# An Asymmetric Encryption-Based Key Distribution Method for Wireless Sensor Networks

Yuan Cheng *, Yanan Liu, Zheng Zhang, Yanxiu Li

Posted Date: 26 June 2023

doi: 10.20944/preprints202306.1731.v1

Keywords: WSN; Security; Key distribution; Cryptography

*Article*

# An Asymmetric Encryption-Based Key Distribution Method for Wireless Sensor Networks

**Yuan Cheng [1], Yanan Liu [1], Zheng Zhang [1] and Yanxiu Li [1]**

[1] School of Network Security, Jinling Institute of Technology; chengyuan2018@jit.edu.cn
* Correspondence: chengyuan2018@jit.edu.cn; Tel.: +86 18168092939

**Abstract:** Wireless sensor networks are usually applied in hostile areas where nodes are easy to be monitored and captured by adversary. Key distribution is an essential primitive to provide most of security mechanism. However, the characteristic of limited resources of sensors restricts the direct use of conventional key distribution schemes. In this paper, a complete security key distribution scheme based on asymmetric cryptography technology is proposed in both static and mobile scenarios. Mutual authentication is guaranteed using challenge-response mechanism. The performance evaluation and security analysis show that the proposed scheme with low complexity not only provides better security for wireless sensor networks, but also reduces storage overhead and key exposure risks.

**Keywords:** WSN; security; key distribution; cryptography

## 1. Introduction

Wireless sensor networks (WSNs) have been proven to be suitable for large numbers of applications ranging from industry and security domains such as environment monitoring, fire detection and precision agriculture, to personal use like health supervision. WSNs are composed of a large number of sensors that work independently of each other. These sensors transmit routing information to each other and forward collected application data [1,2]. The major weakness of wireless sensor networks lies in the limitations of resource including memory, battery capacity, data processing and communication capabilities. Sensors and wireless channels are vulnerable to eavesdropping, physical interception, malicious attacks, message tampering, identity impersonation and side channel attacks [3–5]. Therefore, one of the focuses of wireless sensor networks is how to provide high confidentiality for the transmitted application data and control messages to prevent various illegal attacks [6–9]. At present, it is generally believed that encryption is a key technology that can provide confidentiality between the cloud and the end [10–12], which can also be used in WSNs data exchange.

In this paper, we present a security key management scheme for cluster based wireless sensor networks. In our scheme, session keys can be safely distributed and updated among all sensors with the help of base station. Both static and dynamic conditions are studied over the hierarchical networks. In particular, the efficient encrypting algorithm makes it possible to adopt asymmetric encryption in our proposed scheme to guaranty authentication and confidentiality during data transmission.

The rest of our paper is organized as follows. Section II introduces security features and design constraints in WSNs. Section III exhibits the details of the security key management scheme. Section IV evaluates the performance of the proposed security protocols. Finally, conclusion and perspectives are presented in Section V.

 doi:10.20944/preprints202306.1731.v1



## 2. Design Constraints and Security Issues in WSNs

### *2.1. Physical Characteristics and Constraints*

Sensors in most of wireless sensor networks are greatly limited in terms of device size, battery capacity, computing capacity, communication capacity, storage capacity, which makes the development of applications a challenge. A feasible and efficient security protocol should minimize the number of operations needed for calculation, communication, and storage. Therefore, the following characteristics of a WSN should be taken into consideration of protocol design [13,14].

- Limited battery capacity—Sensor networks are usually deployed in outdoor environment. Due to the size limitation, each sensor is usually equipped with a small battery. As a result, a sensor is unable to calculate and communicate when the battery runs out.
- Limited memory—The cache size of a sensor is usually tens of megabytes, which puts forward higher requirements for the length and number of keys stored.
- Limited bandwidth—Due to power limitation, most sensors use narrowband signal transmission, and the transmission rate generally does not exceed 10KB/s.
- Limited calculation power—In order to reduce the power consumption of CPU, most sensor nodes only use 8-bit 4-MHz microcontrollers.
- Good scalability—Wireless sensor network must allow new legal nodes to join the existing network at any time. At the same time, the failure of any node will not affect the normal operation of the network.
- Variability of network topology—Since sensors are often installed on mobile devices, the topology of wireless sensor networks often changes as well. Thus, network stability and nodes connectivity should be ensured in all protocols design.
- Environment—Some wireless sensor networks are expected to be used for remote control and reconnaissance, and are deployed in insecure and unstable environments, which makes them subject to many attacks, such as spoofing attacks, physical damage and any other mechanical failure associated with environmental factors.

### *2.2. Security Issues in WSNs*

In addition to the above characteristics of wireless sensor networks, security is also an important part of the Internet of things. Since WSNs use wireless medium for data transmission, sensors are more vulnerable to various malicious attacks based on wireless channels. The typical malicious attacks in WSNs include eavesdropping, data modification, sink hole, spoofing attacks, denial of service attacks, sybil attacks and node capture. Take node capture as an example. The attacker accesses the hardware and software of one or more sensors through the network [19]. After successful intrusion into the sensor, the attacker steals all the cryptographic keys and algorithms. Thus, it is possible for the attackers to eavesdrop and tamper with messages, even pretend to be legal terminals to forward data to hackers.

