ID-based Sensor Node Authentication for Multi-Layer Sensor Networks

Soonhwa Sung and Jaecheol Ryou

Abstract: **Despite several years of intense research, the security and cryptography in wireless sensor networks still have a number of ongoing problems. This paper describes how identification (ID)-based node authentication can be used to solve the key agreement problem in a three-layer interaction. The scheme uses a novel security mechanism that considers the characteristics, architecture, and vulnerability of the sensors, and provides an ID-based node authentication that does not require expensive certificates.**

The scheme describes the routing process using a simple ID suitable for low power and ID exposure, and proposes an ID-based node authentication. This method achieves low-cost communications with an efficient protocol. Results from this study demonstrates that it improves routing performance under different node densities, and reduces the computational cost of key encryption and decryption.

Index Terms: **Identification (ID)-based key agreement, ID-based node authentication, ID-based node authentication protocol, routing function, sensor node.**

I. INTRODUCTION

The positions of the sensor nodes in a wireless sensor networks (WSN) do not need to be engineered or predetermined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, it also indicates that the sensor network protocols and the algorithm must possess self-organizing capabilities. Instead of sending raw data to the nodes responsible for fusion, they use their processing abilities to locally perform out simple computations and transmit only the required, partially processed data [1].

These features require fast routing, limited power consumption, simple computation, low-memory, and secure capability. The security challenges of WSNs lie in the conflict between minimizing resource consumption and maximizing security. The capabilities and constraints of the sensor node hardware influence the type of security mechanisms that sensor node platforms are able to host. Because the amount of additional energy consumed to protect each message is relatively small, the greatest consumer of energy in the security realm is key establishment [2].

Protocols such as identification (ID)-based symmetric keying have limited application until the network's routing infrastructure has been sufficiently established. Individually, other protocols such as public-key group and pairwise keying protocols consume too much energy. In addition, [3] is based on public key encryption and thus requires all the nodes to be able to perform the necessary computations. This may not be feasible for energy-limited sensor nodes.

Recently, a number of studies have sought a practical way to use public-key cryptography (PKC) in WSNs [4]–[7]. Their studies focused primarily on the optimization of PKC. Although computing cost is still a crucial problem for PKC systems, the results in [5] indicate that elliptic curve cryptography (ECC) provides some advantages with respect to memory and computing cost, and hence is suitable for WSNs. Therefore, given the studies regarding public key systems, it would be interesting to investigate the use of ID-based encryption (IBE) in a WSN. However, key authentication in the context of a WSN is still an open problem, because this type of network cannot manage the computational demands of a conventional public key infrastructure (PKI). Furthermore, the proposed techniques are not applicable to every context.

Previous ID-based key management methods for fast node identification and lightweight key management have provided secure communications only after node identification keys have been generated. Recent ID-based research [8], [9] has proposed an ID-based key authentication method that does not require pre-distribution; however it has a disproportionate responsibility for delegating authentications. An IBE using a Weil pairing [10] attempted to decrease the computational costs, but the cost remained similar to a previous public key system because of the Weil pairing.

Analyzing encryption overhead for the sensor network nodes [11] requires key pre-distribution and thus does not provide perfect communication. Public key cryptography [12] requires a significant amount of computation to authenticate and encrypt sensor nodes. Thus, to locate an alternative, IBE has recently been studied [8]–[10], [13]–[15]. IBE was first proposed by Shamir [16], and IBE using the Weil pairing [10] was introduced in 2001. Subsequently, extensions including encryption, signature, and authentication schemes from the Weil pairing have been proposed [17]–[19].

Motivated by this, [20] and [21] have used IBE for key distribution. However, none of the studies demonstrated the feasibility of computing IBE primitives in resource constrained nodes.

Therefore, this paper proposes an ID-based node authentication and a novel routing scheme that uses a simple ID for simple

Manuscript received May 7, 2014.

This work was supported by the National Research Foundation of Korea (NRF) and the Center for Women In Science, Engineering and Technology (WISET) Grant funded by the Korean Governmaent (Program for Returners into R&D by the Ministry of Science, ICT & Future Planning (MSIP)).

This research was partly supported by the R&D program of MSIP (Ministry of Science, ICT and Future Planning) [Project No. 10047528] and the National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development.

Soonhwa Sung is with Software Research Center (SOREC), Chungnam National University Republic of Korea, email: shsung@cnu.ac.kr.

Jaecheol Ryou is with Dept. Computer Science and Engineering, Chungnam National University, Republic of Korea, email: jcryou@cnu.ac.kr.

Digital object identifier 10.1109/JCN.2014.000065

key management.

The remainder of this paper is organized as follows: Related work is presented in Section II, a novel routing scheme with three layers is presented in Section III, ID-based node authentication is detailed in Section IV, and the methods are evaluated in Section V. Section VI concludes the paper.

II. RELATED WORK

The security requirements for sensor nodes include authentication, integrity, freshness, availability, and confidentiality.

Authentication is the process of verifying the identity of someone or something. The three types of cryptographic functions used for authentication are hash functions, secret key functions, and public key functions. In WSNs, it is usually assumed that public key cryptography cannot be used because of its elaborate constraints. This suggests that two communicating entities must use secret key functions and a hash function.

