Identity-Based Key Agreement and Encryption For Wireless Sensor Networks

Geng Yang[1], Chunming Rong[2], Christian Veigner[2], Jiangtao Wang[1], and Hongbing Cheng[1]

[1]Department of Computer Sciences & Technology, Nanjing University of Posts & Telecommunications, Nanjing, 210003, China.

[2]Department of Electrical and Computer Engineering, University of Stavanger, N-4036, Norway,

***Abstract*—It is an important challenge to find out suitable cryptography for wireless sensor networks due to limitations of power, computation capability and storage resources. Many schemes based on public or symmetric key cryptography are investigated. Recently, a practical identity-based encryption technique is proposed. In this paper, we present an identity-based key agreement and encryption scheme for wireless sensor networks. The scheme is an elliptic curve cryptography type algorithm. We review briefly about identity-based encryption and decryption first, particularly, the Boneh-Franklin algorithms. Then we describe a key agreement and encryption scheme based on the Boneh-Franklin algorithms for wireless sensor networks. We discuss the efficiency and security of our scheme by comparing with traditional public key technique and symmetric key technique.**

***Index Terms*—identity-based cryptography, security, wireless sensor network.**

I. INTRODUCTION

Wireless Sensor Network (WSN) has received considerable attention during last decade [1,2,3,4] (see, for example, the proceedings of the ACM and IEEE Workshops on WSN). It has been developed for a wide variety of applications, including military sensing and tracking, environment and security monitoring, equipment and human monitoring and tracking, etc. Sensor networks usually consist of a large number of ultra-small autonomous devices. Each device, called a node, is battery powered and equipped with integrated sensors, digital signal processors (DSPs) and radio frequency (RF) circuits. Because of special characteristics and limitations of wireless sensor networks, we face an important challenge in security issue, particularly for the applications where wireless sensor networks are developed in a hostile environment or used for some crucial purposes. For example, an adversary can easily listen to the traffic and mislead communications between nodes. In order to establish a secure network, we have to design secure protocols to deal with problems about key agreement and encryption in communications.

Three types of key agreement schemes have been studied in general network environments: trusted-server schemes, public-key schemes, and key predistribution schemes [5]. There is a trusted server in *Trusted-server* schemes for key agreement between nodes. This type of scheme is not suitable for sensor networks because nodes are with limited power and low computing capability. *Public-key* schemes depend on asymmetric cryptography. By the same reasons, this type of scheme is not a desirable choice. The third approach to establish keys is via *predistribution*, where key information is distributed to all sensor nodes prior to deployment. Such schemes have been extensively investigated [6,7,8,9,10,11].

As we all known, most sensor network is deployed in a random mode. Nodes do not have any information about neighbors and topology of the network a priori. Therefore, a naive approach to key distribution is to let all nodes store an identical secret *master key*. Any pair of nodes can use this master secret key to securely establish a new pairwise key. However, this scheme does not exhibit desirable network resilience: if a single node is compromised, the security of the entire sensor network is compromised. Even though, it is possible to store the master key in tamper-resistant hardware to reduce the risk, but this increases the cost and energy consumption of each sensor. Furthermore, tamper-resistant hardware might not always be safe [12]. At the other extreme, one might consider a key predistribution scheme in which each sensor node stores N -1 keys (where N is the number of nodes in the network). This scheme guarantees perfect resilience because compromised nodes do not leak information about keys shared between two noncompromised nodes. Unfortunately, this scheme is impractical for sensors with an extremely limited amount of memory because N can be very large.

Recently, Eschenauer and Gligor proposed a probabilistic key sharing for wireless sensor networks [6]. Pietro et al presented a random key predistribution scheme [9]. Its operation is briefly described as follows. A random pool of keys is selected from the key space. Each sensor node receives a random subset of keys from the key pool before deployment. Any two nodes able to find one common key within their respective subsets can use that key as their shared secret to initiate communication. One problem in this random scheme is that it only guaranties a common key between two nodes with some probability *p*. Based on this scheme, Chan et al. developed the q-composite key predistribution and the random pairwise keys schemes [10]. It makes two sensors share at least q predistributed keys to establish a pairwise key. This process improves the resilience of the network and requires an attacker to compromise many more nodes in order to compromise communication links. It is shown that, by increasing the value of *q*, network resilience against node capture is improved for certain ranges of other parameters [10]. Based on the random schemes, Du et al. proposed another key predistribution scheme which improves the resilience of the network compared to previous schemes [5]. Their scheme combines the key predistribution scheme of Blom [13,14] with the random key predistribution scheme. A location-based pairwise key establishment for static sensor network is discussed in [11].

