A Hybrid Key Pre-distribution Scheme for Wireless Sensor Networks

Chao Xiong*, Shuquan Li, Lei Liu, Rui Li and Yanjiao Jin

School of Information and Software Engineering, University of Electronic Science and Technology of China Chengdu, China

*Email: xiong_c@163.com

Abstract. Key management in wireless sensor networks (WSNs) is the basic service for deploying security policies. In this paper, we combine q-composite scheme with polynomial scheme to propose a hybrid key pre-distribution scheme for WSNs. The characteristic of the scheme is that a partial polynomial is preloaded for each sensor node, and then a polynomial share stored by each node is used to generate a corresponding key pool. Perform the corresponding hash operation on the generated keys to hide part of the key information of the nodes. Then select the corresponding number of processed generation keys to distribute to each node. The scheme introduces a random key method, which avoids the “t-secure” problem faced by the conventional polynomial scheme and achieves the significant improvement of network security. The corresponding key is generated using the polynomial share preloaded on the sensor node, which has a higher connectivity than the polynomial scheme. Theoretical and simulation results show that the propose scheme not only enhances the network secure connectivity, but also improves the node’s anti-capture attack capability when compared to other schemes.

1. Introduction

A wireless sensor network is a vast number of battery-powered sensor nodes distributed in space using wireless communication technology. These nodes use the sensors they carry to sense the information in the environment[1], and then send the perceived information to the sink node for processing. WSNs are widely used in military and civilian applications, such as health care, military detection, forest fire monitoring, and seismic monitoring[2].

Sensor nodes are limited by various resources and complex deployment environments. Therefore, many security protection methods in traditional computers are difficult to directly use in WSNs. Security key management has become an important research topic in WSNs. The main problem of key management is that the communication keys are generated by neighbour nodes using a preloaded key or key factor according to certain rules[3]. Since the infrastructure of wireless sensor networks is difficult to improve, there is usually no trusted third party to establish a secure pairing key for neighbour nodes. The key pre-distribution scheme perfectly solves this problem[4]. The key management scheme design aiming at providing secure and reliable secure communication is the most basic and important content of WSNs security research[5];

In this paper, we exploit the use of the q-composite scheme[6], in conjunction with the polynomial pool scheme to establish a hybrid key pre-distribution scheme. The preload key in the sensor nodes are not composed of a single key but is composed of two parts. First, each sensor node is assigned a
corresponding polynomial shares, and then each node is assigned a generated key processed by a hash operation. This scheme provides a further compromise between security and node capture. The scheme introduces a random key method, avoiding the "t-secure" problem faced by the conventional polynomial scheme, and using the polynomial t-threshold attribute. The security of the traditional basic random key scheme is improved, and the network security is greatly improved. By generating a corresponding key using a polynomial share preloaded on the sensor node, it has a higher connectivity than a polynomial scheme.

The full text of this paper is organized as follows. In section 2, the related works of key pre-distribution schemes are introduced, and the corresponding scheme is briefly summarized. In the third section, the design of the hybrid key pre-distribution scheme proposed in this paper is analyzed. In section 4, we show the experimental results of the pre-distribution proposed scheme. Finally, we give our conclusions on the article in section 5.