Recently, a hierarchical WSN security protocol was proposed in [22]. This scheme employs hash functions, hash key chains, and symmetric keys. Each sensor and the base station share a secret hash key chain. The sensor encrypts the data and sends it to the cluster head (CH). The CH collects the data from the sensor nodes and then retrieves the secret keys from the base station. The CH decrypts the encrypted message and then sends it to the base station.

This scheme has several advantages. First, it reduces the storage overhead, as each sensor node only stores three keys. Second, it reduces the probability of a successful guessing attack, because the sensor nodes change keys once per transmission. Finally, it uses a two-way challenge and response authentication method to prevent replay attacks.

However, this scheme also has several disadvantages. First, CHs can disclose all the secret keys of the sensor nodes in their cluster. A single compromised CH could affect a large number of sensor nodes. Second, the CHs must retrieve the sensor node's secret key for every data transfer. This results in communication overhead. Third, the sensor nodes must frequently change the secret keys for each instance of data collection [23].

In order to mitigate these disadvantages, we propose an IBE scheme that results in a lightweight system that does not have the key distribution problem.

IBE specifies the cryptosystem in which both the public and private keys are based on the identities of the users. The idea of IBE as formulated by Shamir, affirms that a user's public key is an easily calculated function of their identity, while a user's private key can be calculated for them by a trusted authority, called a private key generator (PKG).

While classic PKI schemes use certificates to bind identities to their public keys, IBE schemes have an implicit binding between an identity and its public key. The main idea in IBE is to eliminate a public key that is derivable from some known aspect of a user's identity, such that public key directories are unnecessary.

Therefore, the authentication of identities within the system is crucial to its overall security, because the public keys are derived from identity. This reliance on authentication enables all parties to verify the signatures of any member in the system without maintaining a dedicated database for the keys of other parties, resulting in a lighter system. In addition, IBE systems eliminate the key distribution problem because the public key required to verify a signature is derivable from the identity. In most use cases, identity is readily available to a verifying party, as are all other parameters required for verification.

Compared to traditional PKI, IBE has a comparable or higher security level. Specifically, private keys in IBE are derived from the identities assigned by the PKG using a master key, while in PKI both the public and private keys are created by the users themselves. This is one reason why PKI is not considered a good choice for key agreement and encryption in WSNs.

Boneh and Franklin [10] presented an IBE scheme based on the properties of bilinear map pairings on elliptic curves, which is the first fully functional, efficient, and provably secure IBE scheme. Subsequently, numerous cryptographic schemes based on this work have been proposed.

Many schemes have focused on robustness against possible attacks or IBE performance compared to traditional symmetric cryptography, and claim that IBE would significantly improve performance. However, they have not provided details for local key generation and transmission, or specified how packets are encrypted and signed.

Therefore, this scheme details how sensor nodes can use IBE for transmission and authentication.

III. NOVEL ROUTING WITH THREE LAYERS

A. Perspective

It is very inefficient for every sensor node to report back its raw data, because every data packet must traverse many hops to reach the base station. In addition, sensor nodes are often constrained by scarce memory, computation, communication, and power resources. Thus, reporting raw data is often undesirable.

One of the main challenges in WSNs is determining how to efficiently process and aggregate the data in the network, instead of wasting energy by sending a large amount of raw data in response to a query. To process and aggregate the data in the network, every node should be assigned a simple ID, because data must be routed rapidly. In addition, to route efficiently, all sensor nodes should establish straightforward data communication paths.

The routing function temporarily hides the ID from the raw data, which prevents possible attacks. In addition, a simple ID enables the scheme to reduce the cost and power required for WSNs. Instead of sending the raw data, it classifies IDs into four types and transmits only to the nodes required for data fusion. Thus, the enrire system temporarily hides the ID and minimizes its exposure to attack.

B. Three-Layer Interaction

This scheme has distributed interaction in a zone (i.e., a set of nodes located close to each other). Clusters (i.e., a group of loosely coupled nodes that work closely together) are built within each zone. Inter-zone clusters are not allowed. A zone in a WSN has a three-layer network, as shown in Fig. 1. Each layer in Fig.1 interacts according to requirements. Three layers support the preparation of routing and authentication for secure

Table 1. Notation.

Symbol	Meaning	Symbol	Meaning
ID	node ID	RPC	Clustering response message
N_{new}	New node	ERQC	Emergency clustering request message
N_m	Mobile node	ERPC	Emergency clustering response message
N_R	Random node	K_{pub}	Public key
R( )	Routing function	K_{priv}	Private key
H( )	Hash function	K_{pub} sequence	Public key sequence
CH	Cluster head	KGC	Key generation center
RQA	Authentication request message	NBR (N)	Number of neighbor nodes
RPA	Authentication response message	NBR (CH)	Number of neighbor CH
RQC	Clustering request message	ACL	Access control list
s	Master key	APM	Access privilege mask

interactions.