Note that, in all above schemes, symmetric key technique is used in authentication and encryption. Compared with asymmetric key system, the main benefit with symmetric key system is low computing cost [15]. But the drawbacks are that it needs a key predistribution process and does not guaranty a perfect connectivity and communication (in random key distribution sehemes, neighboring nodes share a common key in terms of probability). Recently a number of studies have been conducted to find out a practical way to use Public-Key Cryptography (PKC) in sensor networks [16,17,18,19]. Their studies focus mostly on optimization of PKC. Though computing cost is still a crucial problem for PKC system, results in [17] indicate that Elliptic Curve Cryptography (ECC) has some advantages in memory requirement and computing cost and that it is suitable for sensor networks.

In 1984 Shamir proposed the idea of Identity-Based Encryption (IBE) [20]. The idea of an identity-based encryption is that the public key can be an arbitrary string, for example, an email address, a name or a role. Soon after, various identity-based techniques were proposed [21,22] but a fully-functional identity-based encryption scheme was not found until recently by Boneh and Franklin [23]. Since then the ideas of IBE have been used to design several other identity-based schemes for different purposes [24,25,26,27]. Note that IBE-based algorithms are types of ECC. According to the studies about public key system, therefore, it is interesting to investigate the possibility to apply IBE in wireless sensor networks. This is the objective of our paper.

The rest of the paper is organized as follows. Section 2 describes basic ideas and properties of identity-based encryption, particularly, the Boneh-Franklin scheme. Section 3 proposes a key distribution and encryption scheme for sensor networks based on IBE. Section 4 gives a detail analysis of the schemes in terms of efficiency and security. Section 5 concludes this paper and points out some future research topics. Analysis results show that the IBE-based algorithms are suitable for wireless sensor networks in terms of key management, security and storage requirement.

II. IDENTITY-BASED ENCRYPTION

In this section, we briefly review the identity-based encryption and the Boneh-Franklin IBE scheme.

A. Basic of IBE

The concept of identity-based cryptography was first proposed in 1984 by Adi Shamir [20]. In his paper, Shamir presented a new model of asymmetric cryptography in which the public key of any user is a characteristic that uniquely identifies himself/herself, like an e-mail address. In such a scheme there are four algorithms: (1) setup generates global system parameters and a master-key, (2) extract uses the master-key to generate the private key corresponding to an arbitrary public key string $ID \in \{0, 1\}^*$ (3) encrypt encrypts messages using the public key ID, and (4) decrypt decrypts messages using the corresponding private key.

Shamir's original motivation for identity-based encryption was to simplify certificate management in e-mail systems. When Alice sends mail to Bob at bob@company.com she simply encrypts her message using the public key string “bob@company.com”. There is no need for Alice to obtain Bob's public key certificate. When Bob receives the encrypted mail he contacts a third party, which we call the Private Key Generator (PKG). Bob authenticates himself to the PKG in the same way he would authenticate himself to a Center of Authentication (CA) and obtains his private key from the PKG. Bob can then read his e-mail. Note that unlike the existing secure e-mail infrastructure, Alice can send encrypted mail to Bob even if Bob has not yet setup his public key certificate. Also note that key escrow is inherent in identity-based e-mail systems: the PKG knows Bob's private key.

The distinguishing characteristic of identity-based encryption is the ability to use any string as a public key. The functions that compose a generic IBE are thus specified as follows.

Setup: takes a security parameter t_s and returns t_g (system parameters) and *master-key*. The system

parameters include a description of a finite message space M, and a description of a finite ciphertext space C. Intuitively, the system parameters will be publicly known, while the *master-key* will be known only to the Private Key Generator (PKG).

Extract: takes as input t_g, *master-key*, and an arbitrary $ID \in \{0, 1\}^*$, and returns a private key K. Here ID is an arbitrary string that will be used as a public key, and K is the corresponding private decryption key. The Extract algorithm extracts a private key from the given public key.