2. Related worked
In this article, we focus on the key pre-distribution scheme of WSNs. E-G[7] first proposed a random key pre-distribution for WSNs by introducing random graph theory. The q-composite scheme can only establish a communication key with a neighbour node that has more than q shared keys. All the same keys generate a key \( K = \text{hash}(k_1 || k_2 |\cdots|| k_q) \) as the communication key between the two nodes. This scheme improves the anti-capture of the network compared to the E-G scheme. Liu et al.[8] combines the E-G scheme with Blundo's polynomial-based key scheme proposed a new scheme which has a large storage overhead, but has high security when attacking a few nodes. W.Bechkit et al.[9] used the one-way of the Hash function to improve the q-composite scheme, making it impossible to break through the rest of the chain through the backtracking mechanism. The results show that the program increases the resistance to node capture. E.A.M.Anita et al.[10] proposed a scheme combining q-composite and polynomial key setting in the key generation phase. A triple key is established between communication nodes in the network, thereby improving the flexibility of network to node capture. J.Zhang et al.[11] gave the hybrid key establishment. The node allocation method is not fixed. The keys of some nodes are a polynomial shares, and the key of some nodes are a key generated by using polynomial shares. This solution effectively improves the anti-capture of the network. P.Ahlawat et al.[12] proposed a novel hybrid method combining random key pre-distribution and path vulnerability matrix. Building a path vulnerability matrix, the main advantage of this matrix is to combine various standards to maximize attack efficiency and reduce the resource expenditure of the opponent. In recent research, more and more hybrid solutions have been proposed, which effectively combine the advantages of different schemes to improve the performance of WSNs.

3. Improved key management scheme
As with the schemes in[6], [8], we roughly divide our scheme into three phase described below. This section will describe our approach in detail, and analyze the design concept of this paper, and give a simple model as shown in Figure 1.

3.1. Key pre-distribution phase
This is the initialization phase of key management, which is performed offline by the Key Distribution Server (KDS) before network deployment. We roughly divide this stage into three steps.

- **Step 1**: The KDS generates a key pool with \( S_p \) symmetric binary t-degree polynomials \( f_k = \sum_{i,j=0}^{t} a_{ij}x^iy^j \) in the finite field \( GF(q) \) range, with \( f(x, y) = f(y, x) \), and then assigns a unique identifier \( Id_p \) to each polynomial. Then randomly select g polynomials from \( S_p \) to assign to the sensor nodes. Since q is a sufficiently large prime number, the finite field space is large
enough that the probability of any two of the produced $S_p$ polynomials containing identical coefficients is almost 0.

- Step 2: The KDS calculates the generated key $K = H(b_i || ID_u)$, which is derived from the shared polynomial in sensor $u$. Where $0 \leq i \leq t$, KDS generates $t+1$ keys for the sensor $u$, and the share $f(u, y)$ of the polynomial is the coefficient $b_i$ of $y^i$. Here we denote $f(u, y)$ as a univariate $t$-degree polynomial. The unique identifier for each generated key that is assigned is 3-tuple $(ID_p, i, ID_u)$. These keys generated by polynomials form a key pool for subsequent random distribution. For a generated key $K$, we use a hash function to preload the key in each node $(N \mod Z)$ times, where $N$ is the identifier of the node and $Z$ is the modulo parameter of the scheme. Then node $u$ uses the previously preloaded polynomial to generate the corresponding key roughly expressed as $key(u) = [H^{N \mod Z}(K_{u_1}), H^{N \mod Z}(K_{u_2}), \ldots, H^{N \mod Z}(K_{um})]$. 

- Step 3: The KDS randomly selects $r$ hash-prepared keys from the generated key pool, and loads the keys and their corresponding identifiers into the nodes. Here, the key identifier of $ld_p$ in the selected 3-tuple is different from the polynomial identifier loaded in the node.

3.2. Shared-key discovery phase

Any two nodes in the WSNs want to establish communication directly, first by the party that initiated the communication to broadcast information within its communication range. The information contains its own identity identifier, the key identifier of the polynomial, and the identity of the generated key. After receiving the sensor information, the other node will match and compare the sensor identity information of the node itself, and send the sensor identity information stored by itself to the other node, so that the two nodes will calculate the encrypted use when communicating. The process of establishing a communication key by two neighboring nodes is as follows:

- When the two neighbor nodes select a polynomial from the polynomial key pool with a common polynomial, we choose a polynomial with the same identity $ID_p$. Set the identifiers of two neighbor nodes to $u$, $v$, and use the same identifier in the later stages. Then the communication key of the two neighbor nodes is $K = f_i(ID_v, ID_u) = f_i(ID_v, ID_u)$.

