Sensors **2014**, *14*, 17937-17951; doi:10.3390/s141017937

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

An Efficient Identity-Based Key Management Scheme for Wireless Sensor Networks Using the Bloom Filter

Zhongyuan Qin [1,2,*], Xinshuai Zhang [1,†], Kerong Feng [1,†], Qunfang Zhang [3,†] and Jie Huang [1]

[1] School of Information Science and Engineering, Southeast University, Nanjing 210096, China; E-Mails: shuaishuaizhang@yahoo.com.cn (X.Z.); fengkerong@163.com (K.F.); jhuang@seu.edu.cn (J.H.)

[2] Key Lab of Information Network Security, Ministry of Public Security, Shanghai 201204, China

[3] Computer Department, Nanjing Institute of Artillery Corps, Nanjing 211132, China; E-Mail: zhangqunfang520@163.com

[†] These authors contributed equally to this work.

[*] Author to whom correspondence should be addressed; E-Mail: zyqin@seu.edu.cn; Tel.: +86-139-5103-1560.

External Editor: Leonhard M. Reindl

Received: 24 June 2014; in revised form: 6 August 2014 / Accepted: 15 September 2014 / Published: 26 September 2014

Abstract: With the rapid development and widespread adoption of wireless sensor networks (WSNs), security has become an increasingly prominent problem. How to establish a session key in node communication is a challenging task for WSNs. Considering the limitations in WSNs, such as low computing capacity, small memory, power supply limitations and price, we propose an efficient identity-based key management (IBKM) scheme, which exploits the Bloom filter to authenticate the communication sensor node with storage efficiency. The security analysis shows that IBKM can prevent several attacks effectively with acceptable computation and communication overhead.

Keywords: key management; identity-based cryptography; bloom filter; security; wireless sensor network

---

1. Introduction

Wireless Sensor Networks (WSNs) are ripe for wide adoption in several applications, such as military, healthcare, automotive, research, and so on. For applications such as military, higher requirements on WSN security is needed. However, WSN security is a challenging problem, because of the openness of WSNs' network architectures, which enables adversaries to easily eavesdrop, intercept, inject and alter transmitted information. Besides, the existing computer network security mechanisms cannot be adopted in WSNs because of the restricted node resources and low communication bandwidth. Therefore, it is urgent to put forward low consumption key management schemes for WSNs.

Until now, key management schemes in WSNs were mainly based on symmetric cryptographic algorithms or public key cryptography algorithms. For the symmetric cryptographic algorithm, pool-based key predistribution [1] and the related improved schemes, [2,3] proposed probabilistic key pre-distribution schemes for pairwise key establishment. Their basic idea is that each node randomly picks a set of keys from a key pool before deployment, so that any two sensor nodes have a certain probability to share at least one common key. Blom [4] and Blundo *et al.* [5] proposed a matrix and a polynomial key generation schemes, respectively. However, most of these schemes cost much memory and significant communication overhead with a low security level. On the other hand, the key management schemes based on public key cryptography algorithms could provide much simpler solutions with much stronger security resilience compared with those based on symmetric cryptographic algorithms. However, public key cryptography algorithms require more computing capacity, and for this reason, the public key system was generally considered not applicable for energy-constrained WSNs. Recent works [6–9] have demonstrated the feasibility of public key cryptography algorithms on the resource-constrained sensor nodes. Specially, Oliveira *et al.* [7] implement pairings for sensor nodes based on the 8-bit/7.3828-MHz ATmega128L microcontroller (e.g., MICA 2 and MICAz motes), and they argue that pairing-based cryptography is indeed viable in resource-constrained nodes. Usually, they use Public Key Cryptography (PKC) schemes for bootstrapping security in WSNs, *i.e.*, for generating symmetric keys to communicate or key distribution. TinyPK [6] exploits the RSA cryptosystem to provide authentication and key exchange between an external party and a sensor network, but it needs a certificate authority (CA), and certificates are eliminated instead of the challenge-response protocol in which the public key is signed by CA's private key. Kui *et al.* [10] addressed the multiuser broadcast authentication problem in WSNs by designing PKC-based solutions. Their schemes are built upon the integration of several cryptographic techniques, including the Bloom filter, the Merkle hash tree, *et al.*, However, they use the Bloom filter between the base station and the network user, where the network users refer to personnel or devices that use the WSN; they are not sensor nodes. In our scheme, the Bloom filter is used among the sensor nodes in WSN to provide an efficient authentication.

There are several problems in [6–9]. For example, how does one verify the validness of a public key? Conventional solutions, such as Public Key Infrastructure (PKI) and a certificate, are non-implementable in WSNs, due to their constrained resource. How does one apply Identity-Based Encryption (IBE) in WSNs efficiently and securely with the integrity of a public key? Public key validness is hard to be verified in present IBE schemes, because it usually depends on the certificate

and CA. Additionally, the certificate will result in a large communication overhead and expensive signature verification operations, which consume more energy [10].

