On Broadcast Authentication in Wireless Sensor Networks

Kui Ren[1], Wenjing Lou[1], Kai Zeng[1] and Patrick J. Moran[2]

[1]Worcester Polytechnic Institute, Worcester, MA 01609

{kren, wjlou, kzeng }@ece.wpi.edu

[2]AirSprite Technologies, Inc., Marlborough, MA 01752

pmoran@airsprite.com

Abstract

Broadcast authentication is a critical security service in wireless sensor networks (WSNs), since it enables users to broadcast the WSN in an authenticated way. Symmetric key based schemes such as μTESLA and multilevel μTESLA have been proposed to provide such services for WSNs; however, these schemes all suffer from serious DoS attacks due to the delay in message authentication. This paper presents several effective public key based schemes to achieve immediate broadcast authentication and thus overcome the vulnerability presented in the μTESLA-like schemes. Several cryptographic techniques, including Merkle hash tree and identity-based signature scheme, are adopted to minimize the scheme overhead regarding the costs on both computation and communication. A quantitative energy consumption analysis of the proposed schemes is given in detail. We believe that this paper can serve as the start point towards fully solving the important multisender broadcast authentication problem in WSNs.

Index: Security, Wireless Sensor Network, Broadcast Authentication, Multisender

I. Introduction

Wireless sensor networks (WSNs) have enabled data gathering from a vast geographical region, and present unprecedented opportunities for a wide range of tracking and monitoring applications from both civilian and military domains [1], [2], [10], [24]. In these applications, WSNs are expected to process,

The preliminary version of this paper appears in [42].

store and provide the sensed data to the network users upon their demands. As the most common communication paradigm, the network users are expected to issue the queries to the network before obtaining the information of their interest. Furthermore, in wireless sensor and actuator networks (WSANs) [2], the network users may even need to issue their commands to the network (probably based on the information he received from the network). In both cases, there could be a large number of users in the WSNs, which could be either mobile or static. And the users may use their mobile clients to query or command the WSNs from anywhere in the network. Obviously, broadcast/multicast[1] operations are fundamental to the realization of these network functions. Hence, it is also highly important to ensure broadcast authentication for the security purpose.

Broadcast authentication in WSNs has been first addressed by μTESLA in [5]. In μTESLA, the user of WSNs is assumed to be one or a few fixed sinks, which are always assumed to be trustworthy. The scheme adopts a one-way hash function $h()$ and uses the hash preimages as keys in a Message Authentication Code (MAC) algorithm. Initially, sensor nodes are preloaded with $K_0 = h^n(x)$, where x is the secret held by the sink. Then, $K_1 = h^{n-1}(x)$ is used to generate MACs for all the broadcast messages sent within time interval 1. At time interval 2, the sink broadcasts K_1, and sensor nodes verify $h(K_1) = K_0$. The authenticity of messages received during time interval 1 is then verified using K_1. This delayed disclosure technique is used for the entire hash chain and thus demands loosely synchronized clocks between the sink and sensor nodes. μTESLA is later enhanced in [6], [7] to overcome the length limit of the hash chain. Most recently, μTESLA is also extended in [8] to support multiuser scenario at the cost of higher communication overheads per message.

It is generally held that μTESLA-like schemes have the following shortcomings even in the single-user scenario: 1) all the receivers have to buffer all the messages received within one time interval; 2) they are subject to Wormhole attacks [9], where messages could be forged due to the propagation delay of the disclosed keys. Furthermore, here we point out a more serious vulnerability of μTESLA-like schemes when they are applied in multi-hop WSNs. Since sensor nodes have to buffer and forward all the messages received within one time interval, an adversary can hence flood the whole network arbitrarily. All he has to do is to claim that the messages belong to the current time interval which should be buffered and forwarded for authentication until next time interval. Since wireless transmission is very expensive in

[1]For our purpose, we do not distinguish multicast from broadcast in this paper.

WSNs[2], and WSNs are extremely energy constrained, the ability for an attacker to flood the network arbitrarily could cause devastating DoS attacks. Moreover, this type of DoS attacks become even more devastating in multiuser scenario, since the adversary can easily generate more bogus messages without being detected. Obviously, all these attacks are due to authentication delay of the broadcast messages. In [9], TIK is proposed to achieve immediate key disclosure and hence immediate message authentication based on precise time synchronization between the sink and receiving nodes. However, this technique is not applicable in WSNs as pointed out by the authors. Therefore, the problem of broadcast authentication still remains wide open in WSNs.

In this paper, we resort to public key cryptography for effective solutions. We approach broadcast authentication problem in WSNs under multiuser scenario by designing PKC-based solutions with minimized computational and communication costs. On the one hand, we aim to achieve immediate message authentication and be immune to DoS attacks in the presence of both user revocation and node compromise. On the other hand, we want to optimize both computational and communication costs.

We propose three different PKC-based approaches and provide in-depth analysis on their pros and cons. In all three approaches, the users are always authenticated through their public keys. We first propose a straightforward certificate-based approach and point out its inherent vulnerability on certificate revocation management when applied in WSNs. To avoid certificate revocation problem, we further propose a Merkle hash tree based scheme to manage user public keys. In this way, the storage overhead at sensor nodes is a single hash value with L bytes; however, the additional communication overhead per hop is $L * \log_2 N$ bytes, where N is the number of network users. The Merkle hash tree based scheme is further enhanced to have a $L * m$-byte storage overhead and a $L * \log_2 \frac{N}{m}$-byte communication overhead, where m is the number of hash values that need to be stored by sensor nodes. Since the WSN under consideration is usually very large and thus has many hops, $L * \log_2 \frac{N}{m}$ bytes additional communication overhead per hop could still be very high, when N is large. To eliminate the additional communication overhead, we further propose an ID-based authentication technique. The scheme is based on ID-based cryptography, in which a user's public key is his ID information, and only a valid user can have the corresponding private key. Therefore, the ID-based scheme is highly efficient in communication; however, it suffers from high computation cost. We analyze the pros and cons of all the proposed schemes quantitatively with respect

[2]Wireless transmission of a bit can require over 1000 times more energy than a single 32-bit computation, as shown in [24].

to both computational and communication cost.