- When two neighbor nodes do not have the same polynomial identification, but the same generated key identifier 3-tuple $(ID_p, i, ID_u)$. The two sensor nodes have a common key $K_{b_i, i, ID_u} = H(b_i || ID_u)$ that generates the key. The key generated by preloading in node $u$ is

![Figure 1. Improved key management scheme](image-url)
\[ H^{(ID_v \mod Z)}(K_{ld_p, ID_u}), \] and the key generated by preloading in node v is \[ H^{(ID_v \mod Z)}(K_{ld_p, ID_u}). \]

When \( (ID_u \mod Z) > (ID_v \mod Z) \), the communication key of the u and v is \( K_v = H^{(ID_v \mod Z)-(ID_u \mod Z)}(K_{ld_p, ID_u}) \). Otherwise, when \( (ID_u \mod Z) < (ID_v \mod Z) \), then the communication key of the u and v is \( K_v = H^{(ID_u \mod Z)-(ID_v \mod Z)}(K_{ld_p, ID_u}) \).

- Under the condition that the above two cases are not true, but the polynomial identity \( Id_p \) of one node is the same as the \( Id_v \) in generated key identifier in another nod, such as \( (Id_p, i, ID_u) \). These two sensor nodes can be calculated to obtain the same generated key \( K_{ld_p, i, ID_u} = H(b_i \mid i \mid ID_u) \). The \( K_{ld_p, i, ID_u} = H(b_i \mid i \mid ID_u) \) key can then be calculated as the communication key.

3.3. Path-key establishment phase

After the network is deployed, if a communication link cannot be directly established between two nodes in the communication range, the communication key needs to be established through an indirect method through negotiation by using an neighbour node. Under the condition that the network topology is stable, it is always possible to find a secure multi-hop transmission path negotiation to solve the problem of communication between each other.

4. Performance analysis of the scheme

This section will analyze the performance of this paper's scheme and use the Matlab to plot our results.

4.1. Analysis of secure connectivity rates

The probability that two neighbor sensor nodes will establish a communication key directly is an important indicator for evaluating key pre-distribution scheme.

Assume that \( A_1(u, v) \) is an event that two neighbour nodes have a common polynomial, and \( A_2(u, v) \) is that u and v are both calculated by polynomial to obtain the same key. Assume that \( A_3(u, v) \) is an event that the polynomial identification \( Id_v \) in one node is the same as the \( Id_u \) in the generated key identifier in another node. These three events are mutually exclusive events and there is \( P(A_1(u, v)) + P(A_2(u, v)) + P(A_3(u, v)) = \Omega \). Here \( \Omega \) represents the probability space. According to the description in the previous section, the probability that communication can be formed is \( P_c \):

\[
P_c = P(A_1(u, v) + A_2(u, v)A_3(u, v)A_1(u, v) + A_2(u, v)A_3(u, v)A_1(u, v))
\]

(1)

We deform the formula (1) by the relevant formula of probability:

\[
P_c = P(A_1(u, v) + \overline{A_2(u, v)A_3(u, v)A_1(u, v)}) = P(A_1(u, v)) + P(\overline{A_2(u, v)A_3(u, v)A_1(u, v)}) + P(A_2(u, v)A_3(u, v)A_1(u, v))
\]

(2)

We can then be calculated as the communication key.

The probability \( P(A_1(u, v)) \) represents that don't have the same polynomial between the two sensor nodes. The \( P(A_2(u, v) \mid A_1(u, v)) \) represents that u and v have no same generated key. Where \( P(A_3(u, v) \mid A_2(u, v)A_1(u, v)) \) represents that there are no same polynomials in both nodes u and v and have no same generated keys. And no same key which can be calculated from a polynomial and the generated key. Through the above description we can get the expression of probability \( P_c \) as follows:

\[
P_c = 1 - \frac{m - g}{g} \left( \frac{ng(t+1) - 2g(t+1) - r}{r} \right) \left( \frac{ng(t+1) - 2g(t+1) - r}{r} \right)
\]

(3)
According to formula (3), the size of the polynomial key pool is m, where g represents the number of polynomial stored in the node. Each node uses the polynomial share to generate g(t+1) keys in WSNs. Finally, a total of ng(t+1) keys are generated, the key pool size is n=2000, the selected polynomial is g=2, m selects different values, but the number of keys stored in each node is 200. We can get the relationship between the number of different generated keys r and the connectivity of the network as shown in Figure 2.