The advantages of the proposed three layers are as follows:

1. The assigned nodes in a cluster can maintain network communications without insertions into, or deletions from, the routing table, because every sensor node in a cluster, except the CH, has a simple ID assigned to it and has a unique ID represented by "1" by routing function. The scheme does not require directories for public keys such as ID, and minimizes the number of bytes required to code IDs.
2. The routing function temporarily hides the ID from the raw data, thereby preventing fraud from possible attacks. Therefore, the entire system, which temporarily hides the ID, minimizes the exposure of the ID to an attack.
3. The simple ID enables the scheme to improve the low-cost and low-power performance for sensor networks. It classifies the four ID types, and only transmits to the required nodes responsible for the fusion, instead of sending the raw data.
4. The distributed interactions of three layers improve weaknesses such as easy failures in a sensor network, because the interactions continue to operate the *K*-proxies algorithm, routing function, and ID-based node authentication protocol, even though the nodes of a cluster can easily fail, leave, join, or die.
5. The service reduces the number of unnecessary IDs carried by each node, because each layer communicates using the virtuous cycle requirements of a cluster.

C. Proxy Candidate Layer

This layer contains the raw data and a simple ID. ID services include ID issue, renewal, and revocation. The serviced ID acts as a proxy candidate.

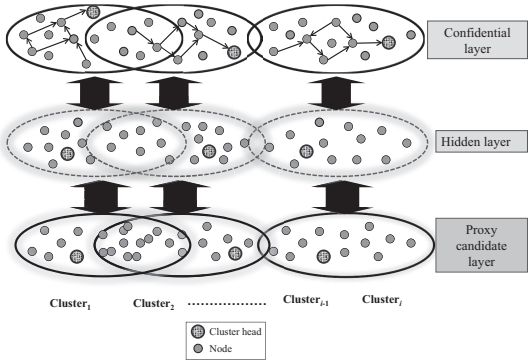

Fig. 1. Interactions of the three layers in a zone.

- **ID issue**
  Technically, this operation is the same as ID renewal. However, it raises additional security concerns. Once an entity obtains its initial ID, it earns the trust of the entire network. Hence, a well-defined ID issuing policy is required. During the network bootstrapping phase, the entities can obtain their initial IDs (1, *i*, -1, and -*i*) from a trusted organization, i.e., the key generation center (KGC). The KGC is the trusted third party in certificateless public key cryptography, comparable to the PKG in IBE.
- **ID renewal**
  Once the initial ID is issued to an entity, it must be renewed within time Trenew. The entity may also need to renew its ID after it updates the personal key pair. To renew its ID, a network entity must present a current valid ID and future expiration time $T <$ (current time + Trenew) for the new ID.
- **ID revocation**
  If an ID is considered to be compromised, a counter-ID is flooded over the network. Each node only needs to maintain a subset of the counter-IDs within the past Trenew.

Each node operates the *K*-proxies algorithm that caches the message information of the sensors to assign the ID. Subsequently, each of the neighboring nodes routes the ID and responds to the original node, according to the results of the routing function.

The *K*-proxies algorithm for ID assignment is as follows:

***K*-proxies Algorithm:** Generate *k* request to discover k proxies

1. **Define:**
2. *w, x, y, z*: four end-nodes to set up ID assignment
3. v: proxy candidate
4. ID_1: ID for the node *w*
5. ID_i: ID for the node *x*
6. ID_1: ID for the node *y*
7. ID_1: ID for the node *z*
8. ID_{self}: ID for itself
9. N_x: 1-hop neighbors of any node *x*
10. Re: request to set up ID assignment
11. *K*: number of proxies that must be found
12. ACK: acknowledge a node to be true
13. **proxies(*k*):** executed at randomly selected node *w, x, y,* or *z*
14. for $i = 1$ to k do
15. randomly select a node in N_x and send Re

16. end for
17. if receive positive ACK from node v then
18. register v as a proxy
19. end if
20. **Check(**Re**):** executed at all nodes receiving Re
21. if Re is not seen before then
22. if $\mathrm{ID}_{\mathrm{self}} \bigcap \mathrm{ID}_w$ is not empty then
23. if $\mathrm{ID}_{\mathrm{self}} \bigcap \mathrm{ID}_x$ is not empty then
24. if $\mathrm{ID}_{\mathrm{self}} \bigcap \mathrm{ID}_y$ is not empty then
25. if $\mathrm{ID}_{\mathrm{self}} \mathrm{ID}_z$ is not empty then
26. register itself as a proxy for nodes *w, x, y,* and *z*
27. send back positive ACK to nodes *w, x, y,* and *z*
28. exit the procedure
29. end if
30. end if
31. end if
32. end if
33. end if
34. randomly select a neighbor other than the sender to forward Re.

D. Hidden Layer

The sensor nodes of this layer are assigned a unique ID represented by "1" by a routing function R( ). The input of R( ) is a variable number of IDs. Each node N of the network entities selects one ID (1 or *i* or ?1 or ?*i*). This input ID is generated by the proxy candidate layer. The input of R( ) is not fixed because sensor nodes leave, join, die, and fail, regardless of their lifetimes. Its output is a fixed unique ID represented by "1" because each node selects an R( ). The nodes can select any ID, regardless of all nodes' location and lifetime; therefore, the output has a fixed number of IDs.