Because of the absence of PKI and a certificate, there is no authentication in the state-of-the-art IBE schemes, which are subject to many attacks, such as the Sybil attack, the man-in-the-middle attack, *etc.* Focused on addressing these problems, we propose an efficient identity-based key management scheme (IBKM) in this paper, which adopts an identity-based cryptosystem to distribute session keys between nodes without the complicated operations of the public key certificate; specifically, we exploit the Bloom filter to provide authentication with storage efficiency. A Bloom filter is a simple space-efficient randomized data structure based on a hash function for representing a set in order to support membership queries. Although Bloom filters allow false positives, for many applications, e.g., WSNs, the space savings outweigh this drawback when the probability of an error is sufficiently low [11].

The main contributions of this paper are as follows:

(1) To the best of our knowledge, we are the first one to exploit the Bloom filter to authenticate sensor nodes in WSNs. The sensor node's public key in IBE is verified using the Bloom filter together with its ID.
(2) We come up with a security analysis, as well as quantitative memory, computation and communication overhead to demonstrate the effectiveness and efficiency of IBKM. The computation overhead that is brought by the Bloom filter is quite small, as hash operations are negligible compared with the bilinear pairing.

The rest of this paper is organized as follows. Section 2 introduces related works. Section 3 gives some preliminaries on bilinear pairing, bilinear computational Diffie–Hellman (BCDH) and the Bloom filter. We describe IBKM in Section 4. In Section 5, a security analysis of IBKM is given to prove its resilience against various types of attacks. Section 6 gives the evaluation of the storage, computation and communication overhead. Finally, Section 7 concludes the paper.

2. Related Works

The notion of identity-based public key schemes was firstly introduced by Adi Shamir [12], who presented an identity-based signature scheme. As compared with the traditional certificate-based public-key cryptosystems, the ID-based system utilizes the users' identity (for example, name or email address) as the public key; therefore, additional computations to verify the corresponding certificates are not needed.

Until now, several cryptography algorithms based on identity have been proposed; however, these solutions cannot completely meet the requirements of practical use, especially in WSNs. In 2001, Franklin and Boneh [13] proposed an identity-based encryption scheme from Weil pairing. They also showed that their scheme can gain security against an adaptive chosen cipher text attack in the random oracle model. Their work is based on bilinear pairings on elliptic curves and led to high research activity in this field.

Yang *et al.* [14] propose an approach based on identity-based encryption and Diffie–Hellman algorithms, which provides authenticated key agreement between pairs of sensor nodes. However, its

computation and memory overhead are too high to be practically applied. Zhang *et al.* [15] propose a new security scheme based on LOCK [16] and ID-based secure group key management. They use the exclusion basis system (EBS) [17] for key agreement between the gateway and node, while ID-based key management between the base station and gateway.

Since Boneh *et al.* [18] proposed a signature and encryption scheme based on identity from pairing; many schemes [19–22] attempt to apply pairing on WSNs. Yang *et al.* [22] proposed IBAKA using pairing-based cryptography. Their scheme achieves significant improvements in terms of security strength, communication and storage overhead. Later in this paper, we will compare our scheme with theirs. However, the pairing costs too much computation overhead for WSNs. Barreto [23] proposed an efficient approach to compute pairings on supersingular curves, which can be used for elliptic and hyperelliptic curves with very efficient results. Manel *et al.* [24] propose a scheme based on identity, which supports the establishment of pair-wise keys and cluster keys. However, their scheme does not verify the authenticity of the identity before the key agreement between two nodes; also, nodes in this scheme store a bunch of other nodes' public key and the identity value, which increases the storage overhead. Cheng *et al.* [25] presented EKAES, which is an ID-based key agreement and encryption scheme for WSNs, but their scheme has an expensive communication overhead. Chatterjee *et al.* [26] propose an ID-based key management scheme using bilinear pairings. The nodes in their scheme verify the authenticity of other nodes through the cluster, which causes a heavy communication overhead.

Kui *et al.* [10] presented several public-key-based schemes to achieve immediate broadcast authentication in WSNs, and the Bloom filter is used. However, in their schemes, the broadcast messages are initiated by network users, which are personnel or devices that use the WSN; they are not sensor nodes. Instead, our scheme is used to authenticate among the sensor nodes in a WSN.

Altogether, state-of-the-art identity-based approaches do not verify the authenticity of the corresponding node before the key agreement, because certificate verification usually needs extensive computation, which causes much computation overhead on the sensor nodes; besides, more entities, such as CA, should be setup. In this paper, we address this problem by adopting the Bloom filter with minimized computational and storage costs to cope with the resource-constrained nature of WSNs.

3. Preliminaries

In this section, we give a brief introduction to bilinear pairing, the bilinear Diffie–Hellman (BDH) problem and the Bloom filter.

Bilinear Pairing: Let G_1 be a cyclic additive group of prime order q and G_2 be a cyclic multiplicative group of the same order q. $e: G_1 \times G_1 \to G_2$ between these groups is a bilinear pairing if it satisfies the following properties:

1. Bilinear: We say that a map $e: G_1 \times G_1 \to G_2$ is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$, while $a, b \in Z$.
2. Non-degenerate: There exists $P, Q \in G_1$, such that $e(P, Q) \neq 1$.
3. Computable: There exists an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Bilinear Diffie–Hellman (BDH) problem: For given $\{P, aP, bP, cP\}$, it is a hard problem to compute $e(P, P)^{abc}$ for some $a, b, c \in Z$.

