A key management scheme realising location privacy protection for heterogeneous wireless sensor networks

Erdong Yuan and Liejun Wang*

School of Information Science and Engineering,
Xinjiang University,
Urumqi 830046, China
Email: 731398345@qq.com
Email: iejunaang@sina.com
*Corresponding author

Abstract: Key management is the core of wireless sensor network (WSN) security management technology. At present, the protection of the source location information of nodes has also received great attention. In this paper, we combine identity-based encryption (IBE) algorithm with elliptic curve cryptography (ECC) based digital signature authentication to achieve more secure authentication. Then we use the double encryption method to realise the location privacy protection. Finally, we adopt a new routing update scheme to prevent attackers from initiating sinkhole attacks. In addition, we add timestamps to messages transmitted between nodes to defend against resend attacks. The scheme we proposed occupies a small amount of key storage space, consumes relatively more energy to protect the location information of nodes in the heterogeneous sensor network (HSN), and prevents attackers from initiating sinkhole attacks and resend attacks, thereby enhancing network security.

Keywords: key management; location privacy protection; sinkhole attacks; IBE algorithm; HSN; heterogeneous sensor network.

Reference to this paper should be made as follows: Yuan, E. and Wang, L. (2020) ‘A key management scheme realising location privacy protection for heterogeneous wireless sensor networks’, *Int. J. Sensor Networks*, Vol. 32, No. 1, pp.34–41.

Biographical notes: Erdong Yuan is now in the 2nd year for his MS in Information and Communication Engineering at Xinjiang University. He has a strong interest in the research of wireless sensor network security.

Liejun Wang received his PhD in the School of Information and Communication Engineering from the Xi’an Jiaotong University in 2012. He is also a Member of the Education Information Teach and Teaching Committee, Member of the expert group for promoting domestic cryptography in key areas in Xinjiang, Director of the Advisory Committee of Educational Information Technology Experts in Xinjiang, Director of the main node of the China Education and Scientific Research Network in Xinjiang, and Deputy Director of Network Centre. His current research interests include wireless sensor network, encryption algorithm and image intelligent processing.

1 Introduction

Wireless sensor networks (WSNs) have a wide range of applications in various fields, such as smart home, traffic management, medical care, environmental monitoring etc. Security issues have been the focus of attention in many applications. As the core of WSN security management technology, key management has also become a difficult point of research. The main difficulty is that in the case of WSN nodes with limited resources, the key management scheme must be designed to ensure security without taking up a lot of storage and energy consumption. A better compromise brings a certain challenge to the designer.

At present, HSN has become a hotspot in WSNs, which are expected to achieve better performance (Yu et al., 2007). The privacy protection of some important data has also received intense attention. For example, the scheme (Yu and Wang, 2017) involves the protection of user privacy, while our scheme focuses on the privacy protection of node location information during the key establishment process. Gura et al. (2004) measured the time of 160-bit elliptic curve cryptography (ECC) point multiplication on the 8 MHz Atmel ATmega128 for 0.81s, demonstrating that the ECC public key cryptography algorithm can be used for small WSN nodes. Du et al. (2007a, 2009b) combined routing structure with key management to open a new branch of key management scheme (labelled as the original key management scheme). The advantages of this scheme are mainly reflected in two aspects. First, it uses the communication neighbour (c-neighbour) concept, that is, only the communication key needs to be established for the c-neighbour nodes, thus reducing the communication and computational overhead used for key establishment. Second, the ECC digital signature (Rivest et al., 1992) is used to ensure the reliability of the message source, and the Diffie-Hellman key exchange idea (Maurer and Wolf 1998) is used to generate the shared key for the c-neighbour nodes. Wang et al. (2014) found that Du et al.'s scheme lacked authentication between c-neighbour nodes during the shared key establishment process and then used identity encryption technology to authenticate the message source (labelled as Wang et al.'s distributed key management scheme). We find that there is a hypothesis problem about the process of establishing the communication key in Wang et al.'s scheme. Wang et al.'s scheme assumes that each node knows its neighbours. However, this assumption does not match the random deployment situation because the randomly deployed L-sensors cannot know which nodes will become neighbours within the communication range before deployment (Xiao et al., 2007). In addition, we find that Wang et al.'s scheme had the risk of leaking the private key. Moreover, the attacker could provide enough energy for the compromised nodes to initiate sinkhole attacks (Mathew and Terence, 2017). According to Wang et al.'s scheme, when one node is captured, the private keys of other nodes in the cluster will be exposed. Wang et al.'s scheme poses a significant risk in protecting the location information uploaded by nodes. The specific analysis will be carried out later.

