for improving the efficiency of calculations. Additionally, the system exploits a rapid and irreversible hash function to generate HMAC for verifying the integrity of transmitted data. Table 1 presents the system notations and variables.

Table 1. System notations and variables

Variable	Definition
ID_i	Identity of sensor node i .
SK_i	Secret-key of node i .
$E_{K_i}[M]$	Encrypting message M using ID_i .
M	Sensed message.
HMAC	Generate HMAC for message M_i using secret-key SK_i .
$M_1 M_2$	Aggregate message M_1 and M_2 .
ACK	Acknowledge message.
$SeqNo$	Sequence number of a message.

The system provides a dual-mode data transmission model with direct and aggregate modes. The direct mode equips urgent data transmissions and acknowledgement transmissions. In the aggregate mode, sensor nodes deliver collected data and aggregate data along the routing path to the base station. In Figure 2, source node S gathers data, and delivers data to the base station D . The direct transmission mode is first introduced below.

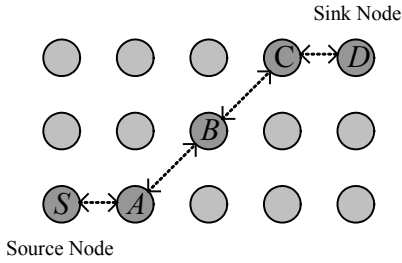


Figure 2. Secure data routing transmissions

(1) Direct transmission mode :

Source node S collects sensed data, and delivers results to the next node within the radio range. For instance, the next node A receives data from S , and then passes data along the path $B \rightarrow C \rightarrow D$. The secure data transmission procedures are as follows:

Node S :

Node S senses data, then encrypts the sensed data using destination node ID_d as $[ID_s, ID_d]EK_d$, and places the encrypted result into the first field. This operation determines whether data arrive at their destinations. Subsequently, node S encrypts transmitted data using ID_s , including a timestamp and a globally unique sequence number with messages as $[Time, SeqNo, M]EK_s$; combines its own ID_s (for recording the routing path) with the above encrypted data, and encrypts the entire data using the next node identify ID_a as $[ID_s[Time, SeqNo, M]EK_s]EK_a$. The result is placed in the second field. The format of the completely transmitted data is given as follows:

$S \rightarrow A$:

$$\{[ID_s, ID_d]EK_d, [ID_s[Time, SeqNo, M]EK_s]EK_a\}$$

Upon receiving data, node A tries to decrypt the first field of the received data, if it cannot decrypt the entire data. In this case, A is not the destination node, but is one of nodes in the routing path. Therefore, A adds its own identity ID_a into the routing path. Node A then encrypts the second field data using ID_b as $[ID_a[ID_s[Time, SeqNo, M]EK_s]EK_a]EK_b$; places the result into the second field, and then delivers the encrypted data to the next node B . The complete data format is as follows:

$A \rightarrow B$:

$$\{[ID_s, ID_d]EK_d,$$

$$[ID_a[ID_s[Time, SeqNo, M]EK_s]EK_a]EK_b\}$$

Node B receives data; repeats the same operations to check the first field, and adds ID_b into the routing path. Subsequently, B encrypts the second field using the next node identity ID_c , and then forwards the result to the next node C .

$B \rightarrow C$:

$$\{[ID_s, ID_d]EK_d, [ID_b[ID_a[ID_s[Time, SeqNo, M]EK_s]EK_a]EK_b]EK_c\}$$

Node C repeats the above operations, and forwards the encrypted data to the node D .

$C \rightarrow D$:

$$\{[ID_s, ID_d]EK_d, [ID_c[ID_b[ID_a[ID_s[Time, SeqNo, M]EK_s]EK_a]EK_b]EK_c]EK_d\}$$

After receiving data, since node D can decrypt the first field data, D is the destination node (base station or sink node). Consequently, D figures out secret keys of node D , C , B and A using the master key and each node ID , and then decrypts the encrypted data from outside to inside using the secret key of each node. Eventually, the base station D gains the plain message M , timestamp and sequence number.