In recent years, a lot of research work has focused on security problems in WSNs. An asymmetric key predistribution scheme called AP was firstly proposed for hierarchical sensor networks in [15]. The famous "probabilistic" schemes had low computational complexity and communication load. However, this scheme can not guarantee accurate sharing of pairwise keys between any two sensors. Based on Blom matrix, a key management scheme is proposed by Boujelben in [16] to improve the resilience against node capture. However, complex matrix operation means high resource consumption for ordinary sensors. Lee presented a key renewal approach for authentication based on modular exponentiation in clustered WSNs [17]. Although this scheme improved the connectivity of the network, public key-based encryption brought about large amount of computation. Tian presented a blockchain based trusted key management approach in [18], which realizes key management in WSNs through a secure cluster formation algorithm and a node mobility algorithm. In literature [20], a novel key management model for hierarchical sensor networks based on public

 doi:10.20944/preprints202306.1731.v1



key infrastructure (PKI) was proposed. However, the key distribution issues in case of movement were not investigated.

### *2.3. Aasymmetric Cryptography in WSNs*

Asymmetric encryption uses key pairs to encrypt and decrypt data for both sides of communication. Any message encrypted with the public key can only be decrypted by one with the private key. The private key is secretly held by its holder, and the public key can be obtained by the required communication entity through public channel. Asymmetric cryptography can provide confidentiality, integrity, and authentication for different kinds of networks. Although information encryption based on asymmetric key has been proved to be applicable to sensor networks, its application is still limited by its complex computation. Furthermore, taking the actual sensor chip as an example, the time taken for asymmetric encryption is still in the order of seconds, which may not be suitable for those applications with strict real-time performance.

Fortunately, in recent years, the new cryptographic algorithms have showed great energy efficiency and reached the same security level as traditional algorithms. For example, the Elliptic Curve Cryptography (ECC) [21] is the representative one of those algorithms. From the perspective of energy consumption and computational complexity, ECC is promising to be used for data encryption in WSNs. It provides comparative security with a smaller key, which also reduces energy of computation and communication in WSNs. Based on it, a new security key management scheme and an authentication approach are proposed in Section 3.

## 3. The Key Management Scheme for Cluster based WSNs

In this part, a security key management scheme for wireless sensor networks based on public key cryptography is presented. To avoid long term attack through which attackers can analyze the encrypted traffic over the network for a long time, a key update approach is specifically designed.

### *3.1. Network Model and Assumptions*

At present, wireless sensor networks commonly used in the industry mainly include two kinds of architectures, namely hierarchical structure and flat structure. A hierarchical architecture is usually used for large-scale WSNs due to its good scalability. A clustered hierarchical network is composed of base stations (BS), a large number of sensor nodes and a small number of cluster head (CH). BS is not limited by resources. The base station is responsible for managing all nodes of the network and receiving the service data collected by the sensor nodes. It is assumed that the cluster head has a higher configuration than the sensors, including battery capacity, memory size, communication and computing capacity. Like the gateway, the cluster head assists in data transmission between the sensors and the base station. In the hierarchical architecture, sensors are divided into non overlapping clusters, which collect data from the surrounding environment and send the original data to the base station. In this article, we focus on hierarchical architecture of WSNs.

In our scheme, asymmetric encryption is used to realize the authentication between the base station, the CHs and the sensor nodes. The public key is pre-loaded into each sensor before network deployment. With the public key system, the proposed scheme not only realizes end-to-end identity authentication, but also provides security for subsequent key distribution process.

In our hierarchical WSN model, we make a few assumptions as follows:

- The base station has more energy power for calculations and communications compared to sensors.
- The base station owns a pair of keys (a public key and a private key).
- The network is divided into several cluster region. In each cluster, there is only one cluster head node, and its location remains unchanged. Each cluster head can be recognized as the gateway of its cluster.
- In terms of security and ease of management, each cluster generates different session key for dialogs between sensor nodes and cluster head.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



- Both asymmetric and symmetric cryptography are used for each sensor. The former provides mutual authentication and key distribution, the latter ensures the confidentiality of traffic transmitted.
- As an optional technology in our scheme, MAC (message authentication code) provides data integrity.
- Public key is preloaded into each sensor and cluster head by an off-line dealer.
- Each sensor can store at least one public key and several session keys in its memory.
- Each sensor can randomly move among different clusters with a low speed.

### *3.2. Network Initializtion and Definitions*

In the network, there are $n$ sensors, denoted as $S_{0,\ldots,n-1}$, and $m$ cluster heads (CH), denoted as $CH_{0,\ldots,m-1}$. Each sensor has a unique identification code *ID_si* with length of 2 bytes stored in the chip. When the network is initialized, sensor nodes are randomly distributed into $m$ groups or cluster. There is only one CH and $n/m$ sensors in each cluster. Figure 1 shows a typical network of three clusters. Each cluster contains one CH and three sensors.

