Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 4393721, 17 pages
http://dx.doi.org/10.1155/2016/4393721

Hindawi

Research Article

Security Analysis and Improvements of Session Key Establishment for Clustered Sensor Networks

Jiye Kim, Jongho Moon, Jaewook Jung, and Dongho Won

College of Information and Communication Engineering, Sungkyunkwan University, Suwon-Si 16419, Republic of Korea

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 2 February 2016; Revised 19 March 2016; Accepted 10 April 2016

Academic Editor: Fei Yu

WSN (wireless sensor network) is one of the main technologies in IoT (Internet of Things) applications or services. To date, several schemes have been proposed to establish a pair-wise key between two nodes in WSN, and most of them are designed to establish long-term keys used throughout the network lifetime. However, in the near future, if WSN will be used for information infrastructures in various fields such as manufacturing, distribution, or public facilities management and its life cycle can be as long as that of other common networks, it will definitely be advantageous in terms of security to encrypt messages using session keys instead of long-term keys. In this paper, we propose a session key establishment scheme for clustered sensor networks that is based on elliptic curve Diffie-Hellman (ECDH) key exchange and hash chain. The proposed scheme eliminates vulnerabilities of existing schemes for WSN and has improved security. The proposed scheme is efficient in terms of energy costs compared to related schemes.

1. Introduction

A wireless sensor network (WSN) is composed of dozens to thousands of sensor nodes and more than one gateway and is employed with the objective of collecting data regarding the conditions or changes in the target area [1, 2]. WSN is one of the key technologies in IoT (Internet of Things) applications or services and is expected to be employed in various applications in fields such as military, healthcare, public facilities management, manufacturing, distribution, and agriculture in the near future [1, 3–5]. However, WSN is vulnerable to attacks such as node impersonation attacks, man-in-the-middle (MITM) attacks, and denial-of-service (DoS) attacks by eavesdropping or altering of the messages transmitted in wireless channels, as are other common wireless networks [6–8]. Therefore, WSN should employ security techniques to meet the security requirements of data confidentiality and integrity, availability of services, and node authentication [9].

The key establishment scheme is one of the most fundamental and feasible security techniques [10]. Lai et al.'s BROSK [11], Eschenauer and Gligor's random key pool-based scheme [12], and so forth provide the function of establishing a pair-wise key between sensor nodes [13]. Such schemes are designed with the objective of establishing a long-term key to be used throughout the lifetime of WSN under the assumption that the life cycle of WSN is much shorter than the life cycle of other networks [14]. For example, if WSN is installed to monitor a hostile environment that is not easily accessible to people, such as a battlefield or a disaster area, its life cycle is shorter than the attack time needed to determine the cryptographic keys. In this case, it is more effective for the cryptographic keys not to be rekeyed after being established, except when adding new nodes or eliminating existing nodes. However, if WSN is used for information infrastructures in fields such as manufacturing, distribution, or public facilities management, its life cycle may be long. In this case, there is a need for a session key establishment scheme that continuously renews cryptographic keys according to a cycle or an event [14].

In an information and communication system, the message sender encrypts the confidential data and transmits it in the form of ciphertext to the message receiver. However, if an attacker obtains the decryption key by hacking, he/she can obtain the plaintext or additionally perform other serious attacks using the key. In order to decrease the damage caused by such key exposure, a cryptographic key known

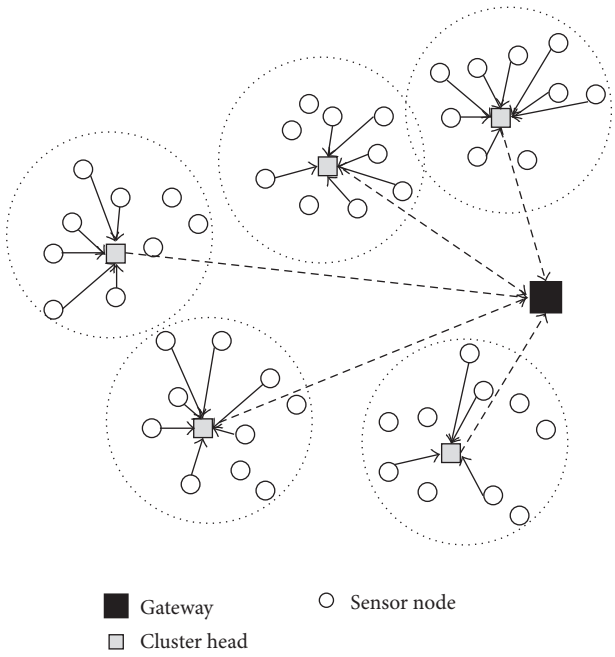


Figure 1: The flow of the sensed data in a clustered sensor network. In practice, the data is transmitted hop-by-hop between nonneighboring nodes.

as a session key is used only for a limited period of time. In communication protocols based on session keys, even if an attacker obtains one of the session keys, the number of ciphertexts he/she can decrypt with it is limited. Also, he/she needs more pairs of plaintext and ciphertext for cryptanalysis or needs to obtain more session keys for other attacks. Therefore, encryption of messages using session keys is definitely advantageous in terms of security [15].

In this paper, we focus on WSNs applied to applications such as healthcare, public facilities management, and industrial automation systems. Applying WSNs to such systems is more advantageous in terms of network performance and management costs compared to applying wired networks [18]. However, in such systems, WSNs should be operated for a long period of time and are security-critical. Moreover, for easy network management, such applications can employ clustered and hierarchical sensor networks, as shown in Figure 1 [19, 20]. When employing clustered sensor networks for such applications, the communication between the gateway and the cluster head requires stronger security than the communication between the cluster head and the sensor node; this is because the cluster head collects the data sensed by sensor nodes in its cluster and transmits it to the gateway [17]. Therefore, it is appropriate to apply the session key to the communication between the gateway and the cluster head in order to increase security. However, we found that existing session key establishment schemes for WSNs [16, 17] have several security flaws; they do not provide mutual authentication between two nodes and are vulnerable to node impersonation attacks and MITM attacks. In addition, neither scheme can guarantee secrecy of future session keys if the long-term keying materials stored in the cluster head are exposed to an attacker.

In this paper, we propose a scheme to establish a session key between the gateway and the cluster head in order to enable the cluster head to transmit encrypted data to the gateway. Our proposed scheme should eliminate the weaknesses of existing schemes in order to achieve improved security. Moreover, not only the security but also the energy costs should be considered when designing the scheme because the nodes in WSNs are battery-powered. To meet these design requirements, the proposed scheme establishes session keys based on elliptic curve Diffie-Hellman (ECDH) key exchange [21, 22], an effective asymmetric key technique. Also, it employs hash chain [23–27] in order to provide mutual authentication between the gateway and the cluster head, verification of message integrity, and session key establishment, considering energy costs.

Our major contributions are as follows: first, the proposed scheme is secure against possible attacks in key establishment schemes for WSN, such as session key attacks, replay attacks, and node capture attacks. Also, it resists both node impersonation attacks and MITM attacks through mutual authentication of two communication parties and verification of message integrity. Second, compared to long-term key establishment, there is less research on session key establishment between two nodes in WSN, and the studies are relatively more recent. Third, computation and communication costs incurred by a cluster head affect its energy consumption [28–30]. Therefore, the proposed scheme is designed to minimize the number of messages transmitted between two nodes for efficiency in terms of communication costs. Also, even though it employs asymmetric key techniques, it is more efficient in terms of computation costs compared to other schemes with similar design requirements and key establishment techniques.

The remainder of the paper is organized as follows. Section 2 reviews several key establishment schemes between nodes in WSN. Section 3 describes the assumptions, design requirements, and main ideas of our proposed scheme. Section 4 proposes the improved scheme and describes its phases in detail. Section 5 analyzes the security of the proposed scheme against possible attacks in key establishment schemes for WSN. Section 6 is devoted to analyzing its energy costs compared to other schemes with similar design requirements and key establishment techniques. Finally, Section 7 concludes this paper.

2. Review of Related Works

A few key establishment schemes have been proposed to establish a pair-wise key between sensor nodes and to provide the rekeying function in case of additions of new sensor nodes or revocation of existing sensor nodes [11, 12, 31–34]. In Lai et al.'s BROSK [11], all sensor nodes share only one master key, and each sensor node establishes a pair-wise key with its neighboring nodes using that master key. This scheme is very efficient and simple, but the entire network can become vulnerable if even one sensor node in the network is compromised by an attacker. Eschenauer and

Gligor proposed a pair-wise key establishment scheme based on a random key pool [12]. In the predeployment phase of their scheme, keys are randomly chosen from one key pool and are preloaded in the sensor node. After deploying sensor nodes to the field, if a sensor node determines it has the same key as its neighboring node, it sets the same key to be the pair-wise key between two nodes. In this scheme, if an attacker compromises another sensor node that has the pair-wise key between two sensor nodes, he/she can decrypt the message transmitted between these two sensor nodes. Several modified schemes have been proposed in order to compensate for this weakness [32–34]. Based on Eschenauer and Gligor's scheme, Chan et al. proposed a scheme where a pair-wise key can be established only when two sensor nodes share multiple keys instead of one key [32]. On the other hand, Du et al. proposed a scheme that combines the random key pool-based method with Blom's method [33], which establishes a pair-wise key between two nodes using the symmetric matrix K in $K = A \cdot G$, where matrix G is the public information, and matrix A is private information in a finite field [35]. Also, Liu et al. proposed a scheme that combines Eschenauer and Gligor's method with the polynomial-based method [34] that establishes a pair-wise key between two nodes using t-degree polynomial $f(x, y)$ that satisfies $f(i, j) = f(j, i)$ [36]. All of Chan et al., Du et al., and Liu et al.'s schemes are proposed to securely protect the links between uncompromised nodes unless a threshold number of nodes are compromised [32–34].