Simultaneously, to ensure data acceptance, the base station D replies ACK to the source node S , and adjusts the first field and second field to $[ID_d, ID_s]EK_s$ and $[ACK, SeqNo]$ respectively. Node D adds the identification of routing nodes; encrypts ACK and $SeqNo$ using each node ID along the routing path as $ID_c[ID_b[ID_a[ID_s[ACK, SeqNo]EK_s]EK_a]EK_b]EK_c$, and transmits the encrypted data to the predecessor C in the following format:

$D \rightarrow C$:

$$[ID_d, ID_s]EK_s, [ID_c[ID_b[ID_a[ID_s[ACK, SeqNo]EK_s]EK_a]EK_b]EK_c\}$$

Node C is not the source node, since it cannot decrypt the first field. Therefore, C decrypts the second field using its secret key, and obtains the decrypted data as $ID_b[ID_a[ID_s[ACK, SeqNo]EK_s]EK_a]EK_c$. Node C recognizes that the predecessor is B , and transmits the decrypted data to B .

$C \rightarrow B$:

$$[ID_d, ID_s]EK_s, ID_b[ID_a[ID_s[ACK, SeqNo]EK_s]EK_a]EK_c\}$$

Since node B cannot decrypt the first field, it is not the source node. Therefore, B decrypts the outside layer of the second field using its secret key. Node B obtains the decrypted data as $ID_a[ID_s[ACK,SeqNo]EK_s]EK_a$, and recognizes the predecessor node A . It then forwards the decrypted data to A .

$B \rightarrow A :$

$$ID_a[ID_s[ACK,SeqNo]EK_s]EK_a$$

Node A , after receiving data, repeats the above steps until the data are delivered to the source node S .

$A \rightarrow S :$

$$[ID_a, ID_s]EK_s, ID_s[ACK,SeqNo]EK_s$$

Since node S can decrypt the first field of received data, S is the source node. Node B decrypts $[ACK,SeqNo]EK_s$ using its secret key, and obtains the $SeqNo$ of the original data and ACK from destination node D . Node S eventually confirms that the original messages are delivered accurately to the base station.

The direct transmission mode is peer to peer transformations. Each node in the routing path only encrypts the transmitted data using an ID-based scheme, and does not decrypt or aggregate the transmitted data. Therefore, the proposed scheme works more rapidly than the asymmetric cipher system, making it suitable for urgent data transmissions and confirmations.

(2)Data aggregation mode:

Since many sensor nodes require simultaneous deliveries of data, the system switches to data aggregation mode to save bandwidth and resources in multiple data transmissions. The proposed method adopts HMAC to ensure the accuracy and integrity of data transmissions. Figure 3 presents a detailed diagram of HMAC. Each sender combines the message with HMAC during the transmission, and then delivers the resulting data to the next node. The transmission procedure is as the follows:

Source node S initially hashes the plain message M_s as an output $HMAC(M_s)$, and then encrypts the aggregation of M_s and $HMAC(M_s)$ using the next node identification ID_a as the follows:

$S \rightarrow A :$

$$[[M_s|HMAC(M_s)]EK_s]EK_a$$

Node A , after receiving data, decrypts $[[M_s|HMAC(M_s)]EK_s]EK_a$ using its own secret key, and aggregates $[M_s|HMAC(M_s)]EK_s$ with the transmitted data $[M_a|HMAC(M_a)]EK_a$ of A as $[M_s|HMAC(M_s)]EK_s||[M_a|HMAC(M_a)]EK_a$. Node A then encrypts the transited data using next node identification ID_b , and forwards the encrypted data to the next node B .

$A \rightarrow B :$

$$[[M_s|HMAC(M_s)]EK_s||[M_a|HMAC(M_a)]EK_a]EK_b$$

Node B receives the transmitted data; decrypts the data using its secret key, and aggregates the decrypted result with the wanted aggregation data as $[M_s|HMAC(M_s)]EK_s||[M_a|HMAC(M_a)]EK_a||[M_b|HMAC(M_b)]EK_b$. Then B encrypts the transmitted data using the next node identification ID_c , and forwards the aggregated results to the next node C .

$B \rightarrow C :$

$$[[M_s|HMAC(M_s)]EK_s||[M_a|HMAC(M_a)]EK_a||[M_b|HMAC(M_b)]EK_b]EK_c$$

Node C , after receiving data, repeats the above same steps.