Encrypt: takes as input t_g, ID, and $m \in M$. It returns a ciphertext $c \in C$.

Decrypt: takes as input t_g, $c \in C$, and a private key K. It return $m \in M$. These algorithms must satisfy the standard consistency constraint, namely when K is the private key generated by algorithm Extract when it is given ID as the public key, then $\forall\ m \in M$: Decrypt(t_g, c, K) = m where c = Encrypt(t_g, ID, c)

B. The Boneh-Franklin IBE scheme

The scheme is based on IBE technique and proposed by Honeh and Franklin [23]. From here on we use Zq to denote the group $\{0, \ldots, q-1\}$ under addition modulo q. For a group G of prime order we use G^* to denote the set $G^* = G \setminus O$ where O is the identity element in the group G. We use Z^+ to denote the set of positive integers. We describe first some definitions and then the Boneh-Franklin IBE scheme.

Definition 2.1 An map $\hat{e}: G_1 \times G_1 \to G_2$ is called a bilinear pairing if, for all $x, y \in G_1$ and all $a, b \in Z$, we have $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$.

Definition 2.2 The Bilinear-Diffie-Hellman problem (BDH) for a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$ such that $|G_1| = |G_2| = q$ is prime is defined as follows: given g, g^a, g^b, $g^c \in G_1$, compute $\hat{e}(g, g)^{abc}$, where g is a generator and $a, b, c \in Z$. An algorithm $\mathcal{A}$ is said to solve the BDH problem with advantage ε if

$$\Pr[\mathcal{A}(g, g^a, g^b, g^c) = \hat{e}(g, g)^{abc}] \geq \varepsilon,$$

where the probability is over the random choice of a, b, c, g, and the random bits of $\mathcal{A}$.

Definition 2.3 A randomized algorithm $\mathcal{G}$ that takes as input a security parameter $k \in Z^+$ is a BDH parameter generator if it turns in time polynomial in k and outputs the description of two groups G_1, G_2 and a bilinear function $\hat{e}: G_1 \times G_1 \to G_2$, with $|G_1| = |G_2| = q$ for some prime q. Denote the output of the algorithm by $\mathcal{G}(1^k) = < G_1, G_2, \hat{e}, q >$.

Definition 2.4 We say that $\mathcal{G}$ satisfies the BDH assumption if no probabilistic polynomial algorithm $\mathcal{A}$ can solve BDH with non-negligible advantage**.**

We now give the Boneh-Franklin IBE algorithm for identity-based encryption based on bilinear pairings on elliptic curves.

Algorithm 2.1 The full Boneh-Franklin IBE scheme

1) Setup: Given a security parameter $k \in Z^+$, the algorithm works as follows.

Step 1: Run $\mathcal{G}$ on input k to generate a prime q, two groups G_1, G_2 of order q, and an admissible bilinear map $\hat{e}: G_1 \times G_1 \to G_2$. Choose a random $\alpha \in G_1$.

Setp 2: Pick a random $s \in Z_q^*$ and set $\beta = \alpha^s$.

Step 3: Choose cryptographic hash functions for some n, $H_1: \{0, 1\}^* \to G_1^*$, $H_2: G_2 \to \{0, 1\}^n$, $H_3: \{0, 1\}^n \times \{0, 1\}^n \to Z_q^*$, $H_4: \{0, 1\}^n \to \{0, 1\}^n$. For the security proof, we view the all hash functions as random oracles. The message space is $M = \{0, 1\}^n$.

The ciphertext space is $C = G_1^* \times \{0, 1\}^*$. The output system parameters are $\pi = \{q, G_1, G_2, \hat{e}, n, \alpha, \beta, H_1, H_2, H_3, H_4\}$. The master key is $s \in Z_q^*$.

2) Extract: For a given string $Id \in \{0, 1\}^*$ the algorithm does:

Step 4: Computes $Q_{Id} = H_1(Id) \in G_1^*$.

Step 5: Sets the private key K_{Id} to be $K_{Id} = (Q_{Id})^s$ where s is the master key.

3) Encrypt: To encrypt $m \in$ M under the public key Id do the following:

Step 6: Compute $Q_{Id} = H_1(Id) \in G_1^*$.

Step 7: Choose a random $\sigma \in \{0, 1\}^n$.

Step 8: Set $r = H_3(\sigma, m)$.