This paper includes three main contributions. Firstly, we find that schemes (Du et al., 2007a, 2009b; Wang and Wang, 2014) have a lack of location privacy protection for L-sensors. When the location information of the L-sensors is acquired by the attacker, the routing structure information of the network is exposed. An attacker can use the location information of the L-sensors to capture them. Many papers (Miao et al., 2018; Singh et al., 2016; Zhou et al., 2013) have noticed the importance of location privacy protection for nodes. Secondly, at different stages of message source authentication, we combine IBE with ECC-based digital signature authentication to achieve more secure authentication. Finally, we adopt a new routing update scheme to prevent attackers from initiating sinkhole attacks. In addition, we add timestamps to messages transmitted between nodes to defend against resend attacks.

The rest of this paper is organised as follows: In Section 2, we describe some of the existing key preload management schemes. In Section 3, we explain some of the theoretical knowledge of mathematics and the process of using this theoretical knowledge to generate a network routing structure. In Section 4, we outline our key management scheme. In Section 5, we give the performance evaluation of our key management scheme. In Section 6, we conclude this paper.

2 Related works

Some schemes are proposed to research key management in WSNs, but each scheme has its own focus. Eschenauer and Gligor (2002) first proposed a random preloading key management scheme (labelled as E-G scheme) for homogeneous WSNs. The scheme first establishes a large key pool, and each node randomly preloads the same number of keys from the key pool. Then, the identical key could be found as a shared secret between any two nodes through preloading keys. When the identical key does not exist, the path key could be established through the neighbour node. The advantage of this scheme is that the computational complexity is low, but the disadvantage is that the invulnerability deteriorates when the number of captured nodes increases.

Chan et al. (2003) proposed a q-composite scheme based on the E-G scheme. When the same number of keys between two nodes is not less than q, the same keys are operated by a one-way hash function to generate a shared key. The advantage of this scheme is that the invulnerability is better than the E-G scheme when there are fewer nodes captured, but the disadvantage is that the invulnerability rapidly deteriorates when more nodes are captured.

Cheng et al. (2005) proposed efficient pairwise key establishment and management in static WSNs. This scheme first generates a large key pool P through the server and selects n keys from P to form a matrix K, where n is the number of nodes in the network. Then randomly selecting a row and a column from the matrix K constitutes a key chain, each key chain corresponding to a key chain identifier, and each node is pre-loaded with a different keychain and corresponding keychain identity identifier. The node performs shared key discovery by broadcasting the identity identifier and the keychain identifier. When each node discovers the common key part with the neighbour nodes through the key chain identifier, then the shared key is obtained by performing an OR operation. The advantage of the scheme is that good connectivity can be achieved, but the disadvantage is that it is vulnerable to node capture attacks. When some nodes are captured, the attacker can use the shared keys of the compromised nodes to derive the shared keys of the normal nodes (Chien et al., 2008).

Gandino et al. (2016) proposed fast hierarchical key management scheme with transitory master key for WSNs, in which two neighbour nodes (say u and v) establish shared key. The process is as follows: Firstly, the u node obtains the master key (K_v) of the v node by using the transitory master key (K_{IN}) and the v node identifier (ID_v) in combination with the pseudo-random function controlled by K_{ID}, Similarly, the v node obtains the master key (K_u) of the u node. Secondly, the transitory master key is deleted in time after generating the master key of the neighbour nodes. Finally, u and v obtain the shared key (K_{uv}) by combining the same identity identifier (ID_v) with the pseudo-random function controlled by K_u. The master keys of the neighbour nodes are deleted in time after both nodes generated the shared key. The advantage of scheme is to increase the overall level of security by reducing the time required to delete transitory keys. The disadvantage is that no network deployment knowledge is utilised, which consumes more energy during the key establishment process, while there is still a risk of transitory master key leak.

3 Mathematical theoretical knowledge and generating network routing structure

According to the scheme (Szczechowiak et al., 2010), identity-based cryptography is a better security solution than the standard public key and the symmetric key mechanism, so we also use the IBE algorithm to complete the key management scheme design. In this section, we will elaborate on the following three aspects: Section 3.1 explains the bilinear Diffie-Hellman problem on which the security of IBE algorithm depends. Section 3.2 explains identity-based encryption (IBE) and decryption algorithm. Section 3.3 is the process of generating a network routing structure using the above mathematical theory in our scheme.

3.1 Bilinear Diffie-Hellman problem

Step 1: The offline private key generator (PKG) selects a k-bit long prime number p and uses p to obtain a finite field F_p. Selecting a safe elliptic curve E over the finite field F_p (labelled as $E(F_p)$). An addition exchange group of q-order (the generator is P) is generated with a unit of ∞, and a q-order multiplicative loop group is generated with a unit of 1. The bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ meets the following three conditions:

Bilinear: $\hat{e}(ap, bQ) = \hat{e}(P, Q^{ab}) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a$,

where $P, Q \in G_1\ a, b \in Z_q^*$

Non-degeneracy: P is the generator of G_1, bilinear mapping $\hat{e}(P,P) \neq 1$.

