Multi-User Broadcast Authentication in Wireless Sensor Networks

Kui Ren, *Member, IEEE*, Shucheng Yu, *Student Member, IEEE*,
Wenjing Lou, *Senior Member, IEEE*, and Yanchao Zhang, *Member, IEEE*

***Abstract*—Broadcast authentication is a critical security service in wireless sensor networks (WSNs), as it allows mobile users of WSNs to broadcast messages to multiple sensor nodes in a secure way. Although symmetric-key-based solutions such as μTESLA and multilevel μTESLA have been proposed, they all suffer from severe energy-depletion attacks resulting from the nature of delayed message authentication. This paper presents several efficient public-key-based schemes to achieve immediate broadcast authentication and thus avoid the security vulnerability that is intrinsic to μTESLA-like schemes. Our schemes are built upon the unique integration of several cryptographic techniques, including the Bloom filter, the partial message-recovery signature scheme, and the Merkle hash tree. We prove the effectiveness and efficiency of the proposed schemes by a comprehensive quantitative analysis of their energy consumption in both computation and communication.**

***Index Terms*—Broadcast authentication, multiuser, security, wireless sensor networks (WSNs).**

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 the civilian and military domains [2]–[8]. In these applications, WSNs are expected to process, store, and provide the sensed data to the network users upon their demands [9]. As the most common communication paradigm, the network users are expected to issue the queries to the network to obtain the information of their interest. Furthermore, in wireless sensor and actuator networks [3], network users may need to issue their commands to the network (probably based on the information that they received from the network). In both cases, there could be a large number of users in the WSNs, which might be either mobile or static, and the users may use their mobile clients to query or command the sensor nodes from anywhere in the WSN. 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 security purposes.

Broadcast authentication in WSNs was first addressed by μTESLA [10]. In μTESLA, users of WSNs are 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, the 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 I_1. During time interval I_2, the sink broadcasts K_1, and the sensor nodes verify $h(K_1) = K_0$. The authenticity of messages received during time interval I_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 was later enhanced in [11] to overcome the length limit of the hash chain. Most recently, μTESLA was also extended in [12] to support a multiuser scenario, but the scheme assumes that each sensor node only interacts with a very limited number of users.

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 [13], where messages could be forged due to the propagation delay of the disclosed keys. However, here, we point out a much more serious vulnerability of μTESLA-like schemes when they are applied in multihop WSNs. Since the sensor nodes buffer all the messages received within one time interval, an adversary can hence arbitrarily flood the whole network. All the adversary has to do is to claim that the flooding messages belong to the current time interval, which should be buffered for authentication until the next time interval. Since wireless transmission is very expensive in WSNs and WSNs are extremely energy constrained, the ability to arbitrarily flood the network could cause devastating Denial of Service (DoS) attacks. Moreover, these types of energy-depletion DoS attacks become more devastating in a multiuser scenario as the adversary can now

Manuscript received May 14, 2008; revised November 19, 2008 and February 23, 2009. First published April 3, 2009; current version published October 2, 2009. This work was supported in part by the U.S. National Science Foundation under Grant CNS-0831963, Grant CNS-0626601, Grant CNS-0716306, Grant CNS-0831628, and Grant CNS-0716302. An earlier version of this paper appeared in *Proc. SECON*, pp. 223–232, Jun. 2007. The review of this paper was coordinated by Prof. J. Li.

K. Ren is with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616 USA (e-mail: kren@ece.iit.edu).

S. Yu and W. Lou are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609-2280 USA (e-mail: yscheng@ece.wpi.edu;wjlou@ece.wpi.edu).

Y. Zhang is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07103 USA (e-mail: yczhang@njit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2019663

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

have more targets and, hence, more chances to generate bogus messages without being detected. Obviously, all these attacks are due to the delayed authentication of the broadcast messages. In [13], the TIK protocol 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 to WSNs, as pointed out by the authors. Therefore, multiuser broadcast authentication still remains a wide-open problem in WSNs.

When μTESLA was proposed, sensor nodes were assumed to be extremely resource constrained, particularly with respect to computation capability, bandwidth availability, and energy supply [10]. Therefore, public key cryptography (PKC) was thought to be too computationally expensive for WSNs, although it could provide much simpler solutions with much stronger security resilience. At the same time, the computationally efficient one-time signature schemes are also considered unsuitable for WSNs, as they usually involve intense communications [10]. However, recent studies [14]–[16] showed that, contrary to widely held beliefs, PKC with even software implementations is only very viable on sensor nodes. For example [14], elliptic curve cryptography (ECC) signature verification takes 1.61 s, with 160-bit keys on an ATmega128 8-MHz processor, which is the processor used in the current Crossbow motes platform [17]. Furthermore, the computational cost is expected to decrease faster than the cost to transmit and receive. For example, ultralow-power microcontrollers such as the 16-bit MSP430 from Texas Instruments Incorporated [18] can execute the same number of instructions at less than half the power required by the 8-bit ATmega128L. The benefits of transmitting shorter ECC keys and, hence, shorter messages/signatures will, in turn, be more significant. Moreover, next-generation sensor nodes are expected to combine ultralow-power circuitry with so-called power scavengers such as Heliomote [19], which allow continuous energy supply to the nodes. At least 8–20 μW of power can be generated using microelectromechanical-systems-based power scavengers [20]. Other solar-based systems are even able to deliver power up to 100 mW for the MICA Motes [19], [21]. These results indicate that, with the advance of fast-growing technology, PKC is no longer impractical for WSNs, although it is still expensive for current-generation sensor nodes, and its wide acceptance is expected in the near future [15].

Having this observation and knowing that symmetric-key-based solutions such as μTESLA are insufficient for broadcast authentication in WSNs, we resort to PKC for more effective solutions. In this paper, we address the multiuser broadcast authentication problem in WSNs by designing PKC-based solutions with minimized computational and communication costs.

Objectives of this Paper: We focus on providing multiuser broadcast authentication in WSNs, where the broadcast messages are initiated by a number of network users. Please note that the network users in this paper refer to personnel or devices that use the WSN; they are not sensor nodes. On the one hand, we aim to achieve immediate message authentication and resist 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.

Overview of this Paper: In this paper, we propose four different public-key-based approaches and provide in-depth analysis of their advantages and disadvantages. In all the four approaches, the users are always authenticated through their public keys. We first propose a straightforward certificate-based approach and point out its high energy inefficiency with respect to both communication and computation costs. We then propose a direct-storage-based scheme, which has high efficiency but suffers from the scalability problem. A Bloom-filter-based scheme is further proposed to improve the memory efficiency over the direct-storage-based scheme. Further techniques are also developed to increase the security strength of the proposed scheme. Finally, we propose a hybrid scheme to support a larger number of network users by employing the Merkle hash tree technique. We give an in-depth quantitative analysis of the proposed schemes and demonstrate their effectiveness and efficiency in WSNs in terms of energy consumption.

Contributions of this Paper: This paper makes the following contributions: 1) We identify the problem of multiuser broadcast authentication in WSNs and point out a serious security vulnerability that is inherent to the symmetric-key-based μTESLA-like schemes. 2) We come up with several PKC-based schemes to address the proposed problem with minimized computational and communication costs. We achieve our goal by integrating several cryptographic building blocks, such as the Bloom filter, the partial message-recovery signature scheme, and the Merkle hash tree, in an innovative manner. 3) We analyze both the performance and security resilience of the proposed schemes. A quantitative energy consumption analysis is given in detail and demonstrates the effectiveness and efficiency of the proposed schemes.

Organization of this Paper: The remaining part of this paper is given as follows: In Section II, we introduce the cryptographic mechanisms to be used. Section III presents the system assumption, adversary model, and security objectives. In Section IV, we introduce two basic schemes. We next propose two advanced schemes and detail the underlying design logic in Section V. Section VI discusses some further enhancements of the proposed schemes. In Section VII, we analyze the performance and security strength of the proposed schemes. Finally, Section VIII presents the conclusion.