All of the schemes mentioned above have been proposed to establish a long-term key used throughout the life cycle of WSN [14]. Compared to such schemes, session key establishment schemes between nodes in WSN have been proposed more recently. References [14, 37, 38] proposed EBS-based rekeying schemes. Eltoweissy et al. proposed the exclusion basis system (EBS), which updates a group key for normal nodes when it evicts malicious nodes from a communication group [39]. An EBS-based scheme has a key pool of size $k+m$ ($1 < k, m < n$, where n is the number of nodes in a group). k administrative keys from the key pool are assigned to each node. When the scheme evicts some malicious nodes from the group, only m messages are needed to update a group key because the messages are encrypted using unknown keys to malicious ones. Chen and Lin proposed a session key establishment scheme for grid-based sensor networks [40]. This scheme is based on one-way hash function, mutual authentication between communication parties, and symmetric key encryption as follows: first, secret parameters (a_i, a_{i-1}) and (b_j, b_{j-1}) are preloaded to the sensor node S_i and the cluster head C_j, respectively. Then, the scheme encrypts the messages transmitted from S_i to C_j using the key $K_i = h(a_i \parallel a_{i-1})$ and the ones from C_j to the gateway using the key $K_j = h(b_j \parallel b_{j-1})$. After a period of time, K_i and K_j are replaced with $K_i' = h(K_i \| a_i \| RN_1)$ and $K_j' = h(K_j \| b_j \| RN_2)$, respectively, where both RN_1 and RN_2 are generated by the cluster head C_j. Eldefrawy et al. proposed a session key agreement scheme based on asymmetric key techniques [41]. In this scheme, the gateway receives random numbers from all sensor nodes in a cluster in order to compute a session key for communication between member nodes in the cluster. The scheme encrypts the random numbers transmitted from sensor nodes to the gateway based on RSA [42] and the session keys from the gateway to sensor nodes based on elliptic curve cryptography [21]. Meanwhile, [43–45] proposed polynomial secret-sharing-based session key establishment schemes to address the node compromise problem.

Chen and Li's scheme [16] and Lee and Kim's scheme [17] employ different key establishment techniques to establish session keys between the gateway and the cluster head in clustered sensor networks. Chen and Li's scheme establishes the $(i+1)$th session key by computing $sk_{i+1} = h(sk_i \parallel sk_{i-1})$, where sk_{i-1} and sk_i represent the $(i-1)$th and ith session keys, respectively [16]. However, if an attacker obtains sk_i and sk_{i-1} of CH_j, the future session keys to be generated in the ith and the following sessions can be computed. In other words, Chen and Li's scheme does not guarantee the secrecy of future session keys. Lee and Kim applied a modified Diffie-Hellman key exchange (DHKE) technique [46] to their scheme in order to consider the efficiency in terms of computation costs of cluster heads [17]. However, because all cluster heads in this scheme share only one private key, which is a long-term key used throughout the life cycle of the WSN, it can also be compromised by an attacker. Therefore, this scheme cannot guarantee the secrecy of future session keys. Furthermore, we found that their scheme is vulnerable to node impersonation attacks or MITM attacks. In Appendices A through D, we review Chen and Li's scheme and Lee and Kim's scheme in detail and analyze their security.

3. Design Outline of the Proposed Scheme

We consider the applications of WSNs such as healthcare, public facilities management, and industrial automation systems. The WSNs utilized for such applications should be operated for a long period of time and are security-critical.

3.1. Network Model. Regarding the WSN that employs the proposed scheme, we assume the following:

(i) The WSN is a clustered sensor network divided into several clusters; it consists of three types of nodes: sensor nodes, cluster heads, and a gateway. In a cluster, the sensor nodes sense the conditions or change regarding the target area and transmit the data to their cluster head. The cluster heads not only control the sensor nodes in respective clusters [13] but also collect the data sensed by the sensor nodes and transmit the data to the gateway [17].

(ii) Sensor nodes have limited resources such as power, computation and communication capability, memory, and transmission range [1, 47–50], whereas the gateway has an abundance of these resources.

(iii) Cluster heads are fixed and not selected from ordinary sensor nodes because resources of cluster heads are richer than those of ordinary sensor nodes. Nevertheless, our scheme can still be also applied to WSNs

that perform cluster head selection [51]. This will be discussed at greater length in Section 4.2.

(iv) A sensor node or a cluster head communicates with a nonneighboring node in a hop-by-hop fashion. We assume that the intermediate nodes between the cluster head and the gateway are not required to read the message contents exchanged between two nodes. Therefore, though the cluster head transmits its message hop-by-hop to the nonneighboring gateway, the message is encrypted/decrypted only at the two nodes; that is, the message encryption/decryption is performed end-to-end.

(v) In WSNs, sensor nodes or cluster heads are usually battery-powered. In this study, because the WSN nodes have a long life cycle, their batteries should be replaced or charged once every few years of system operation [18].

(vi) Sensor nodes or cluster heads can be randomly scattered in a target area or deployed according to a defined network topology. We assume that their spatial distribution depends on the application.

(vii) All nodes in the WSN, that is, sensor nodes, cluster heads, and the gateway, are static. That is, they are not mobile.

3.2. Adversary Capabilities. We assume that an attacker can eavesdrop on or modify transmitted messages. Sensor nodes and cluster heads are vulnerable to physical attacks because they are usually deployed without tamper-proof devices in unattended environments [30, 52–54]. Therefore, an attacker can perform node capture attacks, that is, the capture of a node in a WSN and the extraction of secret parameters for use in subsequent attacks. The gateway is a trusted node that is not compromised and is secure against privileged-insider attacks or stolen-verifier attacks.

3.3. Design Requirements. The goal of our proposed scheme is for the cluster head to securely transmit the data to the gateway. For this goal, the proposed scheme provides functions to establish a session key between the cluster head and the gateway and encrypt/decrypt the data using it. In addition, the security weaknesses of existing schemes described in Section 2 will be addressed in the proposed scheme. The design requirements of the proposed scheme are as follows:

(i) Because the proposed scheme protects the data using a session key, the session key should not be exposed to an attacker attempting to eavesdrop on transmitted messages. Furthermore, although long-term parameters in the cluster head are exposed to an attacker, the attacker should be unable to compute future or past session keys.

(ii) To achieve confidentiality and integrity of the data transmitted between the gateway and the cluster head, the proposed scheme should be designed such that it is secure against possible attacks on key establishment schemes such as node impersonation attacks, MITM attacks, and replay attacks.

(iii) The security protocols alone cannot perfectly prevent node capture attacks; however, the proposed scheme should be designed to minimize the effects of such attacks [7]. That is, even if some sensor nodes are compromised by node capture attacks, it should have no effect on the communication with other normal nodes or the security of the entire network [9].

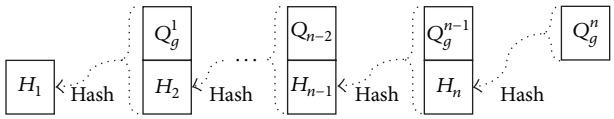
(iv) Sensor nodes or cluster heads are battery-powered and their batteries should be replaced or charged once every few years of system operation [18]. This implies that the resources of cluster heads in our network model can be relatively richer than those of sensor nodes in other sensor networks; however, they are still limited. Therefore, the proposed scheme should be designed to consider the energy consumption and security. For this, the scheme will be designed to be efficient in terms of computation and communication costs.

3.4. Notations. Notations section shows the notations used in the remainder of the paper:

(i) A pair of private and public keys for RSA signature [42] $(k_{g,pr}, k_{g,pub})$ is generated as follows: the scheme chooses two large primes p and q and computes $m = p \cdot q$. It chooses $e \in \{1, 2, \ldots, \Phi(m) - 1\}$ which fulfills the notion that $gcd(e, \Phi(m)) = 1$, where $\Phi(m) = (p-1)(q-1)$. Then, it computes d which fulfills the notion that $d \cdot e \equiv 1 \bmod \Phi(m)$. Here, the public key $k_{g,pub}$ is m and e, and the private key $k_{g,pr}$ is d. In this paper, $SIG_{k_{g,pr}}(x)$ denotes the signing of a message x with the private key $k_{g,pr}$, and it means $x^d \bmod \Phi(m)$. $VER_{k_{g,pub}}(s, x)$ denotes the verifying of a message x and its signature s with the public key $k_{g,pub}$. It computes $x^* = s^e \bmod \Phi(m)$ and then compares x with x^*. If $x^* = x$, then the signature s is valid; otherwise, it is invalid.

(ii) A pair of private and public keys for ECDH [21, 22] (d_j^i, Q_j^i) is generated as follows: the scheme chooses a large prime p and defines the elliptic curve E over Z_p $(p > 3)$ which is the set of all pairs (x, y) which fulfills the notion that $y^2 \equiv x^3 + ax + b \bmod p$ and an imaginary point of infinity O, where $4a^3 + 27b^2 \neq 0 \bmod p$ $(a, b \in Z_p)$. When P is a primitive element on the elliptic curve E and "$\times$" denotes an elliptic curve multiplication, the scheme chooses an integer d_j^i $(0 < d_j^i < n$, where n is the number of points on $E)$ and computes $Q_j^i = d_j^i \times P$. Here, Q_j^i is another element on E.

3.5. Main Ideas. Symmetric key-based session key establishment schemes are efficient with regard to computation costs; however, one of their persisting issues is the sharing and updating of the symmetric key, that is, the session key



The gateway generates n public keys $Q_g^1, Q_g^2, \ldots, Q_g^{n-1}, Q_g^n$

Then, it computes the following values in order:

$$H_n = h(Q_g^n),$$
$$H_{n-1} = h(Q_g^{n-1} \| H_n),$$
$$H_{n-2} = h(Q_g^{n-2} \| H_{n-1}),$$
$$\vdots$$
$$H_2 = h(Q_g^2 \| H_3),$$
$$H_1 = h(Q_g^1 \| H_2)$$

Figure 2: Hash chain generation in the proposed scheme.

encryption key (KEK) by two nodes [15]. Moreover, if the KEK is a long-term key, it is futile to employ the session key because it can be exposed to an attacker. Meanwhile, another method to establish a session key is to generate the next session key using keying materials stored in the previous session, similar to Chen and Li's scheme [16]. However, in such schemes, if an attacker obtains keying materials in a session, the past or future session keys can be computed.