![Figure 2](image)

**Figure 2.** The probability of communication key between neighbour nodes

From Figure 2, the probability $P_c$ of neighbour nodes is inversely proportional to the polynomial key pool. As m decreases, the connectivity probability of the network increases. When each node is assigned the same number of keys and the number of generated keys r is increased, the connectivity of the network can be increased. Our solution has improved the connectivity of the networks.

### 4.2. Anti-capture analysis of nodes

An attacker can get all the pairs of keys in an infected node, so they can break many secure links. Node capture attacks are also one of the most serious threats in unsupervised WSNs.

We assume that K is the communication key of two neighbor nodes that are not compromised. Let $D_1$ represent that the communication K is an event based on a polynomial key, and $D_2$ is a key that generates a key pool based on a polynomial. Let $E_1$ represent a joint event in which the key is a polynomial-based key and the key has been compromised, and let $E_2$ represent that the key is a joint event that generates a key pool based on the polynomial and that the key has been compromised. We use the symbol $K \in D_1$ to indicate "Key K is a polynomial-based key" and $K \in D_2$ means "Key K is a key pool-based generation key". When the x node is compromised, the possibility that the communication key K is compromised is $P_c$.

Suppose $C_x$ is an event where the x node has been compromised. We get formula $P(K_{\text{compromised}} \mid C_x) = P(E_1 \cup E_2 \mid C_x)$. Because events $E_1$ and $E_2$ are mutually exclusive, we get $P(K_{\text{compromised}} \mid C_x) = P(E_1 \mid C_x) + P(E_2 \mid C_x)$ for the above formula conversion.

$$P(K_{\text{compromised}} \mid C_x) = P(K \in D_1)P(D_1 \text{ is compromised})\mid C_x) + P(K \in D_2)P(D_2 \text{ is compromised})\mid C_x) \quad (4)$$

Since the preloaded key in the node is a one-way function value of the polynomial share, the attacker cannot infer the polynomial share from the preloaded key in the damaged node. We have formula $P(K \in D_1) = (P(A_1(u,v))/P_c = (1 - P(A_1(u,v)))/P_c$ and $P(K \in D_1)+P(K \in D_2) = 1$. 

---

**Figure 2.** The probability of communication key between neighbour nodes

From Figure 2, the probability $P_c$ of neighbour nodes is inversely proportional to the polynomial key pool. As m decreases, the connectivity probability of the network increases. When each node is assigned the same number of keys and the number of generated keys r is increased, the connectivity of the network can be increased. Our solution has improved the connectivity of the networks.

### 4.2. Anti-capture analysis of nodes

An attacker can get all the pairs of keys in an infected node, so they can break many secure links. Node capture attacks are also one of the most serious threats in unsupervised WSNs.

We assume that K is the communication key of two neighbor nodes that are not compromised. Let $D_1$ represent that the communication K is an event based on a polynomial key, and $D_2$ is a key that generates a key pool based on a polynomial. Let $E_1$ represent a joint event in which the key is a polynomial-based key and the key has been compromised, and let $E_2$ represent that the key is a joint event that generates a key pool based on the polynomial and that the key has been compromised. We use the symbol $K \in D_1$ to indicate "Key K is a polynomial-based key" and $K \in D_2$ means "Key K is a key pool-based generation key". When the x node is compromised, the possibility that the communication key K is compromised is $P_c$.