This layer protects the ID from attacks. Because the nodes in this layer are only assigned a unique ID represented by "1", an attacker cannot determine whether they are true or false.

A CH collects sensor readings from surrounding nodes and forwards the unique ID representing "1" to another CH.

In Fig. 2, R( ) does not require a routing table to code these IDs. Each assigned ID carries out a fourth degree operation. That is, $1\times1\times1\times1=1, i\times i\times i\times i=1, -1\times-1\times-1\times-1=1, -i\times-i\times-i\times-i=1$, and R( ) is as follows:

$$\begin{aligned} R_1(1) &= R(1)\times R(1)\times R(1)\times R(1)=1, && (1)\\ R_i(i) &= R(i)\times R(i)\times R(i)\times R(i)=1, && (2)\\ R_{-1}(-1) &= R(-1)\times R(-1)\times R(-1)\times R(-1)=1 && (3)\\ R_{-i}(-i) &= R(-i)\times R(-i)\times R(-i)\times R(-i)=1, && (4)\\ R_{x_k}() &= \prod_{k=1}^{4} R_{x_k}x=1,i,-1,-i, \\ k &= k+1(1\le k\le 4). && (5) \end{aligned}$$

As a consequence of R( ), the same ID is assigned to the nodes in the hidden layer. Therefore, an attacker cannot determine whether they are true or false.

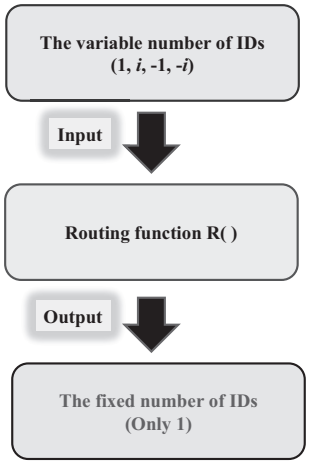

Fig. 2. Diagram of R().

E. Confidential Layer

Because a CH interfaces WSNs to the outside world, the compromise of a significant number of them could render the entire network useless. For this reason, a CH may be trusted by the communication components. Nodes may rely on the routing information from a CH and trust that the data sent to it will be accurately combined with other data when it is forwarded to another CH. Therefore, the communication among all clusters must be trusted for efficient routing.

For secure communications among clusters, confidential data aggregation for transmission is supported by the scheme described in [24]. This scheme scheme enables the CH to perform the aggregation directly on cipher texts. To preserve the security of the cluster, including the CH in the confidential layer, this paper proposes ID-based node authentication.

IV. ID-BASE NODE AUTHENTICATION

A. Key Generation

The key agreements of the formed WSNs [25] proceed between each node and the KGC, while the proposed key agreements proceed between the CH and the KGC. A cluster has one CH, the node with the most neighbors. A CH also includes a secret master key to derive the ID-based private keys. The CH secret key and master key are under the control of the KGC.

The sensor node receives a public key (ID address) and private (secret) key from the KGC, and communicates with the CH as a legitimate node for access. A CH receives the cluster master key from the KGC that derives the identities of the nodes in a cluster. The KGC generates keys only; the CH manages them.

Because the topology of WSNs changes dynamically, this scheme updates keys (re-keying) periodically. The re-keying period is calculated based on a mobility factor [30]. In the re-keying scheme, two keys are updated. First, the cluster master keys are shared between each CH and all its cluster members. Second, the pairwise keys are shared between the public and private key of a cluster. The pairwise keys (public and private) can be shared by any sensors in a cluster among themselves and used to generate cluster master keys. Therefore, data aggrega-

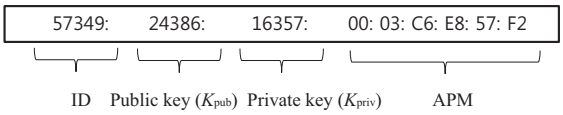

Fig. 3. ACL example.

tion security can be achieved through the cluster master keys.

In each cluster, the CH broadcasts a "Hello" message that includes its ID and the $K_{\rm pub}$ sequence (a sequence number used only once in the sensor network's lifetime) to all the neighboring sensors. When a neighboring sensor receives this message, it replies to the CH with a message authentication code (MAC) encrypted with a pairwise key. Once all the pairwise keys in a cluster have been updated, the new cluster master key can be transmitted to each cluster member through the corresponding pairwise key.

The KGC generates a public key and a private key from $\{0,1\}^*$, i.e., the bit strings generated by a pseudorandom binary sequence generator. In addition, the KGC generates the cluster master key to produce a session key for the private keys of the nodes in a cluster, while the CH manages the access control list (ACL) for legitimate nodes.

In Fig. 3, the ACL is composed of the ID, public key, private key, and the access privilege mask (APM). It consists of binary bit sets that specify node information and the permission service.

The cluster public key ($K_{\rm pub}$) is produced by hashing after adding each public key in the cluster. It is dependent on the cluster size (number of nodes in a cluster) and the point in time. The cluster private key ($K_{\rm priv}$) is produced by hashing after XORing (exclusive ORing) each private key in a cluster. The cluster master key is produced by hashing after XORing the hash of each private key in a cluster.