This paper makes the following contributions:

- We revisit the problem of multisender broadcast authentication in WSNs, and for the first time, point out a serious security vulnerability inherent to the existing symmetric-key based μTESLA-like schemes.
- We come up with several PKC-based schemes to address the problem. Both computational and communication costs are analyzed in depth in the scheme. Several novel cryptographic techniques are adopted to minimize the costs, including Merkle hash tree authentication technique and ID-based signature scheme.
- We analyze both the performance and security resilience of the proposed schemes. A quantitative energy consumption analysis is given in detail.

The remaining part of this paper is organized as follows. In Section II, we introduce the cryptography mechanisms used in the paper. Section III presents the system assumptions, adversary model and security objectives of this paper. Then in Section IV, we introduce our proposed schemes and detail the underlying design logic. Section V is the scheme analysis. We finally conclude the paper in Section VI.

II. PRELIMINARIES

A. Merkle hash tree technique

We illustrate the construction and application of the Merkle hash tree [22] through an example. To authenticate data values $n_1, n_2, ..., n_w$, the data source constructs the Merkle hash tree as depicted in Fig. 1, assuming that $w = 4$. The values of the four leaf nodes are the message hashes, $h(n_i), i = 1, 2, 3, 4$, respectively, of the data values under a one-way hash function $h()$ (e.g., SHA-1 [28]). The value of each internal node is derived from its child nodes. For example, the value of node A is $h_a = h(h(n_1)|h(n_2))$. The data source completes the levels of the tree recursively from the leaf nodes to the root node. The value of the root node is $h_r = h(h_a|h_b)$, which is used to commit to the entire tree to authenticate any subset of the data values n_1, n_2, n_3, and n_4 in conjunction with a small amount of auxiliary authentication information `AAI` (i.e., $\log_2 N$ hash values with N as the number of leaf nodes). For example, a user, who is assumed to have the authentic root value h_r, requests for n_3 and requires the authentication of the received n_3. Besides n_3, the source sends the `AAI` $:< h_a, h(n_4) >$ to the user. The user can then check

the authenticity of the received n_3 by first computing $h(n_3)$, $h_b = h(h(n_3)|h(n4))$ and $h_r = h(h_a|h_b)$, and then checking if the calculated h_r is the same as the authentic root value h_r. The user accepts n_3, only if this check is positive.

B. ID-based cryptography

Identity-based cryptography (IBC) is receiving extensive attention as a powerful alternative to traditional certificate-based cryptography. Its main idea is to make an entity's public key directly derivable from its publicly known identity information. Although the idea of IBC dates back to 1984 [25], only recently has its rapid development taken place due to the application of the *pairing* technique outlined below.

Properly select two large primes p and q, and let $\mathbb{E}/\mathbb{F}_p$ indicate an elliptic curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. We denote by $\mathbb{G}_1$ a q-order subgroup of the additive group of points of $\mathbb{E}/\mathbb{F}_p$, and by $\mathbb{G}_2$ a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^i}$ ($i = 2, 3$ or 6). The Discrete Logarithm Problem (DLP) is required to be hard[3] in both $\mathbb{G}_1$ and $\mathbb{G}_2$. For us, a pairing is a mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

1. *Bilinear*: For $\forall\ P, Q, R, S \in \mathbb{G}_1$, $\hat{e}(P + Q, R + S) = \hat{e}(P, R)\hat{e}(P, S)\hat{e}(Q, R)\hat{e}(Q, S)$.
   Consequently, for $\forall\ c, d \in \mathbb{Z}^*_q$, we have $\hat{e}(cP, dQ) = \hat{e}(cP, Q)^d = \hat{e}(P, dQ)^c = \hat{e}(P, Q)^{cd}$, etc.
2. *Non-degenerate*: If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P) \in \mathbb{F}^*_{p^2}$ is a generator of $\mathbb{G}_2$.
3. *Computable*: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

Note that $\hat{e}$ is also *symmetric*, i.e., $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in \mathbb{G}_1$, which follows immediately from the bilinearity and the fact that $\mathbb{G}_1$ is a cyclic group. Modified Weil [26] and Tate [27] pairings are examples of such bilinear maps for which the *Bilinear Diffie-Hellman Problem* (BDHP) is believed to be hard[4]. We refer to [26], [27] for a more comprehensive description of how these pairing parameters should be selected in practice for efficiency and security.

III. Assumptions, Scheme Objectives, and Design Motivation

System model: In this paper, we consider a very large, spatially-distributed WSN, consisting of a fixed sink and a large amount of sensor nodes. The sensor nodes are not necessarily homogenous in their

[3]It is believed to be computationally infeasible to extract the integer $x \in \mathbb{Z}^*_q = \{a | 1 \leq a \leq q - 1\}$, given $P, Q \in \mathbb{G}_1$ (respectively, $P, Q \in \mathbb{G}_2$) such that $Q = xP$ (respectively, $Q = P^x$).

[4]It is believed that, given $< P, xP, yP, zP >$ for random $x, y, z \in \mathbb{Z}^*_q$ and $P \in \mathbb{G}_1$, there is no algorithm running in expected polynomial time, which can compute $\hat{e}(P, P)^{xyz} \in \mathbb{G}_2$ with non-negligible probability.

functionalities and capabilities. The WSN under consideration is aimed to offer information services to a large number of network users that roam in the network, in addition to the fixed sink. These WSN users include mobile sinks, vehicles, and people with mobile clients, and they are assumed to be more powerful than sensor nodes in terms of computation and communication abilities. For example, the network users could include a number of doctors, nurses, medical equipments (acting as actuators) and so on, in the case of CodeBlue [41], where the WSN is used for emergency medical response. These network users broadcast queries/commands through sensor nodes in their vicinity, and expect the replies that reflect the latest sensing results. The network users also directly communicate with sink or the backend server if needed. We assume that the sink is always trustworthy but the sensor nodes are subject to compromise. At the same time, the users of the WSN may be dynamically revoked due to either membership changing or compromise, and the revocation pattern is not restricted. As the μTESLA-like schemes, we also assume that the WSN time is loosely synchronized.

