Fast Authentication in Wireless Sensor Networks

Chafika Benzaid[a], Karim Lounis[a], Ameer Al-Nemrat[b], Nadjib Badache[a], Mamoun Alazab[c]

[a] *Laboratoire des Systèmes Informatique, USTHB, Algérie*
[b] *Architecture, Computing, and Engineering School, UEL, UK*
[c] *Australian National University, AUS*

Abstract

Broadcast authentication is a fundamental security service in wireless sensor networks (WSNs). Although symmetric-key-based μTESLA-like schemes were employed due to their energy efficiency, they all suffer from DoS attacks resulting from the nature of delayed message authentication. Recently, several public-key-based schemes were proposed to achieve immediate broadcast authentication that may significantly improved security strength. However, while the public-key-based schemes obviate the security vulnerability inherent to symmetric-key-based μTESLA-like schemes, their signature verification is time-consuming. Thus, speeding up signature verification is a problem of considerable practical importance, especially in resource-constrained environments. This paper exploits the cooperation among sensor nodes to accelerate the signature verification of vBNN-IBS, a pairing-free identity-based signature with reduced signature size. We demonstrate through on extensive performance evaluation study that the accelerated vBNN-IBS achieves the longest network lifetime compared to both the traditional vBNN-IBS and the accelerated ECDSA schemes. The accelerated vBNN-IBS runs 66% faster than the traditional signature verification method. Results from theoretical analysis, simulation, and real-world experimentation on a MICAz platform are provided to validate our claims.

Keywords: Broadcast authentication, ID-based cryptography, Digital signature, Accelerated verification, Wireless sensor networks

1. Introduction

In wireless sensor networks, message broadcast is an efficient and a common communication paradigm that allows a multitude of users to join in and disseminate messages into the network dynamically in order to obtain information of their interest. Unfortunately, sensor networks are very susceptible for attacks. Due to the nature of wireless communication in sensor networks, adversaries can easily eavesdrop on the traffic, inject bogus data messages or alter the contents of legitimate messages during multihop forwarding. Hence, authentication mechanisms must be provided to ensure that communication at all times is performed between the correct entities.

When the issue of broadcast authentication first appeared, symmetric key cryptography-based μTESLA-like schemes [1, 2] were employed due to their energy efficiency. μTESLA-like schemes provide source authentication and message integrity by using one-way hash chains and delayed disclosure of authentication keys. By using Message Authentication Code (MAC) and one-way hash functions, μTESLA is able to take low computation effort to broadcast. Nevertheless, the μTESLA induces cache delays for broadcast packets which could be exploited by an attacker to launch DoS attacks [3]. The lack of immediate authentication makes μ-TESLA and its variations unsuitable for applications with real-time requirements. In addition, they also require some level of synchronization between all nodes in the network which must be achieved by periodic broadcasting [4]. All these shortcomings make such schemes unsuitable for broadcast authentication.

Public key cryptography (PKC), on the other hand, is desirable for broadcast authentication. Employing PKC for implementing broadcast authentication in WSNs provides simple solutions, strong security resilience, good scalability and immediate message authentication, when compared to symmetric-key based solutions [5]. Although there is a prejudice against the feasibility of PKC in WSN, recent studies [6, 7] have reported that PKC is possible in WSNs. For instance, Elliptic Curve Cryptography (ECC) signature verification takes $1.61s$, with 160-bit keys on an ATmega128 8-MHz processor [6]. Thus, several PKC-based broadcast authentication protocols have been proposed [8, 7, 9, 10]. These protocols are based on several cryptographic techniques, including Merkle Hash Tree [11], public-key ECC-based signature scheme such as ECDSA [12], and ID-based signature scheme [13] with either pairing-free or optimal-pairing. While the PKC-based schemes avoid the security vulnerability intrinsic to μTESLA-like schemes, the rela-

tively slow signature verification in public-key cryptosystems causes high energy consumption and long verification delay for broadcast authentication in WSNs. Thus, speeding up signature verification is a problem of considerable practical importance, especially in resource-constrained environments. Fan and Gong [5] proposed a method to accelerate ECDSA signature verification in WSNs by exploiting the cooperation among sensor nodes. The speedup results from each node probabilistically forwarding a partially-calculated signature to its neighbors. Then many sensor nodes can use the received intermediate computation results to accelerate their signature verifications. Unlike public-key ECC-based authenticated systems, ID-based authenticated systems do not require the transmission of public-key certificates which reduces the certificate overhead and improves computational efficiency. This makes them especially attractive for use in WSNs. However, the verification of ID-based signatures is still slow and computationally expensive for WSNs.