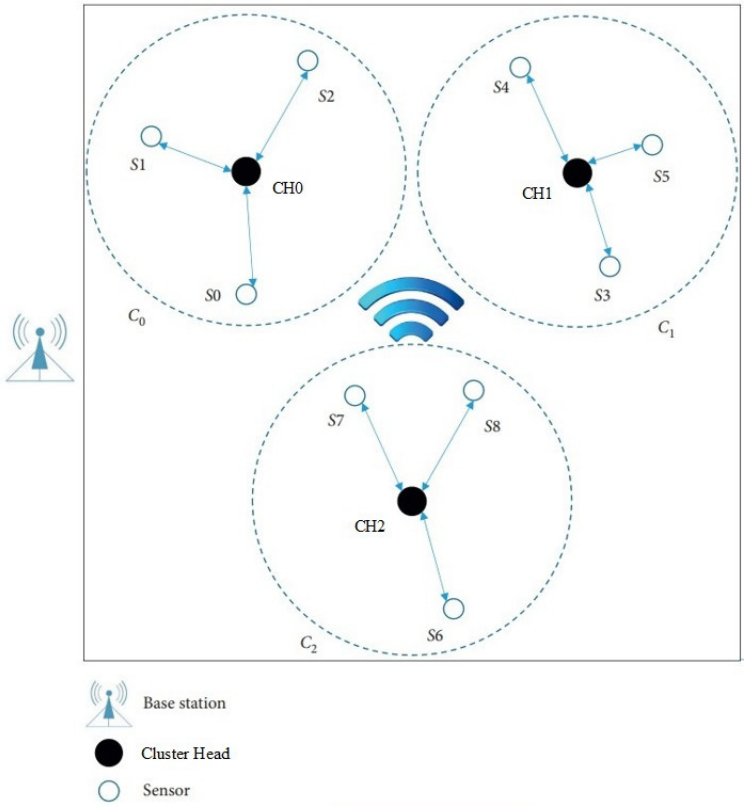


**Figure 1.** The network topology.

After network deployment, each CH runs a cluster forming process, and sensors are divided into clusters with no cross coverage. After a period of operation, some sensor may move into another cluster's region. In this situation, the subsequent key distribution and update process will be done by the CH of the present cluster. In the following part, we will describe the scheme from two aspects, that is static sensors and mobile sensors.

The following definitions will be used in our scheme and analysis.

$SK_i$ denotes the symmetric session key with a length of 128bits shared by the base station and sensors located in $DG_i$.

*PUK* denotes the public key of the BS and *PVK* denotes the corresponding private key. *PUK* can be obtained through public key Infrastructure (PKI).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



The function $E(x,y)$ denotes encryption (symmetric or asymmetric) operation, parameter $x$ denotes encryption key and parameter $y$ denotes the plain message need to be encrypted. The function $D(x,y)$ denotes decryption operation.

$ID\_CH_i$ denotes the identity code of the cluster with a length of 1 byte, and it can be acquired from the CH of that cluster. It is stored in the chip of each CH and tamper proof mechanism is used.

$ID\_si$ denotes the identity code of sensor $S_i$ with a maximum length of 2 bytes. It is stored in the chip of each sensor and tamper proof mechanism is used.

### *3.3. Static Sensors Subscheme for Hierarchical WSNs*

#### 3.3.1. Mutual Authentication and Key Distribution Process

In our clustered architecture network, the CH plays an important role in the process of key management. The key problem here is how to distribute the key among the sensor nodes under many restrictions. We assume all the sensors are static and present the operations of handshake, key distribution, authentication, and key update. The handshake is destined to establish a symmetric key shared by sensors and BS. The operation of handshake includes three steps:

1. **Generation of the $SK_i$:** The $CH_i$ generates a random symmetric key $SK_i$ and a challenge $R$. Then, the $CH_i$ encrypts $SK_i$ , $R$ and $ID\_CH_i$ with $PUK$, and we get:

$$\text{Cipher1}= E(PUK, SK_i \parallel R \parallel ID\_CH_i \parallel timestamp) \quad (1)$$

   The 2-byte timestamp is used to resist replay attacks. $CH_i$ sends Cipher1 to the base station using traditional routing. Here, the $PUK$ is used for authentication and confidentiality of the session key $SK_i$.
2. **Establishment of $SK_i$:** After receiving and decrypting the message, the base station gets $SK_i$, $R$ using its $PVK$ and builds a global table of all the session keys of different clusters. This table is used to identify the cluster and its cluster head on the network. Meanwhile, if $ID\_CH_i$ can be found in the database of legal CHs, the identity of the $CH_i$ can be authenticated by BS.
3. **Completion of the handshake:** The base station encrypts $R$ with the established session key $SK_i$. and gets

$$\text{Cipher2}= E(SK_i, R) \quad (2)$$

Then, the base station sends Cipher2 to $CH_i$, and $CH_i$ decrypts it. When the challenge $R$ is correctly received, a session key is successfully established between BS and $CH_i$. Otherwise, $CH_i$ will reinitiate the handshake. Considering the resource consumption caused by the computational complexity, the message authentication code (MAC) is not added in key distribution scheme.