Suppose $C_x$ is an event where the x node has been compromised. We get formula $P(K_{\text{compromised}} \mid C_x) = P(E_1 \cup E_2 \mid C_x)$. Because events $E_1$ and $E_2$ are mutually exclusive, we get $P(K_{\text{compromised}} \mid C_x) = P(E_1 \mid C_x) + P(E_2 \mid C_x)$ for the above formula conversion.

$$P(K_{\text{compromised}} \mid C_x) = P(K \in D_1)P(D_1 \text{ is compromised})\mid C_x) + P(K \in D_2)P(D_2 \text{ is compromised})\mid C_x) \quad (4)$$

Since the preloaded key in the node is a one-way function value of the polynomial share, the attacker cannot infer the polynomial share from the preloaded key in the damaged node. We have formula $P(K \in D_1) = (P(A_1(u,v))/P_c = (1 - P(A_1(u,v)))/P_c$ and $P(K \in D_1)+P(K \in D_2) = 1$. 

---

**Figure 2.** The probability of communication key between neighbour nodes

From Figure 2, the probability $P_c$ of neighbour nodes is inversely proportional to the polynomial key pool. As m decreases, the connectivity probability of the network increases. When each node is assigned the same number of keys and the number of generated keys r is increased, the connectivity of the network can be increased. Our solution has improved the connectivity of the networks.

### 4.2. Anti-capture analysis of nodes

An attacker can get all the pairs of keys in an infected node, so they can break many secure links. Node capture attacks are also one of the most serious threats in unsupervised WSNs.

We assume that K is the communication key of two neighbor nodes that are not compromised. Let $D_1$ represent that the communication K is an event based on a polynomial key, and $D_2$ is a key that generates a key pool based on a polynomial. Let $E_1$ represent a joint event in which the key is a polynomial-based key and the key has been compromised, and let $E_2$ represent that the key is a joint event that generates a key pool based on the polynomial and that the key has been compromised. We use the symbol $K \in D_1$ to indicate "Key K is a polynomial-based key" and $K \in D_2$ means "Key K is a key pool-based generation key". When the x node is compromised, the possibility that the communication key K is compromised is $P_c$.

Suppose $C_x$ is an event where the x node has been compromised. We get formula $P(K_{\text{compromised}} \mid C_x) = P(E_1 \cup E_2 \mid C_x)$. Because events $E_1$ and $E_2$ are mutually exclusive, we get $P(K_{\text{compromised}} \mid C_x) = P(E_1 \mid C_x) + P(E_2 \mid C_x)$ for the above formula conversion.

$$P(K_{\text{compromised}} \mid C_x) = P(K \in D_1)P(D_1 \text{ is compromised})\mid C_x) + P(K \in D_2)P(D_2 \text{ is compromised})\mid C_x) \quad (4)$$

Since the preloaded key in the node is a one-way function value of the polynomial share, the attacker cannot infer the polynomial share from the preloaded key in the damaged node. We have formula $P(K \in D_1) = (P(A_1(u,v))/P_c = (1 - P(A_1(u,v)))/P_c$ and $P(K \in D_1)+P(K \in D_2) = 1$. 

---

**Figure 2.** The probability of communication key between neighbour nodes

From Figure 2, the probability $P_c$ of neighbour nodes is inversely proportional to the polynomial key pool. As m decreases, the connectivity probability of the network increases. When each node is assigned the same number of keys and the number of generated keys r is increased, the connectivity of the network can be increased. Our solution has improved the connectivity of the networks.

### 4.2. Anti-capture analysis of nodes

An attacker can get all the pairs of keys in an infected node, so they can break many secure links. Node capture attacks are also one of the most serious threats in unsupervised WSNs.

We assume that K is the communication key of two neighbor nodes that are not compromised. Let $D_1$ represent that the communication K is an event based on a polynomial key, and $D_2$ is a key that generates a key pool based on a polynomial. Let $E_1$ represent a joint event in which the key is a polynomial-based key and the key has been compromised, and let $E_2$ represent that the key is a joint event that generates a key pool based on the polynomial and that the key has been compromised. We use the symbol $K \in D_1$ to indicate "Key K is a polynomial-based key" and $K \in D_2$ means "Key K is a key pool-based generation key". When the x node is compromised, the possibility that the communication key K is compromised is $P_c$.