Inspired by the acceleration technique of Fan and Gong [5], we propose in this paper an accelerated verification of digital signatures generated by vBNN-IBS [9], a pairing-free identity-based signature with reduced signature size. However, unlike the original technique and in order to harden the attacker's task, the proposed technique releases the sum of two intermediate computation results rather than releasing them separately. By doing so, it becomes more difficult for an attacker to forge the two intermediate results. It also allows to reduce the message overhead while more intermediate results are released. Another important contribution of our work consists in the extensive performance evaluation through theoretical, simulation and real-world experimental studies. Through this evaluation, we:

- carefully chose the optimization methods used for a scalar point multiplication based on ensuring a satisfactory compromise between the execution time and the required memory size.

- emphasized the impact of energy consumed by the different node's states on the protocol performances; something that was neglected in similar works. We found out that node's states constitutes an important fraction (more than 92%) of the total energy dissipated within the network. Therefore, neglecting this important fraction can lead to erroneous conclusions about the protocol performances.

- demonstrated that the scheme's performances are not only affected by the number of nodes releasing intermediate results, but also by the

deployment topology of nodes and the diffusion pattern followed to broadcast the user packet in the entire network.

The rest of the paper is organized as follows. In Section 3, we give a brief introduction to elliptic curve cryptography and vBNN-IBS [9]. Section 4 describes the proposed acceleration technique for signature verification in WSNs. In Section 5, we improve the scheme and we discuss the selection of system parameters. The security strengths of the proposed scheme are discussed in Section 6. Section 7 presents theoretical, simulation, and real-world experimentation performance results that demonstrate the effectiveness of the proposed acceleration technique. The impact of the network topology on the performance achieved is assessed in Section 8. Finally, Section 9 outlines our concluding remarks and future work directions.

2. Related Work

When the issue of broadcast authentication first appeared, symmetric key cryptography-based μTESLA-like schemes [1, 2, 14] were employed due to their energy efficiency. μTESLA-like schemes provide source authentication and message integrity by using one-way hash chains and delayed disclosure of authentication keys. μTESLA [1] is an extension of the TESLA [15] authentication scheme adapted for WSNs to reduce the computation overhead. The μTESLA scheme employs a chain of authentication keys linked to each other by a one-way hash function; each key in the chain is the hash value of the next key computed using the one-way hash function. The first key of the chain, called the *key-chain commitment*, is securely sent to all the receiving nodes. A sender broadcasts a packet along with a Message Authentication Code (MAC) generated using the current key from the one-way chain, that key will be disclosed after a pre-defined period of time. Upon receiving this packet, each receiver checks if the packet was sent before the disclosure of the key used to calculate the attached MAC. If so, the receiver buffers the packet which will be authenticated when the corresponding disclosed key is received. μTESLA faces a scalability problem because the key-chain commitment has to be unicasted to each node which incurs high communication overhead limiting the network scale. Moreover, the key chain length is limited, and thus cannot support broadcast for a long time.

To address the two aforementioned problems, the multi-level μTESLA [2] technique was proposed. The unicast-based distribution of key chain commitments is bypassed by predetermining and broadcasting the commitments.

To extend the lifetime of authenticated broadcast without requiring a very long key chain, a multi-level key chains are used. The keys in the lowest level key chains are used for authenticating data packets and usually lasts for a relatively short period. Each higher-level key chain is used to distribute the commitments of the immediately lower-level key chains. However, multi-level μTESLA scheme as well as the original μTESLA protocol are not scalable in terms of the number of senders. Subsequently, the tree-based μTESLA [14] scheme was proposed to support a large number of senders over a long period of time by using Merkle hash tree [11]. The basic idea consists first in defining multiple μTESLA instances which may be used by different senders during different periods of time. Then, a Merkle hash tree is built to authenticate and distribute the initial parameters (i.e., the key chain commitment, starting time, duration of each μTESLA interval, etc.) for μTESLA instances. The i-th leaf corresponds to the hash value of the initial parameters for the i-th μTESLA instances. Unfortunately, the tree-based μTESLA scheme supports the multisender scenario at the cost of higher communication overhead per message, limiting thus the number of senders.