II. Preliminaries

A. Digital Signature

A digital signature algorithm is a cryptographic tool for generating nonrepudiation evidence, authenticating the integrity and the origin of a signed message. In a digital signature algorithm, a signer keeps a private key secret and publishes the corresponding public key. The private key is used by the signer to generate digital signatures on messages, and the public key is used by anyone to verify signatures on messages. The digital signature algorithms mostly used are RSA [22] and DSA [23]. ECDSA is referred to as the elliptic curve digital signature algorithm [24]. While RSA with 1024-bit keys (RSA-1024) provides the currently accepted security level, it is equivalent in security strength to ECC with 160-bit keys (ECC-160). Hence,

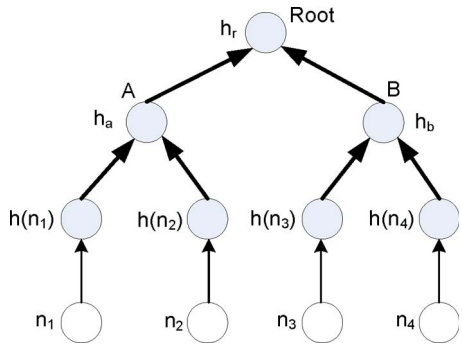


Fig. 1. Example of the Merkle hash tree.

for the same level of security strength, ECDSA uses a much shorter key size and, hence, has a short signature size (320 bit).

B. Bloom Filter and Counting Bloom Filter

A Bloom filter is a simple space-efficient randomized data structure for representing a set to support membership queries [25]. A Bloom filter for representing a set $S = s_1, s_2, \ldots, s_n$ of n elements is described by a vector $\mathcal{V}$ of m bits, which are initially all set to 0. A Bloom filter uses k independent hash functions $h_1, \ldots, h_k$ with range $0, \ldots, m-1$, which map each item in the universe to a random number uniform over $[0, \ldots, m-1]$. For each element $s \in S$, bits $h_i(s)$ are set to 1 for $1 \leq i \leq k$. Note that a bit of $\mathcal{V}$ can be set to 1 multiple times. To check if an item x is in S, we check whether all bits $h_i(x)$ are set to 1. If not, x is not a member of S for certain, i.e., there is no false negative error. If yes, x is assumed to be in S. A Bloom filter may yield a false positive. It may suggest that an element x is in S, even though it is not. The probability of a false positive for an element that is not in the set can be calculated as follows: After all the elements of S are hashed into the Bloom filter, the probability that a specific bit is still 0 is $(1-1/m)^{kN} \approx e^{-kN/m}$. The probability of a false positive f is then $f = (1-(1-1/m)^{kN})^k \approx (1-e^{-kN/m})^k$. We let $f = (1-p)^k$. From now on, for convenience, we use the asymptotic approximations p and f to represent the probability that a bit in the Bloom filter is 0 and the probability of a false positive. Let $p = e^{-kN/m}$, respectively.

The counting Bloom filter is a variation of the Bloom filter, which allows member deletion. In the counting Bloom filter, each entry in the Bloom filter is not a single bit but a small counter that tracks the number of elements that have hashed to that location [26]. When an element is deleted, the corresponding counters are decremented. To avoid overflow, counters must be chosen to be large enough [26].

C. Merkle Hash Tree

A Merkle Tree is a construction introduced by Merkle in 1979 to build secure authentication schemes from hash functions [27]. It is a tree of hashes where the leaves in the tree are hashes of the authentic data values $n_1, n_2, \ldots, n_w$. Nodes further up in the tree are the hashes of their respective children. For instance, assuming that $w = 4$ in Fig. 1, the values of the four leaf nodes are the hashes of the data values $h(n_i), i = 1, 2, 3,$ and 4, respectively, under a one-way hash function $h()$ (e.g., SHA-1 [28]). The value of an internal node A is $h_a = h(h(n_1)\|h(n_2))$, and the value of the root node is $h_r = h(h_a\|h_b)$. h_r 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, where N is the number of leaf nodes). For example, a receiver with authentic h_r requests for n_3 and requires the authentication of the received n_3. The source sends the $\texttt{AAI} : \langle h_a, h(n_4)\rangle$ to the receiver. The receiver can then verify n_3 by first computing $h(n_3)$, $h_b = h(h(n_3)\|h(n_4))$, 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. Only if this check is positive does the user accept n_3. The Merkle hash tree can prevent an adversary from sending bogus data to deceive the client. In the earlier example, an adversary impersonating cannot send a bogus n_3 to the client without being detected. This is because he cannot find h_a and $h(n_4)$ such that $h(h_a\|h(h(n_3)\|h(n_4))) = h_r$, as $h()$ is a one-way function.

III. System Model, Adversary Model, and Design Goals

System Model

In this paper, we consider a large spatially distributed WSN consisting of a fixed sink(s) and a large number of sensor nodes. The sensor nodes are usually resource constrained with respect to memory space, computation capability, bandwidth, and power supply. The WSN is aimed to offer information services to many network users that roam in the network, in addition to the fixed sink(s) [9]. The *network users* may 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 consist of a number of doctors, nurses, medical equipment (acting as actuators), and so on, in the case of CodeBlue [29], where the WSN is used for emergency medical response. These network users broadcast queries/commands through sensor nodes in the vicinity and expect the replies that reflect the latest network information. The network users can also directly communicate with the sink or the backend server without going through the WSN if necessary. 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 dynamically be revoked due to either membership changes or compromise, and the revocation pattern is not restricted. We also assume that the WSN is loosely synchronized.

Adversary Model

In this paper, 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, DoS attacks such as bogus message flooding, aiming at exhausting constrained network resources, is another important focus of this paper. We assume that the adversary is able to compromise both the network users and the sensor nodes. The

adversary could hence exploit the compromised users/nodes for such attacks. However, we assume that the adversary cannot compromise an unlimited number of sensor nodes.

Design Goals

Our security goal is straightforward: All messages broadcasted by the network users of the WSN should be authenticated so that the bogus messages inserted by the illegitimate users and/or compromised sensor nodes can efficiently be rejected/filtered. We also focus on minimizing the overheads of the security design. In particular, energy efficiency (with respect to both communication and computation) and storage overhead are given priority to cope with the resource-constrained nature of WSNs.

IV. Basic Schemes

We explore the PKC domain for the possible solutions to multiuser broadcast authentication in WSNs. The PKC-based solutions realize immediate message authentication and can thus overcome the delayed message authentication problem present in μTESLA-like schemes.

A. Certificate-Based Authentication Scheme (CAS)

CAS works as follows: Each user (not a sensor) of the WSN is equipped with a public/private key pair $(\mathtt{PK}/\mathtt{SK})$ and signs every message he broadcasts with his $\mathtt{SK}$ using a digital signature scheme, such as ECDSA [24]. Note that, in all our designs, we do not require sensors to have public/private key pairs for themselves. To prove the user's ownership over his public key, the sink[2] is also equipped with a public/private key pair and serves as the certification authority. The sink issues each user a public key certificate, which, in its simplest form, consists of the following contents: $\mathtt{Cert}_{U_{\mathrm{ID}}} = U_{\mathrm{ID}}, \mathtt{PK}_{U_{\mathrm{ID}}}, \mathtt{ExpT}, \mathrm{SIG}_{\mathtt{SK}_{\mathrm{Sink}}}\{h(U_{\mathrm{ID}}\|\mathtt{ExpT}\|\mathtt{PK}_{U_{\mathrm{ID}}})\}$, where U_{ID} denotes the user's identification (ID), $\mathtt{PK}_{U_{\mathrm{ID}}}$ denotes its public key, $\mathtt{ExpT}$ denotes the certificate expiration time, and $\mathrm{SIG}_{\mathtt{SK}_{\mathrm{Sink}}}\{h(U_{\mathrm{ID}}\|\mathtt{ExpT}\|\mathtt{PK}_{U_{\mathrm{ID}}})\}$ is a signature over $h(U_{\mathrm{ID}}\|\mathtt{ExpT}\|\mathtt{PK}_{U_{\mathrm{ID}}})$ with $\mathtt{SK}_{\mathrm{Sink}}$. Hence, a broadcast message is now of the form

$$\left\langle M, tt, \mathrm{SIG}_{\mathtt{SK}_{U_{\mathrm{ID}}}}\left\{h\left(U_{\mathrm{ID}}\|tt\|M\right)\right\}, \mathtt{Cert}_{U_{\mathrm{ID}}}\right\rangle. \quad (1)$$

Here, M denotes the broadcast message, and tt denotes the current time. For the purpose of message authentication, sensor nodes are preloaded with $\mathtt{PK}_{\mathrm{Sink}}$ before network deployment, and message verification involves two steps: 1) user certificate verification and 2) message signature verification.