Through the above steps, the mutual authentication between the base station and $CH_i$ is completed. After that, each sensor in the cluster needs to achieve the session key $SK_i$ generated by $CH_i$. Thus, sensor node $S_i$ builds a message encrypted by the $PUK$, denoted as follows:

$$\text{Cipher3}= E(PUK, ID\_CH_i \parallel ID\_si \parallel timestamp \parallel SK\_si \parallel R) \quad (3)$$

where $SK\_si$ is a symmetric key generated by sensor $S_i$. For sensor $S_i$, the Cipher3 is used to apply for session key and identity authentication at the same time.

When the BS receives Cipher3, it picks out the corresponding session key ($SK_i$) according to $ID\_CH_i$. At the same time, if the $ID\_si$ can be found in the list of legal sensor nodes, the authentication of $S_i$ is accomplished as well.

To secure the session key, the base station encrypts $SK_i$ with the session key $SK\_si$ and builds the Cipher4 as follows:

$$\text{Cipher4}= E(SK\_si, SK_i \parallel R). \quad (4)$$

Then, the Cipher4 is sent to $S_i$, and $S_i$ will decrypt it by the symmetric key $SK\_si$. Finally, all the sensors in the same cluster share the same session key $SK_i$ with its cluster head. Through the above key distribution subscheme, the confidentiality of traffic between the cluster head and the sensor is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



guaranteed. Besides, mutual authentication between the BS and $S_i$ is successfully done. The detailed key distribution process is depicted in Figure 2.

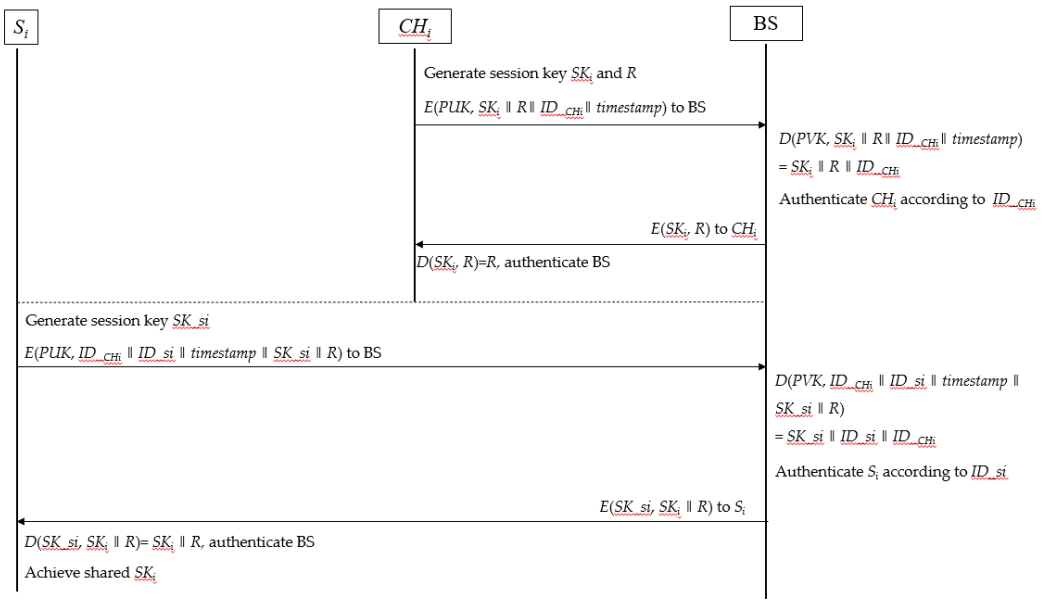


**Figure 2.** The process of authentication and key distribution in static subscheme.

3.3.2. Session Key Update Process

To protect the nodes from long term attacks, a periodic key update mechanism is designed. The steps of the key update are given as follows.

1. The new session key $SK_i'$ is generated by the cluster head $CH_i$ at a certain moment.
2. $CH_i$ notifies the base station to update the session key.
3. Using the proposed handshake operation, the new session key $SK_i'$ is distributed between the BS and the $CH_i$. After that, the $CH_i$ notifies all the sensors to update their session key in its cluster with a broadcasting message. Sensors will stop encrypting sessions until they receive new session key $SK_i'$.
4. After the establishment of $SK_i'$, the $CH_i$ distributes $SK_i'$ encrypted with the original session key $SK_i$ to all the sensors by broadcasting cipher5, denoted as

$$\text{Cipher5} = E(SK_i, SK_i'). \quad (5)$$

5. Each sensor in the cluster decrypts the cipher5 with the old session key $SK_i$ and substitutes with the $SK_i'$. The subsequent dialog is decrypted by the new session key.

*3.4. Mobile Sensors Subscheme for Hierarchical WSNs*

3.4.1. Mutual Authentication and Key Distribution Process

Since sensor nodes have a high probability of moving between different clusters of the network, the dynamic subscheme for hierarchical architecture is more complicated. In Figure 3, $S_0$ moves from the cluster $C_0$ into another cluster named $C_2$. Because the location of each CH is assumed to be unchanged, the process of authentication and key distribution between CH and BS is the same as the static subscheme. The main difference between the static subscheme and the mobile one lies in the key distribution process.

The key distribution process of the mobile scene includes six steps.