Step 9: Set the ciphertext to be

$$c = \langle r\alpha, \sigma \oplus H_2(g_{Id}^r),\ m \oplus H_4(\sigma) \rangle,$$

where $g_{Id} = \hat{e}(Q_{Id}, \beta) \in G_2$

4) Decrypt: Let $c = <U, V, W>$ be a ciphertext encrypted using the public key Id. If $U \notin G_1^*$ reject the ciphertext. To decrypt c using the private key $K_{Id} \in G_1^*$ do:

Step 10: Compute $V \oplus H_2(\hat{e}(K_{Id}, U)) = \sigma$.

Step 11: Compute $W \oplus H_4(\sigma) = m$.

Step 12: Set $r = H_3(\sigma, m)$. Test that $U = r\alpha$. If not, reject the ciphertext.

Step 13: Output m as the decryption of c.

This completes the description of a full version of

Boneh-Franklin IBE algorithm. This full version consists of four hash functions. There is a basic version that contains only two hash functions without H_3 and H_4. But, the full version provides higher security level than the simple version in terms of security.

C. Security of the Boneh-Franklin IBE algorithm

The following theorem shows that the Boneh-Franklin IBE algorithm is a chosen ciphertext secure IBE (i.e. IND-ID-CCA) and the basic version is one-way identity-based encryption scheme (ID-OWE), assuming BDH is hard in groups generated by G

Theorem 2.1. Let the hash functions H_1, H_2, H_3, and H_4 be random oracles. Then the Boneh-Franklin IBE algorithm is a chosen ciphertext secure IBE (IND-ID-CCA) assuming BDH is hard in groups generated by G.

More details about the security of the Boneh-Franklin IBE algorithm can be found in [23,24].

III. Identity-Based Key Agreement & Encryption Scheme for WSN

Based on the Boneh-Franklin IBE algorithm presented above, this section focuses on designing an efficient scheme for key agreement and encryption/decryption in wireless sensor networks.

Our scheme consists of the following steps.

1) *Initialization phase*

In the initialization phase, we calculate all public parameters and private keys, and contribute them to sensors.

- Computing public parameters

We use the Setup function of Algorithm 2.1 to get all system-wide parameters. The output system parameters of the function Setup are

$$\pi = \{q, G_1, G_2, \hat{e}, n, \alpha, \beta, H_1, H_2, H_3, H_4\},$$

where q is a prime number, G_1 and G_2 are two groups of order q, $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ is a bilinear map, n is the length of plaintext, $\alpha \in G_1$, $\beta = \alpha^s$, $s \in Z_q^*$ is the master key, H_1, H_2, H_3, and H_4 are four hash functions with random oracles respectively. The master key should be kept in a secret place and the parameters π can be distributed to all nodes.

Note that, in wireless sensor networks, this phase should be down prior to the nodes deployment. It could be realized in two modes. One is to use a base station to run the Setup function and distribute all the parameters to nodes. The other is to distribute all the parameters to all sensors in manufactory phase. In the first mode, the base station is only needed to generate parameters and send them to all nodes. After that, it exists no longer in a sensor network. Therefore, the first mode can be considered as a special case of the second mode. The second mode is mostly like the MAC address in a network adapter. As we all know, MAC address in network adapter is fixed and unique. There is a one-to-one relationship between MAC and IP addresses. We can store a unique sensor Id in each sensor according to a worldwide identity or a customized identity.

- Computing private keys

In this step, we run the Extract function of Algorithm 2.1 to obtain private keys. The inputs are public parameters obtained in the above step and public key. The public key could be an arbitrary string $Id \in \{0, 1\}^*$. The private key will be distributed to a sensor. This process can be down in the same period as the first step. If a base station is used to perform the calculation, the private key is only known by the base station and the corresponding sensor. If the sensor calculates the private key itself, only the sensor knows its private key. The master key s, in this case, certainly cannot be stored in the sensor after being used, because all private keys can be extracted according to the public parameters, sensors' Ids and the master key s.

Note that, from the administration point of view, this step could be performed within a scope of users of the sensor networks, for example, a military unit, a fire department, a company, etc. The master key is only stored in the base station of an organization. When a new sensor is needed to add or to replace one node in a network, an administration system completes the initialization process and puts it into the networks. This enhances effectively the security of the sensor networks. It will be discussed in detail in the security analysis in the following section.