CAS suffers from two main drawbacks: First, it is not efficient in communication, as the certificate has to be transmitted along with the message across every hop as the message propagates in the WSN. A large per-message overhead will result in more energy consumption on every single sensor node. In CAS, the per-message overhead is as high as $|tt| + |\mathrm{SIG}_{\mathtt{SK}_{U_{\mathrm{ID}}}}\{h(U_{\mathrm{ID}}\|M)\}| + |\mathtt{Cert}_{U_{\mathrm{ID}}}| = 128$ B. As in [14], the user certificate is at least 86 B, when ECDSA-160 [24] is used. Here, we assume that tt and U_{ID} are both 2 B, in which case, the scheme supports up to 65 535 network users. Moreover, $|\mathrm{SIG}_{\mathtt{SK}_{U_{\mathrm{ID}}}}\{h(U_{\mathrm{ID}}\|M)\}| = 40$ B, when ECDSA-160 [24] is assumed. Second, to authenticate each message, it always takes two expensive signature verification operations. This is because the certificate should always be authenticated in the first place.

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

B. Direct-Storage-Based Authentication Scheme (DAS)

One way to reduce the per-message overhead and the computational cost is to eliminate the existence of the certificate. A straightforward approach is then to let sensor nodes simply store all the current users' ID information and their corresponding public keys. In this way, a broadcast message now only contains the following contents:

$$\left\langle M, tt, \mathrm{SIG}_{\mathtt{SK}_{U_{\mathrm{ID}}}}\left\{h\left(U_{\mathrm{ID}}\|tt\|M\right)\right\}, U_{\mathrm{ID}}, \mathtt{PK}_{U_{\mathrm{ID}_1}}\right\rangle. \quad (2)$$

Verifying the authenticity of a user public key is reduced to finding out whether the attached user/public key pair is contained in the local memory. Upon user revocation, the sink simply sends out ID information of the revoked user, and every sensor node deletes the corresponding user/public key pair in its memory.

The drawbacks of DAS are obvious. Given a storage limit of 5 kB, only 232 users can be supported at most; even with a memory space of 19.5 kB, DAS can only support up to 1000 users. At the same time, CAS can support up to 2560 users, given the same storage limit of 5 kB. The reason is that, in CAS, only the ID information of the revoked users is stored by the sensor nodes. Therefore, DAS is neither memory efficient nor scalable. However, the advantage of DAS over CAS is also significant. It successfully reduces the per-message overhead down to $|tt| + |\mathrm{SIG}_{\mathtt{SK}_{U_{\mathrm{ID}}}}\{h(U_{\mathrm{ID}}\|M)\}| + |U_{\mathrm{ID}}| + |\mathtt{PK}_{U_{\mathrm{ID}}}| = 64$ B. The preceding analysis clearly shows that more advanced schemes are needed other than DAS and CAS. In addition, the direction to seek is to improve storage efficiency while retaining or further reducing the per-message overhead.

V. Advanced Schemes

In this section, advanced schemes are proposed to simultaneously achieve both storage efficiency and communication efficiency. The proposed schemes significantly outperform the previous basic schemes through a novel integration of several cryptographic techniques.

A. Bloom-Filter-Based Authentication Scheme (BAS)

1) Scheme Overview: For the purpose of memory efficiency, BAS does not directly preload each sensor node with the original (ID, public key) pairs of the network users. Instead, each sensor node just stores these pairs' hash mappings employing the Bloom filter. When the sensor node receives a broadcast message, it verifies the authenticity of the user public key by checking if its corresponding hash mapping is contained in the local memory. The negative answer results in the dropping of the message. As the hash mappings of the (ID, public key) pairs are extremely concise, they consume much less memory, compared with that of the direct storage scheme. One drawback

of using hash mapping is the possibility of a false positive, i.e., accepting an (ID, public key) pair that is actually not valid. This is caused by the fact that hash functions are usually subjective. BAS addresses this issue by using a Bloom filter with which the probability of a false positive can be controlled by selecting appropriate system parameters. To support user revocation/addition, the sink possesses an extra data structure, i.e., a counting Bloom filter, which uses more bits to represent the (ID, public key) pairs, compared with regular Bloom filters, and thus has less probability of statistical errors. Our message signature and authentication algorithms take a similar idea as a variant of ECDSA [30] and are provably secure.

2) Scheme in Detail: BAS can be described by four phases.

System preparation: The sink generates the public keys for all network users and constructs the set

$$\mathbf{S} = \{\langle U_{\mathrm{ID}_1}, \mathtt{PK}_{U_{\mathrm{ID}_1}}\rangle, \langle U_{\mathrm{ID}_2}, \mathtt{PK}_{U_{\mathrm{ID}_2}}\rangle, \ldots\}$$

where $\#\{\mathbf{S}\} = N$, and $\#\{\}$ denotes the cardinality of the set. Using the Bloom filter, the sink can apply k system-wide hash functions (cf. Section II-B) to map the elements of $\mathbf{S}$ (each with $L+2$ bytes, i.e., $|U_{\mathrm{ID}}| = 2$ B, and $|\mathtt{PK}_{U_{\mathrm{ID}}}| = L$ bytes) to an m-bit vector $\mathcal{V}$, with $\mathcal{V} = v_0 v_1 \cdots v_{m-1}$, where we have $m < N(L+2)$ to reduce the filter size and $m > kN$ to retain a small probability of a false positive. These k hash functions are known by every node and the sink. For each v_i, $i \in [0, m-1]$, we have

$$v_i = \begin{cases} 1, & \text{if } \exists\, l \in [1,k], j \in [1,N] \\ & \text{s.t. } h_l(U_{\mathrm{ID}_j} \| \mathtt{PK}_{U_{\mathrm{ID}_j}}) = i \\ 0, & \text{otherwise.} \end{cases}$$

Additionally, the sink constructs a counting Bloom filter $\overline{\mathcal{V}}$ of $m * c$ bits, with $\overline{\mathcal{V}} = \overline{v}_0 \overline{v}_1 \cdots \overline{v}_{m-1}$, where each $\overline{v}_i$, $i \in [0, m-1]$ is a c-bit counter, i.e., $|\overline{v}_i| = c$ bits. The value of $\overline{v}_i$ is determined as follows:

$$\overline{v}_i = \#\big\{(ID_j, \mathtt{PK}_{U_{\mathrm{ID}_j}}) | h_l(U_{\mathrm{ID}_j} \| \mathtt{PK}_{U_{\mathrm{ID}_j}}) = i \quad \text{for } \exists\, l \in [1,k], j \in [1,N]\big\}.$$

In addition, $c = \lceil \log_2(\max(\overline{v}_i, i \in [0, m-1])) \rceil$ bits, which is usually of 4 bits for most applications [26]. The preceding operations are shown in Fig. 2. The sink finally preloads each sensor node with $\mathcal{V}$ (not including $\overline{\mathcal{V}}$), as well as the sink's public key and the common domain parameters of the ECDSA signature scheme.

Message signing and authentication: Let $\mathtt{PK}_{U_{\mathrm{ID}}} = sG$ be the public key of user U_{ID}, where s is the private key of the signer, and G is the generator of a subgroup of an elliptic curve group of order r. Let $S_K(\cdot)$ be a symmetric key cipher such as the Advanced Encryption Standard. To broadcast a message M ($|M| \geq 10$ B), U_{ID} takes seven steps, following [30], which is a variant of ECDSA with the partial message recovery property.