To meet the requirements described in Section 3.3, our proposed scheme is designed as follows:

(i) The proposed scheme establishes a session key based on asymmetric key techniques in order to resist session key attacks and provide secrecy of past or future session keys. To take into account computation costs and energy consumption of cluster heads, the proposed scheme chooses an efficient key exchange technique, ECDH [21, 22], from asymmetric key techniques with the same security level.

(ii) To resist node impersonation attacks, MITM attacks, and so forth, the proposed scheme should provide mutual authentication between the gateway and the cluster head and verify message integrity. To realize this, the proposed scheme is designed based on the hash chain containing the digests of public keys generated by the gateway, as shown in Figure 2. The gateway transmits one element of the hash chain to the cluster head for each session. Using the received hash chain element, the cluster head can authenticate the message sender and verify the integrity of the message. In our scheme, the cluster head can perform these processes efficiently in terms of computation and communication costs by computing only a single hash value.

4. Description of the Proposed Scheme

Our scheme is composed of the following three phases: predeployment phase, hash chain setup phase, and key establishment phase. The predeployment phase is performed before cluster heads are deployed in the field. After that, the hash chain setup phase and the key establishment phase are performed. Each of these phases is described in detail from Section 4.1 to Section 4.3.

4.1. Predeployment Phase. Keying materials include information or algorithms required for key establishment. Not only in the proposed scheme but also in many secure protocols for WSN, keying materials are preloaded into nodes before they are deployed in the field [16, 17, 33, 55]. There are two reasons for preloading the keying materials. First, WSN is difficult to be equipped with secure channels such as mail compared to other common networks. Second, computation or communication costs can be reduced by skipping the initialization process after nodes are deployed in the target area. The predeployment phase of our scheme is as follows (steps (P-1) to (P-4)):

(P-1) The scheme generates a pair of private and public keys for RSA signature [42] ($k_{g,pr}$, $k_{g,pub}$) as described in Section 3.4.

(P-2) The two keys ($k_{g,pr}$, $k_{g,pub}$) are preloaded into GW. The private key $k_{g,pr}$ is stored only in GW and is not shared with other nodes. The public key $k_{g,pub}$ is preloaded into all cluster heads. In the hash chain setup phase described in Section 4.2, GW signs the first element of the hash chain using $k_{g,pr}$, and CH_j verifies the signature using $k_{g,pub}$.

(P-3) The scheme generates a pair of private and public keys for ECDH [21, 22] (d_j^1, Q_j^1) as described in Section 3.4.

(P-4) The two keys d_j^1 and Q_j^1 are preloaded into CH_j. d_j^1 is not shared with any cluster heads or sensor nodes other than CH_j. Q_j^1 are stored in the database of GW. In the hash chain setup phase described in Section 4.2, (d_j^1, Q_j^1) are used for CH_j to establish a session key based on ECDH [21, 22].

When this phase is completed, ($k_{g,pr}$, $k_{g,pub}$) and Q_j^1 are preloaded into GW. (d_j^1, Q_j^1) and $k_{g,pub}$ are preloaded into CH_j. The private key of GW, $k_{g,pr}$, and the private key of CH_j, d_j^1, are secret parameters that cannot be shared with other nodes.

4.2. Hash Chain Setup Phase. In the hash chain setup phase, GW generates a hash chain to be used in the key establishment phase discussed in Section 4.3. If the number of elements in the hash chain is n, during n sessions, the hash chain setup phase is performed once only in the first session, and the key establishment phase is performed $(n-1)$ times in total, once in each session from the second to the nth session. In this phase, when GW transmits the first element of the hash chain, H_1, with its signature to CH_j, CH_j verifies that H_1 is generated by GW and is not altered during the transmission using the signature. Figure 3 depicts the hash chain setup phase. The detailed process of this phase is as follows (steps (H-1) to (H-11)):

(H-1) GW generates n private keys ($d_g^n, d_g^{n-1}, \ldots, d_g^2, d_g^1$) used for ECDH [21, 22] of n sessions. Then, GW

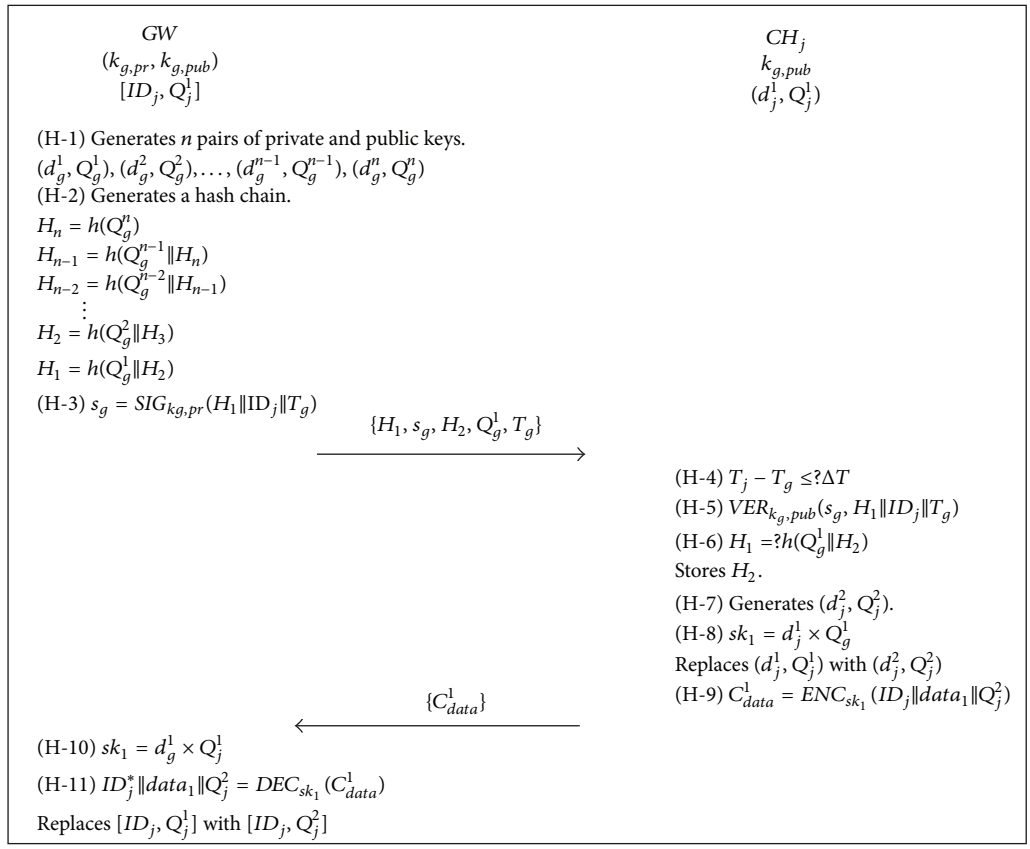


Figure 3: Hash chain setup phase of the proposed scheme.

computes n public keys $(Q_g^n, Q_g^{n-1}, \ldots, Q_g^2, Q_g^1)$ corresponding to the private keys.

(H-2) GW generates a single hash chain containing n elements, as shown in Figure 2, using the public keys $(Q_g^n, Q_g^{n-1}, \ldots, Q_g^2, Q_g^1)$. First, GW computes the hashed value of Q_g^n; that is, $H_n = h(Q_g^n)$, and it then computes the following values in order, $H_{n-1} = h(Q_g^{n-1} \| H_n), H_{n-2} = h(Q_g^{n-2} \| H_{n-1}), \ldots, H_2 = h(Q_g^2 \| H_3), H_1 = h(Q_g^1 \| H_2)$:

$$\begin{aligned} H_n &= h\left(Q_g^n\right), \\ H_{n-1} &= h\left(Q_g^{n-1} \| H_n\right), \\ H_{n-2} &= h\left(Q_g^{n-2} \| H_{n-1}\right), \\ &\vdots \\ H_2 &= h\left(Q_g^2 \| H_3\right), \\ H_1 &= h\left(Q_g^1 \| H_2\right). \end{aligned} \tag{1}$$

(H-3) GW signs the first element of the hash chain (H_1) using its private key $k_{g,pr}$; that is, $s_g = SIG_{k_{g,pr}}(H_1 \| ID_j \| T_g)$, where ID_j is the identity of CH_j, and T_g is the current timestamp of GW system. Then, GW transmits the message $\{H_1, s_g, H_2, Q_g^1, T_g\}$ to CH_j.

(H-4) CH_j determines if $(T_j - T_g) \leq \Delta T$, where T_j is the current timestamp of CH_j system, and ΔT is the maximum permitted transmission delay time. If $(T_j - T_g) \leq \Delta T$, then the next step proceeds; otherwise, this phase is aborted.

(H-5) CH_j verifies s_g using the preloaded public key $k_{g,pub}$; that is, CH_j performs $VER_{k_{g,pub}}(s_g, H_1 \| ID_j \| T_g)$. If the verification is successful, then the next step is performed.

(H-6) CH_j compares the hashed value of Q_g^1 and H_2 with H_1. In $H_1 = h(Q_g^1 \| H_2)$, it is very difficult to compute Q_g^1 or H_2 from H_1 because of the characteristics of the one-way hash function. Therefore, CH_j can verify that Q_g^1 and H_2 are generated by GW and not altered during the transmission by verifying $H_1 = h(Q_g^1 \| H_2)$. If the verification is obtained, then CH_j stores H_2, and the next step will be performed.

(H-7) CH_j generates a pair of private and public keys (d_j^2, Q_j^2) for ECDH [21, 22] in the next session.

(H-8) CH_j computes the session key $sk_1 = d_j^1 \times Q_g^1$ for this first session. Then, CH_j replaces (d_j^1, Q_j^1) with (d_j^2, Q_j^2).

(H-9) CH_j encrypts $data_1$ and Q_j^2 using the session key sk_i; that is, it performs $C_{data}^1 = ENC_{sk_1}(ID_j \parallel data_1 \parallel Q_j^2)$, where $data_1$ represents the data that CH_j wants to transmit to GW in this session. Then, CH_j transmits the message $\{C_{data}^1\}$ to GW.

(H-10) Upon receiving the message from CH_j, GW finds $[ID_j, Q_j^1]$ from its database and then computes the session key $sk_1 = d_g^1 \times Q_j^1$.