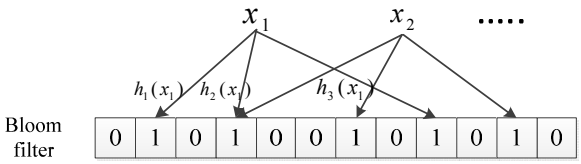
Bloom filter: A Bloom filter [11] is a simple space-efficient randomized data structure; it can be used to succinctly represent a set in order to support membership queries. In our scheme, we use it to authenticate a sensor node while receiving the communication request. A Bloom filter is described by a vector of m bits, which are initially all set to zero. In order to represent a set $S = \{x_1, x_2 \cdots x_n\}$ that contains n elements, we use k independent hash functions to map each item to the m-bits vector. For each element, $x \in S$ bits $h_a(x)$ are set to one. Then, we have:

$$BloomFilter_i = \begin{cases} 1, & if \exists a \in [1,k], b \in [1,\mathrm{n}] \\ & s.t. \quad h_a(x_b) = i \\ 0, & otherwise \end{cases} \tag{1}$$

The initial value: $BloomFilter_i = 0 \quad i \in [1,m]$

A simple example is shown in Figure 1 when $k = 3$. x_1 is hashed by three hash functions, and three corresponding items in the Bloom filter are set to one. Note that a bit of the vector can be set to one multiple times, but only one works.

Figure 1. Example of a Bloom filter.

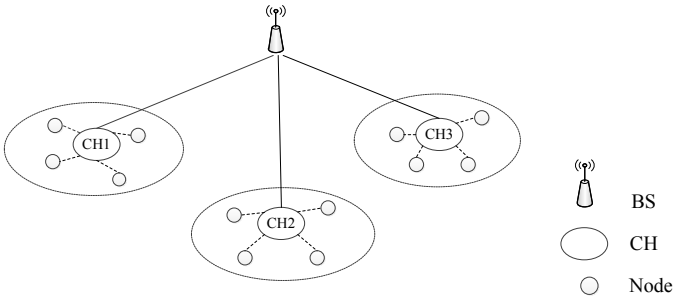


4. IBKM Scheme

Basically, there are two architectures available for WSNs. One is a distributed flat architecture, and the other is a hierarchical architecture. Considering the limitations of WSNs, such as low energy supply, extremely large network size and redundant low-rate data, the hierarchical network model has more operational advantages than the flat homogeneous model for wireless sensors [27].

In this work, we focus on the hierarchical network model, which is shown in Figure 2, as in [28]. It has three different kinds of wireless devices; base station (BS), cluster head (CH) and sensor node (N). We assume that the BS is trusted and that the CH is more capable than normal nodes. In a cluster, the CH collects and aggregates packets from its member nodes and then forwards them to the BS. Normally, a member sensor node can transfer packets to CH through several hops.

Figure 2. Network model of WSN.



IBKM consists of three phases: parameters initialization, node registration and share secret key generation between two nodes. Table 1 displays the notations used in this paper.

Table 1. List of the notations used.

Notation	Description
G_1	A cyclic additive group of prime order q
G_2	A cyclic multiplicative group of prime order q
E	A random elliptic curve
P	A point on E
p, q	Large prime numbers
$e(\,)$	A bilinear mapping function
$H(\,)$	Hash function
$E_k(\,)$	Symmetric encryption with key k
T	A time stamp
r	Random number
ID_{CH}	Identity of cluster head CH
ID_A, ID_B	Identity of node A and node B
Q_{CH}, S_{CH}	Public key and private key of CH
Q_A, S_A	Public key and private key of Node A
K_{C1}, K_{C2}	Shared secret key between cluster head and node
K_{S1}, K_{S2}	Shared secret key between two nodes in a cluster

4.1. Parameters Initialization Phase

BS selects large prime p, q and generates a random elliptic curve E over finite field F_p. One point P on curve E is selected and used as generator to construct an additive group G_1, and $e: G_1 \times G_1 \to F_p^*$ is a bilinear map. $H_1: \{0,1\}^* \to E(F_p)$ are two cryptographic hash functions.

(1) BS selects a random number s and computes $P_{pub} = sP \in G_1$ as the public key. BS broadcasts the public parameters $(G_1, E(F_P), p, q, e, P, P_{pub}, H_1, H_2)$.

(2) BS generates each node's ID and calculates the public and private key pair of the node. Then, BS preloads them into the node. The public key is $Q_N = H_1(ID_N)$, and the private key is $S_N = sQ_N$, where N represents a node in WSNs.

(3) BS generates the CH's ID and calculates the public and private key pair of the CH. Then, BS stores them in the CH, in which the public key is $Q_{CH} = H_1(ID_{CH})$; the private key is $S_{CH} = sQ_{CH}$, where CH is a cluster header in WSNs.

(4) BS keeps a list of all nodes' IDs and their public-private key pairs. BS also keeps all CHs' IDs and public keys for the next steps.

4.2. Node Registration Phase

In this phase, all sensor nodes register to the cluster heads and a session key is generated between each node and their cluster head, as shown in Figure 3.