Suppose $C_x$ is an event where the x node has been compromised. We get formula $P(K_{\text{compromised}} \mid C_x) = P(E_1 \cup E_2 \mid C_x)$. Because events $E_1$ and $E_2$ are mutually exclusive, we get $P(K_{\text{compromised}} \mid C_x) = P(E_1 \mid C_x) + P(E_2 \mid C_x)$ for the above formula conversion.

$$P(K_{\text{compromised}} \mid C_x) = P(K \in D_1)P(D_1 \text{ is compromised})\mid C_x) + P(K \in D_2)P(D_2 \text{ is compromised})\mid C_x) \quad (4)$$

Since the preloaded key in the node is a one-way function value of the polynomial share, the attacker cannot infer the polynomial share from the preloaded key in the damaged node. We have formula $P(K \in D_1) = (P(A_1(u,v))/P_c = (1 - P(A_1(u,v)))/P_c$ and $P(K \in D_1)+P(K \in D_2) = 1$. 

---

**Figure 2.** The probability of communication key between neighbour nodes

From Figure 2, the probability $P_c$ of neighbour nodes is inversely proportional to the polynomial key pool. As m decreases, the connectivity probability of the network increases. When each node is assigned the same number of keys and the number of generated keys r is increased, the connectivity of the network can be increased. Our solution has improved the connectivity of the networks.
It is assumed that the attacker wants to break anyone polynomial \( f \), \( f \) is chosen as the probability \( P_f = g/m \) of the polynomial establishing the key. Suppose the attacker randomly captures \( x \) sensor nodes. In order to achieve the compromise polynomial \( f \), the \( x \) attacker requires that the polynomial \( f \) be selected in this sensor node at least \( t+1 \) times as a polynomial of the negotiation key, so the probability that the polynomial \( f \) is cracked is:

\[
P(D_i \text{ is compromised}) | C_s = 1 - \sum_{i=0}^{x} \left( \frac{g}{m} \right)^i \left( 1 - \frac{g}{m} \right)^{x-i}
\]

(5)

When an attacker captures \( x \) nodes from the network, the \( s \) keys are obtained, and the probability that any one of the keys in the key pool is selected as the key is \( \frac{r}{ng(t+1)} \). The probability of at least one key that has been cracked as a key for communication by the secure node is \( 1 - \left( 1 - \frac{r}{ng(s+1)} \right)^s \).

However, in this paper, the generated key in the node is preloaded with the probability of breaking another key with the same identity through a captured key is:

\[
p(\lambda) = \sum_{i=0}^{x} \frac{1}{Z} \times \frac{Z - \lambda}{Z} = \frac{Z + 1}{2Z}
\]

(6)

So here the \( P(D_2 \text{ is compromised}) \) is:

\[
P(D_2 \text{ is compromised}) = 1 - \left( 1 - \frac{Z + 1}{2Z} \frac{s}{ng(s+1)} \right)^s
\]

(7)

Substituting the above formula (5)(7) into the equation (4), we will obtain equation (8).

\[
p_r = \left( 1 - \sum_{i=0}^{x} \frac{g}{m} \left( 1 - \frac{g}{m} \right)^{x-i} \right) 1 - P(E(u,v)) + \left( 1 - \frac{Z + 1}{2Z} \frac{s}{ng(t+1)} \right) P_r \left( 1 - P(E(u,v)) \right)
\]

(8)

We designed a scheme to ensure that the total number of keys in the node is the same and maintain same connectivity probability, assuming that the total 200 keys in the node and the connectivity probability of each scheme is \( P_r = 0.5 \). We perform experiment under parameters given above.