Computable: For any $P, Q \in G_1$, bilinear mapping $\hat{e}(P,Q)$ is calculated in a polynomial time. It is not feasible to calculate $\hat{e}(abP, Q)$ from the above conditions, and the problem is called the bilinear Diffie-Hellman problem.

Step 2: Selecting two hash functions to satisfy H_0:$\{0,1\}^* \rightarrow G_1^*$, H_1:$G_2 \rightarrow \{0,1\}^n$. The cipher text space is $C = G_1^* \times \{0.1\}^n$.

Step 3: The offline PKG outputs system parameter $\pi = \{q, \hat{e}, n, P, P_{pub}, H_0, H_1\}$ and master key $s \in Z_q^*$ by inputting a security parameter k. where the master public key $P_{\text{pub}} = sP$ can be obtained by s and P.

Step 4: Given the L-sensor (say u) identity identifier $\text{ID}_u \in \{0,1\}^*$ as its own public key, PKG calculates $Q_{ID_u} = H_0(ID_u) \in G_1^*$, and obtains its private key $(K_{ID_u} = sQ_{ID_u})$. According to the irreversibility of the bilinear Diffie-Hellman problem, the attacker cannot calculate master key s even if knows the public key and private key.

3.2 Identity-based encryption and decryption algorithm (Shamir, 1984)

Each L-sensor (say *u*, *v*) is pre-loaded with their own identity identifier, private key, π and *s*. The preloaded materials of H-sensors are the same as the L-sensors. H-sensors have tamper-proof hardware, but L-sensors do not have tamper-resistant hardware due to cost issues (Du et al., 2007a, 2009b).

Identity-based encryption and decryption process (say *u*, *v*):

Step 1: The u node generates a random number $r \in Z_q^*$, and gets *rP* (Yacobi, 2002) through discrete logarithm problem on G_2 group. Setting m be plain text, the specific encryption process is as follows:

$$g(ID_u) = \hat{e}(Q_{ID_u}, P_{pub}) \in G_2^*$$
$$V = m \oplus H_1({g_{ID_u}}^r))$$
$$u = rP$$
$$C = \langle U, V \rangle$$

Step 2: The u node sends cipher text C to the v node.

Step 3: After receiving C, the v node generates K_{ID_u} using ID_u, π and *s*. The specific decryption process is as follows:

$$\hat{e}(K_{ID_u}, U) = \hat{e}(sQ_{ID_u}, rP) = \hat{e}(Q_{ID_u}, P)^{sr} = \hat{e}(Q_{ID_u}, P_{pub})^r = g_{ID}^r$$
$$V \oplus H_1\left(\hat{e}(K_{ID_u}, U)\right) = m$$

Through the above IBE and decryption process, we find that if the v node wants to decrypt C, the v node can obtain K_{ID_u} by ID_u, π and *s*. When one node is captured, π and *s* will be leaked, and the attacker can use π and s in conjunction with the identity identifiers of other nodes to obtain the private keys of these nodes. Therefore, this method carries a great risk. This problem is the defect of Wang et al.'s scheme. If the v node applies for the private key of the u node by a trusted third party (the same PKG exists), there are still two problems: The identity trusted authentication of the v node and the private key of the u node is securely distributed.

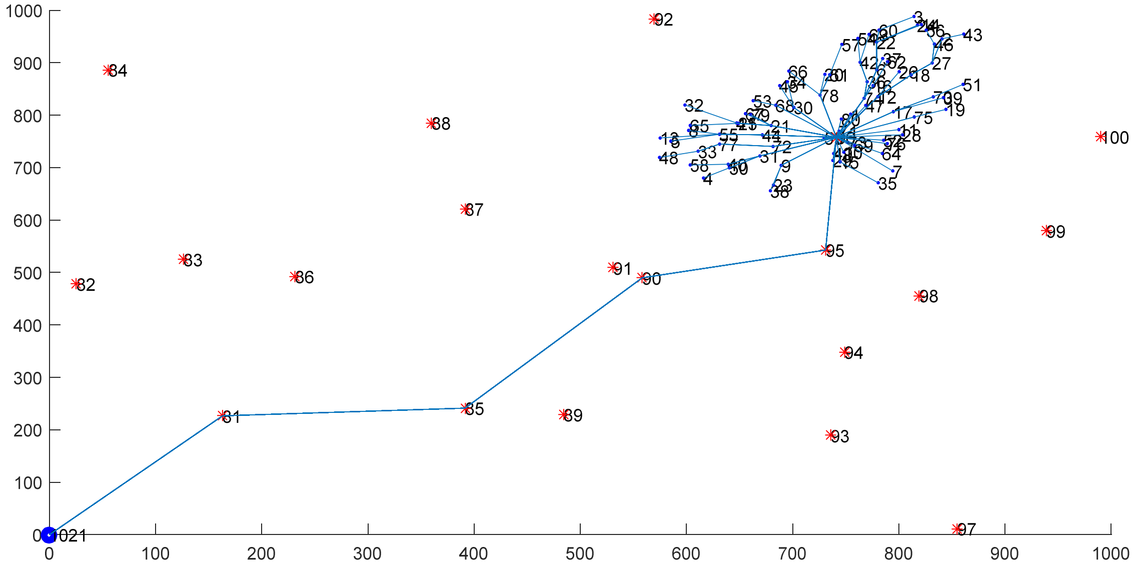
3.3 Network routing structure generation process in our scheme