$C \rightarrow D :$

$$[[M_s|HMAC(M_s)]EK_s||[M_a|HMAC(M_a)]EK_a||[M_b|HMAC(M_b)]EK_b||[M_c|HMAC(M_c)]EK_c]EK_d$$

Eventually, node D receives the transmitted data, and decrypts these data using its own secret key. Since node D is the base station (sink node), it figures out the secret keys for each node using the master key and identifications of ID_s , ID_a , ID_b and ID_c . Then node D decrypts $[M_s|HMAC(M_s)]EK_s$, $[M_a|HMAC(M_a)]EK_a$, $[M_b|HMAC(M_b)]EK_b$ and $[M_c|HMAC(M_c)]EK_c$. The system then adopts HMAC to verify whether data are modified during transmissions.

Since the data aggregation mode can deliver mass messages using a single routing path without multiple paths, it can lower the requirements of bandwidth for WSNs.

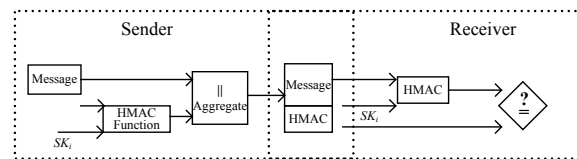


Figure 3. HMAC message authentication code operations

4. Security Analyses

(1) Data confidentiality

Sensor node obtains transmitted data, then employs the ID-based scheme to encrypt data, and forwards encrypted data to the following node. Only the base station has the secret key of each node. The other nodes reveal nothing about secret keys, and therefore they cannot decrypt the encrypted data.

(2) Data accuracy and integrity

This study employs HMAC to ensure the accuracy and integrity of the transmitted data. During the transmissions, the sender computes HMAC and aggregates HMAC with the transmitted data. Once the base station receives the transmitted data, then puts plain messages and secret keys into a hash function, and verifies the resulting HMAC with received HMAC to ensure the data integrity. Since HMAC is an irreversible operation, given a random number y , no ways can compute x such thus $H(x)=y$. Moreover, when $a \neq b$, then $H(a) \neq H(b)$. Therefore, if any sensor nodes modify the transmitted data during transmissions, the base station detect the unmatched HMAC instantly and recognizes the tampered data.

(3) Data En/Decryption Operations

Since the sensor nodes have limited computing power and storage, they cannot employ asymmetric key schemes such as RSA or PKI to encrypt transmitted data. In this study, each node only conserves an individual secret key and a hash function such as HMAC-160 and RIPEMD-160, and therefore consumes few resources during operations. Additionally, hash functions and symmetric operations are highly efficiency, making the proposed scheme extremely suited for WSNs.

5. Computing Evaluation

The proposed method allocates the complex data decryption and verification operations on the base station to prevent the overburdening of sensor nodes, which have insufficient resources. Moreover, the proposed ID-based scheme is clear and simple, and consumes few of CPU resources. This study evaluates the following computation operations.

(1) Secret-key operations: Each sensor node owns a single secret key for encrypting data. In the initial phase, if n nodes exist in the system, the base station calculates n secret keys for nodes using $SH(x)$.

(2) Encryptions: When sensor nodes receive data, nodes encrypt the transmitted data and forward the encrypted data to the next node. Therefore, the operational times depend on the numbers of the received data.

(3) Decryptions: In the direct transmission mode, each node encrypts the transmitted data along the routing path and transmits the encrypted data to the next node. The base station eventually receives the final data; decrypts the outside layer one by one, and then restores the plain messages.

In aggregation mode, each node calculates HMAC; appends it to the required transition data, and then encrypts both using its own ID . The node then aggregates these encrypted data with received data (from previous nodes), and encrypts them again using the ID of the next node. These steps are repeated until the base station is reached. The base station thus receives the transmitted data; figures out the secret key of each node in the routing path; decrypts the encrypted data, and verifies each HMAC. The operational times depend on the passing through nodes of accumulating transmitted data.

(4) Aggregate operations: Each sensor node accumulates transmitted data coming from predecessor nodes, and forwards the encrypted result to the base station.