1) Concatenate $\langle M \| tt \| U_{\mathrm{ID}} \rangle$, and break it into two parts, i.e., M_1 and M_2, where $|M_1| \leq 10$ B.
2) Generate a random key pair $\{u, V\}$, where $u \in [1, r-1]$, $V = uG = (x_1, y_1)$, and $(x_1 \mod r) \neq 0$.

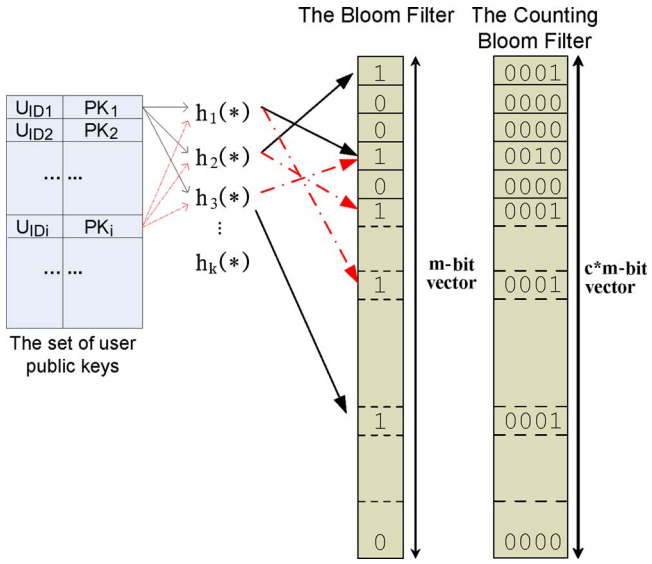


Fig. 2. Example of the Bloom filter and counting Bloom filter.

3) Encode and hash V into an integer I [30].
4) Form F_1 from M_1 by adding the proper redundancy [31].
5) Compute $\mathtt{C} = (I + F_{1q}) \mod r$, and make sure that $\mathtt{C} \neq 0$; otherwise, repeat the preceding steps.
6) Compute $F_2 = h(M_2)$ and $\mathtt{D} = u^{-1}(F_2 + s\mathtt{C}) \mod r$.
7) Repeat all the preceding steps if $\mathtt{D} = 0$; output the signature as $\langle \mathtt{C}, \mathtt{D} \rangle$ otherwise.

Then, U_{ID} broadcasts

$$\langle M_2, \mathtt{C}, \mathtt{D}, \mathtt{PK}_{U_{\mathrm{ID}}} \rangle \tag{3}$$

where tt and U_{ID} are parts of M_2. In addition, this is the simplest known message format that can be achieved using PKC.[3] Now, upon receiving a broadcast message (not from the sink), a sensor node checks the authenticity of the message in two steps: First, it checks the authenticity of the corresponding public key by verifying its membership in $\mathbf{S}$. To do so, the sensor node checks whether $\mathcal{V}[h_l(U_{\mathrm{ID}} \| \mathtt{PK}_{U_{\mathrm{ID}}})] \stackrel{?}{=} 1, l \in [1,k]$, and a negative result will lead to the discarding of the message. We note that, here, a false positive may happen due to the probabilistic nature of the Bloom filter but only with a very small (negligible) probability when appropriate parameters are chosen, as we will analyze later. Second, it verifies the attached signature in seven steps.

1) Discard the message if $\mathtt{C} \notin [1, r-1]$ or $\mathtt{D} \notin [1, r-1]$.
2) Compute $F_2 = h(M_2)$, $\mathtt{H} = \mathtt{D}^{-1} \mod r$, and $\mathtt{H}_1 = F_2 \mathtt{H} \mod r$.
3) Compute $\mathtt{H}_2 = \mathtt{CH} \mod r$ and $P = \mathtt{H}_1 G + \mathtt{H}_2 \mathtt{PK}_{U_{\mathrm{ID}}}$.
4) Discard the message if $P = \mathcal{O}$.

[3]The claim is true only when ID-based cryptography [32] is excluded from consideration, in which case, the user's ID is also his public key. Furthermore, the shortest signature size possibly obtained from pairing is about 22 B [33], which is shorter than the 40 B obtained from ECDSA. However, to apply a pairing-based scheme (i.e., an ID-based signature or short signature) on sensor nodes, the known reachable signature size has to be 84 B, even when a 32-bit microprocessor can be used [34]. In addition, the energy cost is also multiple times higher than that of an ECDSA-160 signature.

5) Encode and hash P into an integer I [30], and compute $F_1 = \mathtt{C} - I \mod r$.
6) Discard the message if the redundancy of F_1 is incorrect.
7) Otherwise, accept M_1 (obtained from F_1) and the signature, and reconstruct $M\|tt\|U_{\mathrm{ID}} = M_1\|M_2$.

Our message signing and authentication algorithms are provably secure under adaptive chosen message attacks. See [30] for the formal security proof.

User revocation/addition: To revoke a user, for example, U_{ID_j}, the sink follows five steps.

1) First, it hashes $h_l(U_{\mathrm{ID}_j}\|\mathtt{PK}_{U_{\mathrm{ID}_j}}) = i$ and decreases $\overline{v}_i$ by 1. It repeats this operation for all $h_l, l \in [1, k]$.
2) From the updated counting Bloom filter $\overline{\mathcal{V}}$, the sink obtains the corresponding updated Bloom filter $\mathcal{V}'$, with $\mathcal{V}' = v'_0 v'_1 \ldots v'_{m-1}$. Here, $v'_i = 1$ only when $\overline{v}_i \geq 1$, and $v'_i = 0$ otherwise.
3) The sink further calculates $\mathcal{V}_\Delta = \mathcal{V}' \oplus \mathcal{V}$ and deletes $\mathcal{V}$ afterward. Here, $\oplus$ denotes bitwise exclusive OR operation. Obviously, $\mathcal{V}_\Delta$ is an m-bit vector with at most k bits set to 1. Hence, $\mathcal{V}_\Delta$ can simply be represented by enumerating its 1-valued bits, requiring $\overline{k}\lceil \log_2 m \rceil$ bits for indexing ($\overline{k} \leq k$). This representation is efficient for a small $\overline{k}$, as will be analyzed in Section VI-B.
4) The sink finally broadcasts $\mathcal{V}_\Delta$ after signing it. The message format follows (3) but with the sink's public key omitted as every sensor already has it.
5) Upon receiving and successfully authenticating the broadcast message, every sensor node updates its own Bloom filter accordingly, i.e., if $v_{\Delta,i} = 1$, then $v_i = 0$, $i \in [0, m-1]$.

BAS also supports simultaneous multiuser revocation. Suppose that N_{rev} users are simultaneously revoked. The sink follows the same manner in constructing $\mathcal{V}_\Delta$, with $\overline{k}$ bits set to 1. Now, we have $\overline{k} \leq kN_{\mathrm{rev}}$. Furthermore, the compressed message for representing $\mathcal{V}_\Delta$ could now theoretically achieve $mH(p)$ bits, where $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$ is the entropy function, and $p = (1-1/m)^{\overline{k}}$ is the probability of each bit being 0 in $\mathcal{V}_\Delta$. As pointed out in [25], using the arithmetic coding technique can efficiently approach this lower bound.

BAS supports dynamic user addition in two ways: First, it enables a later binding of network users and their (ID, public key) pairs. In this approach, the sink may generate more (ID, public key) pairs than needed during system preparation. When a new network user joins the WSN, it will be assigned an unused ID and public key pair by the sink. Second, BAS could add new network users after the revocation of old members. This approach, however, could only add the same number of new users as that of the revoked. This requirement ensures that the probability of a false positive never increases in BAS. To do so, the sink updates its counting Bloom filter by hashing the new user's information into the current Bloom filter. The sink then obtains a $\mathcal{V}_\Delta$ in the same way as in the revocation case and broadcasts it after compression. During this time, if $v_{\Delta,i} = 1$, the sensor nodes will set $v_i = 1$, $i \in [0, m-1]$ to update their current Bloom filters.