(H-11) GW decrypts C_{data}^1 using sk_1. If the decryption is completed and the result values $(ID_j^* \parallel data_1 \parallel Q_j^2)$ are correct, it means that the message sender computed the same session key as sk_1 of GW. Therefore, GW can authenticate CH_j as the message sender and verify that the message is not altered during the transmission by checking the decryption result. GW will replace $[ID_j, Q_j^1]$ with $[ID_j, Q_j^2]$ in its database for the next session.

Our proposed scheme is more suitable for a network model wherein cluster heads are fixed and not selected from ordinary sensor nodes. In this case, the resources of cluster heads are usually richer than those of ordinary sensor nodes. Nevertheless, our scheme can still be applied to WSNs that perform random node deployment, clustering, or cluster head selection [51]. In the predeployment phase, our scheme preloads only three keys, that is, d_j^1, Q_j^1, and $k_{g,pub}$, to the cluster head CH_j. Even though nodes in WSNs have limited memory, they do not require additional memory to store these three keys. Therefore, when the cluster heads are replaced, the scheme preloads three keys to all cluster head candidates in the predeployment phase. Then, only the selected cluster heads perform the hash chain setup phase in the field.

4.3. Key Establishment Phase. After the hash chain setup phase generates a hash chain with n elements in the first session, the key establishment phase is performed for each session from the second session to the last, nth session. GW transmits a key establishment request message including one element of the hash chain to CH_j. Then, CH_j verifies the message, generates the session key based on ECDH [21, 22], encrypts the data using the key, and transmits it as the response message to GW. If all verifications in this phase are passed successfully, GW and CH_j can share the same session key and encrypt/decrypt the data using the key. Figure 4 shows the process of the key establishment phase as follows (steps (K-1) to (K-7)):

(K-1) GW transmits the key establishment request message $\{Q_g^i, H_{i+1}\}$ to CH_j.

(K-2) CH_j computes $h(Q_g^i \parallel H_{i+1})$ and verifies that $H_i = h(Q_g^i \parallel H_{i+1})$, where H_i is stored in the previous session. If the verification is passed, then CH_j replaces H_i with H_{i+1}, and the next step is performed.

(K-3) CH_j computes the session key $sk_i = d_j^i \times Q_g^i$.

(K-4) CH_j generates its new private and public keys d_j^{i+1} and Q_j^{i+1} for the next $(i+1)$th session and replaces (d_j^i, Q_j^i) with (d_j^{i+1}, Q_j^{i+1}).

(K-5) CH_j encrypts $data_i$ and Q_j^{i+1} using the session key sk_i; that is, $C_{data}^i = ENC_{sk_i}(ID_j \parallel data_i \parallel Q_j^{i+1})$, where $data_i$ is the data that CH_j wants to transmit to GW in this session, and ID_j is the identity of CH_j. Then, it transmits the response message $\{C_{data}^i\}$ to GW.

(K-6) When GW receives the message from CH_j, it finds $[ID_j, Q_j^i]$ from its database and computes the session key $sk_i = d_g^i \times Q_j^i$.

(K-7) GW decrypts C_{data}^i using sk_i and determines whether or not the decryption result is correct. If the verification is passed successfully, GW can authenticate CH_j as the message sender and verify that the message was not altered during the transmission. GW replaces $[ID_j, Q_j^i]$ with $[ID_j, Q_j^{i+1}]$ in its database.

After GW exhausts the last element of the hash chain in the key establishment phase for the nth session, the scheme performs the hash chain setup phase for a set of n new sessions.

5. Security Analysis of the Proposed Scheme

The existing schemes are not able to protect past or future session keys if long-term keying materials are exposed to an attacker. The proposed scheme employs asymmetric key techniques to improve this problem, especially ECDH [21, 22], considering computation efficiency in cluster heads. Additionally, it employs the hash chain composed of digests of public keys generated by the gateway in order to resist MITM attacks or node impersonation attacks and to provide mutual authentication of two nodes and the verification of message integrity, considering computation and communication costs:

(i) *Data Encryption Using a Session Key*. If the life cycle of WSN is much longer than the time required for an attacker to obtain cryptographic keys through cryptanalysis or hacking, it is better in terms of security to use the session key instead of a long-term key [15, 16, 56]. In the proposed scheme, CH_j or GW encrypts/decrypts the data using keys renewed in every session. Therefore, it is relatively more difficult for an attacker to guess cryptographic keys in our proposed scheme than in long-term key-based schemes because the information that he/she can obtain by eavesdropping messages is limited and valid in only one session. Furthermore, even when an attacker succeeds in guessing the cryptographic keys,

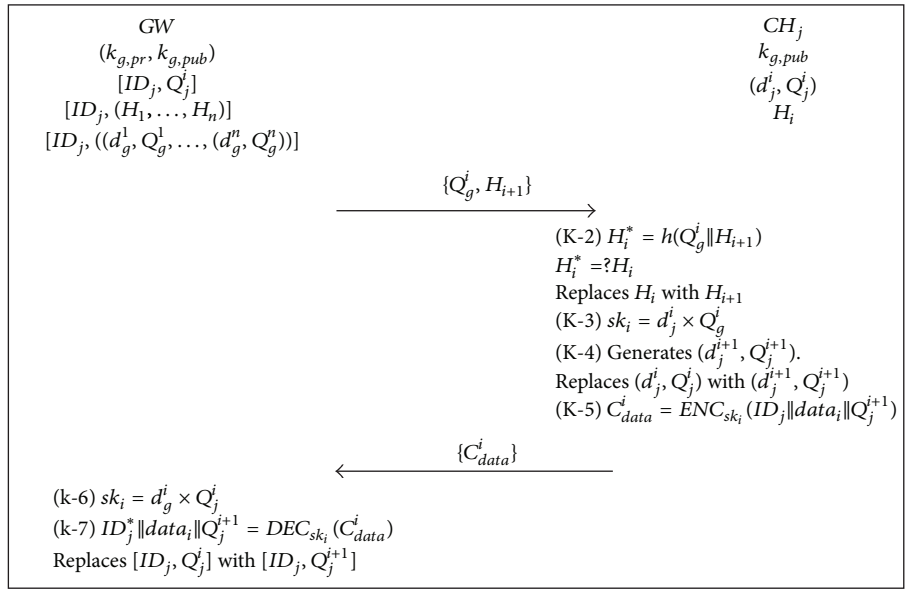


Figure 4: Key establishment phase of the proposed scheme.

the damage is significantly reduced because he/she can decrypt the data in only one session.

(ii) *Session Key Attacks*. This attack is to obtain session keys by eavesdropping the messages exchanged between two nodes. In the key establishment phase of the proposed scheme, even if an attacker eavesdrops the key establishment request message $\{Q_g^i, H_{i+1}\}$ transmitted from GW to CH_j, he/she cannot compute the session key. Even if he/she can extract the public key of GW, Q_g^i $(= d_g^i \times P)$ from the message, it is very difficult to compute the private key of GW, d_g^i, because of the elliptic curved discrete logarithm problem (ECDLP) [21, 22]. Also, the private key of GW or CH_j, d_g^i or d_j^i, respectively, is not transmitted to other nodes in an insecure channel because it is a secure parameter. As a result, the attacker cannot decrypt or alter C_{data}^i because he/she cannot compute the session key sk_i $(= d_g^i \times Q_j^i = d_j^i \times Q_g^i = d_g^i \times d_j^i \times P)$ without knowing anything of the private keys of two nodes.

(iii) *Mutual Authentication*. This means that one node should be authenticated as a legitimate node by another node with which it is in communication.

After GW generates a hash chain in the hash chain setup phase, it transmits the first element of the hash chain, H_1, and its signature, s_g, to CH_j. CH_j verifies H_1 and s_g using the public key of GW, $k_{g,pub}$. If the verification is passed, CH_j can authenticate GW as the sender of H_1. An attacker cannot impersonate GW because he/she cannot forge the signature s_g without knowing the private key of GW, $k_{g,pr}$.

Meanwhile, CH_j generates the public key Q_j^{i+1} for the $(i+1)$th session and transmits it to GW in the ith session. Then, GW stores this Q_j^{i+1} in its database. When GW receives the message $\{C_{data}^{i+1}\}$ from CH_j in the next $(i+1)$th session, it finds the public key of CH_j, Q_j^{i+1}, in its database and computes the session key $sk_{i+1} = d_g^{i+1} \times Q_j^{i+1}$. If GW can decrypt C_{data}^{i+1} using sk_{i+1}, that is, if the result value of decryption is a correct plaintext, then GW can authenticate CH_j as the sender of the message $\{C_{data}^{i+1}\}$. However, if GW fails to decrypt C_{data}^{i+1} or the result value of decryption is a meaningless random value, the session key sk_{i+1} is the wrong value. In this case, GW cannot be sure that the message sender is CH_j.

(iv) *Node Impersonation Attacks*. Node impersonation attacks in WSN mean that an attacker communicates with a legitimate node by impersonating a gateway, a sensor node, or a cluster head. In the proposed scheme, if GW or CH_j receives a message, it performs the authentication process of the message sender. Therefore, an attacker cannot impersonate GW or CH_j.

(v) *MITM Attacks*. This means that a malicious node decrypts or alters the messages transmitted between two legitimate nodes. The proposed scheme resists MITM attacks by the mutual authentication between GW and CH_j and the verification of the received messages integrity.

In the hash chain setup phase, when CH_j receives the message $\{H_1, s_g, H_2, Q_g^1, T_g\}$ from GW, it checks if H_1 is the first element of the hash chain generated by GW; that is, it verifies the signature of H_1, s_g. If this verification is passed successfully, it means that the message sender is GW and that the value of H_1 is not altered during the transmission. Each element of

the hash chain, H_i, is the digest of Q_g^i and H_{i+1}, that is, $h(Q_g^i \parallel H_{i+1})$. Other nodes except GW are not able to compute Q_g^i or H_{i+1} from H_i because $h(\cdot)$ is a one-way hash function. Therefore, after CH_j completes one verification of H_1 and s_g, the following $(n-1)$ key establishment request messages can be successively verified using H_1. That is, whenever CH_j receives the key establishment request message $\{Q_g^i \parallel H_{i+1}\}$, it compares H_i with the digest of Q_g^i and H_{i+1} to verify the message integrity. As a result, an attacker cannot alter the first element of the hash chain, H_1, because he/she cannot forge the signature of GW, s_g. Also, he/she cannot alter the rest of the elements from H_2 to H_n because of the characteristics of the one-way hash function.