1. When $S_0$ moves into *cluster2*, it will send a cluster-entry request to $CH_2$. The cluster forming and cluster head detection process is not described here, please refer to [24].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



2. $CH_2$ detects and receives this message. Then, $CH_2$ replies to $S_0$ with a message including its identification code $ID\_{CH2}$.
3. $S_0$ updates the identification of the present cluster, replacing $ID\_{CH0}$ with $ID\_{CH2}$.
4. $S_0$ applies for the latest session key $SK_2$ from the base station with the cipher6 denoted as follows:

$$\text{Cipher6}= E(PUK, ID\_{CH2} \| ID\_{S0} \| timestamp \| SK\_{S0} \| R) \quad (6)$$

5. The BS decrypts cipher6 with the $PVK$ and gets $ID\_{CH2}$, $SK\_{S0}$, and $ID\_{S0}$ from
Plain6 = $D(PVK, \text{Cipher6})= D(PVK, E(PUK, ID\_{CH2} \| ID\_{S0} \| timestamp \| SK\_{S0} \| R)) = ID\_{CH2} \| ID\_{S0} \| SK\_{S0} \| R$.
The latest session key $SK_2$ can be picked out in terms of $ID\_{CH2}$, and the $S_0$ is authenticated by BS according to $ID\_{S0}$. Then, the cipher7 will be sent to $S_0$. The cipher7 is built as follows:

$$\text{Cipher7}= E(SK\_{S0}, SK_2 \| R). \quad (7)$$

6. $S_0$ decrypts the cipher7 with the symmetric key $SK\_{S0}$ and successfully gets $SK_2$.

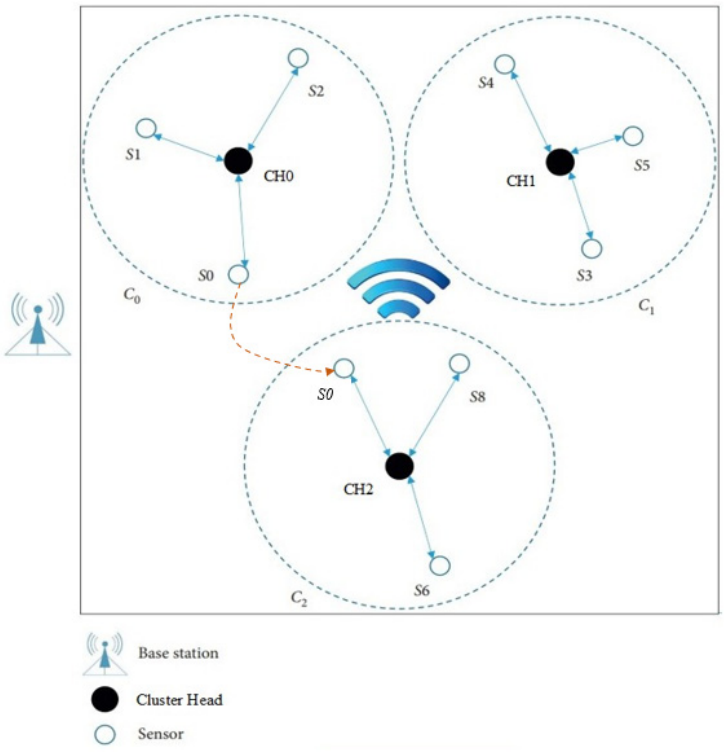


**Figure 3.** Sensor $S_0$ moves from *Cluster0* to *Cluster2*.

Thus, the mobile sensor is able to achieve the latest session key of the present cluster and send encrypted traffic to the corresponding cluster head. The detailed key distribution process in mobile subscheme is depicted in Figure 4.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