2) *Encrypting message*

Once the initialization is finished, a sensor network is deployed. A node has its private key and the public parameters. If a node A wants to send a message to another node B with identity Id_B, it can run the function Encryp of Algorithm 2.1 to get ciphertext, where the public key is the identity Id_B. Therefore, unlike traditional application of public-key infrastructure, a Certification Authority (CA) could be eliminated in identity-based cryptography for sensor networks, and the problem of impersonation could be resisted by using an identity-based signcryption scheme [24,25].

3) *Decrypting message*

Plaintext can be recovered by running the Decrypt function of Algorithm 2.1 with the node's private key. In a sensor network without base station, only the node knows its private key.

IV. Analysis of Our Scheme

This section focuses on analysis of efficiency and complexity of our scheme. By comparing with other key agreement schemes and encryption algorithms, we discuss benefits and drawbacks of the scheme in security and efficiency.

A. Efficiency Analysis

1) Comparison with PKI

IBE has some special characteristics and properties compared with PKI. We have

(1) Public keys in IBE are arbitrary strings or "identities". They can be names, roles, email addresses, etc. This makes it possible for a sender to send a message whenever he wants, while in PKI public keys should be generated and distributed to senders before sending a message. Our key predistribution scheme for wireless sensor networks benefits from this property. In fact, we can generate private keys in initialization phase. No key preditribution is needed in this case.

(2) Private keys in IBE are derived from the identities by a trusted Private Key Generator (PKG) using a master key, while in PKI both public and private keys are created by users themselves. This gives one reason that why PKI is not considered as a good choice for key agreement and encryption in wireless sensor networks. In a system with RAS algorithm, an authentication process is executed before establishment of a secure communication, whereas this process is unnecessary in IBE-based algorithms.

(3) The most common criticism on using PKI in sensor networks is its computational complexity and communication overhead. Recently, a number of studies have been conducted to address PKC for sensor networks [18,19,28,29]. For example, Gura et al. show that Elliptic Curve Cryptography (ECC) signature verification takes 1.62s with 160-bit keys on ATmega128 8MHz processor, a processor used for Crossbow motes platform [17]. These results indicate that ECC-based algorithms have some advantages and will soon be available for sensor networks, in despite of comparing with the symmetric key cryptography, PKC is still much more expensive.

As we all known, IBE algorithms are based on ECC. Research results show that the traditional RSA algorithm with 1024-bit key (RSA-1024) provides the currently accepted security level, and is equivalent in strength to ECC with 160 bit keys (ECC-160) and to symmetric key with 80 bit [30]. Therefore, the length of the keys is much more short than that of the traditional RSA algorithms. As a result, it economizes the storage resources and computing cost.

2) Comparison with symmetric key encryption

Applications of symmetric key system in wireless sensor networks have been widely investigated. Compared to IBE algorithms, in symmetric key system, an extra key distribution must be performed prior to deployment of a sensor network. Secret keys are stored in nodes after distributing operation. There are two extreme cases in storing secret keys. One is to let each sensor keep in memory only one secret key (a global master secret key) shared by all nodes in a sensor network. The other is to let each node carry all *N-1* secret pairwise keys, where *N* is the total number of nodes in a sensor network. Evidently, these two mechanisms are impractical. A random key pre-distribution scheme and its variants are proposed [6,9,10], where at least *q* keys selected from a key pool are stored in each node. When a node wants to communicate with another node, a key discovery operation should be performed. However, in IBE algorithms, each node stores only public parameters and owner private key. Neither key predistribution nor key discovery is needed. At the same time, IBE algorithms with 160 bit keys provide currently a sufficient security level. Therefore, in terms of memory requirement and key discovery in wireless sensor networks, our algorithm has a better performance than symmetric key encryption algorithms. But in encrypting and decrypting operations it seems that symmetric key algorithms offer a better performance in computing cost. A detail comparison could be an interesting future work. A glance at the computation cost gives that our scheme in encryption with the full version of IBE algorithm mainly requires four hash-function evaluations, two XOR operations, and one map computation. Similarly, for the basic version, computation cost only consists of two hash-function evaluations, one XOR operation and one map computation (see Table 1).