Meanwhile, the message $\{C_{data}^i\}$ transmitted from CH_j to GW is secure, unless the session key is exposed to the attacker because it is encrypted using the session key. Also, an attacker cannot alter this message without knowing the session key.

(vi) *Secrecy of Past or Future Session Keys*. This means that an attacker should not be able to compute past or future session keys that were already used in the previous sessions or will be generated in the following sessions even when he/she obtains long-term keying materials. In the proposed scheme, GW and CH_j exchange their public keys, Q_g^i and Q_j^i, and generate the session key sk_i $(= d_g^i \times Q_j^i = d_j^i \times Q_g^i)$ based on ECDH [21, 22]. The parameters stored in CH_j are $k_{g,pub}$ and (d_j^i, Q_j^i), where $k_{g,pub}$ is a long-term key, and (d_j^i, Q_j^i) are values renewed in each session. When they are exposed to an attacker, past or future session keys are protected as follows.

Even though an attacker obtains $k_{g,pub}$, he/she is not able to compute the private key $k_{g,pr}$ because of the integer factorization problem [42]. That is, because he/she cannot forge the signature of GW, s_g, he/she cannot alter Q_g^i transmitted from GW to CH_j.

d_j^i and Q_j^i are ephemeral keys renewed in each session. This means that CH_j replaces d_j^i and Q_j^i with d_j^{i+1} and Q_j^{i+1}, respectively, in the end of the ith session. Assume the worst scenario in which an attacker obtains the private key, d_j^{i+1}, between the ith and $(i+1)$th sessions through some methods. Even in this case, the proposed scheme can protect the data securely transmitted before and after the $(i+1)$th session. For example, if an attacker knows the private key of CH_j, d_j^{i+1}, and eavesdrops on the message of $\{Q_g^{i+1}, H_{i+2}\}$ transmitted from GW to CH_j in the $(i+1)$th session, he/she can compute the session key $sk_{i+1} = d_j^{i+1} \times Q_g^{i+1}$ and decrypt the message C_{data}^{i+1} using sk_{i+1}. However, he/she cannot obtain any more information to restore the other session keys except sk_{i+1} from the decryption result of C_{data}^{i+1}, $(data_{i+1} \parallel Q_j^{i+2})$. As a result, the proposed scheme can assure the confidentiality of the data transmitted in all other sessions except the $(i+1)$th session.

(vii) *Replay Attacks*. This means an attacker stores messages transmitted on security protocols and transmits them again later. The proposed scheme resists replay attacks as follows:

In the proposed scheme, the message transmitted from GW to CH_j is the message $\{H_1, s_g, Q_g^1, H_2, T_g\}$ in the hash chain setup phase or the message $\{Q_g^i, H_{i+1}\}$ in the key establishment phase. The former contains the current timestamp of GW system, T_g, and is verified by the message receiver CH_j. The latter consists of the values that depend on the former because both are elements of a hash chain. Therefore, an attacker is not able to perform replay attacks using these messages.

The message $\{C_{data}^i\}$ is transmitted from CH_j to GW as a response to the hash chain setup request of GW or to the key establishment request of GW. Therefore, an attacker cannot use this message for replay attacks.

(viii) *Node Capture Attacks*. This means that an attacker physically captures some nodes deployed in WSN and extracts secret parameters from them for other attacks. In the proposed scheme, each cluster head generates a unique session key. Therefore, the links between uncompromised nodes are still secure even when one cluster head is compromised by node capture attacks. For example, assume that an attacker captures CH_j and extracts d_j^i, Q_j^i, or $k_{g,pub}$ from it. The public key of GW, $k_{g,pub}$, is preloaded into not only CH_j but also all cluster heads. However, an attacker cannot use it for any other attacks because he/she is not able to compute the private key $k_{g,pr}$ from $k_{g,pub}$. Also, d_j^i and Q_j^i are not shared with other nodes except CH_j, so the attacker cannot obtain other session keys except a session key between CH_j and GW using these two values.

Table 1 shows the comparison of the security in the proposed scheme and that in other schemes that have design requirements similar to ours. Table 1 shows which scheme is secure against possible attacks in key establishment schemes or provides security functionalities. This table shows that the proposed scheme is clearly improved in terms of security. In Appendices A through D, we review and analyze the security of the schemes proposed by Chen and Li and Lee and Kim.

6. Energy Cost Analysis of the Proposed Scheme

In this section, we analyze the efficiency of the proposed scheme in terms of computation and communication costs. Computation costs refer to the number of times each operation is performed on a cluster head or a gateway system

Table 1: Security comparison of the proposed scheme.

Security attacks or features	Chen and Li's scheme [16]	Lee and Kim's scheme [17]	The proposed scheme
Data encryption using a session key	Partially	Partially	Yes
Session key attacks	Partially	Partially	Yes
Mutual authentication	No	No	Yes
Node impersonation attacks	Partially	No	Yes
MITM attacks	Partially	No	Yes
Secrecy of past session keys	Yes	Yes	Yes
Secrecy of future session keys	No	No	Yes
Replay attacks	Yes	No	Yes
Node capture attacks	Yes	No	Yes

Yes: the scheme resists the attacks or provides the functionality; No: the scheme does not resist the attacks or provide the functionality. Partially: "Yes" under the condition that the secret parameters stored in CH_j have not been exposed to an attacker.

Table 2: Computation cost analysis of the proposed scheme (during n sessions).

Phases	Nodes	Operations						
		1024-bit RSA		SHA1	64-bit AES		163-bit ECDH	
		Signing	Verification		Encryption	Decryption	Key generation	Key exchange
Hash chain setup	Gateway	1	0	n	0	1	n	1
(one time)	Cluster head	0	1	1	1	0	1	1
Key establishment	Gateway	0	0	0	0	$n-1$	0	$n-1$
($n-1$ times)	Cluster head	0	0	$n-1$	$n-1$	0	$n-1$	$n-1$

in a scheme. Communication costs refer to the number of messages exchanged between two nodes in a scheme. In a WSN, these two costs affect the energy consumption of nodes [28–30]. In addition, we compare the computation and communication costs of our scheme with those of existing schemes that are similar to ours in terms of design requirements or key establishment techniques.

We focus on the repeatedly performed phases, that is, the hash chain set up phase and the key establishment phase, and exclude the predeployment phase. The predeployment phase does not directly affect the efficiency because it is performed only prior to the deployment of sensor nodes and cluster heads in the field.

6.1. Computation Costs. Table 2 shows the kinds of operations and the number of times they are performed on a cluster head or gateway system in the proposed scheme during n sessions. In the proposed scheme, the hash chain setup phase is performed once, and the key establishment phase is performed $(n-1)$ times:

(i) When the hash chain setup phase is performed once, the gateway performs one signing of RSA signature ((H-3) in Figure 3), data decryption ((H-11) in Figure 3), and ECDH key exchange ((H-10) in Figure 3) each. Moreover, the one-way hash operation ((H-2) in Figure 3) and ECDH key generation ((H-1) in Figure 3) are performed n times each.

(ii) When the hash chain setup phase is performed once, the cluster head performs one verification of RSA signature ((H-5) in Figure 3), one-way hash operation ((H-6) in Figure 3), data encryption ((H-9) in Figure 3), ECDH key generation ((H-7) in Figure 3), and ECDH key exchange ((H-8) in Figure 3) each.

(iii) When the key establishment phase is performed $(n-1)$ times, the gateway performs data decryption ((K-7) in Figure 4) and ECDH key exchange ((K-6) in Figure 4), $(n-1)$ times each.

(iv) When the key establishment phase is performed $(n-1)$ times, the gateway performs one-way hash operation ((K-2) in Figure 4), data encryption ((K-5) in Figure 4), ECDH key exchange ((K-3) in Figure 4), and ECDH key generation ((K-4) in Figure 4) $(n-1)$ times each.

Table 3 also shows the types of operations and the number of times they are performed on a cluster head or gateway system in Lee and Kim's scheme [17] during n sessions. In terms of design requirements and key establishment techniques, our scheme is similar to that of Lee and Kim.

To analyze the energy costs of the proposed scheme, we define several notations as follows:

(i) RSA_s: the energy cost of performing a signing of 1024-bit RSA signature.

(ii) RSA_v: the energy cost of performing a verification of 1024-bit RSA signature.

(iii) H: the energy cost of performing a SHA1.

(iv) E: the energy cost of performing a 64-bit AES encryption.

Table 3: Computation cost analysis of Lee and Kim's scheme [17] (during n sessions).

Phases	Nodes	Operations				
		SHA1	64-bit AES		163-bit ECDH	
			Encryption	Decryption	Key generation	Key exchange
Procedure 2	Gateway	n	n	n	n	n
(n times)	Cluster head	0	n	n	0	n

Table 4: Energy costs comparison of the proposed scheme (during n sessions).

Schemes	Nodes	Total energy costs	Energy cost comparison* (mJ)
The proposed scheme	Gateway	$RSA_s + nH + nD + nEC_g + nEC_e$	$546.50 + 440.20n + 0.00121l$
	Cluster head	$RSA_v + nH + nE + nEC_g + nEC_e$	$15.97 + 440.20n + 0.00121l$
Lee and Kim's scheme [17]	Gateway	$nH + nE + nD + nDH_g + nDH_e$	$1922.46n + 0.00242l$
	Cluster head	$nE + nD + nDH_e$	$1046.50n + 0.00242l$

*Energy cost comparison based on the experimental results in [28]. We assume that the cluster head transmits the total of l byte data to the gateway during n sessions.

Table 5: Energy costs of cryptographic algorithms [28].