Cluster public key:

$$K_{\rm pub} = H\Big(\sum_{k=1}^{i/4} K_{\rm pub}(1) + \sum_{k=1}^{i/4} K_{\rm pub}(-1) + \sum_{k=1}^{i/4} K_{\rm pub}(i) + \sum_{k=1}^{i/4} K_{\rm pub}(-i)\Big). \quad (6)$$

Cluster private (secret) key:

$$K_{\rm pub} = H\Big(K_{\rm pub(1)} \oplus K_{\rm pub(2)} \oplus K_{\rm pub(3)} \oplus \cdots \oplus K_{\rm pub(i)}\Big). \quad (7)$$

Cluster master key:

$$Km_{\rm priv} = H\{(H(K_{\rm priv(1)}) \oplus H(K_{\rm priv(2)}) \oplus H(K_{\rm priv(3)}) \oplus \cdots \oplus H(K_{\rm priv(i)}))\}. \quad (8)$$

In Fig. 4, the ID-based key distribution supports the generation of the zone keys in a tree type. The zone keys are identical to the cluster keys in terms of the key generation methods, but

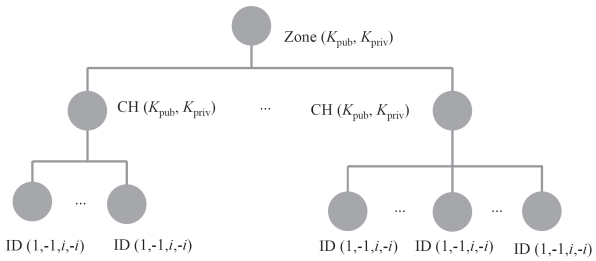

Fig. 4. ID-based key distribution.

they are different from the cluster keys in the derivation of the private keys. The cluster private keys are identified by the PKG using a master key, whereas zone private keys are identified by the KGC in certificateless public key cryptography. This is because zone private keys conform to ID-based key distribution.

Zone public key:

$$K_{\rm pub} = H\Big(\sum_{k=1}^{i/4} K_{\rm pub}(1) + \sum_{k=1}^{i/4} K_{\rm pub}(-1) + \sum_{k=1}^{i/4} K_{\rm pub}(i) + \sum_{k=1}^{i/4} K_{\rm pub}(-i)\Big). \quad (9)$$

Zone private (secret) key:

$$K_{\rm pub} = H\Big(K_{\rm pub(1)} \oplus K_{\rm pub(2)} \oplus K_{\rm pub(3)} \oplus \cdots \oplus K_{\rm pub(i)}\Big). \quad (10)$$

Zone master key:

$$Km_{\rm priv} = H\{H(K_{\rm priv(1)}) \oplus H(K_{\rm priv(2)}) \oplus H(K_{\rm priv(3)}) \oplus \cdots \oplus H(K_{\rm priv(i)})\}. \quad (11)$$

B. ID-based Encrypyion Algorithm

In this scheme, the four phases of an ID-based key agreement form the IBE Algorithm.

Set up phase:

A new node $N_{\rm new}$ receives its ID from the KGC and broadcasts its "Hello" message to the neighboring nodes. After they receive the message, they exchange IDs to verify that it is a legitimate node. If the ID is compromised, it revokes itself from the counter-ID withdrawal component of the KGC. This is because the CH supervises the ACL based on the KGC information, and the ID changes whenever the topology changes.

If $K_{\rm priv} = (K_{\rm pub})^s$ of an ID, where s is the CH master key, the entity intends to establish a communication channel.

To e-key the private key for security, the KGC selects the random $K'_{\rm priv}$ of a new node with a short lifetime and broadcasts it to all the nodes in a cluster. If $K'_{\rm priv} = (K'_{\rm pub})^s$ of the ID, the entity intends to establish a communication channel.

If a random node NR receives a public key (ID address) and

a private key from the KGC, it takes the security parameter $k \in Z^+$ and returns the k system parameter and master key. The system parameters include descriptions of the finite message space M and finite ciphertext space C. Intuitively, the system parameters will be publicly known, while the master key will be known only to the KGC and CH.

Extract phase:

A mobile node N_m takes input k, the master key, and arbitrary ID and returns private key K_{priv} to another node. The arbitrary ID is an arbitrary string that is used as a public key, and K_{priv} is the corresponding private decryption key.

Encryption phase:

The node takes the inputs *k*, *D*, and $m \in M$

(where m is a random message and M is the message group).

It returns ciphertext $c \in C$ (where c is a random ciphertext and C is the ciphertext group).

Decryption phase:

The node takes the inputs $k, c \in C$, and private key K_{priv}.

It returns $m \in M$.

$\forall m \in M$: $\text{Decrypt}(k, c, K_{\text{priv}}) = m$ where $c = \text{Encrypt}(k, m, K_{\text{pub}})$.

ID-based encryption algorithm

Algorithm notation is as follows: G^*: the set $G^* = G|O$ (where O is the identity element in group G) Z^+: the set of positive integers H: hash function.