Table 1 Computation cost

	Full version		Basic version [23]	
	encryption	decryption	encryption	decryption
$\hat{e}$ evaluation	1	1	1	1
Hashing	4	3	2	1
XOR computing	2	2	1	1
multiplication	1	1	1	0
exponentiation	1	0	1	0

The complexity of a $\hat{e}$ evaluation is $O(\log_2 p)$[23], while a hashing function costs about O(n). It gives thus the complexity of our scheme is bout $O(\log_2 p)+O(n)$.

B. Security Analysis

Theorem 2.1 shows that the Boneh-Franklin IBC algorithm is a chosen ciphertext secure IBE (IND-ID-CCA) under some assumption. The results in [23] indicate that it is also a semantically secure identity based encryption scheme (IND-ID-CPA). Furthermore, in symmetric key system

private keys are stored in at least two nodes, while in our scheme private keys are stored only in one node. This can enhance security level of sensor networks with IBE algorithms.

In order to add new node in wireless sensor networks with symmetric key technique, some private keys have to be distributed to the new node. Also, some index information has to be changed in case a node is deleted. But in our scheme, based on IBE algorithms, adding or deleting a node does not affect other nodes, because only identities of nodes are used as public keys. The scheme is independent of network size. Moreover, it is easy to reach a time-stamped identity by using "bob@company || 03" as a public key [24].

V. SIMULATION

Simulating an IBE scheme, e.g. for exchanging public keys in a sensor network is a vital part of our work. We will examine the CPU time and memory requirement by comparing our scheme with DES and RAS schemes. In our simulation a sensor network contains 50 nodes. Each of them has an Id number. The configuration of a computer system is Intel Pentium M 1.73GHz, 768MB RAM and the TinyOS operating system which provides low-level event and task management..

TinyOS [31] was initially developed by the U.C. Berkley EECS Department, and is an event based open-source operating system designed for use with embedded networked sensors. More specifically, it is designed to support the concurrency intensive operations required by networked sensors with minimal HW requirements. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by severe memory constraints inherent in sensor networks..

Table 2 and Table 3 give the average CPU times of all nodes of IBE scheme and RAS scheme in both encryption and decryption respectively. It shows that, for a RAS scheme, the computation time increases with the length of keys. At the same security level, an IBE scheme with 160 –bit key takes 6.8s, while a RAS scheme needs 29s. Moreover, the management of keys in RAS is more complex than that in IBE. As for a DES scheme with 64-bit key, it takes only 0.00139s. However, an extra key distribution and a key management must be performed.

The memory requirement is given in Table 4. The IBE scheme needs 738 byts RAM.

Table 2 CPU time for IBE scheme (second)

key length	encryption	decryption
64	4	3.3
128	6	5.2
160	6.8	5.2
256	9.5	7.2

Table 4 CPU time for RSA scheme (second)

key length	encryption	decryption
128	0.07	0.12
256	0.47	0.82
512	3.3	6
1024	29	47

Table 5 Memory requirment of DES、IBE and RSA schemes (byts)

	IBE	DES	RSA
RAM	738	26114	1796
ROM	54658	13482	86176

VI. CONCLUSION

Wireless sensor networks are being deployed for a wide variety of applications. It is an important challenge to find out practical security protocols for wireless sensor networks due to limitation of power, computation and storage resources.

Symmetric key techniques are attractive due to their energy efficiency. But the drawbacks of symmetric key techniques are evident in terms of key management and security. Public key infrastructure is considered to be not suitable to provide security for wireless sensor networks because of complexity. But some studies on elliptic curves cryptography indicate that algorithm based on this kind of cryptography could be a potential choice. Fortunately, a practical identity-based cryptography is proposed recently, which gives a possibility to employ elliptic curves cryptography in wireless sensor networks. Compared with the traditional asymmetric and symmetric key techniques, the distinguishing characteristic of identity-based encryption is the ability to use any string as a public key, for example, an email address, a name, etc. Based on the Boneh-Franklin IBE algorithms, we proposed an identity-based key agreement and encryption scheme for wireless sensor networks. Analysis shows that our scheme has some advantages in terms of key management, storage requirement and security.

Our future work will focus on a comprehensive analysis and comparison of our scheme with others. Particularly, the complexity analysis should be an interesting topic.