By using MAC codes and one-way hash functions, μTESLA-like schemes are able to take low computation effort to broadcast. Nevertheless, those schemes induce cache delays for broadcast packets which could be exploited by an attacker to launch DoS attacks [3]. The lack of immediate authentication makes μTESLA and its variations unsuitable for applications with real-time requirements. In addition, they also require some level of synchronization between all nodes in the network which must be achieved by periodic broadcasting [4]. All these shortcomings make such schemes unsuitable for broadcast authentication.

Schemes proposed in [16, 17, 18, 19, 20] are based on one-time signature mechanism. Those schemes are computationally efficient as signatures only rely on one-way functions without trapdoors. However, some of these schemes require large storage space and produce large signatures leading to high communication overhead. Furthermore, the security level of such schemes depends on the collision resistant of the one-way function used. Indeed, the security strength is inversely proportional to the number of signatures generated.

Public key cryptography (PKC), on the other hand, is desirable for broadcast authentication. Employing PKC for implementing broadcast authentication in WSNs provides simple solutions, strong security resilience, good scalability and immediate message authentication, when compared to symmetric-

key based solutions [5]. Although there is a prejudice against the feasibility of PKC in WSN, recent studies [6, 7] have reported that PKC is possible in WSNs. Thus, several PKC-based broadcast authentication protocols have been proposed [7, 8, 9, 10].

A straightforward authentication scheme based on certificate was proposed in [7]. The scheme is inefficient in terms of communication and computation costs. A certificate has to be transmitted along with each signature which induces a larger per-message overhead. Moreover, extra computation operations are needed to validate the received certificate; the certificate validation is equal to a signature verification. Consequently, the energy consumption during the communication and computational processes is increased. Another concern with this scheme consists in its inefficiency to support user revocation, because the certificate revocation list stored in each sensor node requires a storage space linear to the total number of revoked certificates. To avoid the storage overhead of the certificate-based scheme, a Merkle hash tree-based authentication scheme [7] was proposed. In this scheme, a Merkle hash tree with leaves corresponding to the current users is constructed. Each leaf of the Merkle tree contains the binding between the corresponding user ID and his public key. With only the hash value of the root node of the hash tree stored on each sensor node, a revoked or an invalid user public key can never pass the verification. Nevertheless, the Merkle hash tree-based scheme is communication inefficient when the number of users becomes large. This is because the size of AAI (Auxiliary Authentication Information) grows logarithmically with the number of users. This scheme was improved by increasing the number of stored hash values on sensor nodes in order to reduce the size of AAI [7]. Although this improvement decreases the communication cost, it increases in parallel the storage cost. Ren *et al.* [8] proposed a multi-user broadcast authentication scheme called HAS. To reduce the computational and communication costs while increasing the number of users, HAS uses several cryptographic building blocks, such as the Bloom filter [21], a variant of ECDSA with the partial message recovery [22], and the Merkle tree [11]. Nevertheless, HAS does not support user scalability due to the use of Merkle hash tree; a new user can be added to WSN only after the revocation of an old one.

ID-based authentication schemes [7, 9, 10] have recently attracted substantial attention due to their communication efficiency and support of scalability. The authors in [7] proposed a broadcast authentication scheme using ID-based signatures [23]. This scheme decreases the communication overhead

as it removes the need of certification transmission and provide sound scalability. However, it incurs a very high computational overhead as is based on the costly bilinear pairing operations. Cao *et al.* [9] proposed IMBAS, a more efficient ID-based multi-user broadcast authentication scheme based on a variant of the pairing-free ID-based signature scheme BNN-IBS [24] with reduced signature size. IMBAS achieves better scalability and lower energy consumption compared to scheme in [7]. The $\mathcal{EIBAS}$ scheme presented in [10] uses a pairing-optimal identity-based signature scheme with message recovery, where the original message of the signature is not required to be transmitted. The $\mathcal{EIBAS}$ scheme requires the shortest broadcast message size compared to the two aforementioned ID-based schemes. However, $\mathcal{EIBAS}$ still relies on bilinear pairing operations which are computationally expensive.