In the HSN, a layered network structure consists of three types of sensors. Routing is divided into two phases:

- *Intra-cluster routing*: L-sensors send data to the nearest H-sensors.
- *Inter-cluster routing*: A backbone network is established between the H-sensors to transfer data collected from multiple L-sensors to the sink.

Figure 1 is simulation diagram of routing structure in HSN.

Figure 1 Simulation diagram of routing structure in HSN (see online version for colours)



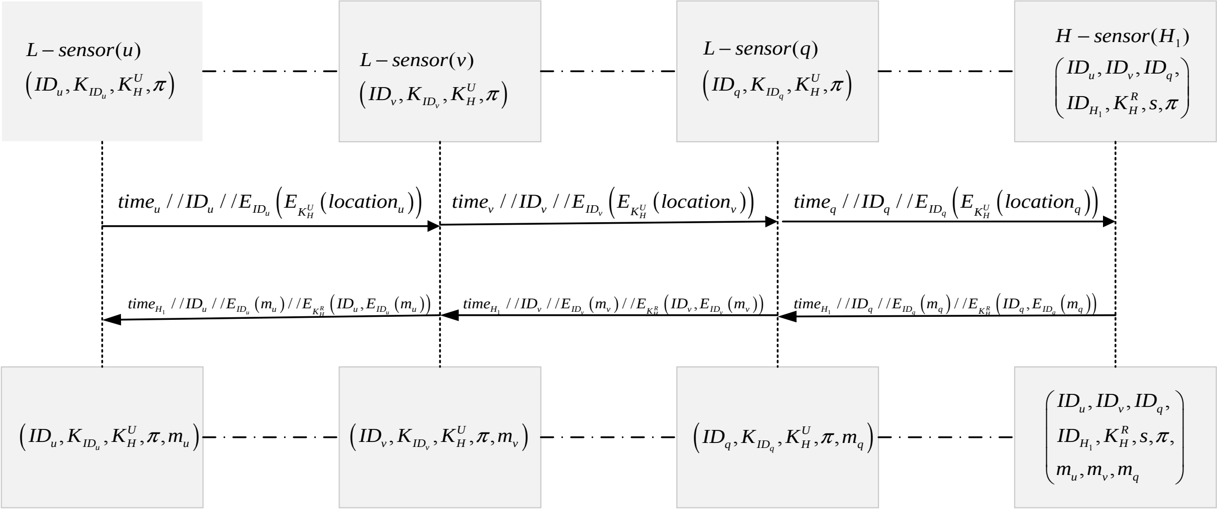
Taking the intra-cluster routing as an example. Firstly, according to the routing protocol algorithm of scheme (Du et al., 2007a, 2009b), L-sensors can obtain their own location information by using secure location services (such as (Lazos and Poovendran, 2004)). Secondly, L-sensors (say *u*, *v* and *q*, with the u node as the farthest node, the v node as the next far node, and the q node as the nearest node analysis.) select the H-sensors closest to themselves and send key request messages to the nearest H-sensor (say H_1) according to the shortest path distance. Finally, we need to protect the security of the location information in the key request messages uploaded by L-sensors. In distributed key establishment scheme, each L-sensor (say u) encrypts the location information for the first time using the H_1 node's public key (K_H^U), and then encrypts it again with ID_u. Next, each L-sensor in the cluster forwards the encrypted key request message and timestamp to the H_1 node via a greedy geographical routing protocol (such as (Karp, 2000)). After receiving the encrypted key request message and timestamp of the u node, the H_1 node uses the timestamp of the u node to verify the validity of the message, thus avoiding the resend attack. The H_1 node obtains K_{ID_u} using ID_u, π and *s* to decrypt and verify the validity of the message source. After verifying the legitimacy of the message source, the H_1 node decrypts it again by its own private key (K_H^R) to obtain the location information of the u node. In centralised key establishment scheme, the location information of L-sensors only needs to be encrypted once with their own public keys, and other operations are the same as the distributed key establishment scheme. After obtaining the location information of all L-sensors in the cluster, the H_1 node uses their location information to generate routing structures and the communication keys or routing materials of L-sensors. In order to ensure the security of the communication keys or routing materials of L-sensors during the distribution process, the H_1 node first encrypts the communication key or routing material with ID_u of the corresponding L-sensor (say *u*) in the cluster, and then digitally signed (Rivest et al., 1992) it with K_H^R . The H_1 node unicasts the digital signatures generated above to L-sensors. After receiving its own digital signature information, each L-sensor verifies the message source using K_H^U . After the message source is verified, each L-sensor decrypts the encrypted communication key or routing material by using its own private key. In Figure 2, the intra-cluster L-sensors upload key request messages to the H_1 node. In addition, the H_1 node unicasts the communication key or routing structure to the L-sensors.