Adversary model: We assume that the adversary's goal is to inject bogus messages into the network, attempt to deceive sensor nodes, and obtain the information of his interest. Additionally, Deny of Service (DoS) attacks such as bogus message flooding, aiming at exhausting constrained network resources, is another important focus of the paper. We assume that the adversary is able to compromise both network users and sensor nodes. The adversary hence could exploit the compromised users/nodes for such attacks. More specifically, we consider the following types of attacks: 1) The adversary may directly broadcast bogus messages to the WSN by himself; 2) The adversary may use one or more compromised nodes to propagate bogus messages to the WSN by pretending that the messages are initiated by legitimate network users; 3) The adversary may use one or more compromised users to broadcast messages to the WSN. However, we do assume that adversary cannot compromise an unlimited number of sensor nodes. Neither can they break any cryptographic primitive on which we base our design. Otherwise, it is unlikely for any feasible security solution to be designed.

Security objectives: Given the adversary model above, our security objective is straightforward. First, user authentication is needed so that illegitimate users will be excluded from injecting bogus messages. Second, user revocation mechanisms have to be implemented so that sensor nodes could deal with user revocations. Third, the authenticity of any message broadcast by a user should be able to be verified by every receiving node. In summary, all messages being broadcast to the WSN should be authenticated so

that any bogus ones issued by the illegitimate users and/or compromised sensor nodes can be efficiently and deterministically rejected/filtered.

Design motivation: At the time when μTESLA was proposed, sensor nodes were assumed to be extremely resource constrained, especially with respect to computation capability, bandwidth availability, and energy supply [5]. Therefore, public key cryptography (PKC) was thought to be forbiddingly computationally expensive, although it could provide much simplified solutions with much stronger security strengths. However, recent studies [16], [17] showed that, contrary to widely held beliefs, PKC with software implementations only is very viable on sensor nodes. For example in [16], it was reported that Elliptic Curve Cryptography (ECC) signature verification takes 1.61s with 160-bit keys on ATmega128 8MHz processor, a processor used for the current Crossbow motes platform [18]. The benefits of transmitting smaller ECC keys and hence smaller messages/signatures[5] will in turn be more significant. Moreover, next generation sensor nodes are expected to combine ultra-low power circuitry with so-called power scavengers such as Heliomote [15], [19], which allows continuous energy supply to the nodes. At least $8-20\mu$W of power can be generated using MEMS-based power scavengers [12], [13]. Other solar-based systems are even able to deliver power up to 100mW for the Mica Motes [19], [20]. These results indicate that, with the advance of fast growing technology, PKC is no longer impractical for WSNs, although still expensive for the current generation of sensor nodes. And its wide acceptance is expected in the near future [17].

IV. The Proposed Schemes

PKC-based solutions can realize immediate message authentication and thus overcome the delayed authentication problem presented in μTESLA-like schemes. However, the straightforward solutions such as certificate-based approach can not be directly applied in WSNs due to their high scheme overhead as we analyze below. More advanced techniques have to be adopted to achieve a desirable scheme performance.

A. The Certificate-Based Authentication Scheme

The scheme: Each user of the WSN is equipped with a public/private key pair (PK/SK), and signs every message he broadcasts with his SK using a digital signature scheme such as RSA or DSA [29], [30]. To

[5]To provide the same level of security strength, RSA requires a key of 1024-bit, while ECC requires a key size of 160-bit.

prove the user's ownership over his public key, the sink[6] is also equipped with a public/private key pair and serves as the certificate authority (CA). The sink issues each user a public key certificate, and such a certificate, to its simplest form, consists of the following contents:

$$\texttt{Cert}_{U_{ID}} = U_{ID}, \mathsf{PK}_{U_{ID}}, \texttt{ExpT}, \text{SIG}_{\mathsf{SK}_{Sink}}\{h(U_{ID}||\texttt{ExpT}||\mathsf{PK}_{U_{ID}})\}, \texttt{para},$$

where U_{ID} denotes the user's ID, $\mathsf{PK}_{U_{ID}}$ denotes his public key, $\texttt{ExpT}$ denotes certificate expiration time and $\text{SIG}_{\mathsf{SK}_{Sink}}\{h(U_{ID}||\texttt{ExpT}||\mathsf{PK}_{U_{ID}})\}$ is a signature signed over $h(U_{ID}||\texttt{ExpT}||\mathsf{PK}_{U_{ID}})$ with SK_{Sink}. Hence, a broadcast message is now of the form as follows:

$$< M, tt, \text{SIG}_{\mathsf{SK}_{U_{ID}}}\{h(U_{ID}||tt||M)\}, \texttt{Cert}_{U_{ID}} > \qquad (I)$$

Here, M denotes the broadcast message and tt denotes the current time. Then, sensor nodes are enabled to verify the authenticity of the received messages by preloading PK_{Sink} before the network deployment. The verification contains two steps: the certificate verification and the signature verification.

Analysis: This straightforward scheme suffers from many severe drawbacks. Firstly and most importantly, it is inefficient to support user revocation in this scheme. In order to support user revocation and hence certificate revocation, sensor nodes have to receive and store a certificate revocation list (CRL). Clearly, the CRL requires a storage space linear to the total number of revoked certificates at each sensor node. However, this is practically infeasible due to the stringent storage limitation of sensor nodes, especially given a large number of users or a highly dynamic membership changing scenario. For example, assuming that a public key is 20-byte long, a CRL containing only $1,000$ revoked certificates is at least of size 19.5 KB even in the simplest format (i.e., containing only the public key). At the same time, resorting to the sink on-demand for CRL verification is obviously inefficient either, because this could introduce too much communication cost. Embedding validity interval into the certificate does not really help reduce the storage overhead much, since the revocation pattern is not available a priori. Secondly, to authenticate each message, it always takes two signature verification operations, instead of one. This is because the certificate should always be authenticated in the first place.

[6]We assume that the sink represents the network planner.

B. The Basic Merkle Hash Tree Based Authentication Scheme

Having observed the CRL problem inherent to the first scheme, we next propose a Merkle hash tree based authentication scheme, which is highly storage efficient.