While the PKC-based schemes avoid the security vulnerability intrinsic to μTESLA-like schemes, the relatively slow signature verification in public-key cryptosystems causes high energy consumption and long verification delay for broadcast authentication in WSNs. Thus, speeding up signature verification is a problem of considerable practical importance, especially in resource-constrained environments. Fan and Gong [5] proposed a method to accelerate ECDSA signature verification in WSNs by exploiting the cooperation among sensor nodes. The speedup results from each node probabilistically forwarding a partially-calculated signature to its neighbors. Then many sensor nodes can use the received intermediate computation results to accelerate their signature verifications. Unlike public-key ECC-based authenticated systems, ID-based authenticated systems do not require the transmission of public-key certificates which reduces the certificate overhead and improves computational efficiency. This makes them especially attractive for use in WSNs. However, the verification of ID-based signatures is still slow and computationally expensive for WSNs.

Inspired by the acceleration technique of Fan and Gong [5], we propose in this paper an accelerated verification of digital signatures generated by vBNN-IBS [9]. As mentioned above, vBNN-IBS is a variant of Bellare Namprempre Neven Identity-Based Signature (BNN-IBS) scheme [24].

3. Preliminaries

In this section, we first give a brief introduction to elliptic curve cryptography, followed by a variant of ID-based signature scheme with reduced

signature vBNN-IBS [9].

3.1. Elliptic curve cryptography

Elliptic curves used in cryptography are typically defined over a prime finite field $\mathbb{F}_q$, where q is a large prime number. It is defined by a cubic equation

$$y^2 = x^3 + ax + b \tag{1}$$

with $a,\ b \in \mathbb{F}_q$ are constants such that $4a^3 + 27b^3 \not\equiv 0$ [25]. An elliptic curve E over $\mathbb{F}_q$ consists of the set of all pairs of coordinates (x, y) that satisfy the equation (1) along with a *point at infinity* $\mathcal{O}$.

$$E(\mathbb{F}_q) = (x, y) \in \mathbb{F}_q * \mathbb{F}_q \; where \; y^2 = x^3 + ax + b \cup \mathcal{O} \tag{2}$$

$P \in E(\mathbb{F}_q)$ is a point of order p and $\mathbb{G}$ is a group generated by P. Note that p is a prime number with p^2 does not divide the order of $E(\mathbb{F}_q)$. $\mathbb{G}$ forms a cyclic group under the point addition " + " defined as follows: Let P, $Q \in E(\mathbb{F}_q)$, l be the line containing P and Q (tangent line to $E(\mathbb{F}_q)$ if $P = Q$), and R, the third point of intersection of l with $E(\mathbb{F}_q)$. Let l' be the line connecting R and $\mathcal{O}$. Then P" + "Q is the point such that l' intersects $E(\mathbb{F}_q)$ at R, $\mathcal{O}$ and P"+"Q. Point multiplication over $E/\mathbb{F}_q$ can be computed as follows: $nP = \underbrace{P + P + \cdots + P}_{n \; times}$.

The problem of finding n given nP and P is called the Elliptic Curve Discrete Logarithm Problem (ECDLP). It is computationally infeasible to solve ECDLP for appropriate parameters [25]. The hardness of ECDLP allows several cryptographic schemes based on elliptic curves.

3.2. vBNN-IBS signature scheme

vBNN-IBS [9] is a pairing-free ID-based signature scheme with reduced signature size, for securing users' broadcasts in wireless sensor networks. vBNN-IBS [9] is an ID-based cryptosystem, where a user's public key is directly derivable from his/her publicly known identity information, and the user's private key is calculated by a trusted party, called PKG (Private Key Generator). A user's identifier serves as the user's public key, and the user's private key is the user's public key certificate. vBNN-IBS is implemented as follows:

Setup Given the security parameter k, PKG takes the following steps:

1. Specify $E/\mathbb{F}_q$ and a point P of order p.
2. Select a system secret key x at random from $\mathbb{Z}_p$ and set the system public key $P_0 = xP$.
3. Choose two cryptographic hash functions $H_1 : \{0,1\} \times \mathbb{G}_1^* \longrightarrow \mathbb{Z}_p$ and $H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_p$.
4. Publish system parameters $(E/\mathbb{F}_q, P, p, P_0, H_1, H_2)$ and keep x secret.