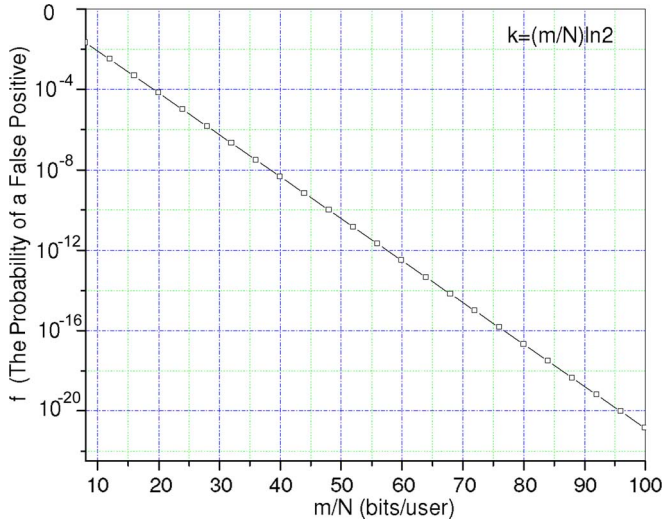


Fig. 3. Minimum probability of a false positive regarding m/N.

B. Minimize the Probability of a False Positive

Since the Bloom filter provides probabilistic membership verification only, it is important to make sure that the probability of a false positive is as small as possible.

Theorem 1: Given the number of network users N and the storage space m bits for a single Bloom filter, the minimum probability of a false positive f that can be achieved is 2^{-k}, with $k = m/N \ln 2$, i.e.,

$$f = (0.6185)^{\frac{m}{N}}.$$

Proof: Since $f = (1-(1-1/m)^{kN})^k \approx (1-e^{-kN/m})^k$, we then have $f = e^{k\ln(1-e^{-kN/m})}$. Let $g = k\ln(1-e^{-kN/m})$. Hence, minimizing f is equivalent to minimizing g with respect to k. We find

$$\frac{dg}{dk} = \ln(1-e^{-kN/m}) + \frac{kN}{m}\frac{e^{-kN/m}}{1-e^{-kN/m}}.$$

It is easy to check that the derivative is 0 when $k = m/N \ln 2$. In addition, it is not hard to show that this is a global minimum [25]. Note that, in practice, k must be an integer. ■

Fig. 3 shows the probability of a false positive f as a function of m/N, i.e., bits per element. We see that f sharply decreases as m/N increases. When m/N increases from 8 to 96 bits, f decreases from $2.1 * 10^{-2}$ to $9.3 * 10^{-21}$. Obviously, f determines the security strength of our design. For example, when $m/N = 92$ bits, the adversary has to generate about $2^{63.8}$ public/private key pairs on average before finding a valid pair to pass the Bloom filter. This is almost computationally infeasible, at least within the lifetime of the WSN (which is usually, at most, several years). However, when $m/N = 64$ bits, the adversary is now expected to generate about $2^{44.4}$ public/private key pairs before finding a valid pair. The analysis given here shows the time and cost of the attack. To generate a public/private key pair in ECDSA-160, a point multiplication operation has to be performed, for which the fastest known

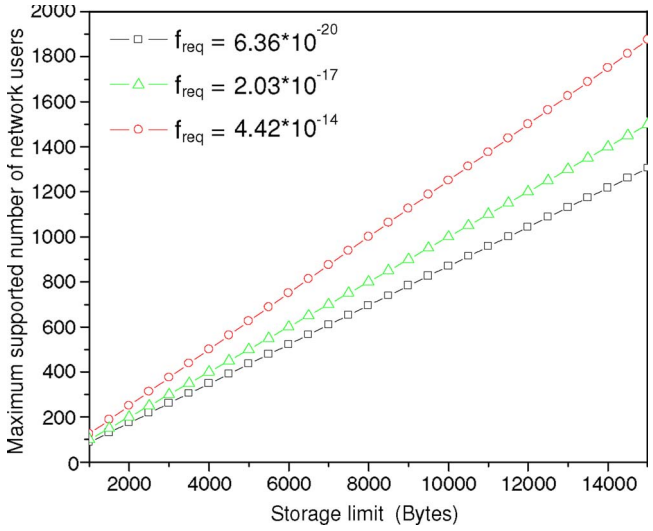


Fig. 4. Maximum supported number of network users with respect to storage limit.

implementation speed is 0.21 ms through a specialized field-programmable gate array (FPGA) design [35]. Suppose that the adversary could afford 100 000 such FPGAs, which would cost no less than $1 million. Then, by simultaneously executing 100 000 FPGAs, generating one valid key pair still roughly takes 13.2 h. With the preceding analysis, we suggest careful selection of the value of f according to the security requirements of the different types of applications. Given a highly security sensitive military application, we suggest that f be no larger than $6.36 * 10^{-20}$, i.e., $m/N \geq 92$ bits. On the other hand, when the targeted applications are less security sensitive as in the civilian scenario, we can tolerate a larger f. This is because the adversary is now generally much less resourceful, compared with the former case.

C. Maximum Number of Network Users Supported

It is important to know how many network users can be supported in BAS so that the WSN can be well planned. Theorem 2 provides the answer.

Theorem 2: Given the storage space m bits for a single Bloom filter and the required probability of a false positive $f_{\text{req}}(f_{\text{req}} \in (0,1))$, the maximum number of network users that can be supported is $-m(\ln 2)^2/\ln f_{\text{req}}$, i.e.,

$$N \leq \frac{-0.4805m}{\ln f_{\text{req}}}.$$

Proof: Since the minimal probability of a false positive $f = 2^{-k}$ is achieved with $k = m/N \ln 2$, we have $f_{\text{req}} = 2^{-m/N \ln 2}$. Then, we can easily get $N = -m(\ln 2)^2/\ln f_{\text{req}}$ in this case, and this is the maximum number of users that can be supported, given f_{req} and m. ■

Fig. 4 shows the maximum supported number of network users as a function of the storage limit. Fig. 4 shows that BAS supports up to 1250 users when $f_{\text{req}} = 4.42 * 10^{-14}$, 1000 users when $f_{\text{req}} = 2.03 * 10^{-17}$, and 869 users when $f_{\text{req}} = 6.36 * 10^{-20}$ for a storage space of 9.8 kB. Obviously, BAS also allows a tradeoff between the maximum supported number of network users and the probability of a false positive, given a fixed storage limit.

D. Supporting More Users Using the Merkle Hash Tree: Hybrid Authentication Scheme (HAS)

Through the preceding analysis, we know that the maximum supported number of network users is usually limited, given the storage limit and the probability of a false positive. For example, if $f_{\text{req}} = 6.36 * 10^{-20}$ and the storage limit is 4.9 kB, the maximum number of users supported by BAS is 434. Therefore, an additional mechanism has to be employed to support more users when necessary. HAS achieves this goal by employing the Merkle hash tree technique, which trades the message length for the storage space. That is, by increasing the per-message overhead, HAS can support more network users. Specifically, HAS works as follows:

The sink first calculates the maximum number of users supported in case of BAS according to the given storage limit and the desired probability of a false positive. It then collects all the public keys of the current network users and constructs a Merkel hash tree. In fact, the sink constructs 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 his public key, i.e., $h(U_{\text{ID}}, \texttt{PK}_{U_{\text{ID}}})$. The values of the internal nodes are determined by the method introduced in Section II-C. The sink further prunes the Merkle hash tree into a set of equal-sized smaller trees. We denote the value of the root node of a small hash tree as $h_r^i, i = 1, \ldots, |\mathbf{S}|$, where $|\mathbf{S}|$ is equal to the maximum number of supported users the sink calculates in BAS.