1. **Setup:** Given a security parameter $k \in Z^+$, the algorithm proceeds as follows.
   Step 1: Select a random $\alpha \in G$.
   Step 2: Select a random $s \in Z^+$.
   Step 3: Select cryptographic hash functions for some n (where n is the length of the plaintext).
   H: $\{0,1\}^* \to G^*$ H: $G \to \{0,1\}^n$ For the security proof, we view all hash functions as random oracles. The message space is $M = \{0,1\}^n$.
   The ciphertext space is $C = G^* \times \{0, 1\}^n$.
   The output system parameters are $= \{G, n, \alpha, H1, H2\}$.
   The master key is $s \in Z^+$.
2. **Extract:** For a given string $ID \in \{0,1\}^*$, the algorithm extracts as follows:
   Step 4: Compute $K_{\text{pub}} = \text{H(ID)} \in G^*$.
   Step 5: Set the private key K_{priv} to be $K_{\text{priv}} = (K_{\text{priv}})^s$, where s is the master key.
3. **Encrypt:** To encrypt $m \in M$ under the public key ID, the algorithm performs the following:
   Step 6: Compute$K_{\text{pub}} = H(ID) \in G^*$.
   Step 7: Select a random $\sigma \in \{0,1\}^n$.
   Step 8: Set the cipher to be $\forall c \in C$
   c = $(k, K_{\text{pub}}, m \oplus H(K_{\text{pub}})^s)$, where $ID \in G$, k is a system parameter.
4. **Decrypt:** To decrypt $c \in C$ using the private key $K_{\text{pub}} \in G^*$, the algorithm proceeds as follows:
   Step 9: $K_{\text{priv}} = (K_{\text{pub}})^s = (H(ID))^s$
   Step 10: Compute $c \oplus H(K_{\text{priv}}) = \sigma$.
   Step 11: Compute $c \oplus H(\sigma) = m$.
   Step 12: Set $\gamma = H(\sigma, m)$. Test that $c = \gamma\alpha$. If not, reject the ciphertext.
   Step 13: Output m as the decryption of c.

The following section proposes an ID-based node authentication protocol, in which the CH can independently authenticate keys without key agreement from the KGC before the secure communication.

C. ID-based Node Authentication Protocol

In a general cryptographic system, if a master key is exposed, all of the private keys of the users are exposed. However, in this scheme, none of the node private keys are exposed, if the master key is exposed, and vice versa. This is because an RSA algorithm-based system authenticates the key after a secure communication is established, but this scheme authenticates the keys before a secure communication is established. That is, the CH supervises the ACL in a one-way function to authenticate the key before the establishment of a secure communication, and the KGC only generates the keys that are not required for the establishment of a secure communication. Therefore, the CH can independently authenticate keys without KGC key agreement before secure communication.

Fig. 5 illustrates this process in more detail (see Table 1 for notation): (1) After the KGC generates the keys (K_{pub}, K_{priv}, master key(s)), it sends the key (ID, K_{pub}, K_{priv}) to N_R (N_{new}) and N_m, and sends the master key(s) to the CH. (2) A mobile node N_R sends a clustering request message (RQC) to the CH and (3) the CH sends the key information including the ID address of N_R to N_R. (4) After the KGC confirms the private key including the ID address of NR, if it accepts the key information, it sends the authentication response message (RPA) to the CH; (5) if not, it sends the clustering response message (RPC) to the CH. (6) When a new node N_{new} is added to a cluster, if it has the same ID of a legitimate NR, then CH sends the RPA to N_{new}.

(7) If the CH is unsure whether N_m is an attacker, it sends an emergency RQC (ERQC) to the KGC. (8) The KGC then sends the emergency RPC (ERPC) with the key information including the ID address of N_m to the CH, which authenticates the private key of Nm using the master key. (9) If the CH authenticates the node, it sends the RPA to N_m, and if not, N_m deletes itself.

In the same manner, (10) if N_{m1} is unsure whether N_{m2} is an attacker, N_{m1} sends an ERQC to the CH.

(11) The CH sends the ERQC to the KGC, which compares the values of the ACL with the values of N_{m2}. (12) If the values of N_{m2} are not the same as the values of the ACL, the KGC sends the ERPC to the CH and N_{m2} deletes itself. (13) If the values of N_{m2} are the same as the values of ACL, the KGC sends an RPC with the updated key information of N'_{m2} to the CH.

V. EVALUATION

The computer system used for simulating the proposed IBE scheme was an Intel Pentium E2220 2.40 GHz with 1.75 GB RAM, running the TinyOS [27]operating system that provides low-level event and task management. The default simulation testbed had 30 sensors randomly distributed over a 2 × 2 m area. Each simulation ran for 600 s, and each result was averaged over five random network topologies created by QualNet [28].