Scheme initialization: The sink collects all the public keys of the current network users and constructs a merkel hash tree. Specifically, we construct N leaves with each leaf corresponding to a current user of the WSN. For our problem, each leaf node contains the binding between the corresponding user ID and the public key of the user, that is, $h(U_{ID}, \mathsf{PK}_{U_{ID}})$. The values of the internal nodes are determined with the same method as in Section II.A. We denote the value of the final root node of the hash tree as h_r. Then, the sink preloads/broadcasts each sensor node with this value before network deployment or during the network operation time. However, if h_r is broadcast during the network operation time, h_r should be signed by the sink to prove its authenticity. Of course, in this case, sensor nodes should be preloaded with the sink's public key. At the same time, each user should obtain its $\mathtt{AAI}$ according to his corresponding leaf node's location in the Merkle hash tree. Let T denote all the nodes along the path from a leaf node to the root (not including the root). Then A is defined as the set of nodes corresponding to the siblings of the nodes in T; and $\mathtt{AAI}$ further corresponds to the values associated with the nodes in A. Obviously, $\mathtt{AAI}$ is $(L * \log_2 N)$ bytes, where the hash value is L bytes in length.

Message authentication: Now a message sent by a user U_{ID} is of form

$$< M, tt, \mathrm{SIG}_{\mathsf{SK}_{U_{ID}}}\{h(U_{ID}||tt||M)\}, U_{ID}, \mathsf{PK}_{U_{ID}}, \mathtt{AAI}_{U_{ID}} > \quad (II)$$

Each node verifies such a message in two steps. First, it verifies $\mathsf{PK}_{U_{ID}}$ using $\mathtt{AAI}_{U_{ID}}$ attached in the message and h_r stored by itself. The verification operation is a chain of hash operations with the final value equal to h_r as we demonstrated in section II.A. A different final value other than h_r suggests the invalidity of the corresponding public key. Second, the sensor node verifies $\mathrm{SIG}_{\mathsf{SK}_{U_{ID}}}\{h(U_{ID}||tt||M)\}$ using $\mathsf{PK}_{U_{ID}}$. Upon user revocation and/or addition, the sink updates the Merkle hash tree and obtains a new h_r. This new h_r is then signed by the sink using SK_{Sink} and broadcast to sensor nodes immediately. Furthermore, each current user also obtains his updated $\mathtt{AAI}_{U_{ID}}$ from the sink.

Analysis: In this scheme, a user does not need a certificate to prove the binding to his public key. Instead, a Merkle hash tree technique is used. A revoked or invalid user public key will never pass the verification, as long as the user holds the up-to-date root node value h_r. Hence, in this scheme, certificates are no longer

necessary and can be eliminated. Furthermore, the user revocation problem (i.e., certificate revocation problem) is now reduced to the problem of updating sensor nodes a single hash value h_r, which requires a storage space of only L bytes. Assuming that SHA-1 [28] is used, $L = 20$ bytes. However, the scheme is communication inefficient when N becomes large. This is because the size of AAI grows logarithmically with N. Assume $L = 20$ bytes, AAI alone is of size 200 bytes, when $N = 1,024$; and $|\texttt{AAI}| = 260$ bytes, when $N = 8,192$.

C. The Enhanced Merkle Hash Tree Based Authentication Scheme

In the above scheme, the storage overhead is only one hash value, i.e., L bytes, but the communication overhead is no less than $L * \log_2 N$ bytes. We hence, want to make a compromise between the storage and communication overheads. That is, we increase the number of stored hash values to reduce the size of AAI.

We illustrate how to do it through an example. In Fig.1, h_r is made public and stored by the authenticator. Hence, the user corresponding to leaf node n_3 must have $\texttt{AAI} :< h_a, h(n_4) >$. However, if both h_a and h_b are made public and stored by the authenticator, the corresponding AAI now contains $h(n_4)$ only. Therefore, by trimming down the Merkle hash tree constructed in the above scheme, we can have a set of smaller Merkle hash trees. If each sensor node is loaded with all the values of the root nodes corresponding to these smaller trees, then the size of AAI can be reduced to the height of the smaller trees multiplying L bytes. In fact, if we remove k levels of the original Merkle tree, the communication overhead is reduced by $k * L$ bytes. However, the storage cost increases to $2^k * L$ bytes. Note that if we require sensor nodes to store all the leaf values, the scheme is reduced to the trivial memorize-all-keys case, which demands $N * L$ bytes storage space.

Analysis: Since sensor nodes are storage constrained, the value of k is obviously limited. Given that $m = 2^k$ hash values can be stored by each sensor node, the size of AAI is now $(L * \log_2 \frac{N}{m})$ bytes. If $N = 1,024$ and $m = 32$, this is 100 bytes; and if N is increased to $8,192$, this is 160 bytes. If m is made to be 64, then the size of AAI will be 80 bytes, given $N = 1,024$, and 140 bytes, given $N = 8,192$. This result is much improved as compared to the above basic scheme. When $N = 8,192$, the message overhead in this enhanced scheme is 120 bytes less than that of the basic Merkle hash tree based scheme. This gain comes at the cost of increased storage overhead, which is now $64 * 20 = 1,280$ bytes $= 1.25$

KB. Therefore, this scheme is still communication inefficient when N is large. We defer the detailed analysis to Section V.

D. ID-Based Authentication Scheme

In this section, we propose an ID-based authentication scheme. In contrast to the Merkle hash tree based schemes, the proposed ID-Based authentication scheme requires sensor nodes to memorize the revoked user IDs only, and adopts an automatic public key update technique.

In our ID-based authentication scheme, the time is divided into consecutive time intervals, denoted by v_1, v_2,..., and we assume that sensor nodes and users are loosely synchronized. We then adopt $U_{ID}||v_i$ as user U_{ID}'s public key under an ID-based signature scheme [32]. In this way, before a user wants to authenticate itself to the sensor nodes, he has to firstly obtain its private key from the sink. And since each obtained private key is valid only within the current time interval, every user has to obtain a new private key from the sink at the beginning of each time interval. Now upon user revocation, the sink only needs to broadcast the corresponding user IDs to the sensor nodes. Each sensor node stores a local copy of such revoked IDs only within the current interval and dumps them afterwards. The scheme works as follows.