Next, the sink constructs a Bloom filter $\mathcal{V}$, following the same way as described in the last section. The difference is that now, the member set $\mathbf{S} = \{h_r^1, h_r^2, \ldots, h_r^{|\mathbf{S}|}\}$. Then, the sink preloads each sensor node with $\mathcal{V}$. At the same time, each user should obtain its $\texttt{AAI}$ according to his corresponding leaf node's location in the smaller Merkle hash tree. Let T denote all the nodes along the path from a leaf node to the root (not including the root) and A be the set of nodes corresponding to the siblings of the nodes in T. Then, $\texttt{AAI}$ further corresponds to the values associated with the nodes in A. Obviously, $\texttt{AAI}$ is of size $(\overline{L} * \log_2 N/|\mathbf{S}|)$ bytes, where $\overline{L}$ is the length of the hash values. Upon user revocation, the sink simply updates all the sensor nodes with the ID information of the revoked users. In addition, each node directly stores the revoked IDs, as described earlier. Now, a message sent by a user U_{ID} is of the form

$$\langle M_2, \texttt{C}, \texttt{D}, \texttt{PK}_{U_{\text{ID}}}, \texttt{AAI}_{U_{\text{ID}}} \rangle. \tag{IV}$$

Each node verifies the authenticity of a user public key in two steps: First, it calculates the corresponding root node value h_r^i using $\texttt{AAI}_{U_{\text{ID}}}$ attached in the message. Second, it checks whether the calculated h_r^i is a member of $\mathcal{V}$ stored by itself. By checking Message (4), we can easily find that HAS doubles the maximum supported number of user, compared with BAS, at the cost of per-message overhead with 20 more bytes, assuming SHA-1 is used [28]. In addition, the number can further be doubled with per-message overhead with 40 more bytes.

VI. Further Enhancements

A. Dealing With Long Messages

The messages broadcast in WSNs are usually short, due to the application-specific nature of WSNs. The query or command messages can be less than 100 B. However, there are a few cases in which long messages may be required to be broadcast in WSNs. For example, the sink may broadcast code images to the sensor nodes for the purpose of retasking WSNs [36]. The size of such code images can be on the order of kilobytes. In this case, it is not desirable to directly apply the proposed BAS or HAS scheme by signing the whole message (i.e., the message hash) only once or signing on every single packet otherwise. This is because of two reasons: First, if we sign the whole message once, then each sensor node can authenticate a message only after it obtains the entire message. That is, the sensor nodes have to buffer a large number of received packets before it can authenticate them. This obviously introduces a severe vulnerability that could result in message flooding attacks. Second, if we sign every packet belonging to the same message, the scheme overheads will significantly increase with respect to both computation and communication. This is because now, every packet is attached with a signature, which is 40 B in our setting.

Fortunately, several solutions were proposed to solve this problem in the context of code update in WSNs [37], [38]. The first solution is suitable for lossless network environments, which employs an offline hash chain technique to amortize the cost of a single digital signature over multiple packets and allow for incremental message authentication and packet pipelining [37], [39]. The second solution is aimed at tolerating packet losses. This solution makes use of a signed hash tree technique and trades message overhead for potential packet losses [38]. Both solutions can directly be superimposed with BAS and HAS in dealing with long messages. We omit the details of these solutions for lack of space.

B. Reducing the Probability of a False Positive

In [40], a method that uses two families of k hash functions, instead of one, is introduced. In addition, an element is in the set if either family gives back all the 1s from the filter. The trick is to adaptively choose one of the two families of the hash functions: Choose which family of hash functions to use for each element of your set in such a way as to keep the number of 1s in the filter as small as possible. As such, a smaller false positive probability in the same space can be achieved at the cost of more hashing. This method can reduce the probability of a false positive to half under certain conditions using the same storage space. This technique can be exploited by BAS so that we achieve a desirable probability of a false positive with a smaller storage space.

C. Optimization on Constructing the Merkle Hash Tree

Different types of network users may have different broadcast frequencies in practice. This fact can be exploited by HAS, when supporting a vast number of network users is a must. Instead of pruning the user Merkle hash tree into a set of equal-sized smaller trees, the tree can now be trimmed into the same number of different-sized smaller trees based on user broadcast frequency. The higher the frequency, the smaller the hash tree the user is grouped in. This way, the energy efficiency can be improved in the overall sense, as more messages being broadcast contain only smaller `AAI` sizes. This is similar to the idea introduced in [15].

VII. Performance Analysis

In this section, we analyze the performance of BAS and HAS with respect to communicational and computational overheads (in terms of energy consumption) and security strength. We give a quantitative analysis of the schemes and compare them with the other two basic schemes.

A. Communication Overhead

We study how the message size affects the energy consumption in communication in a WSN. We investigate the energy consumption as a function of the size of the WSN (denoted as W). We denote the hop-wise energy consumption to transmit and receive 1 B as $E_{\rm tr}$. As reported in [14], a Chipcon CC1000 radio used in Crossbow MICA2DOT motes consumes 28.6 and 59.2 μJ to receive and transmit 1 B, respectively, at an effective data rate of 12.4 kb/s. Furthermore, we assume a packet size of 41 and 32 B for the payload and 9 B for the header [14]. The header, ensuing an 8-B preamble, consists of source, destination, length, packet ID, cyclic redundancy check, and a control byte [14]. We also assume that $|M| = 20$ B.

Then, for BAS, the signature size is still the same as that for ECDSA, but only part of the message now has to be transmitted, with savings of up to 10 B. Therefore, the per-message overhead of BAS is 54 B, which is 10 B less than that of DAS. As Message (III) is 74 B, there should be three packets in total, two of which are 41 B, and one is 19 B. Therefore, there should be $41 * 2 + 19 * 1 + 8 * 3 = 125$ B for transmission (including 8-B preamble per packet). Hence, the hop-wise energy consumption of message transmission is $125 * 59.2\ \mu\text{J} = 7.40$ mJ, and the energy consumption of message reception is $125 * 28.6\ \mu\text{J} = 3.58$ mJ. For each message broadcast, every sensor node should retransmit the message once and receive w' times of the same message, assuming blind flooding is used.[4] Here, w' denotes node density in terms of the total number of sensor nodes within one unit disc, where a unit disc is a circle area with radius equal to the transmission range of sensor nodes.[5] Hence, the total energy consumption in communication will be $W * (7.4 + 3.58 * w')$ mJ.

Fig. 5 shows the energy consumption in communication as a function of W with $w' = 20$. Clearly, BAS consumes much less energy compared with others. For example, when $W = 15\,000$, CAS always costs 2.20 kJ, whereas BAS costs only 1.18 kJ.

[4]In an idealized lossless network, blind flooding, i.e., every node always retransmits exactly once for every unique message it receives, is wasteful, as individual nodes are likely to receive the same broadcast multiple times. In practice, however, blind flooding is a commonly used technique, as its inherent redundancy provides some protection from unreliable (lossy) wireless networks [41].

[5]We assume a uniform transmission range for all sensor nodes.

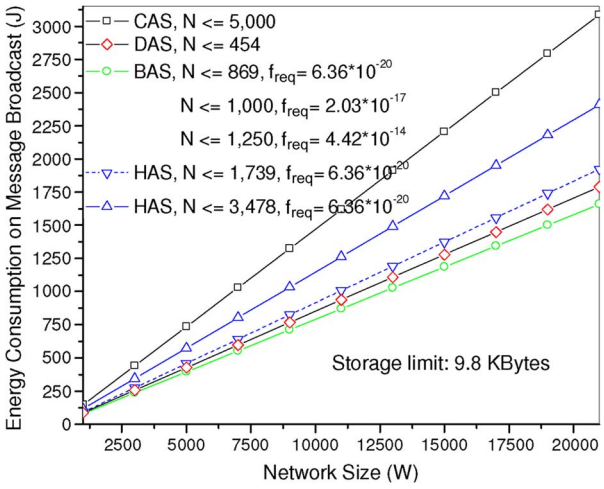


Fig. 5. Energy consumption in communication regarding different schemes.

The energy savings for a single broadcast can be more than 1000 J between BAS and CAS. Note that, although DAS also consumes much less energy than CAS, DAS only supports up to $10\,000/22 \approx 454$ users. At the same time, BAS can handle 869 users, even when $f_{\text{req}} = 6.36 * 10^{-20}$. CAS handles more users than BAS and DAS, however, at the cost of much higher energy consumption. Moreover, HAS can handle a large number of users but with a much lower energy consumption, compared with CAS. In summary, BAS demonstrates the highest communication efficiency, as well as desirable storage efficiency. From Fig. 5, we also find that the energy consumption in communication is the critical cost for WSNs, as a single broadcast of a message of only 20 B in length could cost energy on the order of kilojoules. This also exposes the severe vulnerability of the μTESLA-like schemes, as they allow the adversary to arbitrarily flood the WSN.