4 Key establishment process between L-sensors in our scheme

4.1 Centralised communication key establishment

When L-sensors (say *u*, *v*) are determined as communication neighbour nodes, the H-sensor (say H_1) generates a communication key (K_{uv}) through a pseudo-random function. Through the above analysis, we know that m_u and m_v both represent K_{uv} between the u node and the v node.

Figure 2 Obtaining communication key or routing structure process between L-sensors (see online version for colours)


$L-sensor(u)$
$(ID_u, K_{ID_u}, K_H^U, \pi)$
$L-sensor(v)$
$(ID_v, K_{ID_v}, K_H^U, \pi)$
$L-sensor(q)$
$(ID_q, K_{ID_q}, K_H^U, \pi)$
$H-sensor(H_1)$
$\begin{pmatrix} ID_u, ID_v, ID_q, \\ ID_{H_1}, K_H^R, s, \pi \end{pmatrix}$
$time_u // ID_u // E_{ID_u}(E_{K_H^U}(location_u))$
$time_v // ID_v // E_{ID_v}(E_{K_H^U}(location_v))$
$time_q // ID_q // E_{ID_q}(E_{K_H^U}(location_q))$
$time_{H_1} // ID_u // E_{ID_u}(m_u) // E_{K_H^R}(ID_u, E_{ID_u}(m_u))$
$time_{H_1} // ID_v // E_{ID_v}(m_v) // E_{K_H^R}(ID_v, E_{ID_v}(m_v))$
$time_{H_1} // ID_q // E_{ID_q}(m_q) // E_{K_H^R}(ID_q, E_{ID_q}(m_q))$
$(ID_u, K_{ID_u}, K_H^U, \pi, m_u)$
$(ID_v, K_{ID_v}, K_H^U, \pi, m_v)$
$(ID_q, K_{ID_q}, K_H^U, \pi, m_q)$
$\begin{pmatrix} ID_u, ID_v, ID_q, \\ ID_{H_1}, K_H^R, s, \pi, \\ m_u, m_v, m_q \end{pmatrix}$

4.2 Distributed shared key establishment

When L-sensors (say *u*, *v*) are determined to be communication neighbour nodes, through the above analysis, we know that m_u and m_v represent their routing materials, respectively. In our scheme, the routing materials of L-sensors contain the identity identifiers and private keys from the optimal neighbour nodes and the sub-optimal neighbour nodes (the identity identifiers and private keys of multiple backup neighbour nodes may also be included as needed). Each L-sensor first establishes shared keys with the optimal neighbour nodes. Still taking the u node and the *v* node as examples, in Section 3.2, r_uP and r_vP are obtained (Yacobi, 2002). The *u* node encrypts r_uP with ID_u and sends it to the v node. Similarly, the v node encrypts r_vP with its own identifier (ID_v) and sends it to the u node. After receiving the encrypted r_uP, the v node decrypts and verifies the message source with K_{ID_u} distributed by the H_1 node. The u node performs the same operation. Finally, the shared key is obtained via the Diffie-Hellman key exchange idea. In our scheme, even if the attacker captures a node and obtains the private keys of the neighbour nodes, cannot knows their location information through the obtained private keys. Because K_H^R has already encrypted their location information.

4.3 Addition of new nodes and revocation of dead or compromised nodes

Due to some specific nodes in the network become the optimal communication neighbour nodes of multiple L-sensors, these specific nodes will lead to premature death, and even some specific nodes will be attacked by attackers. The death or compromise of these nodes usually leads to network partitioning, which is not conducive to uploading data. Therefore, we must undo the dead or compromised nodes while adding new sensor nodes. Firstly, each H-sensor distributes the identity identifiers and private keys of the new nodes to L-sensors of a certain area in advance, and the new nodes are pre-loaded with identity identifiers and private keys of the nodes with the communication neighbour relationship. Secondly, the addition process of the new nodes still uses the shared key establishment principle described above to establish a shared key. It is assumed that the H-sensors can detect which specific nodes are dead or compromise by the detection mechanism and broadcast the identity identifiers of these nodes. Finally, after receiving these nodes' identity identifiers, each L-sensor filters the identity identifiers of the neighbour nodes to delete these nodes' identity identifiers and deletes the previously established shared keys.