Cryptographic algorithms	Energy costs*
Signing of 1024-bit RSA signature	546.50 mJ
Verification of 1024-bit RSA signature	15.97 mJ
SHA1	0.76 μJ/byte
64-bit AES encryption	1.21 μJ/byte
64-bit AES decryption	1.21 μJ/byte
1024-bit DHKE key generation	875.96 mJ
1024-bit DHKE key exchange	1046.50 mJ
163-bit ECDH key generation	276.70 mJ
163-bit ECDH key exchange	163.50 mJ

*These values are the experimental results in [28], in which the cryptographic algorithms were developed on a Compaq iPAQ H3670 equipped with a 206 MHz Intel SA-1110 StrongARM processor and 64 MB RAM.

(v) D: the energy cost of performing a 64-bit AES decryption.

(vi) DH_g: the energy cost of performing a 1024-bit DHKE key generation.

(vii) DH_e: the energy cost of performing a 1024-bit DHKE key exchange.

(viii) EC_g: the energy cost of performing a 163-bit ECDH key generation.

(ix) EC_e: the energy cost of performing a 163-bit ECDH key exchange.

Potlapally et al. described the energy consumption of well-known cryptographic algorithms and security protocols using the experimentation results in [28] (Table 5).

Table 4 shows the energy costs of our scheme and Lee and Kim's scheme based on computation cost analysis of the two schemes and Potlapally et al.'s experimentation results. Assume that the cluster head transmits the total of l byte data to the gateway during n sessions. To perform the proposed scheme, the gateway uses about $546.50+440.20n+0.00121l$ mJ ($= RSA_s + nH + nD + nEC_g + nEC_e$), and the cluster head uses about $15.97 + 440.20n + 0.00121l$ mJ ($= RSA_v + nH + nE + nEC_g + nEC_e$). Under the same conditions, to perform Lee and Kim's scheme, the gateway uses about $1922.46n + 0.00242l$ mJ ($= nH + nE + nD + nDH_g + nDH_e$), and the cluster head uses about $1046.50n + 0.00242l$ mJ ($= nE + nD + nDH_e$).

Given that the cluster heads are battery-powered, we have to focus more on the energy costs in the cluster head than in the gateway. Table 4 shows that the energy cost of the cluster head in our scheme is smaller than that in Lee and Kim's scheme ($15.97 + 440.20n + 0.00121l$ mJ $< 1046.50n + 0.00242l$ mJ). Therefore, in terms of energy consumption based on computation costs, the proposed scheme is more efficient than Lee and Kim's scheme. This can be attributed to the difference in the energy costs of the two key exchange algorithms, that is, 163-bit ECDH and 1024-bit DHKE; 163-bit ECDH and 1024-bit DHKE schemes have the same security level, but the energy consumption of the former is only one-quarter that of the latter (276.70 + 163.50 mJ < 875.96 + 1046.5 mJ in Table 5) [28]. Meanwhile, the verification of RSA signature in the proposed scheme does not significantly affect the total energy costs of CH_j, even though the scheme is an asymmetric key technique. This is because, for n sessions, the operation is performed only once in the hash chain setup phase and the verification is performed more efficiently than the signing in RSA signature (15.97 mJ < 546.50 mJ in Table 5) [15, 28].

6.2. Communication Costs. Communication costs as well as computation costs affect the energy costs of cluster heads [29, 30]. In our scheme, the messages $\{H_1, s_g, H_2, Q_g^1, T_g\}$ and $\{C_{data}^1\}$ are exchanged between the cluster head and the gateway in the hash chain setup phase, while the messages $\{Q_g^i, H_{i+1}\}$ and $\{C_{data}^i\}$ are exchanged in the key establishment

phase. That is, in the proposed scheme, two message exchanges are needed between the two nodes during one session, which is same as the number of messages in Lee and Kim's scheme and less than the three messages in Chen and Li's scheme. The proposed scheme minimizes the number of messages, considering that it provides all functions of session key establishment, node authentication, and data encryption.

7. Conclusion

In this paper, we propose a session key establishment scheme for clustered sensor networks based on ECDH [21, 22] and hash chain [23–27]. Our proposed scheme is secure against the possible attacks in key establishment schemes of WSN such as session key attacks, node impersonation attacks, MITM attacks, replay attacks, and node capture attacks. The scheme eliminates vulnerabilities of existing session key establishment schemes for WSN and provides secrecy of past or future session keys. Additionally, the proposed scheme is designed to minimize the number of messages for efficiency in terms of communication costs. Also, it is more efficient in terms of computation costs compared to other schemes based on asymmetric key techniques. Because of the efficiency of the proposed scheme, the cluster head requires less energy to operate.

Appendix

A. Review of Chen and Li's Scheme

In Chen and Li's scheme [16], two secret parameters a_j and a_{j-1} are preloaded in CH_j before deploying nodes to the field. GW knows every secret parameter of cluster heads and sensor nodes in the network. After the nodes are deployed to the field, CH_j performs the following (CL-1) to (CL-10) in order to transmit the data to GW (in [16], Chen and Li's scheme is composed of two parts of data transmission from the sensor node to the cluster head and from the cluster head to the gateway. Section II reviews only the latter considering our topic). In the first session, all the steps of (CL-1) to (CL-10) are performed. After the second session, the steps except (CL-1) and (CL-3) are repeated in each session. Figure 5 shows session key establishment between the gateway and the cluster head in Chen and Li's scheme:

(CL-1) CH_j computes $sk_1 = h(a_j \parallel a_{j-1})$ using its secret parameters a_j and a_{j-1}. CH_j will use the result value sk_1 as the session key to communicate with GW.

(CL-2) CH_j transmits the message $\{C_1, ID_j\}$ to GW to request the keys to decrypt the data received from the sensor nodes. Here, $C_1 = ENC_{sk_1}(sensor\_list \parallel ID_j \parallel RN_j)$, where $sensor\_list$ is the list of sensor nodes that sent the data, ID_j is the identity of CH_j, and RN_j is a random number generated by CH_j.

(CL-3) When GW receives the request message $\{C_1, ID_j\}$, it finds the secret parameters a_j and a_{j-1} of CH_j in its database and computes the session key sk_1.

(CL-4) GW decrypts C_1 using sk_1; that is, $sensor\_list \parallel ID_j \parallel RN_j = DEC_{sk_1}(C_1)$.

(CL-5) GW computes $C_g = ENC_{sk_1}(ID_g \parallel key\_list \parallel RN_j \parallel RN_g)$, where ID_g is the identity of GW, key_list is the decryption key list in regard to $sensor\_list$, and RN_g is a random number generated by GW. Then, it returns the response message $\{C_g, ID_g\}$ to CH_j.

(CL-6) When CH_j receives the response message $\{C_g, ID_g\}$ from GW, it decrypts the messages using key sk_1; that is, $ID_g \parallel key\_list \parallel RN_j^* \parallel RN_g = DEC_{sk_1}(C_g)$. Then, CH_j compares RN_j with RN_j^*, where RN_j is a random number generated in step (CL-2), and RN_j^* is a part of the decrypted results ($ID_g \parallel key\_list \parallel RN_j^* \parallel RN_g$). If the verification is passed, the next step is performed.

(CL-7) CH_j decrypts each data received from sensor nodes using the decryption keys in the key_list. Then, CH_j derives $data_1$ from the decrypted results to transmit them to GW.

(CL-8) CH_j encrypts $data_1$ using session key sk_1; that is, $C_{data}^1 = ENC_{sk_1}(data_1 \parallel RN_g \parallel ID_j)$. Then, it transmits the message $\{C_{data}^1, ID_j\}$ to GW.

(CL-9) GW decrypts C_{data}^1 using sk_1 when it receives the message $\{C_{data}^1, ID_j\}$; that is, $data_1 \parallel RN_g^* \parallel ID_j = DEC_{sk_1}(C_{data}^1)$. Then, GW compares RN_g^* with RN_g. If the verification is passed, GW can use $data_1$.

(CL-10) CH_j and GW separately compute the next session key $sk_2 = h(sk_1 \parallel a_j)$ and replace secret parameters a_j and a_{j-1} with sk_1 and a_j.

B. Cryptanalysis of Chen and Li's Scheme

In the ith session of Chen and Li's scheme, CH_j or GW encrypts the message using the session key sk_i and then transmits it to the other node. Before the end of the session, two nodes separately compute the new session key $sk_{i+1} = h(sk_i \parallel sk_{i-1})$ for the next session and replace secret parameters sk_i and sk_{i-1} with sk_{i+1} and sk_i, respectively. The following analyzes the security of their scheme against possible attacks in key establishment schemes for WSN:

(i) Session key attacks and MITM attacks: session key attacks mean that an attacker obtains session keys by eavesdropping the messages exchanged between two nodes. MITM attacks refer to attacks in which an attacker eavesdrops or alters the messages transmitted between two legitimate nodes. In Chen and Li's scheme, an attacker cannot compute the session key sk_{i+1} ($= h(sk_i \parallel sk_{i-1})$) using only the transmitted messages without knowing the secret parameters sk_i and sk_{i-1}, stored in CH_j.