Scheme initialization: Prior to network deployment, we assume that the sink does the following operations:

1. Generate the pairing parameters $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e})$, as described in Section II.B. Select an arbitrary generator P of $\mathbb{G}_1$.
2. Choose two cryptographic hash functions: H, mapping strings to non-zero elements in $\mathbb{G}_1$, and h, mapping arbitrary inputs to fixed-length outputs, e.g., SHA-1 [28].
3. Pick a random number $\kappa \in \mathbb{Z}_q^*$ as the network master secret and set $P_{pub} = \kappa P$.
4. Preload each sensor node with the public system parameters $(p, q, \mathbb{E}/\mathbb{F}_p, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, H, h, P, P_{pub})$.
5. Preload each user U_{ID} with the private key $\mathsf{SK}_{U_{ID}} = \kappa H(U_{ID}||v_1)$

Message broadcast authentication: Assume that user U_{ID} wants to broadcast a message M. He first obtains its private key as $\mathsf{SK}_{U_{ID}} = \kappa H(U_{ID}||v_i)$, where v_i is the current time interval. U_{ID} then picks a random $\alpha \in \mathbb{Z}_q^*$ and computes $\theta = \hat{e}(P, P)^{\alpha}$. U_{ID} further computes

$$U_{x,y} = h(M \parallel tt \parallel \theta)\mathsf{SK}_{U_{ID}},$$

and

$$\sigma_{x,y} = U_{x,y} + \alpha P.$$

$< \sigma_{x,y}, h(M \parallel tt \parallel \theta) >$ is the signature on message M. And the broadcast message is now of form

$$< U_{ID}, tt, M, \sigma_{x,y}, h(M \parallel tt \parallel \theta) > \qquad (III)$$

Upon receiving Message (III), each sensor node verifies its authenticity in the following way: It checks the current time $\overline{tt}$ and determines whether or not the received message is fresh. Assume δ is the predefined message propagation time limit. Then, we should have $\overline{tt} - tt \leq \delta$. If so, the sensor node further computes,

$$\theta' = \hat{e}(\sigma_{x,y}, P)\hat{e}(H(U_{ID}||v_i), -P_{pub})^{h(M\|tt\|\theta)},$$

using the current time interval v_i. If the message is authentic, we will have

$$\begin{aligned}
\theta' &= \hat{e}(\sigma_{x,y}, P)\hat{e}(H(U_{ID}||v_i), P_{pub})^{-h(M\|tt\|\theta)} \\
&= \hat{e}(h(M \parallel tt \parallel \theta)\mathsf{SK}_{U_{ID}} + \alpha P, P)\hat{e}(H(U_{ID}||v_i), \kappa P)^{-h(M\|tt\|\theta)} \\
&= \hat{e}(h(M \parallel tt \parallel \theta)\mathsf{SK}_{U_{ID}} + \alpha P, P)\hat{e}(\kappa H(U_{ID}||v_i), P)^{-h(M\|tt\|\theta)} \\
&= \hat{e}(\mathsf{SK}_{U_{ID}}, P)^{h(M\|tt\|\theta)}\hat{e}(P, P)^{\alpha}\hat{e}(\mathsf{SK}_{U_{ID}}, P)^{-h(M\|tt\|\theta)} = \theta.
\end{aligned} \qquad (1)$$

Therefore, if $h(M \parallel tt \parallel \theta') = h(M \parallel tt \parallel \theta)$, a sensor node considers the message authentic. If the above verification fails, a sensor node considers the message a fabricated or replayed one, and simply dumps it. Otherwise, it propagates the message to the next hop.

Analysis: The pros of the ID-based authentication scheme are two-fold: First, it eliminates the existence of certificate or auxiliary authentication information. Therefore, the resulted message size can be reduced. Second, it requires much smaller storage space to support user revocation, since now only the revoked user IDs have to be stored. Assuming a WSN supporting up to 65,535 users, then two bytes are enough for the length of a user ID. Hence, accumulating the same 1,000 revoked users, now only 2,000 bytes = 1.95 KB storage space is needed. However, the cons of the ID-based authentication scheme are also obvious, since it has a very high computation cost due to the pairing operation involved.

V. Quantitative Performance Comparison

In this section, we present a quantitative performance comparison with respect to the above proposed schemes. Our main concern is the energy consumption spent on message propagation and computation. We start from analyzing the message sizes provided by different schemes, since the message size is directly related to the energy consumption on message propagation.

A. Message size

- The Certificate-based Authentication Scheme: The total message size of form (I) equals to

$$|M| + |tt| + |\mathsf{SIG}_{\mathsf{SK}_{U_{ID}}}\{h(U_{ID}||M)\}| + |\mathtt{Cert}_{U_{ID}}|,$$

  where $|\bullet|$ denotes the size of '$\bullet$' in byte. In its simplest form, the size of a certificate can be significantly larger than that of the message in WSNs generally. As in [35], $\mathtt{Cert}_{U_{ID}}$ is at least of size 86-bytes, even if ECDSA is used[7] [34]. The total message size of form (I) is then 148 bytes, assuming M 20 bytes, tt 2 bytes, and that ECDSA is used.

- The Merkle Hash Tree Based Authentication Scheme: The total message size of form (II) equals to

$$|M| + |tt| + |\mathsf{SIG}_{\mathsf{SK}_{U_{ID}}}\{h(U_{ID}||M)\}| + |U_{ID}| + |\mathsf{PK}_{U_{ID}}| + |\mathtt{AAI}_{U_{ID}}|$$

  Assuming that SHA-1 is used, U_{ID} is 2 bytes, and all the other settings remain the same as above, we have the total message size equal to $(20+2+40+2+20+20*\log_2 N) = 84+20*\log_2 N$ bytes. For its enhanced scheme as presented in Section III.C, the total message size of form (II) is further reduced to $84+20*\log_2 \frac{N}{m}$, as $\mathtt{AAI}$ is now $(L*\log_2 \frac{N}{m})$ bytes. If $N = 1,024$ and $m = 32$, this is 184 bytes; and if N is increased to $8,192$, this is 244 bytes. If m is made to be 64, then the total message size will be 164 bytes, given $N = 1,024$, and 224 bytes, given $N = 8,192$. Note that RSA-1024 is obviously not a choice here, since total message size of form (II) will be $280 + 20*\log_2 \frac{N}{m}$ bytes in this case.