5 Performance evaluation

In this section, we will compare the original key management proposed by Du et al. and the distributed key management scheme proposed by Wang et al. We will compare the storage space occupied by the pre-loaded keys of the nodes and computational overhead of key establishment process in Sections 5.1 and 5.2, respectively. Section 5.3 performs a safety analysis.

5.1 Key pre-loaded storage comparison

The original centralised key management scheme is adopted in references (Du et al., 2007a, 2009b), which do not encrypt location information of the key request messages of L-sensors and do not digitally sign the key request messages uploaded by L-sensors. Each H-sensor is pre-loaded with K_H^R, public keys of all L-sensors and the key (K_H) used for communication between H-sensors. Each L-sensor (say u) is pre-loaded with K_H^U and its own private key (K_U^R). The number of H-sensors and L-sensors set in the HSN is M and N, respectively, and is usually M << N. The same is true for the number configuration of different types of nodes in other key management schemes in this paper. In original centralised key management scheme, the total number of preloaded keys is:

$$M\times(N+3)+2\times N=(M+2)N+3M \tag{1}$$

It should be noted that we can deduce two possibilities by observing the number of keys preloaded into each L-sensor. One possibility is that the public keys of all H-sensors preloaded into the L-sensor are the same one. Another possibility is that L-sensors in the cluster already know which H-sensor is closest to themselves before the cluster formation. The scheme uses the idea of the paper (Du and Lin, 2005) that all L-sensors know the location of the H-sensor after nodes deployment. However, the L-sensors cannot determine which H-sensor is closest to themselves before random deployment. Therefore, the public key of the closest H-sensor cannot be stored in advance to the L-sensors. We consider the first possibility.

The original distributed key management scheme (Du et al., 2009) adds signature and verification process for the key request message uploaded by L-sensors. Each H-sensor is pre-loaded with the public keys of all L-sensors, K_H^R and K_H. Each L-sensor (say u) is pre-loaded with K_U^R and K_H^U. In original distributed key management scheme, the total number of preloaded keys is:

$$M\times 3+N\times 2=3M+2N \tag{2}$$

Wang et al.'s distributed key management scheme, which uses the hypothetical interpretation method to think that L-sensors can directly know the identity identifier of their neighbour nodes, but this assumption does not match the random deployment situation. Each H-sensor is pre-loaded with K_H^R and K_H. Each L-sensor (say u) only is pre-loaded with K_{ID_u}. However, according to Wang et al.'s scheme description, the third-party has the ability to decrypt data exchanged between nodes. We can infer that both the H-sensors and the L-sensors are pre-loaded with the same master key s. In Wang et al.'s distributed key management scheme, the total number of correct preloaded keys is:

$$M\times 3+N\times 2=3M+2N \tag{3}$$

Our centralised and distributed key management schemes encrypt location information of L-sensors and digitally signs the key request messages uploaded by L-sensors. In addition, the total number of keys preloaded by our centralised and distributed key management schemes is the same. Each H-sensor is pre-loaded with K_H^R, s and K_H. Each L-sensor (say u) is pre-loaded with K_H^U and K_{ID_u}. In our centralised and distributed key management schemes, the total number of preloaded keys is:

$$M\times 3+N\times 2=3M+2N \tag{4}$$

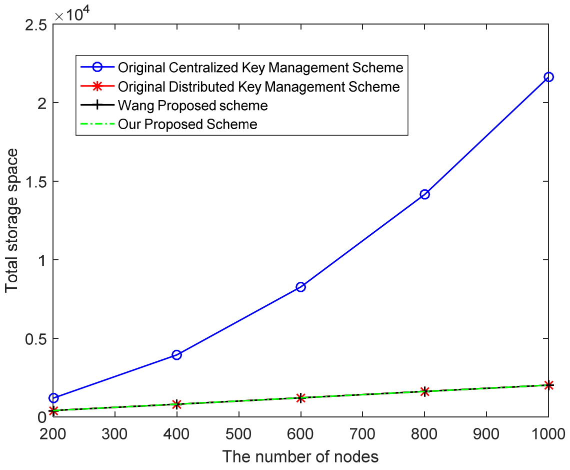
It should be noted that Wang et al.'s scheme is very dangerous for protecting the location information of the L-sensors by encryption. When one L-sensor is captured, the private keys of all L-sensors are exposed, even the private keys of all H-sensors. Table 1 shows the configuration of nodes in HSN.

Table 1 The configuration of nodes in HSN

Node type	*1*	*2*	*3*	*4*	*5*
Sink	1	1	1	1	1
H-sensor	4	8	12	16	20
L-sensor	196	392	588	784	980

Figure 3 shows the comparison of the number of key preloads between schemes (our scheme, the original key management scheme and Wang et al.'s distributed key management scheme). The x-axis is the number of nodes, and the y-axis represents the total storage space (in terms of key length) required to preload the keys.