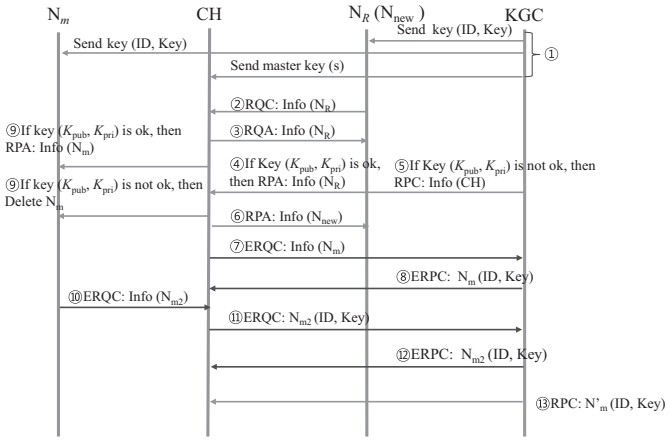

Fig. 5. ID-based node authentication protocol.

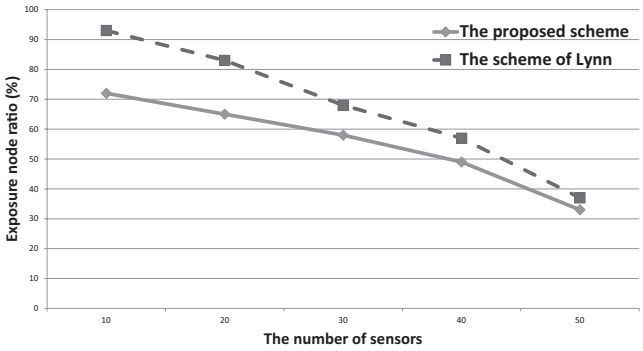

Fig. 6. Exposure node ratios under different node densities.

Table 2. CPU times for the proposed and Yang *et al.* schemes.

Key length (bit)	Encryption (s)		Decryption (s)	
	Proposed scheme	Yang *et al.* scheme	Proposed scheme	Yang *et al.* scheme
64	3.0		2.9	
128	4.9	6.0	4.6	5.2
160	5.1	6.8	4.9	5.2
256	7.6	9.5	6.8	7.2
512		10.7		8.2

The proposed IBE scheme was compared with the IBE schemes of Yang *et al.* [8] and Lynn [18]. Table 2 shows the average CPU encryption and decryption times of all nodes for the proposed and Yang *et al.* schemes. The results show that, for the Yang *et al.* scheme, the computation time increases with the length of the keys. At the same security level (160-bit key), the proposed scheme required 5.1 s for encryption, while the Yang *et al.* scheme required 6.8 s. Moreover, key management in the Yang *et al.* scheme is more complex than in the proposed scheme, because it uses key agreement and an encryption scheme based on elliptic-curve cryptography.

The results in Table 2 suggest that, in the Yang *et al.* scheme, the cost of key computations using elliptic-curve cryptography is much greater than the simple key computations in the proposed scheme. The Yang *et al.* scheme requires four hash-function evaluations, two XOR operations, and one map computation for encryption, whereas the proposed scheme requires two hash-function evaluations, one XOR operation, and one map computation. In addition, the Yang *et al.* scheme does not provide mutual authentication.

For a fixed 2×2 m routing area, the number of sensors was varied from 10 to 50 in increments of 10. Fig. 6 shows the exposure node ratio for different node densities under the proposed and Lynn IBE schemes. It also shows that the exposure node ratio of both the proposed and Lynn schemes decreases with the increase of sensor density.

In the proposed scheme, when the sensor density increases, there are more sensors in each cluster and more candidates to relay the packets to the CH, and hence they require more routing functions. This is why the exposure node ratio under the proposed IBE scheme decreases with a sensor density increase.

In Fig. 6, as the number of sensors increase, the two exposure node ratios approach each other. This is because the proposed scheme efficiently routes with simple routing functions and quickly prepares sensor mobility. Although the Lynn scheme has a high level of security with new encryption and decryption algorithms, it is less efficient than the proposed scheme because encryption and signing are separate operations.

VI. CONCLUSION

The goal of public key authentication is to ensure that the binding between an identity and a public key is authentic. The certificate approach is designed for users who do not have a pre-established trust relationship that enables them to authenticate each other's public key. They achieve this using a third party, the certificate authority (CA), with whom they both have a trust relationship. However, if the two users already have a trust relationship, it is not necessary to use the certificates. In WSNs, the nodes have previously authenticated their deployment, because these nodes usually belong to the same administrative entity.

Therefore, this scheme facilitates a novel security mechanism in which the KGC functions as the trusted third party of certificateless public key cryptography after the three layers have interacted. It authenticates keys between the CH and the KGC, while previous cryptographic schemes authenticated keys between the KGC and all sensors in a cluster. Thus, the private keys of all the nodes are not exposed, although a master key is exposed. This is because the CH supervises the node keys from the ACL, and the KGC only generates them using the ID-based node authentication system.

This scheme has advantages in terms of key management, routing, and CPU time for encryption and decryption. Future work should study the sham attack on ID-based key authentication in WSNs.

ACKNOWLEDGMENTS

The authors appreciate Prof. Cheong Youn and Prof. Eunbae Kong for their helpful research supporting.