(1) CH broadcasts a message that contains its own identity and a public key to all neighboring sensor nodes:

$$CH \xrightarrow{E_{S_{CH}}(ID_{CH} \| Q_{CH})} All\ Nodes$$

(2) Upon the receipt of CH's messages, each sensor node sends its ID and public key to the CH with whom it wants to join.

$$All\ Nodes \xrightarrow{E_{Q_{CH}}(ID_N \| Q_N)} CH$$

(3) After receiving the ID and public key of a node, CH calculates the session key K_{C2}.

$$K_{C2} = e(S_{CH}, Q_N)$$

(4) Node calculates the same session key with CH.

$$K_{C1} = e(S_N, Q_{CH})$$

$K_{C1} = K_{C2}$ can be proved as follows:

$$\begin{aligned} K_{C1} &= e(S_N, Q_{CH}) = e(S_N, H_1(ID_{CH})) = e(sQ_N, H_1(ID_{CH})) \\ &= e(sH_1(ID_N), H_1(ID_{CH})) = e(sH_1(ID_{CH}), H_1(ID_N)) \\ &= e(sQ_{CH}, H_1(ID_N)) = e(S_{CH}, H_1(ID_N)) = e(S_{CH}, \mathrm{Q}_N) \\ &= K_{C2} \end{aligned} \tag{2}$$

We let $K_C = K_{C1} = K_{C2}$.

(5) CH generates a Bloom filter of all nodes' IDs and public keys within its cluster and sends the Bloom filter encrypted by the session key generated before to all nodes in the cluster. Figure 4 shows the generation of the Bloom filter.

$$CH \xrightarrow{E_{K_C}(Bloom\ Filter)} All\ Nodes$$

Figure 3. Procedure of node registration.

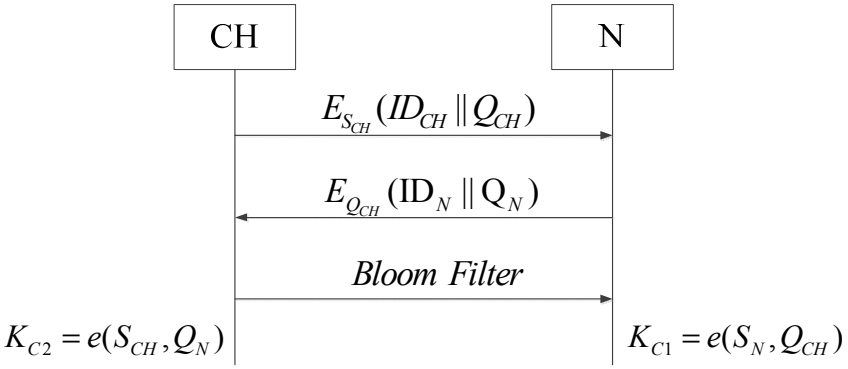
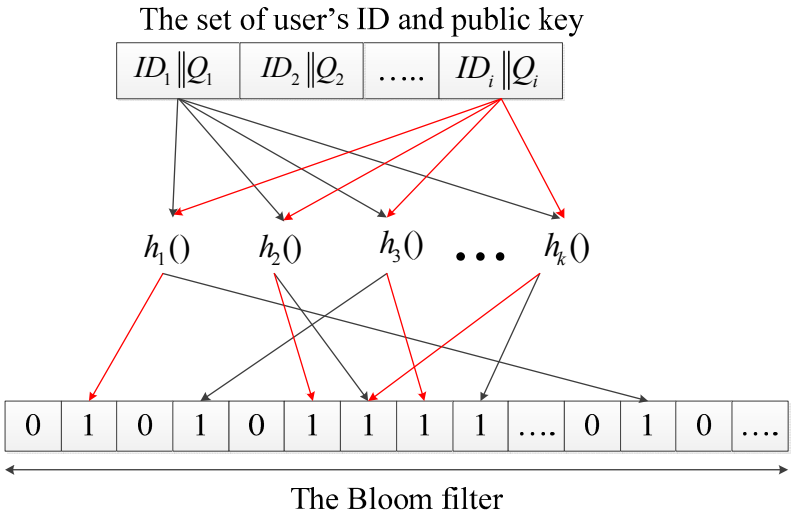


Figure 4. CH generates its own cluster Bloom filter.



4.3. Share Secret Key Generation between Two Nodes

(1) Sensor Node A chooses a random number r_1 and broadcasts a message that contains its ID, public key and a time stamp encrypted by its own private key to neighboring nodes after it registers to the CH.

$$A \xrightarrow{ID_A \| E_{S_A}(r_1 Q_A \| T)} Neighbor\ Nodes$$

(2) When the neighboring Node B receives the message, it verifies the authenticity of A by checking if the hash mapping of (ID_A, Q_A) is contained in the Bloom filter obtained from CH. A negative answer means authentication failure. Our node authentication algorithm takes a similar idea as in [10] and is provably efficient. If the authentication is passed, B chooses its random number r_2 and returns its ID, public key and a time stamp encrypted by its own private key. Then, B calculates the session key K_{S1}.