Figure 3 The comparison of the number of key preloads between schemes (see online version for colours)



As can be seen from Figure 3, the key storage space occupied by our centralised key management establishment is much lower than the key storage space occupied by the original centralised key management establishment. The key storage space occupied by our distributed key management establishment is the same as the key storage space occupied by the original distributed key management establishment and Wang et al.'s distributed key management scheme. However, our distributed key management scheme can protect the node location privacy information, private keys and routing structures when the number of preloaded keys is the same.

5.2 *Comparison of computational overhead during key establishment*

According to Figure 1, there are 80 nodes in the cluster. We use the MATLAB software tool to calculate the computational overhead of the cluster's L-sensors key establishment process. L-sensor type is MICAz, MICAz integrates 8-bit, 8MHz, ATmega128L processor, working voltage is 3 V, working current is 8 mA, which executes once Tate pairing operation for 2.66 s and consumes

63.84 mJ, and executes once elliptic curve point multiplication operation for 0.81 s and consumes 19.44 mJ. When sending and receiving receipts, sending one byte consumes 0.052 mJ, and receiving one byte consumes 0.019 mJ. H-sensor type is Imote2, Imote2 integrates 32-bit, 104MHz, PXA271 Xscale processor. Working voltage is 0.95 V. Working current is 66 mA. Imote2 executes once Tate pairing operation for 0.06 s and consumes 3.762 mJ. Imote2 executes once elliptic curve point multiplication operation for 0.012 s and consumes 0.752 mJ. Imote2 sends or receives one byte consumes approximately 0.002 mJ. We assume that the length of the node ID is 2 bytes, the length of the location information is 4 bytes, and the length of the timestamp is 2 bytes. In addition, they (the hash function output, the encrypted cipher text, the digital signature of the message, and the routing structure information) are the same length. The length is 20 bytes. We only consider the computational overhead of the key establishment process, ignoring the computational overhead of the communication process.

Table 2 Energy consumption of each scheme for completing key establishment in a cluster

Scheme type	*Energy consumption* (mJ)
Original centralised key management scheme	1735.04
Our centralised key management scheme	10989.12
Original distributed key management scheme	6460.48
Wang et al.’s distributed key management scheme	12623.976
Our distributed key management scheme	25155.336

As shown in Table 2, compared to other schemes, our scheme consumes relatively more energy because we have taken more protection measures. We use the sacrificed energy to exchange scheme privacy (the location information and the private keys) protection.

5.3 Security analysis

Firstly, when L-sensors in the cluster upload the key request messages to H-sensor, we protect the location information, which avoids the attacker obtaining the routing structure information or directly captures L-sensors using their location information. Secondly, in our distributed key management scheme, we double-encrypt the location information of the L-sensors to prevent the private keys of the neighbour nodes from leaking and threatening its location privacy information. At the same time, when H-sensor distributes the private keys of the neighbour nodes to L-sensors, we use the method of performing digital signature after encryption to complete the authentication of the message source and the protection of private keys of the neighbour nodes. Finally, H-sensors can determine which L-sensors need to change the communication neighbour nodes according to the remaining energy values uploaded by L-sensors, thus preventing the attacker from initiating sinkhole attacks. When forwarding messages between nodes, we add timestamps to prevent attacker from initiating resend attacks.

6 Conclusion

In this paper, we present a key management scheme realising location privacy protection for heterogeneous WSNs. Firstly, the number of keys preloaded by our centralised key management scheme is lower than the original centralised key management scheme. Then, we use the sacrificed energy to exchange the protection of the L-sensors location information and the private keys of the neighbour nodes to prevent the attacker from initiating a targeted attack or directly utilising the node’s location information to capture nodes. Finally, in our scheme, H-sensors can determine which L-sensors need to change the communication neighbour nodes according to the remaining energy values uploaded by L-sensors, thus preventing the attacker from initiating sinkhole attacks. At the same time, when forwarding messages between nodes, we add timestamps to prevent the attacker from initiating resend attacks. In future work, we will examine the issues involved in key management under the mobile WSN model.

Acknowledgement

This work was supported in part by the National Science Foundation of China under Grant 61771416, the Creative Research Groups of Higher Education of Xinjiang Uygur Autonomous Region under Grant XJEDU2017T002, the CERNET Innovation Project (No. NGII20180320 and No. NGII20180201).

References

Chan, H., Perrig, A. and Song, D. (2003) ‘Random key predistribution schemes for sensor networks’, *2003 Symposium on Security and Privacy*, Vol. 2003, January, pp.197–213.

Cheng, Y. and Agrawal, D.P. (2005) ‘Efficient pairwise key establishment and management in static wireless sensor networks’, *2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, Vol. 2005, pp.544–550.