B. User Revocation/Addition Traffic Overhead

Another important performance metric for the broadcast authentication schemes is the overhead of the user revocation/addition traffic. As analyzed in Section V-A, BAS requires the sink to broadcast $\mathcal{V}_\Delta$ upon user revocation/addition. We have shown that, in the single-user case, $\mathcal{V}_\Delta$ can efficiently be represented by simply enumerating all its 1-valued bits, the length of which is bounded by $\overline{k}\lceil \log_2 m \rceil$ bits. That is, the per-user revocation traffic overhead is upper bounded by $\overline{k}\lceil \log_2 m \rceil$ bits. In addition, the theoretical lower bound obtained from the entropy function is $mH(p)$ bits, where $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$, and $p = (1-1/m)^{\overline{k}}$. It is not hard to see that the expectation value of $\overline{k}$ is about $k/2$, where $k = m/N \ln 2$. Our simulation shows that $\overline{k}$ is always about $k/2$. Hence, for a given $f_{\text{req}} = 6.36 * 10^{-20}$, we will have $\overline{k}\lceil \log_2 m \rceil = 68$ B and $mH(p) \approx 52$ B for $N = 1000$. This implies that the per-user revocation traffic $\mathcal{V}_\Delta$ only ranges from 52 to 68 B, on average, for $N = 1000$, depending on the used coding method.[6] In addition, for $N \leq 11\,000$, $\mathcal{V}_\Delta$ is at most 80 B on average. This overhead is much lower, compared with that of the μTESLA-like scheme that was proposed for supporting multiple users [12]. In [12], the per-user revocation traffic (i.e., a revocation certificate) is no less than $1 + \lceil \log_2 N \rceil$ hash values, which is 220 B for $N = 1000$ and 300 B for $N = 11\,000$, assuming the same hash length of 20 B. We further note that, in contrast to μTESLA-like schemes, BAS does not require periodic key chain update (for running out of available keys) among users and sensor nodes. This is the advantage that is inherent to the PKC-based schemes.

C. Computational Overhead

It was previously widely held that PKC is not suitable in WSNs, as sensor nodes are extremely computation constrained. However, recent studies [14], [15] showed that PKC with only software implementations is very viable on sensor nodes. For example, in [14], ECC signature verification takes 1.61 s with 160-bit keys on an ATmega128 8-MHz processor used in a Crossbow mote. We analyze the computation cost of the proposed schemes to further justify the suitability of PKC-based schemes in WSNs. In all our proposed schemes, the major computational cost is due to the signature verification operation. In the following analysis, we omit the cost of other operations such as hash operations and table lookup, as they are negligible, compared with the signature verification operation [14].

In CAS, two ECDSA signature verifications are needed for each broadcast message. In BAS, verification of a message takes $k = m/N \ln 2$ hash operations and one ECDSA signature verification. It was reported in [14] that an ECDSA-160 signature verification operation costs 45.09 mJ on an 8-bit ATmega128L processor running at 4 MHz. If we assume that the sensor's central processing unit is a low-power high-performance 32-bit Intel PXA271 processor, the energy cost can further be minimized. Note that the PXA271 has widely been used in many sensor products, such as Intel iMote2. It was reported in [42] that it takes 14.49 ms to verify an ECDSA-160 signature on an iMote2 platform at 416 MHz, with optimization switches enabled, and the energy cost is 3.51 mJ. Our experiment on a real iMote2 platform confirms these results. Therefore, we can obtain the computational costs of the proposed CAS and BAS schemes on different sensor platforms.[7] The results are summarized here.

Scheme	ATmega128L	PXA271
CAS	90.18 mJ	7.02 mJ
BAS	45.09 mJ	3.51 mJ

BAS is, obviously, also more computationally efficient than CAS. Furthermore, when we compare the computational cost with the communication cost on hop-wise message transmission, we can find that both are on the same order, which justifies the suitability of PKC-based schemes in WSNs.

D. Security Strength

The Bloom-filter-based public key verification ensures the security strength of the proposed scheme by enabling

[6]We assume that the number of simultaneous network users are always about N.

[7]DAS and HAS consume similar amounts of energy as BAS does, as they both require one signature verification.

immediate message authentication. That is, there is no authentication delay on messages being broadcast. Therefore, it is very hard for the adversary to perform network-wide flooding in the WSN. As we have previously analyzed, by appropriately choosing a suitable value of f_{req}, such as $6.36 * 10^{-20}$ in military applications, it is infeasible to forge a valid public/private key pair during the lifetime of the WSN. Furthermore, by embedding a time stamp into the message, the message replay attack is also effectively prevented, as WSN is assumed to be loosely synchronized [43]. Therefore, the immediate message-authentication capability provided by the proposed schemes can effectively protect the WSN from network-wide flooding attacks. This is the most significant security strength over the μTESLA-like schemes in which network-wide flooding attacks are always possible.

Moreover, since the public key operation is expensive, it is also important that sensor nodes can be resistant to local jamming attacks. Under such attacks, the adversary may simply broadcast random bit strings to the sensor nodes within his transmission range. If these neighbor sensors have to perform the expensive signature verification operation for all received messages, it will be a heavy burden on them. CAS obviously suffers from this type of attacks, as the signature verification operation has to be performed for every received message. However, in both BAS and HAS, such an attack can effectively be mitigated. This is because, in both schemes, a sensor node first verifies the authenticity of the attached user public key through hash operations; therefore, it performs a signature-verification operation for a bogus public key only with a negligible probability (e.g., $6.36 * 10^{-20}$). As reported in [14], the energy cost of SHA-1 is only 5.9 μJ/B on an 8-bit ATmega128L processor, whereas ECDSA-160 could consume 45.09 mJ on signature verification. An adversary may also flood the sensor nodes with forged messages containing valid user public keys, which can be obtained by eavesdropping on the network traffic. In this case, the forged messages can only be discarded after signature verification, and sensor nodes that are physically close to the adversary can thus be abused. We note that this type of attacks is always possible for PKC-based security mechanisms. Intuitively, this attack can still be mitigated in BAS by implementing an alert report mechanism. If a sensor node fails to authenticate the received messages multiple times in a row, it will derive that an attack is going on and alert the sink about the attack. The sink further carries out field investigations or other means to detect the adversary and take corresponding remedy actions that are outside the scope of this paper. Recently, a dynamic window scheme was proposed by Wang *et al.* [44], which is particularly for defending this kind of DoS attacks aimed at PKC-based security schemes. We leave the choice of an appropriate defense method to the system designer, as it is independent of our work.

VIII. Concluding Remarks

In this paper, we have studied the problem of multiuser broadcast authentication in WSNs. We have pointed out that symmetric-key-based solutions such as μTESLA are insufficient for this problem by identifying a serious security vulnerability that is inherent to these schemes: The delayed authentication of the messages can easily lead to severe energy-depletion DoS attacks. We have then come up with several effective PKC-based schemes to address the problem. Both the computational and communication costs of the schemes have been minimized through a novel integration of several cryptographic techniques. A quantitative energy consumption analysis, as well as security strength analysis, has been given in detail, demonstrating the effectiveness and efficiency of the proposed schemes.

References

[1] K. Ren, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," in *Proc. SECON*, San Diego, CA, Jun. 2007, pp. 223–232.

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

[3] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Netw.*, vol. 2, no. 4, pp. 351–367, Oct. 2004.

[4] K. Ren and W. Lou, *Communication Security in Wireless Sensor Networks*. Saarbrücken, Germany: VDM Verlag, 2008.

[5] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 585–598, May 2008.

[6] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in *Proc. IEEE INFOCOM*, 2009, to be published.

[7] S. Yu, K. Ren, and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009, to be published.

[8] R. Zhang, Y. Zhang, and K. Ren, "DP_2AC: Distributed privacy-preserving access control in sensor networks," in *Proc. IEEE INFOCOM*, 2009, to be published.

[9] C. Lu, G. Xing, O. Chipara, C. Fok, and S. Bhattacharya, "A spatiotemporal query service for mobile users in sensor networks," in *Proc. ICDCS*, Washington, DC, Jun. 2005, pp. 381–390.