ACKNOWLEDGMENT

This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant Nos. BK2004218 and BK2003106.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," c*omputer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] A. Perrig, R. Canetti, Briscoe, J. Tygar, and D. Song, "TESLA: Multicast source authentication transform," IRTF draft, draft-irtf-smug-tesla-00.txt, November 2000.

[3] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "SPINS: Security protocols for sensor networks," in *Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, July 2001.

[4] 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, February 2005.

[5] 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, April 2005.

[6] 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. 2002.

[7] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proceedings of the 10th Annual Network and Distributed System Security Symposium (NDSS'03)*, pp. 263–276, February 2003.

[8] D. Liu and P. Ning, "Multi-level μTESLA: Broadcast authentication for distributed sensor networks," *ACM Transactions in Embedded Computing Systems (TECS)*, vol. 3, no.4, pp. 800-836, 2004.

[9] R. D. Pietro, L. V. Mancini, and A. Andmei, "Random key assignment for secure wireless sensor networks," in *ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, pp. 62-71, 2003.

[10] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE symposium on Research in Security and Privacy*, pp. 197–213, 2003.

[11] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *2003 ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, pp. 72-82, 2003.

[12] R. Anderson and M. Kuhn, "Tamper resistance—A cautionary note," in *Proceedings of the 2nd Usenix Workshop on Electronic Commerce*, pp. 1–11, 1996.

[13] R. Blom, "An optimal class of symmetric key generation systems," in *Advances in Cryptology: Proceedings of EUROCRYPT 84*, T. Beth, N. Cot, and I. Ingemarsson, Eds. *Lecture Notes in Computer Science*, vol. 209, Springer-Verlag, Berlin, 335–338. 1985.

[14] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccard, and M. Yung, "Perfectlysecure key distribution for dynamic conferences," *Lecture Notes in Computer Science*, vol. 740, pp. 471–486, 1993.

[15] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, "Analyzing and modeling encryption overhead for sensor network nodes," in *Proceedings of the 1st ACM internationalworkshop on Wireless sensor networks and applications*, San Diego, California, USA, September 19 2003.

[16] G. Gaubatz, J. Kaps, and B. Sunar, "Public keys cryptography in sensor networks – revisited," in *The Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS)*, 2004.

[17] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs," in *Proceedings of the Workshop on Cryptography Hardware and Embedded Systems* (CHES 2004), Boston, August 11-13 2004.

[18] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," in *The First IEEE International Conference on Sensor and Ad Hoc Communications and Networks*, Santa Clara, California, pp. 71-79, October 2004.

[19] W. Du, R. Wang, and P. Ning, "An efficient scheme for authenticating public keys in sensor networks," *MobiHoc'05*, May 25–27, 58-67, UrbanaChampaign, Illinois, USA, 2005.

[20] A. Shamir, "Identity-based cryptography and signature schemes," *Advances in Cryptology*, CRYPTO'84, *Lecture Notes in Computer Science*, vol. 196, pp. 47-53, 1985.

[21] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *J. Cryptology*, vol. 1, pp. 77-94, 1988.

[22] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," In *Proceedings of CRYPTO'86*, pp. 186-194, 1986.

[23] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology*, CRYPTO 2001, *Lecture Notes in Computer Science*, vol. 2139, pp. 213-229,2001.

[24] X. Boyen, "Multipurpose Identity-based signcryption, a Swiss army knife for identity-based cryptography," in *Proceedings of the 23rd Interna. Conf. On Advances in Cryptology, Lecture Notes in Computer Science, vol.* 2729, pp. 383-399, 2003.
[25] L. Chen and C. Kudla, "Identity-based authenticated key agreement protocols from pairings," *Cryptology ePrint Archive*, Report 2002/184, http://eprint.iacr.org/2002/184, 2002.
[26] B. Lynn, "Authenticated identity-based encryption," *Cryptology ePrint Archive*, Report 2002/072, http://eprint.iacr.org/2002/072, 2002.
[27] B. R. Waters, "Efficient Identity-Based Encryption Without Random Oracles," *Cryptology ePrint Archive*, Report 2004/180, http://eprint.iacr.org/2004/180, 2004.
[28] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks SASN'04*, pp. 59-64, October 2004.
[29] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications*, pp.324-328, March 2005.
[30] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 62-67, Feb 2004.
[31] TinyOS, *TinyOS 1.1.0*, http://tinyOS.net.