- The ID-Based Authentication Scheme: The total message size of form (III) equals to

$$|U_{ID}| + |tt| + |M| + |\sigma_{x,y}| + |h(M \parallel tt \parallel \theta)|.$$

  Assuming that everything else is same as above, then we remain to determine the size of the signature. The second part of the signature, i.e, $h(M \parallel tt \parallel \theta)$ is a hash value which should be 20 bytes given SHA-1 is used. The size of the first part, i.e., $\sigma_{x,y}$, however, is variable. In our evaluation, the bilinear map $\hat{e}$ used is the Tate pairing [27]. The elliptic curve $\mathbb{E}$ is defined over $\mathbb{F}_p$. The order q of $\mathbb{G}_1$ and $\mathbb{G}_2$ is a 160-bit prime. According to [26], in order to deliver an equivalent level of security to that of

[7]ECDSA is referred to Elliptic Curve Digital Signature Algorithm [34]. While RSA with 1024-bit keys (RSA-1024) provides the currently accepted security level, it is equivalent in strength to ECC with 160-bit keys (ECC-160). And hence, for the same level of security strength, ECDSA uses a much small key size and hence has a small signature size (320-bit).

1024-bit RSA, p should be a 512-bit prime, if $\mathbb{G}_2$ is a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. Furthermore, p could be 340-bit, given the finite field $\mathbb{F}^*_{p^3}$, and 160-bit, given the finite field $\mathbb{F}^*_{p^6}$. As we already know, $\sigma_{x,y}$ is a point of $\mathbb{E}/\mathbb{F}_p$, only one of its X and Y coordinates needs to be transmitted because the other can be easily derived using the curve equation, resulting in an overhead of $|p|$ bits. Therefore, the total message size of form (III) is $44 + |p|$ bytes, ranging from 64 to 108 bytes.

Fig. 2 shows the total message sizes of the different schemes as a function of the number of network users. In Fig. 2, we see that the ID-based scheme (of any p size) has the smallest message size as compared to the others, when N is larger than 500. At the same time, this message size is independent to the number of network users. However, the computation cost of the ID-based scheme is very high. We further see that the certificate-based scheme has a constant message size of 146 bytes, which is also independent to N. Furthermore, we see that the Merkle hash tree based scheme is efficient only when N is up to several hundreds. For example, when N is 512, the size of Message (II) is 144 bytes. Therefore, the Merkle hash tree based scheme is unsuitable for supporting larger numbers of users.

B. Energy consumption on message broadcast

In this subsection, to quantify the impact of message length regarding broadcast in WSNs, we further evaluate the energy consumption due to broadcast of messages of different sizes. We denote by E_{tr} the hop-wise energy consumption for transmitting and receiving one byte. As reported in [16], a Chipcon CC1000 radio used in Crossbow MICA2DOT motes consumes 28.6 and 59.2 μJ to receive and transmit one byte, respectively, at an effective data rate of 12.4 kb/s. Furthermore, we assume a packet size of 41 bytes, 32 for the payload and 9 bytes for the header [16]. The header, ensuing a 8-byte preamble, consists of source, destination, length, packet ID, CRC, and a control byte [16].

For the certificate-based scheme, $\texttt{Cert}_{U_{ID}}$ is at least 86 bytes [16], even if ECDSA-160 is used. The total message size of form (I) is then 148 bytes, assuming M 20 bytes, tt 2 bytes. Hence, there should be 5 packets in total, among which four of them are of size 41 bytes, and one packet is of size 29 bytes. Therefore, there should be $41 * 4 + 29 * 1 + 8 * 5 = 233$ bytes for transmission (including 8-byte preamble per packet). Hence, the hop-wise energy consumption on transmitting Message (I) equals to $233 * 59.2$ μJ = 13.79mJ; And the energy consumption on receiving Message (I) equals to $233 * 28.6$ μJ = 6.66mJ. To

broadcast a message to the whole WSN, every sensor node should at least retransmit once and receive w' times the same message, when the simple flooding technique is used. Here, w' denotes the neighborhood density (i.e., the number of neighbor nodes one sensor has). Hence, the total energy consumption on message broadcast will be $W * (13.79 + 6.66 * w')$mJ. The energy consumption on message broadcast for the remaining scheme can also be calculated similarly. We summarize the results in Table 1.

Fig. 3 illustrates the broadcast energy consumption as a function of network size W, assuming $w' = 20$. Clearly, we see that the ID-based scheme offers a much lower energy consumption as compared to that of the remaining two schemes. On the other hand, we see that the Merkle hash tree based scheme outperforms of the certificate-based scheme, when N is no more than 512.

C. Energy consumption on computation

In this subsection, we evaluate the computation overhead of the proposed schemes also in terms of energy consumption. In the certificate-based scheme, the computation overhead is mainly due to the verification of two ECDSA signatures. In the Merkle hash tree based scheme, the computation overhead is due to the verification of one ECDSA signature and a number of hash operations. And in the ID-based scheme, the computation cost is due to the verification of the ID-based signature.

We now study the energy consumption of these operations. Assuming $|p| = 512$-bit, we use the following method to quantify the computation time and energy consumption of the Tate pairing used in verifying the ID-based signature. We assume that the sensor CPU is a low-power high-performance 32-bit Intel PXA255 processor at 400 MHz. The PXA255 has been widely used in many sensor products such as Sensoria WINS 3.0 and Crossbow Stargate. According to [36], the typical power consumption of PXA255 in active and idle modes are 411 and 121 mW, respectively. It was reported in [37] that it takes 752 ms to compute the Tate pairing with the similar parameters as ours on a 32-bit ST22 smartcard microprocessor at 33 MHz. Therefore, the computation of the Tate pairing on PXA255 roughly needs $33/400 \times 752 \approx 62.04$ ms, and the energy consumption E_p is approximately 25.5 mJ. Then, to verify the ID-based signature requires one exponentiation in $\mathbb{G}_2$, one hash function evaluation and two evaluations of the Tate pairing. As noted in [32], the pairing evaluation by far takes the most running time of a signature verification operation. Thus, for the sake of simplicity, we use energy consumed on pairing evaluations to approximate that of the signature verification, which ranges from E_p to $2E_p$. Furthermore, it was reported in [21] that it takes

92.4 ms to verify a ECDSA-160 signature with the similar parameters on a 32-bit ARM microprocessor at 80 MHz. Using the same estimation method, we can obtain the energy consumption roughly as 7.6 mJ. Similarly, we omit the energy cost on the hash operations and use 7.6 mJ as the energy cost regarding verification of an ECDSA-160 signature.