[10] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and D. Tygar, "SPINS: Security protocols for sensor networks," in *Proc. MobiCom*, Rome, Italy, Jul. 2001, pp. 189–199.

[11] D. Liu and P. Ning, "Multi-level mTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 4, pp. 800–836, Nov. 2004.

[12] D. Liu, P. Ning, S. Zhu, and S. Jajodia, "Practical broadcast authentication in sensor networks," in *Proc. MobiQuitous*, San Diego, CA, Jul. 2005, pp. 118–132.

[13] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless ad hoc networks," in *Proc. INFOCOM*, San Francisco, CA, Apr. 2003, pp. 1976–1986.

[14] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography on small wireless devices," in *Proc. IEEE PerCom*, Kauai, HI, Mar. 2005, pp. 324–328.

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

[16] K. Ren, K. Zeng, W. Lou, and P. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 4136–4144, Nov. 2007.

[17] *Wireless Sensor Networks*, Crossbow Technol. Inc., San Jose, CA, 2004. [Online]. Available: http://www.xbow.com/

[18] *MSP430 Family of Ultra-Lowpower 16-bit RISC Processors*, Texas Instruments Inc., Dallas, TX. [Online]. Available: http://www.ti.com

[19] A. Kansal, D. Potter, and M. Srivastava, "Performance aware tasking for environmentally powered sensor networks," in *Proc. ACM SIGMETRICS*, New York, Jun. 2004, pp. 223–234.

[20] R. Amirtharajah and A. Chandrakasan, "Self-powered signal processing using vibration-based power generation," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 687–695, May 1998.

[21] A. Kansal and M. Srivastava, "An environmental energy harvesting framework for sensor networks," in *Proc. ACM/IEEE ISLPED*, Seoul, Korea, Aug. 2003, pp. 481–486.

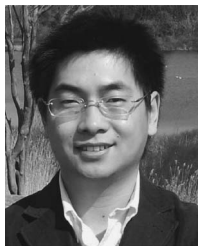
[22] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[23] NIST, "Proposed federal information processing standard for digital signature standard (DSS)," *Fed. Regist.*, vol. 56, no. 169, pp. 42 980–42 982, 1991.
[24] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York: Springer-Verlag, 2004.
[25] M. Mitzenmacher, "Compressed bloom filters," *IEEE/ACM Trans. Netw.*, vol. 10, no. 5, pp. 613–620, Oct. 2002.
[26] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: A scalable wide-area web cache sharing protocol," *IEEE/ACM Trans. Netw.*, vol. 8, no. 3, pp. 281–293, Jun. 2000.
[27] R. Merkle, "Protocols for public key cryptosystems," in *Proc. SP*, Oakland, CA, Apr. 1980, pp. 122–134.
[28] NIST, "Digital hash standard," in *Federal Information Processing Standards Publication 180-1*. Rockville, MD, Apr. 1995.
[29] 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," *Pervasive Comput.*, vol. 3, no. 4, pp. 16–23, Oct.–Dec. 2004.
[30] D. Naccache and J. Stern, "Signing on a postcard," in *Proc. FC*, 2001, pp. 121–135.
[31] *Standard Specifications for Public Key Cryptography,* IEEE Std. P1363a, 2000. [Online]. Available: http://grouper.ieee.org/groups/1363/index.html
[32] A. Shamir, "Identity based cryptosystems and signature schemes," in *Proc. CRYPTO*, Santa Barbara, CA, Aug. 1984, pp. 47–53.
[33] D. Boneh, H. Shacham, and B. Lynn, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, Sep. 2004.
[34] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE J. Sel. Areas Commun.—Special Issue Security Wireless Ad Hoc Netw.*, vol. 24, no. 2, pp. 247–260, Feb. 2006.
[35] T. Itoh and S. Tsujii, "A fast algorithm for computing multiplicative inverse in $gf(2^m)$ using normal bases," *Inf. Commun.*, vol. 78, pp. 171–177, 1988.
[36] J. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proc. ACM SenSys*, Baltimore, MD, Nov. 2004, pp. 81–94.
[37] P. E. Lanigan, R. Gandhi, and P. Narasimhan, "Secure dissemination of code updates in sensor networks," in *Proc. ACM SenSys*, San Diego, CA, Nov. 2005, pp. 278–279.
[38] J. Deng, R. Han, and S. Mishra, "Secure code distribution in dynamically programmable wireless sensor networks," in *Proc. ACM/IEEE IPSN*, Nashville, TN, Apr. 2006, pp. 292–300.
[39] R. Gennaro and P. Rohatgi, "How to sign digital streams," *Inf. Commun.*, vol. 165, no. 1, pp. 100–116, Feb. 2001.
[40] S. Lumetta and M. Mitzenmacher, *Using the Power of Two Choices to Improve Bloom Filters*. Preprint.
[41] J. McCune, E. Shi, A. Perrig, and M. Reiter, "Detection of denial-of-message attacks on sensor network broadcasts," in *Proc. SP*, Oakland, CA, May 2005, pp. 64–78.
[42] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Proc. IPSN*, Apr. 2008, pp. 245–256.
[43] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA CryptoBytes*, vol. 5, 2002.
[44] R. Wang, W. Du, and P. Ning, "Containing denial-of-service attacks in broadcast authentication in sensor networks," in *Proc. MOBIHOC*, New York, Sep. 2007, pp. 71–79.

Kui Ren (M'07) received the B.Eng. and M.Eng. degrees from Zhejiang University, Hangzhou, China, in 1998 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute, Worcester, MA, in 2007.

He has been a Research Assistant with Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China; the Institute for Infocomm Research, Singapore, and Information and Communications University, Daejeon, Korea. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago. His research is supported by the U.S. National Science Foundation. His research interests include network security and privacy and applied cryptography, focusing on security and privacy in cloud computing, lower layer attack and defense mechanisms for wireless networks, and sensor network security.

Dr. Ren is a member of the Association for Computing Machinery.



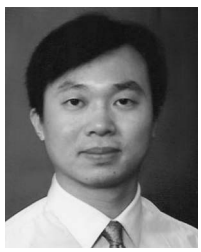
Shucheng Yu (S'07) received the B.E. degree from Nanjing University of Post and Telecommunication, Nanjing, China, in 1999 and the M.E. degree from Tsinghua University, Beijing, China, in 2004, both in computer science and engineering. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA.

From July 1999 to August 2001, he was a Software Engineer with Bright Oceans Corporation, Beijing. He was a Senior Software Engineer with the Beijing R&D Center of Cadence Design Systems from July 2006 to December 2006 and the Chinese National Source Coding Center from July 2004 to June 2006. His research interests include network security and applied cryptography. His current research interests include privacy-preserving access control.

Wenjing Lou (SM'08) received the B.E. and M.E. degrees in computer science and engineering from Xi'an Jiaotong University, Shaanxi, China, the M.A.Sc. degree in computer communications from Nanyang Technological University, Singapore, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville.

From December 1997 to July 1999, she was a Research Engineer with the Network Technology Research Center, Nanyang Technological University. In 2003, she joined the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute (WPI), Worcester, MA, as an Assistant Professor; she is currently an Associate Professor. Her current research interests include ad hoc, sensors, and mesh networks, with emphasis on network security and routing issues.

Dr. Lou has been an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2007. She was named Joseph Samuel Satin Distinguished Fellow in 2006 by WPI. She was the recipient of the U.S. National Science Foundation Faculty Early Career Development Award in 2008.



Yanchao Zhang (M'06) received the B.E. degree in computer communications from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1999, the M.E. degree in computer applications from Beijing University of Posts and Telecommunications, Beijing, China, in 2002, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2006.

He then joined the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, as an Assistant Professor. His research interests include network and distributed system security, wireless networking, and mobile computing.

Prof. Zhang is currently an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and a Feature Editor for IEEE WIRELESS COMMUNICATIONS. He is also a Technical Program Committee Co-Chair for the Communication and Information System Security Symposium (IEEE GLOBECOM 2010).