$$B \xrightarrow{ID_B \| E_{S_B}(r_2 Q_B \| T)} A$$

$$K_{S1} = e(\mathrm{r}_2 S_B, \mathrm{r}_1 H_1(ID_A))$$

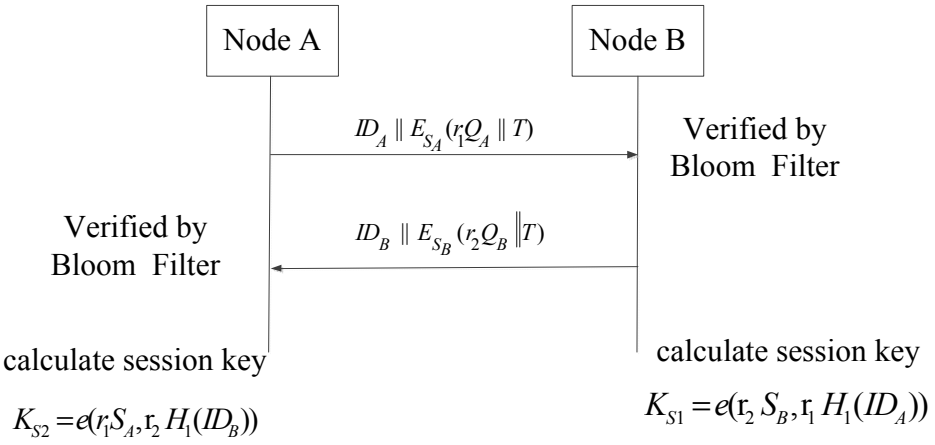
(3) A decrypts the message and get B's ID and public key with its own private key and verifies the authenticity of B using the Bloom filter obtained from CH. If B is authenticated, A calculates the session key K_{S2}.

Thanks to the properties of the bilinear map, we can prove $K_{S1} = K_{S2}$ as follows.

$$\begin{aligned} K_{S1} &= e(\mathrm{r}_2 S_B, r_1 H_1(ID_A)) = e(\mathrm{r}_2 sQ_B, r_1 H_1(ID_A)) \\ &= e(\mathrm{r}_2 sH_1(ID_B), r_1 H_1(ID_A)) = e(r_1 sH_1(ID_A), \mathrm{r}_2 H_1(ID_B)) \\ &= e(r_1 sQ_A, \mathrm{r}_2 H_1(ID_B)) = e(r_1 S_A, \mathrm{r}_2 H_1(ID_B)) = K_{S2} \end{aligned} \quad (3)$$

Afterwards, Nodes A and B can communicate with each other using the shared session key. The shared secret key between two nodes can be decided as shown in Figure 5.

Figure 5. Share secret key generation between two nodes.



5. Security Analysis

Due to the unreliable wireless channel and volatile topology, a key agreement scheme for WSNs is subject to various attacks, such as node-compromise attack, Sybil attack, *etc.* Compared to previous works, our scheme can resist these attacks using the bilinear map and authentication through the Bloom filter.

Sybil Attack: Before node deployment, the BS allocates an ID for each node in the WSNs, and then, the CH generates a Bloom filter of nodes in its own cluster. Therefore, before sharing the secret key between two nodes, they authenticate each other using the Bloom filter generated by CH. Therefore, IBKM can resist Sybil attack because an adversary cannot convince another node that it has a legal ID.

Node-compromise attack: It is easy to capture a node in WSNs and steal secret information about the network stored in the node. Compared to the E-G and other key pre-distribution schemes, IBKM can resist node-compromise attack and ensure the security of the entire network. For the E-G scheme and its variants, if the number of node adversaries captured exceeds a certain threshold, the adversaries will get almost all of the keys of the WSN. However, in our scheme, different node pairs share different keys; even if a node is compromised, it will not affect other node pairs' keys.

Rekeying and forward secrecy*:* IBKM employs a random number r in the process of secret key generation between two nodes. On the one hand, we can stipulate the secret key agreement period; therefore, nodes must renegotiate a new session key in a certain period. In this way, we can enhance the security of the network. On the other hand, the rekeying can provide forward secrecy of the network when a node is captured by the adversary. Even if the adversary gets the current secret key, he cannot deduce the keys used before, because different random numbers generate different secret keys.

HELLO flood attack: In this attack, the main aim of the attacker is to deplete the node energy. In our scheme, every node possesses a Bloom filter for node identity authentication. Therefore, if an adversary sends a HELLO message, the receiver nodes will first check if the message is legitimate or not. If the result is negative, later calculation will not be carried on. Therefore, no more energy of the received node will be consumed.

Man-in-the-middle attack: In our scheme, the adversary cannot calculate the pairwise session key, even if it intercepts the system parameters, since the messages transmitted in our scheme are all encrypted in the public key cryptosystem. On the other hand, the session key is generated by the

private key and the random number. It is assumed to be hard for an adversary to decrypt the message on air or to calculate the session key.

Mutual authentication: Our scheme achieves both identity authentication and key authentication. Before the session key is agreed upon, the nodes verify the authenticity of each other by checking if the corresponding hash mapping is contained in the local Bloom filter. A negative answer means that the node is illegal in this cluster. Then, we verify the identity of the node by the signature of the private key. While, after, Node A and Node B share the same session key, they can realize identity authentication by the session key, because only A and B share the same key. In this way, we can prevent the unauthenticated node from accessing the sensor network.