Fig. 4 illustrates the energy consumption on computation when the message is broadcast under different message forms. Several conclusions can be drawn from Fig. 4. First, for message broadcast, energy cost on propagation is much higher than that of computation. Second, The ID-based scheme incurs a much higher computation cost as compared to the remaining schemes. However, when we consider energy cost on both computation and communication, the ID-based scheme is still relatively efficient especially when W becomes large. Also, as an emerging technique, ID-based cryptography is under rapid development. For example, according to the recent result in [38], the Tate pairing can be evaluated up to ten times faster than previously reported implementations. Recent advances in efficient hardware implementations of the Tate pairing on smartcards, PDAs, and FPGAs are also reported in [37], [39], [40]. Therefore, as the computation cost of ID-based cryptography is expected to continue to decrease, the ID-based scheme can be envisioned to have good application potential in the near future. Third, when W is less than 500, the Merkle hash tree based scheme is the overall best choice, considering both communication and computation cost. Fourth, when W is large, it still remains to find a satisfying scheme which is both computational and communication efficient. We leave this as our future work.

VI. Concluding Remarks

In this paper, we first revisited the problem of multisender broadcast authentication in WSNs. We pointed out that symmetric-key based solutions such as μTESLA are insufficient for this problem by identifying a serious security vulnerability inherent to these schemes: the delayed authentication of the messages can lead to severe DoS attacks, due to the stringent energy and bandwidth constraints in WSNs. We then came up with several effective PKC-based schemes to address the proposed problem. Both computational and communication costs are minimized. We further analyzed both the performance and security resilience of the proposed schemes. A quantitative energy consumption analysis was given in detail. We believe that this paper can serve as the start point towards fully solving the important multisender broadcast authentication problem in WSNs.

ACKNOWLEDGEMENT

This work was supported in part by a research grant from AirSprite Technologies, Inc., Marlborough, MA, USA.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks, IEEE Communications Magazine," Vol. 40, No. 8, pp. 102-116, August 2002.

[2] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," Ad Hoc Networks 2(4): 351-367 (2004)

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," ACM WSNA02, Atlanta GA, September 2002.

[4] N. Xu, "A Survey of Sensor Network Applications," http://enl.usc.edu/ningxu/papers/survey.pdf

[5] 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 (MobiCom'01), July 2001.

[6] 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), 2003, pp. 263-276.

[7] D. Liu and P. Ning, "Multi-level mTESLA: Broadcast authentication for distributed sensor networks," ACMTransactions in Embedded Computing Systems (TECS), vol. 3, no. 4, 2004.

[8] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical Broadcast Authentication in Sensor Networks," To appear in Proceedings of The 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2005), July 2005.

[9] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," In proceedings of IEEE INFOCOM, 2003.

[10] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing Location-aware End-to-end Data Security in Wireless Sensor Networks," In Proc. of IEEE INFOCOM 2006, Spain, 2006

[11] Texas Instruments Inc., "MSP430 Family of Ultra-lowpower 16-bit RISC Processors," http://www.ti.com

[12] S. Meininger, J. Mur-Miranda, R. Amirtharajah, A. Chandrakasan, and J. Lang, "Vibration-to-electric energy conversion," IEEE Transactions on Very Large Scale Integration (VLSI) Systems vol.9, p64-76, 2001

[13] R. Amirtharajah, A. Chandrakasan, "Self-powered signal processing using vibration-based power generation," IEEE Journal of Solid-State Circuits 33 (1998) 687-695

[14] S. Seys, and B. Preneel, "Power Consumption Evaluation of Efficient Digital Signature Schemes for Low Power Devices," In IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB 2005), IEEE, pp. 79-86, 2005.

[15] 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).

[16] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. Chang Shantz. "Energy Analysis of Public-Key Cryptography on Small Wireless Devices," IEEE PerCom, March 2005.

[17] W. Du, R. Wang, and P. Ning "An Efficient Scheme for Authenticating Public Keys in Sensor Networks," In Proceedings of The 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Pages 58-67, May 25-28, 2005.

[18] Crossbow Technology Inc, Wireless sensor networks, http://www.xbow.com/. 2004.

[19] A. Kansal, D. Potter and M. Srivastava, "Performance Aware Tasking for Environmentally Powered Sensor Networks," ACM Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), 2004

[20] A Kansal and MB Srivastava, "An Environmental Energy Harvesting Framework for Sensor Networks," ACM/IEEE Int'l Symposium on Low Power Electronics and Design (ISLPED)-2003.

[21] M. Aydos, T. Yanik, and C. K. Koc. "An high-speed ECC-based wireless authentication protocol on an ARM microprocessor," In proceedings of the 16th Annual Computer Security Applications Conference, pages 401-409, New Orleans, Louisiana, 2000.

[22] R. Merkle, "Protocols for public key cryptosystems," in Proceedings of the IEEE Symposium on Research in Security and Privacy, Apr 1980.

[23] K. Barr and K. Asanovic, "Energy aware lossless data compression," in 1st Int. Conf. Mobile Systems, Applications, and Services (MobiSys'03), San Francisco, CA, May 2003.

[24] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms in wireless sensor networks," IEEE Journal on Selected Areas in Communications, Special Issue on Security in Wireless Ad Hoc Networks, Vol. 24, No. 2, pp. 247-260, Feb., 2006

[25] A. Shamir, "Identity based cryptosystems and signature schemes," in Proc. CRYPTO'84, ser. LNCS, vol. 196. Springer-Verlag, 1984, pp. 47.53.

[26] D. Boneh and M. Franklin, "Identify-based encryption from the weil pairing," in Proc. CRYPTO'01, ser. LNCS, vol. 2139. Springer-Verlag, 2001, pp. 213.229.