Chien, H.Y., Chen, R.C. and Shen, A. (2008) ‘Efficient key pre-distribution for sensor nodes with strong connectivity and low storage space’, *22nd International Conference on Advanced Information Networking and Applications (AINA 2008)*, pp.327–333.

Du, X., Guizani, M., Xiao, Y. and Chen, H.H. (2009) ‘Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks’, *IEEE Transactions on Wireless Communications*, Vol. 8, No. 3, pp.1223–1229.

Du, X. and Lin, F. (2005) ‘Maintaining differentiated coverage in heterogeneous sensor networks’, *Eurasip Journal on Wireless Communications and Networking*, Vol. 2005, No. 4, pp.565–572.

Du, X., Xiao, Y., Ci, S. and Guizani, M. (2007) 'A routing-driven key management scheme for heterogeneous sensor networks', *IEEE International Conference on Communications,* pp.3407–3412.

Eschenauer, L. and Gligor, V.D. (2002) 'A key-management scheme for distributed sensor networks', *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ACM, Washington DC, USA, pp.41–47.

Gandino, F., Ferrero, R., Montrucchio, B. and Rebaudengo, M. (2016) 'Fast hierarchical key management scheme with transitory master key for wireless sensor networks', *IEEE Internet of Things Journal*, Vol. 3, No. 6, pp.1334–1345.

Gura, N., Patel, A., Wander, A., Eberle, H. and Shantz, S.C. (2004) 'Comparing elliptic curve cryptography and RSA on 8-bit cpus', *Cryptographic Hardware and Embedded Systems*, Vol. 3156, pp.119–132.

Karp, B. (2000) 'GPSR: greedy perimeter stateless routing for wireless sensor networks', *Proc. 6th Annual International Conference on Mobile Computing and Networking*, pp.243–254.

Lazos, L. and Poovendran, R. (2004) 'SeRLoc: secure range-independent localization for wireless sensor networks', *Proceedings of the 2004 ACM Workshop on Wireless Security, WiSe*, pp.21–30.

Mathew, A. and Terence, J.S. (2017) 'A survey on various detection techniques of sinkhole attacks', *Proceedings, WSN. of the 2017 IEEE International Conference on Communication and Signal Processing (ICCSP)*, pp.1115–1119.

Maurer, U. and Wolf, S. (1998) 'Diffie-Hellman, decision Diffie-Hellman, and discrete logarithms', *IEEE International Symposium on Information Theory*, p.327.

Miao, X., Han, G., He, Y. and Wang, H. (2018) 'A protecting source-location privacy scheme for wireless sensor networks', *2018 IEEE International Conference on Networking, Architecture and Storage* (*NAS*), IEEE, pp.1–5.

Rivest, R.L., Hellman, M., Anderson, J.C. and Lyons, J.W. (1992) 'Responses to NIST's proposal', *Communications of the ACM*, Vol. 35, No. 7, pp.41–54.

Shamir, A. (1984) 'Identity-based cryptosystems and signature schemes', *Annual International Cryptology Conference*, Vol. 196, LNCS, pp.47–53.

Singh, J.P., Roy, P.K., Singh, S.K. and Kumar, P. (2016) 'Source location privacy using data mules in wireless sensor networks', *2016 IEEE Region 10 Conference (TENCON)*, Vol. 0, pp.2743–2747.

Szczechowiak, P., Scott, M. and Collier, M. (2010) 'Securing wireless sensor networks: an identity-based cryptography approach', *International Journal of Sensor Networks*, Vol. 8, Nos. 3–4, pp.182–192.

Wang, J. and Wang, H. (2014) 'Distributed key management scheme based on ECC for heterogeneous sensor networks', *2014 2nd International Conference on Advanced Cloud and Big Data (CBD)*, pp.235–239.

Xiao, Y., Rayi, V.K., Sun, B., Du, X., Hu, F. and Galloway, M. (2007) 'A survey of key management schemes in wireless sensor networks', *Computer Communications*, Vol. 30, No. 11-12, pp.2314–2341.

Yacobi, Y. (2002) 'A note on the bilinear Diffie-Hellman assumption', *IACR Cryptology ePrint Archive*, Vol. 2002, p.113.

Yu, H. and Wang, L. (2017) 'A security-enhanced mutual authentication scheme with privacy protected in wireless sensor networks', *Cluster Computing*, pp.1–11.

Yu, L., Wang, N., Zhang, W. and Zheng, C. (2007) 'Deploying a heterogeneous wireless sensor network', *2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM. (2007*), pp.2588–2591.

Zhou, L., Wen, Q. and Zhang, H. (2013) 'Protecting sensor location privacy against adversaries in wireless sensor networks', *2013 International Conference on Computational and Information Sciences*, pp.1384–1387.