6. Performance Evaluation

Although security is a critical factor in WSNs, it is also necessary to evaluate the storage, computation and communication consumption of sensor nodes, since they are extremely resource constrained. In this section, we evaluate the performance of IBKM by comparing the storage, computation and communication overhead with two relevant schemes, Yang's scheme [22] and Cheng's scheme [25]. The two schemes were influential ones of the key management protocols proposed for WSNs.

6.1. Performance of Bloom Filter

Since we apply the Bloom filter to provide probabilistic membership verification in our scheme and hash functions have the disadvantage to collision, it is important to evaluate the probability of the false positive.

Theorem 1: Given the number of cluster nodes n and the storage space m bits for a single Bloom filter, assume the number of hash functions in the Bloom filter is k; we can get the minimum probability of the false positive $f = 2^{-k}$ with the number of hash functions around $k = (m/n)\ln 2$.

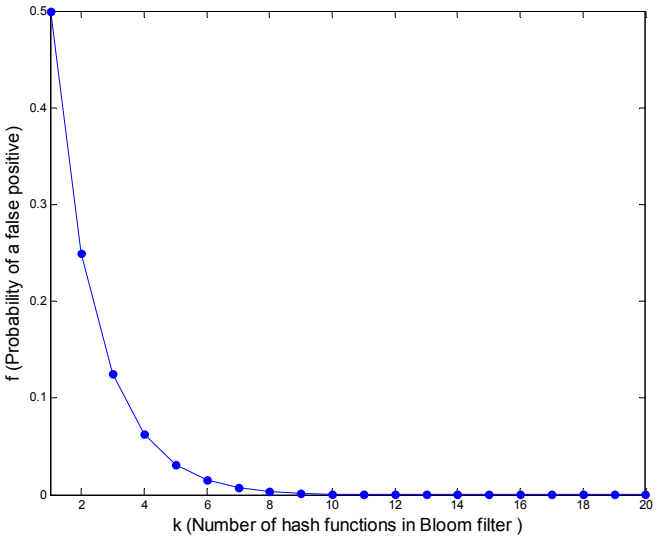
Proof: Since $f = (1-(1-1/\mathrm{m})^{kn})^k \approx (1-\mathrm{e}^{-kn/m})^k$ [10], we can have $f = e^{k\ln(1-\mathrm{e}^{-kn/m})}$. Let $g = k\ln(1-\mathrm{e}^{-kn/m})$; hence, minimizing f is equivalent to minimizing g with respect to k. We find:

$$\frac{dg}{dk} = \ln(1-\mathrm{e}^{-kn/m}) + \frac{kn}{m}\frac{\mathrm{e}^{-kn/m}}{1-\mathrm{e}^{-kn/m}} \tag{4}$$

It is easy to check that the derivative is zero when $k = (m/n)\ln 2$. We substitute $k = (m/n)\ln 2$ into $f = (1-\mathrm{e}^{-kn/m})^k$ and get $f = 2^{-k}$.

In addition, it is not hard to show that this is a global minimum [11]. Now, we can see that the probability of a false positive f is a function of k, *i.e.*, $f = 2^{-k}$. Figure 6 shows that as k increases, the false positive decrements rapidly. Then, we can choose appropriate k according to the different application scenarios of WSNs to achieve an acceptable false positive rate. In our following experiments, we take k as 10, since the false positive f has dropped below 10^{-3}.

Figure 6. Minimum probability of false positives.



6.2. Memory Overhead

It can be obtained from the above analysis that the probability of the false positives reaches the minimum when $m = n(k / \ln 2)$, and the minimum value is $f = 2^{-k}$. For a certain network whose number of nodes n is determinate, thus the probability of the false positive changes after m and k. To maintain the minimum probability of the false positives, we should keep $m = n(k / \ln 2)$. It turns out that when we want smaller f, we should use larger m or k, which means more memory consumption and more hash computation; therefore, we should make a trade-off to choose appropriate k and m in accordance with different scenarios. At this point, we assume a civilian scenario in which f is acceptable when less than 1%, *i.e.*, $k = 10$, we obtain $m = 14.427 * n$. In Cheng's and Yang's scheme, each node is preloaded with other node's public keys, while our scheme only use the Bloom filter to verify the public keys. Therefore, for convenience, to compare the three schemes, we take the whole cluster memory consumption as the measurement. Here, we assume the public key is 128 bits long; thus, the total memory of preloaded keys in the cluster is $m = 128 * n$ for Cheng's and Yang's scheme. Figure 7 shows the performance comparison of IBKM, Cheng's and Yang's schemes. From the comparison, we can see that our scheme costs significantly less memory consumption than their schemes.

6.3. Computational Overhead

We implement our proposed scheme in Microsoft Visual C++ 6.0. The operating system is Windows 7 Ultimate. The computer configuration is as follows: CPU, Intel Core i5 3.2 GHz; memory, 4 GB; hard disc, 1 TB. In our scheme, we need to compute one bilinear pairing, exponentiation on G_T, 160-bit scalar point multiplication and the Bloom filter. Yang's scheme [22] involves the computation of two bilinear pairings, one exponentiation on G_T and one 271-bit scalar point multiplication. Cheng's scheme [25] involves the computations of two bilinear pairings, one exponentiation on G_T, two 160-bit scalar point multiplications and one 271-bit scalar point multiplication. We calculate the time needed for the major operations, which is shown in Table 2.