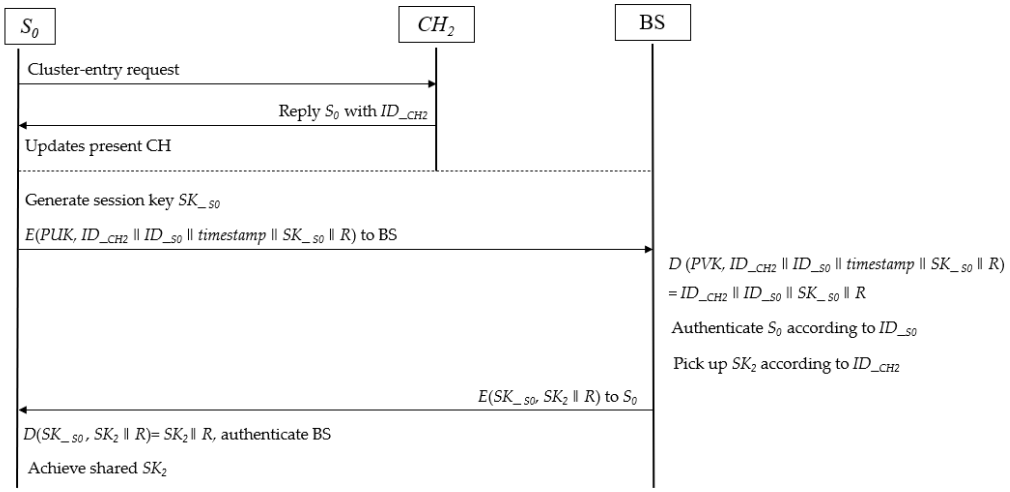


**Figure 4.** The process of authentication and key distribution in mobile subscheme.

3.4.2. Session Key Update Process

In mobile scene, a sensor updates session key like the static model for most of time. However, when $S_0$ moves to the junction of two adjacent clusters, for example $C_0$ and $C_2$ in Fig.2, it may receive key update messages from $CH_0$ and $CH_2$ at the same time. It should be noted that $S_0$ only knows the previous session key $SK_0$ of *cluster0*, but unaware of the previous session key of *cluster2*. Thus, $S_0$ can only decrypt the broadcasting message from $CH_0$ to update $SK_0'$. After joining *cluster2*, $S_0$ can obtain the present session key $SK_2$ from the base station and wait for key updating again.

## 4. Analysis and Comparison

Extensive simulations are provided to verify the performance of our scheme, such as memory consumption, communication overhead, connectivity, recovery capability for node capture. Then, we compare the proposed key management scheme with other schemes from multiple dimensions. We evaluate the performance based on NS-2 [27]. In the performance evaluation, 100 nodes are distributed randomly in a 10 thousand square meters area. Each sensor has a fixed speed of 2 m/s. Each cluster head is in the center of each 10 × 10 m area and its coverage radius is 20m.

### *4.1. Key Storage of Sensor Nodes*

In our scheme, the public key is preload into sensor's memory during the network initialization. Since the strength of encryption with the 256 bits ECC key is equal to that of the 3072 bits RSA key, a public key of 256-bit length is used in our simulation. Besides, two 32-byte session keys are used in the key distribution process. When a sensor receives the refreshed session key, the original will be deleted to save the memory. Therefore, the memory overhead is only 96 bytes and that of the CH is also 96 bytes.

The key distribution in [15] is that $k$ keys are preloaded into each sensor, while $m$ keys ($m \gg k$) are preloaded into each CH. If any two nodes share a pairing key, they can establish a secure link. Thus, the more keys stored, the higher probability of sharing common keys. In [23], the memory is divided into two parts. One is used to store $\alpha$ predistributed keys and the other is for $\beta$ post-deployment keys.

Table 1 presents the key storage overhead in different schemes. For large and medium-sized wireless sensor networks, sensors in our scheme require less storage space than others. But cluster heads require slightly more memory space than Erfani's scheme. Since the number of sensors is much larger than that of CHs, our scheme is valuable for resource limited WSNs.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



**Table 1.** Key storage overhead (bytes) in different schemes.

| | Du[15] | Erfani[23] | Our Scheme |
|---|---|---|---|
| Sensor | 32$l$ | 32($\alpha$+$\beta$) | 96 |
| Cluster Head | 32$m$ | 32 | 96 |

*4.2. Communication Overhead*

The communication overhead in our analysis only considers the payload related to key distribution and key update and does not include the IP packet encapsulation of the network layer.

In the static scheme, a CH usually sends one packet of 128 bytes during initialization handshake, while sends two packets, totally 256 bytes, each time the key is updated. Each sensor only needs to send one packet of 128 bytes in the initialization phase. In the mobile scheme, the communication overhead of CH and Sensor are the same as that of static scheme. If the update frequency of the session key is high, the communication load will increase. That means there is a trade-off between the security of the session key and the communication load.

*4.3. Security Analysis*

4.3.1. Mutual Authentication

In both subschemes, mutual authentication of BS and sensors (including CHs) is assured by the challenge-response mechanism. Terminals without legal identifier ($ID\_{CHi}$ or $ID\_si$) cannot pass the identity authentication. Since the identifier is stored in the chip of each sensor with tamper proof mechanism and it is encrypted for transmission, the confidential and integrity can be guaranteed. Only the authenticated sensors can participate in the wireless sensor network and get the session key for data encryption.

4.3.2. Security Connectivity

The security connectivity is defined as the probability that two nodes successfully establish a session key. Since authentication and key distribution in our proposal are cluster based, we define "inter-cluster connectivity" as the probability that a CH shares a pairwise key with the sensors in its cluster.

In our deterministic key distribution scheme, each authenticated sensor can always successfully share a session key with present cluster head. Compared with the probabilistic key distribution approaches in [15,16,25], the inter-cluster connectivity in our scheme is 100%. Those random schemes, like AP [15], can only achieve higher security connectivity by increasing the amount of key storage. Figure 5 depicts the secure connectivity versus key pool size in the AP. As the number of preloaded keys increases, the performance of the secure connectivity gradually improves. For fixed parameters [$l$, $M$], the security connectivity decreases significantly as the key pool increases.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