REFERENCES

[1] I. Akyildiz *et al.*, "A survey on sensor networks," *IEEE Commun. Mag.*, Aug. 2002.

[2] D. Carman, P. Kruus, and B. Matt, "Contraints and approaches for distributed sensor network security," *NAI Lab., Tech. Rep. #00-010*, June 2000.

[3] L. Zhou and Z. J. Hass, "Securing ad hoc networks," *IEEE Netw.*, vol. 13, no. 6, 1999.

[4] G. Gaubatz, J. Kaps, and B. Sunar, "Public keys cryptography in sensor networks-revisited," in *Proc. ESAS*, 2004.

[5] N. Gura *et al.*, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Proc. CHES*, Aug. 2004.

[6] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in *Proc. IEEE SECON*, Oct. 2004, pp. 71–79.

[7] W. Du, R. Wang, and P. Ning, "An efficient scheme for authenticating public keys in sensor networks," in *Proc. MobiHoc*, USA, May 2005, pp. 58–67.

[8] G. Yang *et al.*, "Identity-based key agreement and encryption for wireless sensor networks," *J. China Universities of Posts and Telecommun.*, China, 2007.

[9] T. T. Huyen and E.-N. Huh, "A reliable 2-mode authentication framework for ubiquitous sensor network," *J. Korean Soc. for Internet Inform.*, 2008.

[10] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Crypto*, 2001, pp. 213–229.

[11] P. Ganesan *et al.*, "Analyzing and modeling encryption overhead for sensor network nodes," in *Proc. WSNA*, USA, Sept. 2003.

[12] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in *Proc. IEEE SECON*, Oct. 2004, pp. 71–79.

[13] X. Boyen, "Multipurpose identity-based signcryption, a Swiss army knife for identity-based cryptography," in *Proc. CRYPTO*, 2003, pp. 383–399.

[14] L. Chen and C. Kudla, "Identity-based authenticated key agreement protocols from pairing," in *Proc. IEEE CSFW*, 2003, pp. 219–233.

[15] B. R. Waters, "Efficient identity-based encryption without random oracles," in *Proc. EUROCRYPT*, 2005, pp. 114–127.

[16] A. Shamir, "Identity-based cryptography and signature schemes," in *Proc. CRYPTO*, Aug. 1985, pp. 47–53.

[17] F. Hess, "Efficient identity based signature schemes based on pairings," in *Proc. SAC*, 2003, pp. 310–324.

[18] B. Lynn, "Authenticated identity-based encryption," *IACR Cryptology ePrint Archive, Report 2002/072*, 2002.

[19] J. Horwitz and B. Lynn, "Toward hierarchical identity-based encryption," in *Proc. EUROCRYPT*, 2002, pp. 466–481.

[20] W. L. Zhang, W. Lou, and Y. Fang, "Securing sensor networks with location-based keys," in *Proc. IEEE WCNC*, 2005, pp. 1909–1914.

[21] B. Doyle *et al.*, "Security considerations and key negotiation techniques for power constrained sensor networks," *Comput. J.*, vol. 49, no. 4, pp. 443–453, 2006.

[22] C. Chen and C. Li, "Dynamic session key generation for wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, 2008.

[23] J. Zhang *et al.*, "A Dynamic Athentication Scheme for Hierarchical Wireless Sensor Networks," in *Proc. MobiQuitous*, 2012, pp. 186–197.

[24] S. Sung, "Confidential aggregation for wireless transmissions," in *Proc. ICOIN*, Feb. 2014, pp. 390–394.

[25] X. H. Le *et al.*, "An energy efficient access control scheme for wireless sensor networks based on elliptic curve cryptography," *J. Commun. Netw.* vol. 11, no. 6, pp. 599–606, Dec. 2009.

[26] F. Hu and S. Kumar, "QoS considerations in wireless sensor networks for telemedicine," in *Proc. ITCom*, 2003, pp. 217–227.

[27] TinyOS, TinyOS 1.1.0, [Online]. Available: http://tinyOS.net

[28] "QualNet network simulator, The Scalable Network Technology" [Online]. Available: http://www.qualnet.com

Soonhwa Sung received a Ph.D. degree in 2005 from the Department of Computer Engineering, Chungnam National University, Republic of Korea. From 2000 to 2005, she taught in the Department of Computer Web Informations, Daeduk College, Republic of Korea. From 2002 to 2005, she taught in the Department of Computer Engineering, Chungnam National University, Republic of Korea. From 2006 to 2011, she was a Visiting Professor at Chungnam National University, Republic of Korea. She was a Visiting Professor at Chungbuk National University, Republic of Korea in 2012. She is currently researching at the Software Research Center (SOREC), Chungnam National University, Republic of Korea. Her research interests include mobile information security, privacy protection, user authentication system for future internet, and sensor network security.

Jaecheol Ryou is a Professor in the Department of Computer Engineering at Chungnam National University in Korea. He received a B.S. degree in Industrial Engineering from Hanyang University in 1985, an M.S. degree in Computer Science from Iowa State University in 1988, and a Ph.D. degree in Electrical Engineering and Computer Science from Northwestern University in 1990. His research interests are Internet security and electronic payment systems.