From this, we can see that the most time-consuming operations are bilinear pairing, exponentiation on G_T and 160-bit point multiplication. The Bloom filter only costs about 1/86, 1/9 and 1/6 of the bilinear pairing, exponentiation on G_T and point multiplication, respectively.

Figure 7. Comparison of memory overhead. IBKM, identity-based key management.

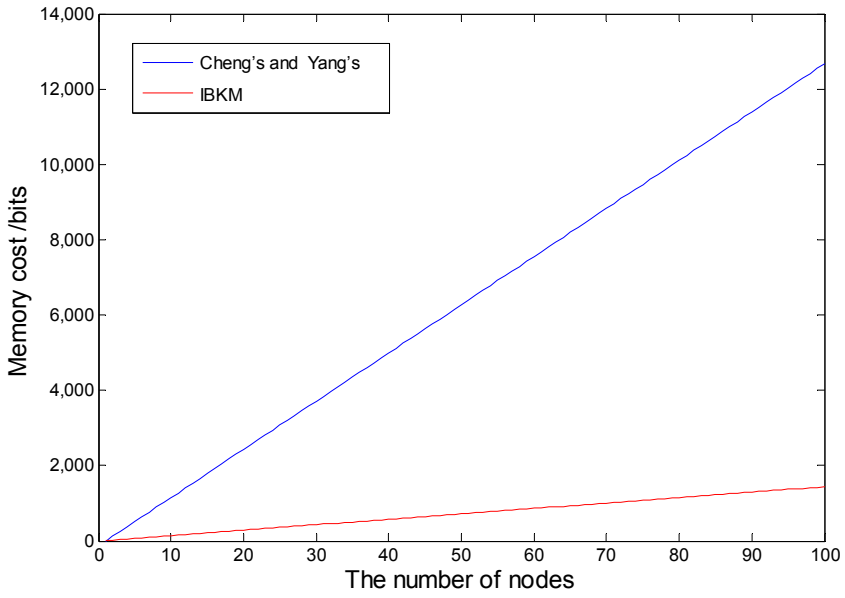


Table 2. Time consumption of major operations.

Operations	Time/ms
Bilinear pairing	14.193
Exponentiation on G_T	1.525
point multiplication	0.940
Bloom filter	0.165

A comparison of total computation overhead with the two schemes is shown in Table 3. From the table, we can see that, since we only use one bilinear pairing, which is the most time-consuming operation, our scheme needs less computation time than the other two schemes; therefore, IBKM saves the energy consumption of node. Notice that, although hash mapping and encryption are introduced in the Bloom filter, they are omitted, since they consume negligible computing power compared with that of bilinear pairing, *etc.*

Table 3. The comparison of computation overhead in the secret key generation of two nodes.

Schemes	Bilinear Airing	Exponentiation on G_T	Point Multiplication	Time/ms
Yang's	2	1	1	30.951
Cheng's	2	1	3	32.831
IBKM	1	0	2	16.438

6.4. Communication Overhead

In IBKM, the secret key generation process comprises two messages: one is sent by A, and the other is sent by B. Each message includes the node's ID, public key and a time stamp, namely the message is $ID \| E(\mathrm{rQ} \| T)$. In this message, rQ is a point on elliptic curve, which given x of rQ, the node can derive y whenever it needs. In accordance with Yang's scheme, Q can be compressed to 34 bytes and two bytes for ID. We take eight bytes for the time stamp in our scheme, and the encryption here does not change the length of the messages. Therefore, the communication overhead of our scheme is 44 bytes. While Yang's scheme needs 61 bytes for the message of $\langle U, V \rangle$ and Cheng's scheme needs 168 bytes for using the 1024-bit modular in Diffie–Hellman key exchange, their schemes both need two messages when nodes share a secret key in the communication process. We show the communication overhead of the three schemes in Table 4. From this table, we can see the superiority of our communication consumption compared with the other two schemes.

Table 4. The comparison of communication overhead in the secret key generation of two nodes.

Scheme	Cheng's	Yang's	IBKM
Communication overhead (bytes)	336	122	88

7. Conclusions

In this paper, we propose IBKM, which is an efficient key management scheme for WSNs. By adopting the Bloom filter into identity-based cryptosystem to distribute session keys between nodes, IBKM achieves the advantages of the node's identity authentication without complex certification verification by the certification authority. The results of the analysis show that our scheme can resist various attacks and has acceptable overhead in storage, computation and communication compared to the existing related schemes.

Acknowledgments

This work is supported by the National High Technology Research and Development Program of China (863 program) under Grant 2013AA014001 and the Key Lab of Information Network Security, the Ministry of Public Security.