(ii) Node impersonation attacks: this attack means an attacker impersonates a gateway or a cluster head to communicate with legitimate nodes. Chen and Li's scheme does not provide any node authentication process. However, an attacker cannot impersonate GW or CH_j without knowing the secret parameters

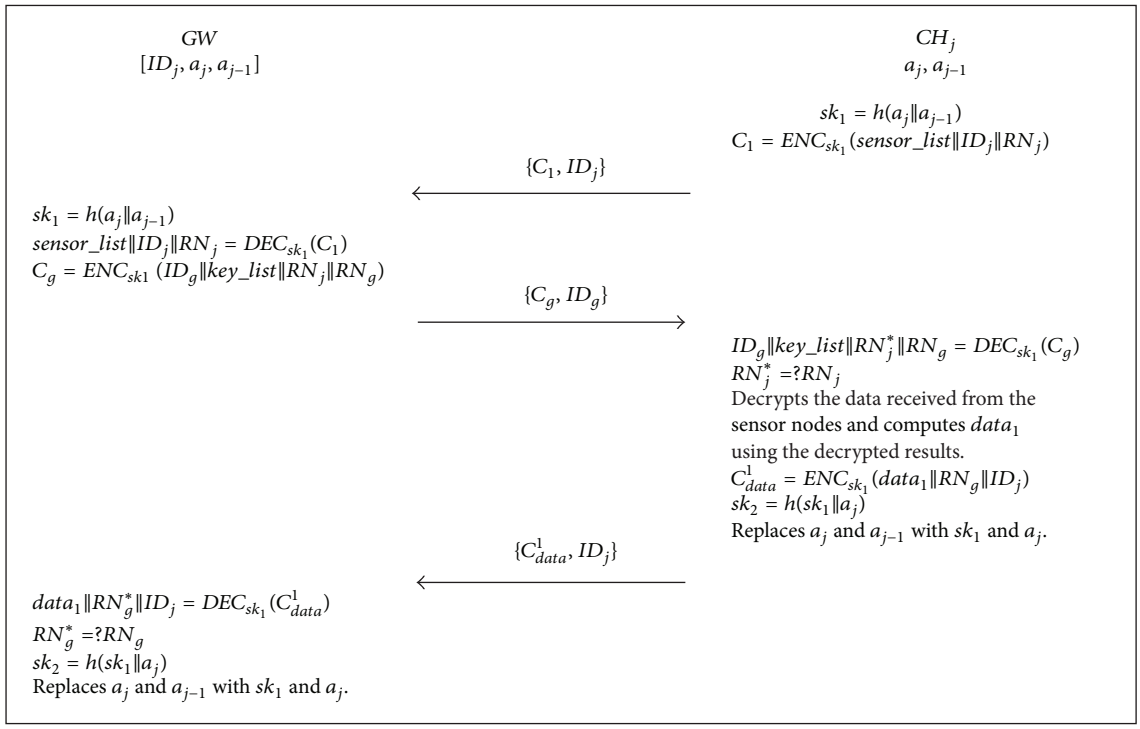


FIGURE 5: Session key establishment between the gateway and the cluster head in Chen and Li's scheme (redrawn from [16]).

such as (a_j, a_{j-1}) or (sk_i, sk_{i-1}) because the secret parameters are unique values for only CH_j and GW, and the two nodes encrypt/decrypt messages using the session keys derived from them.

(iii) Secrecy of past session keys: this means that an attacker should be unable to compute the past session keys already used in the previous sessions even when the long-term keying materials are exposed to the attacker. In Chen and Li's scheme, even if an attacker obtains sk_i and sk_{i-1} from CH_j because of the characteristics of the one-way hash function, he/she cannot recover the past session keys used in the previous sessions, that is, from the first session to the $(i-1)$th session [16].

(iv) Secrecy of future session key: this means that an attacker should be unable to compute the future session keys to be generated subsequent to the current session even when the long-term keying materials are exposed to the attacker. If an attacker obtains sk_i and sk_{i-1} of CH_j, he/she can compute the future session keys to be generated in the ith and the following sessions. That is, their scheme cannot assure the confidentiality or integrity of all messages transmitted, since the ith session until the GW system determines that the secret parameters of CH_j are compromised.

(v) Node capture attacks: this means that an attacker captures sensor nodes or cluster heads deployed in the target field and uses secret parameters extracted from them for other attacks. Because sk_i and sk_{i-1} are derived from unique values a_j and a_{j-1} for CH_j, the link between uncompromised nodes is still secure even when an attacker captures CH_j and extracts the secret parameters sk_i and sk_{i-1} from it.

C. Review of Lee and Kim's Scheme

Lee and Kim proposed a session key establishment scheme based on Diffie-Hellman key exchange (DHKE) technique [46] for secure communication between the gateway and the cluster head [17]. Before nodes are deployed in the field, a large prime for modulus operations, m, and a primitive element, q ($q \in Z_m^*$), are stored in each cluster head and the gateway. After cluster heads are deployed in the field, procedure 1 is performed for the first session and procedure 2 is performed for the second and subsequent sessions. Figure 6 illustrates both procedures. In procedure 1, the following steps ((LK-1) to (LK-5)) are performed for key setup:

(LK-1) The cluster head CH_j computes the hashed value of m, $k = h(m)$. Then, it generates a random number RN_j and encrypts RN_j and its identity ID_j using the key k; that is, $C_j = ENC_k(ID_j \parallel RN_j)$. Then, CH_j transmits the key setup request message $\{C_j, ID_j\}$ to GW.

(LK-2) Upon receiving the message from CH_j, GW computes the key $k = h(m)$ and then decrypts C_j using the key k; that is, $ID_j \parallel RN_j = DEC_k(C_j)$. GW generates a random number RN_g and computes the session key $sk_1 = q^{RN_g \cdot RN_j} \bmod m$.

(LK-3) GW computes $k_j = q^{RN_g} \bmod m$ and encrypts the result k_j and its identity ID_g using the key k; that is,

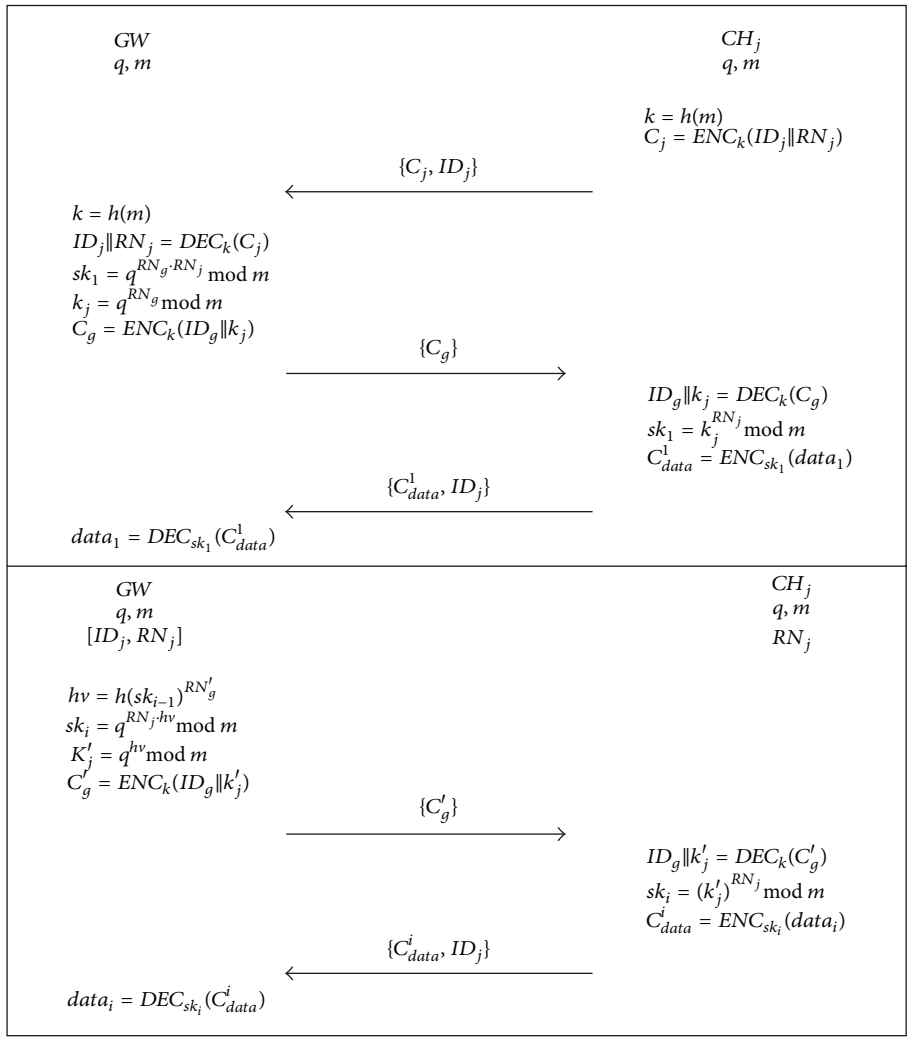


Figure 6: Procedures 1 and 2 in Lee and Kim's scheme (redrawn from [17]).

$C_g = ENC_k(ID_g \parallel k_j)$. Then, it returns the message $\{C_g\}$ to CH_j.

(LK-4) Upon receiving the message $\{C_g\}$ from GW, CH_j decrypts C_g using the key k. Then, CH_j computes $sk_1 = k_j^{RN_j} \bmod m$.

(LK-5) CH_j encrypts $data_1$ using the session key sk_1 and transmits the result to GW. Then, GW decrypts the message to obtain $data_1$.

Procedure 2 comprises the following steps ((LK-6) to (LK-10)) and is performed for CH_j to transmit data to GW for the second and subsequent sessions.

(LK-6) GW generates a new random number RN'_g and computes $hv = h(sk_{i-1})^{RN'_g}$, where sk_{i-1} is the previous session key shared with CH_j.

(LK-7) GW computes a new session key $sk_i = q^{RN_j \cdot hv} \bmod m$, where RN_j is the random number received from CH_j in procedure 1.

(LK-8) GW computes $k'_j = q^{hv} \bmod m$ and encrypts the result k'_j and ID_g using the key k; that is, $C'_g = ENC_k(ID_g \parallel k'_j)$. Then, it sends the message $\{C'_g\}$ to CH_j.

(LK-9) Upon receiving the data request message $\{C'_g\}$ from GW, CH_j decrypts C'_g using the key k. Then, CH_j computes a new session key $sk_i = (k'_j)^{RN_j} \bmod m$.

(LK-10) CH_j encrypts $data_i$ using the session key sk_i and then transmits the result C^i_{data} to GW. Then, GW decrypts C^i_{data} using the key sk_i to obtain $data_i$.

D. Cryptanalysis of Lee and Kim's Scheme

In procedure 1 of Lee and Kim's scheme, CH_j and GW exchange their random numbers RN_j and RN_g in order to share the first session key sk_1 ($= q^{RN_g \cdot RN_j} \bmod m$). In procedure 2, they compute the session key $sk_i = q^{hv \cdot RN_j} \bmod m$

for the second and subsequent sessions, where RN'_g is a new random number of GW, and $hv = h(sk_{i-1})^{RN'_g}$. However, q and m are likely to be exposed to attackers because they are shared by not only CH_j and GW but also all cluster heads in the network, and they are long-term parameters used throughout the lifetime of the network. If q and m are exposed to an attacker, this scheme can be vulnerable to node impersonation attacks and MITM attacks and cannot assure the secrecy of future session keys. The following analyzes the security of Lee and Kim's scheme against possible attacks in key establishment schemes for WSN:

(i) Session key attacks: in this scheme, all the messages exchanged between GW and CH_j are encrypted with the key k. Therefore, an attacker cannot restore session keys using only these messages without knowing secret parameters q and m.