(5) Message Authentication Code: Each node calculates the HMAC of the received data until the data are delivered to the base station. The operational time is the time taken by data to pass through nodes.

(6) Secret-key and HMAC compared times: The base station receives the transmitted data, and employs secret keys of the passing nodes to verify the integrity of each HMAC. If the message is modified, the base station can find which node alters transmitted data instantly. The whole HMAC comparison times are the numbers of the passing through nodes during the transmission.

(7) Comparison of different schemes: Figure 4 presents the numbers of keys in various systems. The ID-based scheme only needs one new secret key for each new node that joins the system. In contrast, a full pair-wise key system needs to save numerous keys for each new node. Figure 5 shows the numbers of keys maintained by each node under different key schemes. In an ID-based scheme, each node only retains one secret key and the required memory does not rise as new nodes join the network. Figure 6 shows the evaluation in different types of operations for sensor nodes and the base station. Consequently, each node

performs a few operations, while the base station performs most of them. Figure 7 presents that each node in the ID-based only requires a small memory capacity to store a secret key, an HMAC function and operations.

Furthermore, the ID-based scheme requires fewer keys and less memory than the other approaches.

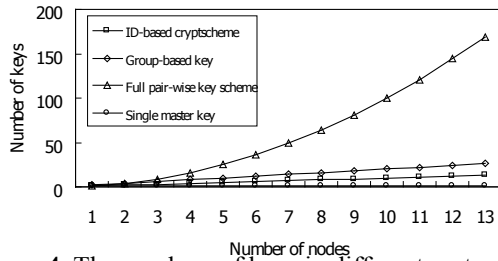


Figure 4. The numbers of keys in different systems

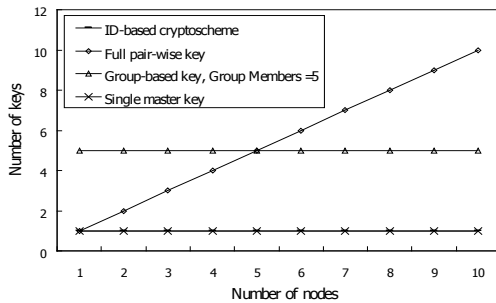


Figure 5. Key usages for each node

Operation	Private Key	Encryption	Decryption	Aggregation	MAC Operation	Comparison of MAC
Base Station	n	0	No. of Passing nodes	0	No. of Passing nodes	No. of Passing nodes
Sensor Node	1	1/Per data	0	1/Per data	1	0

Figure 6. Different types of operations for

Scheme	Single Master Key	Full Pair-wise Key	Random Key	Group Key	ID-based
Communication Key	Single Key	Pair Key	Common Key / Path Key	Direct Key / Path Key	ID
Key Updates	No	$n(n-1)/2$	No	No. of members	No
No. of keys/Per node	1	$n-1$	Key ring size	1	1
No. of Keys for system	1	$n(n-1)/2$	Key pool size	mn (m nodes, n groups)	n

Figure 7. Various key comparisons for different schemes

6. Conclusions

This study proposes an ID-based secure data transmission scheme, and exploits the implied public

key information to reduce the storage requirement. Moreover, this study adopts a symmetric key method to improve the efficiency of the cryptographic system and reduce the number of CPU computations required. The proposed dual-mode transmission model adopts two modes, named direct and aggregate modes, to provide flexible data transmission. The direct mode supports urgent data delivery methods and received data confirmations using ACK. The aggregate mode employs HMAC to ensure the accuracy and integrity of transmitted data. The significant efficiency saving is that the mode can deliver mass messages only using a single routing path, thus significantly reducing the required wireless bandwidth. Additionally, the aggregate mode allocates complex operations on the powerful base station for decryptions and verifications, thus reducing the requirements of storage and CPU computations of a sensor node. The proposed scheme is appropriate 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 Security*, Vol 8, issue 2, May 2005, 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 Security*, 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 predistribution 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] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing", *Advances in Cryptology CRYPTO 2001*, Vol. 32, No. 3, LNCS, 2001, pp. 586-615.

- [8] A. Shamir, "Identity-based cryptosystems and signature schemes" , in Proceedings of CRYPTO 84, Vol. 196, LNCS, 1985, pp. 47-53.