Author Contributions

Xinshuai Zhang proposed using the Bloom filter, which is one of the main contributions in this paper. Kerong Feng helped with the performance analysis, such as storage and computational overhead. Qunfang Zhang helped to implement the prototype system. Jie Huang provided the experimental environment and took part in the paper discussion.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Eschenauer, L.; Gligor, V.D. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washingtion, DC, USA, 18–22 November 2002; pp. 41–47.
2. Ren, K.; Zeng, K.; Lou, W. A new approach for random key pre-distribution in large-scale wireless sensor networks. *Wirel. Commun. Mobile Comput.* **2006**, *6*, 307–318.
3. Chan, H.W.; Perrig, A.; Song, D. Random key predistribution schemes for sensor networks. In Proceedings of 2003 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 11–14 May 2003; pp. 197–213.
4. Blom, R. An Optimal Class of Symmetric Key Generation Systems. In *Advances in Cryptology*; *Theory and Application of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 335–338.
5. Blundo, C.; de Santis, A.; Herzberg, A.; Kutten, S.; Vaccaro, U.; Yung, M. Perfectly-Secure key distribution for dynamic conferences. *Inf. Comput.* **1998**, *146*, 1–23.
6. Watro, R.; Kong, D.; Cuti, S.-F.; Gardiner, C.; Lynn, C.; Kruus, P. TinyPK: Securing sensor networks with public key technology. In SASN'04-Proceedings of the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks, Washington, DC, USA, 25 October 2004; pp. 59–64.
7. Oliveira, L.B.; Aranha, D.F.; Morais, E.; Daguano, F.; López, J.; Dahab, R. Tinytate: Computing the tate pairing in resource-constrained sensor nodes. In Proceedings of Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007), Cambridge, MA, USA, 12–14 July 2007; pp. 318–323.
8. Gura, N.; Patel, A.; Wander, A.; Eberle, H.; Shantz, S.C. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Cryptographic Hardware and Embedded Systems-CHES 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 119–132.
9. Liu, A.; Ning, P. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In Proceedings of the 7th International Conference on Information Processing in Sensor Networks, St. Louis, MO, USA, 22–24 April 2008; pp. 245–256.
10. Ren, K.; Yu, S.C.; Lou, W.J.; Zhang, Y.C. Multi-User broadcast authentication in wireless sensor networks. *IEEE Trans. Veh. Technol.* **2009**, *58*, 4554–4564.
11. Mitzenmacher, M. Compressed bloom filters. *IEEE/ACM Trans. Netw.* **2002**, *10*, 604–612.
12. Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In *Advances in Cryptology*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 47–53.
13. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology—CRYPTO 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
14. Yang, G.; Rong, C.-M.; Veigner, C.; Wang, J.-T.; Cheng, H.-B. Identity-Based key agreement and encryption for wireless sensor networks. *J. China Univ. Posts Telecommun.* **2006**, *13*, 54–60.
15. Zhang, J.; Varadharajan, V. A new security scheme for wireless sensor networks. In Proceedings of IEEE Global Telecommunications Conference, New Orleans, LA, USA, 30 November–4 December 2008; pp. 1–5.
16. Eltoweissy, M.; Tech, V.; Moharrum, M.; Mukkamala, R. Dynamic key management in sensor networks. *IEEE Commun. Mag.* **2006**, *44*, 122–130.

17. Eltoweissy, M.; Heydari, M.H.; Morales, L.; Sudborough, I.H. Combinatorial optimization of group key management. *J. Netw. Syst. Manag.* **2004**, *12*, 33–50.
18. Boneh, D.; Lynn, B.; Shacham, H. Short signatures from the Weil pairing. *J. Cryptol.* **2004**, *17*, 297–319.
19. Oliveira, L.B.; Aranha, D.F.; Gouvêa, C.P.; Scott, M.; Câmara, D.F.; López, J.; Dahab, R. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks. *Comput. Commun.* **2011**, *34*, 485–493.
20. Xu, J.; Zhang, Z.; Feng, D. ID-Based proxy signature using bilinear pairings. In Parallel and Distributed Processing and Applications-ISPA 2005 Workshops; Springer: Berlin/Heidelberg, Germany, 2005; pp. 359–367.
21. Rahman, M.; Nasser, N.; Saleh, K. Identity and pairing-based secure key management scheme for heterogeneous sensor networks. In Proceedings of 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Avignon, France, 12–14 October 2008; pp. 423–428.
22. Yang, L.; Ding, C.; Wu, M. Establishing authenticated pairwise key using Pairing-based Cryptography for sensor networks. In Proceedings of 8th International Conference on Communications and Networking in China (CHINACOM), Guilin, China, 14–16 August 2013; pp. 517–522.
23. Barreto, P.S.; Galbraith, S.D.; Ó'hÉigeartaigh, C.; Scott, M. Efficient pairing computation on supersingular abelian varieties. *Des. Codes Cryptogr.* **2007**, *42*, 239–271.
24. Boujelben, M.; Youssef, H.; Mzid, R.; Abid, M. IKM—An identity based key management scheme for heterogeneous sensor networks. *J. Commun.* **2011**, *6*, 1–5.
25. Cheng, H.; Yang, G. EKAES: An efficient key agreement and encryption scheme for wireless sensor networks. *J. Electron.* **2008**, *25*, 495–502.
26. Chatterjee, K.; De, A.; Gupta, D. An improved ID-Based key management scheme in wireless sensor network. In *Advances in Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 351–359.
27. Cheng, Y.; Agrawal, D.P. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Ad Hoc Netw.* **2007**, *5*, 35–48.
28. Younis, M.; Youssef, M.; Arisha, K. Energy-Aware management for cluster-based sensor networks. *Comput. Netw.* **2003**, *43*, 649–668.