(ii) Node impersonation attacks and MITM attacks: upon receiving a message, GW or CH_j only determines whether the message is encrypted using the key k without the message sender authentication process. Even if an attacker obtains the value of m from other cluster heads excluding CH_j, he/she can compute the key $k = h(m)$ and transmit data request messages to CH_j just like GW or can alter the messages.

(iii) Secrecy of future session keys: RN_j stored in CH_j is a random number but is a long-term parameter that is not updated. If an attacker obtains RN_j after the ith session ended, he/she can compute future session keys between GW and CH_j in the following sessions. In this case, confidentiality and integrity of the data encrypted using these session keys cannot be guaranteed.

(iv) Replay attacks: this means that an attacker resends the messages transmitted on security protocols. In their scheme, CH_j neither checks random numbers or timestamps nor authenticates GW in order to resist replay attacks using the data request messages from GW. Therefore, an attacker can repeatedly broadcast one of the data request messages to cluster heads to cause DoS attacks in WSN.

(v) Node capture attacks: in their scheme, if an attacker extracts the values of q, m, and RN_j from a cluster head in the target area, he/she can compromise even links with other cluster heads. This vulnerability causes more serious problems when new cluster heads are added for expansion or changes in the network. When a new cluster head starts procedure 1 for key setup, GW and the new cluster head exchange their random numbers after encrypting them using the key k. If an attacker already knows the key k through node capture attacks against existing cluster heads, he/she can perform node impersonation attacks, MITM attacks, and so forth by eavesdropping the exchanged messages or altering the random numbers.

Notations

GW:	Gateway node
CH_j:	jth cluster head
ID_g:	Identity of GW
ID_j:	Identity of CH_j
$k_{g,pr}, k_{g,pub}$:	Private and public keys of GW for RSA signature scheme [42]
d_j^i, Q_j^i:	Private and public keys of CH_j for elliptic curve Diffie-Hellman key exchange (ECDH) [21, 22]
$SIG_k(x)$:	Signing of a message x with a key k in RSA signature scheme [42]
$VER_k(s, x)$:	Verification of a message x and its signature s with a key k in RSA signature scheme [42]
$ENC_k(m)$:	Encryption of a plaintext m using a symmetric key k
$DEC_k(c)$:	Decryption of a ciphertext m using a symmetric key k
$h(\cdot)$:	One-way hash function
RN_g or RN_j:	Random number generated by GW or CH_j
$data_i$:	Data that CH_j transmits to GW in the ith session
sk_i:	Session key for the ith session
$\|$:	Concatenation operation
$\leq?$ or $=?$:	Verification operation
T_g or T_j:	Current timestamp of GW or CH_j
Δt:	The maximum of transmission delay time permitted.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (no. R0126-15-1111, The Development of Risk-based Authentication Access Control Platform and Compliance Technique for Cloud Security).

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[3] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Pauthkey: a pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 357430, 14 pages, 2014.

[4] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[5] D. Nyang and M.-K. Lee, "Improvement of das's two-factor authentication protocol in wireless sensor networks," *IACR Cryptology ePrint Archive*, vol. 2009, p. 631, 2009.

[6] J. Nam, M. Kim, J. Paik, Y. Lee, and D. Won, "A provably-secure ECC-based authentication scheme for wireless sensor networks," *Sensors*, vol. 14, no. 11, pp. 21023–21044, 2014.

[7] J. Kim, D. Lee, W. Jeon, Y. Lee, and D. Won, "Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6443–6462, 2014.

[8] W. Jeon, J. Kim, J. Nam, Y. Lee, and D. Won, "An enhanced secure authentication scheme with anonymity for wireless environments," *IEICE Transactions on Communications*, vol. 95, no. 7, pp. 2505–2508, 2012.

[9] S. A. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," Tech. Rep., Rensselaer Polytechnic Institute, Troy, NY, USA, 2005.

[10] Y. Lee, S. Kim, and D. Won, "Enhancement of two-factor authenticated key exchange protocols in public wireless LANs," *Computers & Electrical Engineering*, vol. 36, no. 1, pp. 213–223, 2010.

[11] B. Lai, S. Kim, and I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks," in *Proceedings of the IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES '02)*, p. 7, December 2002.

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

[13] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 2, pp. 2–23, 2007.

[14] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122–130, 2006.

[15] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer Science & Business Media, 2009.

[16] C.-L. Chen and C.-T. Li, "Dynamic session-key generation for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2008, no. 1, Article ID 691571, 2008.

[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] G. Zhao, "Wireless sensor networks for industrial process monitoring and control: a survey," *Network Protocols and Algorithms*, vol. 3, no. 1, pp. 46–63, 2011.

[19] O. K. Sahingoz, "Large scale wireless sensor networks with multi-level dynamic key management scheme," *Journal of Systems Architecture*, vol. 59, no. 9, pp. 801–807, 2013.

[20] B. Premamayudu, K. V. Rao, and P. S. Varma, "A novel pairwise key establishment and management in hierarchical wireless sensor networks (HWSN) using matrix," in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India—Vol I*, pp. 425–432, Springer, 2014.

[21] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[22] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology—CRYPTO '85 Proceedings*, vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, Springer, 1986.

[23] J. Deng, R. Han, and S. Mishra, "Secure code distribution in dynamically programmable wireless sensor networks," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 292–300, ACM, Nashville, Tenn, USA, April 2006.

[24] P. K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the deluge network programming system," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 326–333, Nashvill, Tenn, USA, April 2006.

[25] S. Lee, H. Kim, and K. Chung, "Hash-based secure sensor network programming method without public key cryptography," in *Proceedings of the Workshop on World-Sensor-Web (WSW '06)*, Boulder, Colo, USA, 2006.

[26] P. E. Lanigan, R. Gandhi, and P. Narasimhan, "Sluice: secure dissemination of code updates in sensor networks," in *Proceedings of the 26th IEEE Internationa26th IEEE International Conference on Distributed Computing Systems (ICDCS '06)*, 53 pages, IEEE, July 2006.

[27] M. L. Das and A. Joshi, "Dynamic program update in wireless sensor networks using orthogonality principle," *IEEE Communications Letters*, vol. 12, no. 6, pp. 471–473, 2008.

[28] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 128–143, 2006.

[29] W. Wang, S. Zhang, G. Duan, and H. Song, "Security in wireless sensor networks," in *Wireless Network Security*, pp. 129–177, Springer, Berlin, Germany, 2013.

[30] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless sensor network security: a survey," in *Security in Distributed, Grid, Mobile, and Pervasive Computing*, Y. Xiao, Ed., vol. 1, p. 367, 2007.

[31] G. Wang, S. Kim, D. Kang, D. Choi, and G. Cho, "Lightweight key renewals for clustered sensor networks," *Journal of Networks*, vol. 5, no. 3, pp. 300–312, 2010.

[32] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 197–213, Berkeley, Calif, USA, May 2003.

[33] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, 2005.

[34] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, 2005.

[35] R. Blom, "An optimal class of symmetric key generation systems," in *Advances in Cryptology*, pp. 335–338, Springer, Berlin, Germany, 1985.

[36] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Advances in Cryptology—CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings*, vol. 740 of *Lecture Notes in Computer Science*, pp. 471–486, Springer, Berlin, Germany, 1993.

[37] M. Eltoweissy, A. Wadaa, S. Olariu, and L. Wilson, "Group key management scheme for large-scale sensor networks," *Ad Hoc Networks*, vol. 3, no. 5, pp. 668–688, 2005.

[38] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered

sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2006.

[39] M. Eltoweissy, M. H. Heydari, L. Morales, and I. H. Sudborough, "Combinatorial optimization of group key management," *Journal of Network and Systems Management*, vol. 12, no. 1, pp. 33–50, 2004.

[40] C.-L. Chen and I.-H. Lin, "Location-aware dynamic session-key management for grid-based wireless sensor networks," *Sensors*, vol. 10, no. 8, pp. 7347–7370, 2010.

[41] M. H. Eldefrawy, M. K. Khan, and K. Alghathbar, "A key agreement algorithm with rekeying for wireless sensor networks using public key cryptography," in *Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification (ASID '10)*, pp. 1–6, IEEE, Chengdu, China, July 2010.

[42] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[43] W. Zhang, S. Zhu, and G. Cao, "Predistribution and local collaboration-based group rekeying for wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1229–1242, 2009.

[44] S. Guo and A.-N. Shen, "A compromise-resilient pair-wise rekeying protocol in hierarchical wireless sensor networks," *Computer Systems Science and Engineering*, vol. 25, no. 6, pp. 397–405, 2010.

[45] Y. Zhang, C. Wu, J. Cao, and X. Li, "A secret sharing-based key management in hierarchical wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 406061, 7 pages, 2013.

[46] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[47] J. Nam, K.-K. R. Choo, S. Han, M. Kim, J. Paik, and D. Won, "Efficient and anonymous two-factor user authentication in wireless sensor networks: achieving user anonymity with lightweight sensor computation," *PLoS ONE*, vol. 10, no. 4, Article ID e0116709, 2015.

[48] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[49] C.-T. Li, C.-Y. Weng, and C.-C. Lee, "An advanced temporal credential-based security scheme with mutual authentication and key agreement for wireless sensor networks," *Sensors*, vol. 13, no. 8, pp. 9589–9603, 2013.

[50] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10081–10106, 2014.

[51] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '33)*, p. 10, January 2000.

[52] J. Rehana, "Security of wireless sensor network," Tech. Rep. TKK-CSE-B5, Helsinki University of Technology, Helsinki, Finland, 2009.

[53] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63–75, 2010.

[54] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security*, vol. 1, no. 2, 2010.

[55] S. Lee and K. Kim, "Sensor authentication scheme for clustering routing protocols in wireless sensor networks," in *Proceedings of the IEEE Sensors*, pp. 1819–1822, IEEE, November 2010.

[56] M. Stamp, *Information Security: Principles and Practice*, John Wiley & Sons, 2011.