![Figure 3](image-url)
node and the \( P_c = 0.5 \), we carry out the corresponding simulation experiments. Our scheme is more resistant to capture than the \( q \)-composite \((q=1)\) scheme [6]. We effectively solve the "t-secure" security problem of the Liu scheme[8]. When the number of captured is more than a certain value, the security of the network drops sharply. If an attacker captures 350 sensor nodes, the probability of the network being compromised is 62.9%. Once the attacker captures 600 nodes, the entire network will be paralyzed. Our scheme even if the attacker captures 600 nodes, when: \( t=40, g=3, m=21, n=1000, Z=5 \) reduces the probability of the network being compromised to 81.25%. When: \( t=40, g=2, m=16, n=1000, Z=5 \) only 66.43% of the non-compromised nodes were damaged. Even if the number of nodes being attacked in the network is large, it will not cause network paralysis. When the number of attacks reaches a certain number, the more attacks the nodes are less effective. Our scheme can effectively reduce the probability of network compromise.

5. Conclusion
The key management of WSNs is still a hot and difficult problem, and it is also the focus of contemporary scholars to study WSNs. The preload key in the sensor node is not composed of a single key but is composed of two parts. Each sensor node is assigned a corresponding polynomial shares, and then each node is assigned a generated key processed by a hash operation. The proposed scheme aims to avoid the "t-secure" problem faced by the conventional polynomial scheme and achieves the great improvement of network security. The corresponding key is generated using the polynomial share preloaded on the sensor node, which has a higher connectivity than the polynomial scheme. In the future research, we also plan to find a set of the most suitable parameters for different schemes, so that our scheme can adapt to more situation.

References
[1] S. R. Nabavi and M. S. Mousavi, “A Review of Distributed Dynamic Key Management Schemes in Wireless Sensor Networks,” JCP, vol. 13, pp. 77–89, 2018.
[2] D. S. Kridi, D. G. Gomes, N. De Carvalho, and C. Giovanni, “Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection,” Comput. Electron. Agric., vol. 127, pp. 221–235, 2016.
[3] R. Ezhilarasie, T. Renugadevi, and A. Umamakeswari, “Key management schemes in wireless sensor networks: a survey,” IJAIP, vol. 7, pp. 222–239, 2015.
[4] X. Zhou, B. Tan, and F. Xu, Wireless Sensor Network and Security. National Defend Industry Press, 2007.
[5] H. Song, S. Jeschke, and G. A. Fink, Security and Privacy in Cyber-physical Systems: Foundations, Principles, and Applications. John Wiley & Sons, 2017.
[6] H. Chan, D. Song, and A. Perrig, “Random key predistribution schemes for sensor networks,” in Proceedings - IEEE Symposium on Security and Privacy, 2003, vol. 2003–Janua, pp. 197–213.
[7] L. Eschenauer and V. Gligor, “A key-management scheme for distributed sensor networks,” Proc. 9th ACM Conf. Comput. Commun. Secur. - CCS, pp. 41–47, 2002.
[8] D. Liu, R. Li, and P. Ning, “Establishing pairwise keys in distributed sensor networks,” ACM Trans. Inf. Syst. Secur., vol. 8, no. 1, pp. 41–77, 2005.
[9] W. Bechkit, Bouabdallah Abdelmadjid, and Y. Challal, “A new class of Hash-Chain based key pre-distribution schemes for WSN,” Comput. Commun., vol. 36, no. 3, pp. 243–255, 2013.
[10] E. A. M. Anita and R. G. E. Kannan, “A Novel Hybrid Key Management Scheme Sensor Networks,” Wirel. Pers. Commun., pp. 1419–1433, 2015.
[11] J. Zhang, R. Yang, and Q. Cui, “A Hybrid Key Establishment Scheme for Wireless Sensor Networks,” vol. 10, no. 2, pp. 411–422, 2016.
[12] P. Ahlawat and M. Dave, “A Hybrid Approach for Path Vulnerability Matrix on Random Key Predistribution for Wireless Sensor,” Wirel. Pers. Commun., vol. 94, no. 4, pp. 3327–3353, 2017.