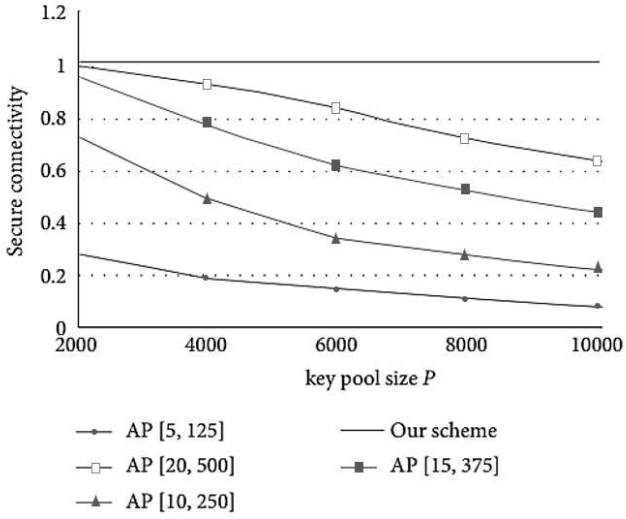


**Figure 5.** Secure connectivity versus key pool size *P*.

4.3.3. Resistance to Attacks

The new scheme provides a set of session keys to secure data exchange between the base station and sensors. Our proposal based on session key and public key can effectively resist common network attacks.

Eavesdropping can be avoided using symmetric encryption as well as the key update mechanism proposed in this article. Spoofing attack is avoided in our scheme through mutual authentication based on public key encryption. Besides, the authenticity of sensors is achieved by challenge-response mechanism and the identity code preloaded before deployment.

Attacks like modification, reply and insertion can be resisted by symmetric encryption and message authentication code added to each message. Only those authenticated nodes can send or modify data packets on the network.

Attackers obtain the secret information by capturing nodes or other physical means. We define resilience against node capture as the probability $F(x)$ that attackers obtain the key from uncaptured node according to a certain number of captured nodes $x$. So we get

$$F(x) = \frac{\text{number of compromised links between uncaptured nodes}}{\text{number of uncompromised links}} \quad (8)$$

Resilience against sensor capture is evaluated firstly. Different from the random key predistribution schemes in [10,11,26], sensors only need to preload a public key in our approach, which saves memory of sensor node. Due to the periodical key update applied, it is too hard for attackers to get the constantly updated session key despite capturing a sensor physically in our proposal. Thus, the probability of resilience against node capture $F(x_s)= 0$, where $x_s$ represents the number of captured sensor nodes. As shown in Figure 6, the resilience performance gets worse with the increasing number of captured nodes for random key predistribution schemes, because of the storage of a large number of session keys. Since the sensors store matrixes instead of keys, the resilience performance of Boujelben's scheme [16] is better than the AP scheme [15]. Simulation results indicates that threat of sensor capture is perfectly eliminated by our scheme.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 June 2023 doi:10.20944/preprints202306.1731.v1



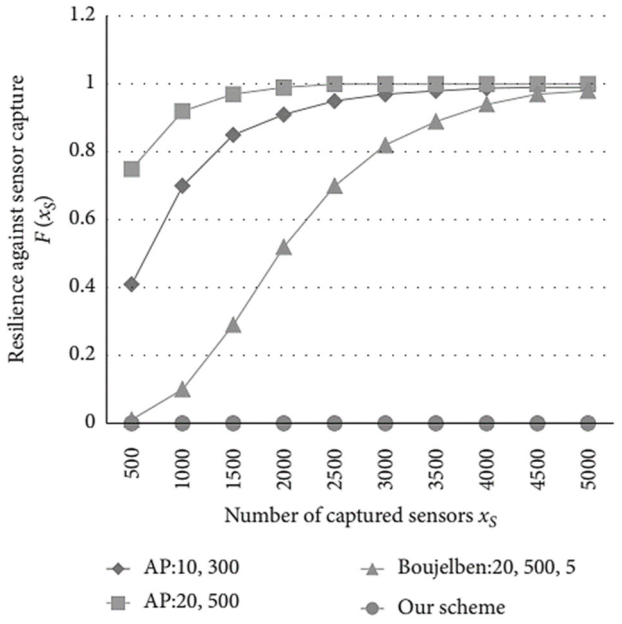


**Figure 6.** The probability of resilience against sensor capture in different schemes.

Finally, Table 2 presents several typical schemes of key management in WSN recent years. In our scheme, we provide a simple and feasible mutual authentication mechanism compared with [15,20,23]. Lee in [17] used an asymmetric encryption algorithm with more computation overhead than [20] and our proposal. Furthermore, our scheme outperforms other schemes in terms of resilience against node capture and resistant to eavesdropping.

**Table 2.** Comparisons of different key distribution solutions.

| Scheme features | Du[15] | Lee [17] | Benamar[20] | Erfani[23] | Our Scheme |
|---|---|---|---|---|---|
| Public key encryption | — | √ | √ | — | √ |
| Key predistribution | √ | x | √ | √ | √ |
| Mobility of sensors | — | x | x | √ | √ |
| Perfect resilience against node capture | x | — | — | x | √ |
| Mutual authentication | x | √ | x | x | √ |
| Resistant to eavesdropping attacks | — | — | √ | √ | √ |

—: Not involved.

## 5. Conclusions

In this article, a new key distribution scheme is designed for dynamic WSNs. From the security point of view, the proposed scheme provides good security for key distribution and update with relatively low communication overhead and memory consumption. Comprehensive comparisons are made between our scheme and other key pre-distribution schemes. Utilizing asymmetric cryptography technology, the proposed approach also reduces the risk of key exposure and provides excellent resistance to many types of attacks.