[27] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in Proc. CRYPTO'02, ser. LNCS, vol. 2442. Springer-Verlag, 2002, pp. 354-368.

[28] NIST, "Digital hash standard," Federal Information Processing Standards PUBlication 180-1, April 1995.

[29] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM 21(2), pp.120-126, 1978.

[30] National Institure of Standards and Technology: Proposed Federal Information Processing Standard for Digital Signature Standard (DSS). Federal Register, vol. 56, no. 169, pp. 42980–42982, 1991

[31] J. Baek, J. Newmarch, R. Safavi-Naini and W. Susilo, "A Survey of Identity-Based Cryptography," Proc. of the 10th Annual Conference for Australian Unix Users Group (AUUG 2004), pp. 95-102, 2004.

[32] F. Hess, "Efficient identity based signature schemes based on pairings," in Proc. SAC'02, St. John's, Newfoundland, Canada, Aug. 2002.

[33] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612.613, 1979.

[34] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography," Springer-Verlag New York, Inc. 2004. ISBN 0-387-95273-X.

[35] A. Wander, N. Gura, H. Eberle, Vipul Gupta, and S. Shantz, "Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks," Third IEEE International Conference on Pervasive Computing and Communication (PerCom 2005), Kauai, Mar. 2005

[36] "Intel PXA255 Processor Electrical, Mechanical, and Thermal Specification," http://www.intel.com/design/pca/applicationsprocessors /manuals/278780.htm

[37] G. Bertoni, L. Chen, P. Fragneto, K. Harrison, and G. Pelosi1, "Computing tate pairing on smartcards," White Paper, STMicroelectronics, 2005. Available: http://www.st.com/ stonline/products/families/smartcard/astibe.htm

[38] P. Barreto, B. Lynn, and M. Scott, “On the selection of pairing-friendly groups,. in Selected Areas in Cryptography, ” In proc. of SAC’2003, LNCS vol. 3006, pp. 17-25, Springer-Verlag, 2004,

[39] M. Scott, “Computing the tate pairing,” in Cryptographers’ Track at the RSA Conference (CT-RSA’05), San Francisco, CA, Feb. 2005.

[40] T. Kerins, W. Marnane, E. Popovici, and P. Barreto, “ Efficient hardware for the tate pairing calculation in characteristic three,” in Proc. Workshop on Cryptographic Hardware and Embedded Systems (CHES’05), Edinburgh, Scotland, Aug./Sep. 2005.

[41] K. Lorincz, D. Malan, T. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, S. Moulton, and M. Welsh, “Sensor Networks for Emergency Response: Challenges and Opportunities,” In IEEE Pervasive Computing, Special Issue on Pervasive Computing for First Response, Oct-Dec 2004.

[42] K. Ren, K. Zeng, W. Lou, and P. Moran, “On Broadcast Authentication in Wireless Sensor Networks,” In Proc. of WASA 2006, LNCS Vol. 4138, p502-514, Xian, China, Aug., 2006

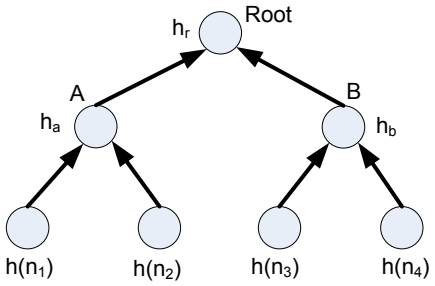

Fig. 1. An example of Merkle hash tree

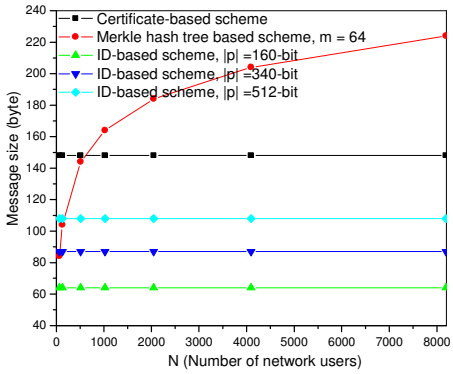

Fig. 2. Message sizes with regard to number of network users

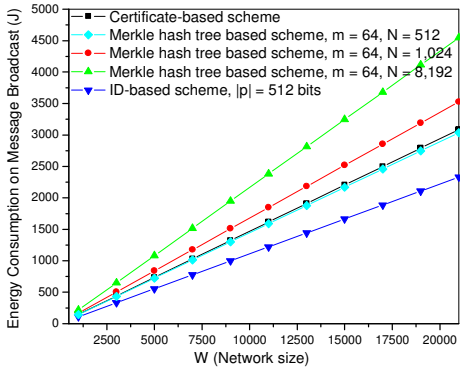

Fig. 3. Energy consumption on message broadcast with regard to network size

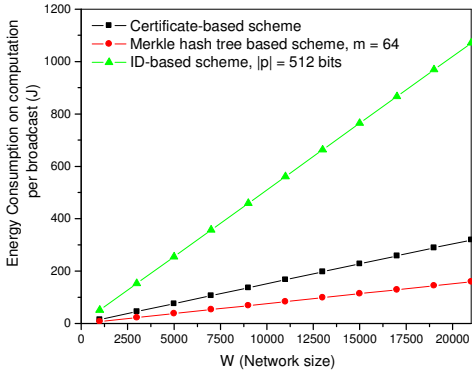

Fig. 4. Energy consumption on computation with regard to network size

Energy cost (mJ)	The certificate-based scheme	The Merkle hash tree based scheme	The ID-based scheme ($\|p\| = 512$ bits)
$N = 512$	$W * (13.79 + 6.66 * w')$	$W * (13.56 + 6.55 * w')$	$W * (10.42 + 5.03 * w')$
$N = 1,024$	$W * (13.79 + 6.66 * w')$	$W * (15.75 + 7.61 * w')$	$W * (10.42 + 5.03 * w')$
$N = 8,192$	$W * (13.79 + 6.66 * w')$	$W * (20.31 + 9.81 * w')$	$W * (10.42 + 5.03 * w')$

TABLE I

ENERGY CONSUMPTION ON MESSAGE BROADCAST (PER NODE)