**Author Contributions:** Conceptualization, Y.Cheng and Yanan.Liu; methodology, Y.Cheng; software, Yanan.Liu; validation, Y.Cheng and Zheng.Zhang; formal analysis, Y.Cheng; investigation, Yanxiu.Li; resources, Y.Cheng and Yanan.Liu; data curation, Yanan.Liu; writing—original draft preparation, Y.Cheng; writing—review and editing, Y.Cheng; supervision, Zheng.Zhang. All authors have read and agreed to the published version of the manuscript.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Dludla, A.G.; Abu-Mahfouz, A.M.; Kruger, C.P.; Isaac, J.S. Wireless sensor networks testbed: ASNTbed. In Proceedings of the 2013 IEEE IST-Africa Conference and Exhibition (IST-Africa), Nairobi, Kenya, 29–31 May 2013; pp. 1–10.
2. Abu-Mahfouz, A.M.; Steyn, L.P.; Isaac, S.J.; Hancke, G.P. MultiLevel Infrastructure of Interconnected Testbeds of Large-Scale Wireless Sensor Networks (MI2T-WSN). In Proceedings of the International Conference on Wireless Networks (ICWN), Athens, Greece, 1–7 January 2012; p. 1.
3. D. Carman, P. Kruus, and B. Matt, Constraints and approaches for distributed sensor network security (final), pp. 1–139, NAI Labs Technical Report, NAI Labs, MD, USA, 2000.
4. Y. Ren, Y. Leng, J. Qi et al., "Multiple cloud storage mechanism based on blockchain in smart homes," Future Generation Computer Systems, vol. 115, pp. 304–313, 2021.
5. J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 922–933, 2021.
6. Aysal TC, Barner KE. Sensor data cryptography in wireless sensor networks. IEEE Transactions on Information Forensics and Security 2008; 3(2): 273–289.
7. Giruka VC, Singhal M, Royalty J, Varanasi S. Security in wireless sensor networks. Wireless Communications and Mobile Computing 2008; 8(1):1–24.
8. Kundur D, Luh W, Okorafor UN, Zourntos T. Security and privacy for distributed multimedia sensor networks. Proceedings of the IEEE 2008; 96(1): 112–130.
9. Wang Y, Attebury G, Ramamurthy B. A survey of security issues in wireless sensor networks. IEEE Communications Surveys & Tutorials 2006; 8(2): 2–23, Second Quarter.
10. G. Liu, Q. Yang, and H. Wang, "Trust assessment in online social networks," IEEE Transactions on Dependable and Secure Computing, vol. 2, pp. 994–1007, 2018.
11. C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 3, p. 1, 2021.
12. C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 4, p. 1, 2020.
13. Jun Zheng and Abbas Jamalipour, "Wireless Sensor Networks: A Networking Perspective", a book published by A John & Sons, Inc, and IEEEE, 2009.
14. Shio Kumar Singh, M P Singh, and D K Singh. "Routing Protocols in Wireless Sensor Networks – A Survey ". International Journal of Computer Science & Engineering Survey, Vol.1, No.2, 2010.
15. X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," Ad Hoc Networks, vol. 5, no. 1, pp. 24–34, 2007.
16. M. Boujelben, O. Cheikhrouhou, M. Abid, and H. Youssef, "Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks," in Proceedings of the 3rd International Conference on Sensor Technologies and Applications Athens, pp. 18–23, Athens, Greece, 2009.
17. S. Lee and K. Kim, "Key renewal scheme with sensor authentication under clustered wireless sensor networks," Electronics Letters, vol. 51, no. 4, pp. 368-369, 2015.
18. Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6193–6202, 2020.
19. N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," Proceedings of the Sixth Workshop on Cryptographic Hardware and Embedded Systems, pp. 119-132, 2004.
20. K. Benamar, F. Mohammed, and M. Abdellah, "Architecture aware key management scheme for wireless sensor networks," International Journal of Information Technology & Computer Science, vol. 4, no. 12, pp. 50–59, 2012.
21. Crossbow Technology Inc., Processor/Radio Modules, 2008. (http://www.xbow.com)

22. U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," ACM Transactions on Embedded Computing Systems, vol. 16, no. 3, pp. 1–25, 2017.
23. S. H. Erfani, H. H. S. Javadi, and A. M. Rahmani, "A dynamic key management scheme for dynamic wireless sensor networks," Security and Communication Networks, vol. 8, no. 6, pp. 1040–1049, 2015.
24. Y. Mohamed, Y. Moustafa, and A. Khaled, "Energy-aware management for cluster-based sensor networks," Computer Networks, vol. 43, no. 5, pp. 649–668, 2003.
25. L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," in Proceedings of the ACM Conference on Computer and Communication Security, pp. 41–47, Washington, DC, USA, November 2002.
26. C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks," Security and Communication Networks, vol. 7, 2014.
27. University of Southern California: "The network simulator – ns-2". Available at http://www.isi.edu/nsnam/ns/, September 2005.