User-Key Extraction Given a user A's unique identifier $ID_A \in \{0,1\}^*$, PKG generates A's private key Pri_A based on Schnorr signature [11] as follows:

1. Choose at random $r \in \mathbb{Z}_p$ and compute $R = rP$.
2. Use system secret key x to compute $s = r + cx$, where $c = H_1(ID_A||R)$.

A's private key is the pair (R, s), and is sent to A by PKG through a secure channel.

Signature Generation User A with identifier ID_A signs a message m with its private key $Pri_A = (R, s)$ as follows:

1. Choose at random $y \in \mathbb{Z}_p$ and compute $Y = yP$.
2. Compute $h = H_2(ID_A, m, R, Y)$ and $z = y + hs$.

The tuple (R, h, z) is A's signature on m.

Signature Verification Given (R, h, z) signature, ID_A and message m, a verifier first computes $c = H_1(ID_A||R)$. Then it checks whether the equation: $h = H_2(ID_A, m, R, zP - h(R + cP_0))$ holds. The signature is accepted if it does and rejected otherwise.

3.3. Message broadcast and authentication

In [9], Cao *et al.* proposed that if a user with identifier ID wants to broadcast a message M, it sends the following packet:

$$< M, tt, ID, Sig\{M, tt, ID\} > \quad (3)$$

where tt denotes the current time and $Sig\{M, tt, ID\}$ is the user's vBNN-IBS signature over $\{M, tt, ID\}$.

Upon the receipt of packet 3, a sensor does the following:

1. Check whether tt is fresh.
2. Verify vBNN-IBS signature if tt is valid, drop the message otherwise.
3. Reject the message and drop it if the signature verification fails; propagate the message to the next hop otherwise.

4. Faster vBNN-IBS signature verification

In the broadcast authentication procedure, as shown previously, all sensor nodes execute the same signature verification after receiving a broadcast packet. To verify vBNN-IBS signature, each node needs to calculate $h = H_2(ID_A, m, R, zP - h(R + cP_0))$, then it must calculate zP, hR, and hcP_0. These three values are scalar multiplications over elliptic curve which incur significant energy consumption. To reduce this cost, we propose an accelerated variant of the vBNN-IBS signature verification which is inspired by the acceleration technique of Fan *et al.* [5]. The key idea of this acceleration technique comes from the observation that all sensor nodes execute the same signature verification procedure during the broadcast authentication. Therefore, some sensor nodes of the network will consume their energy to release some intermediate information, as a result, the signature verification of their neighbors can be accelerated significantly.

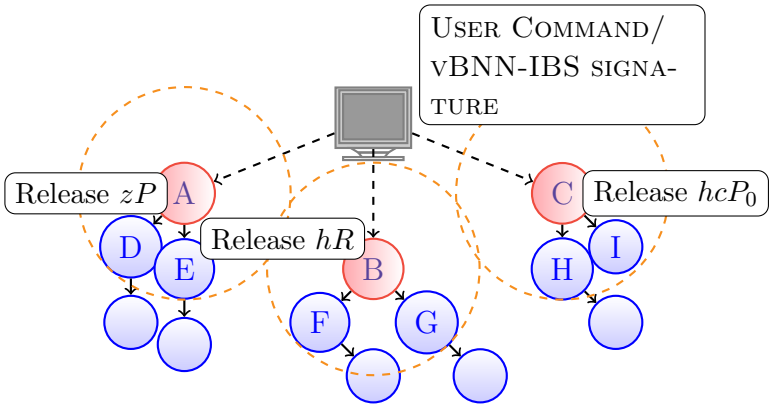


Figure 1: Faster vBNN-IBS digital signature verification through nodes cooperation

In Fig. 1, a user broadcast the packet $< M, tt, ID, Sig\{M, tt, ID\} >$, where $Sig\{M, tt, ID\} = (R, h, z)$ is the corresponding vBNN-IBS signature of (M, tt, ID). When the nodes A, B and C receive the packet and finish the signature verification successfully, they decide to release their intermediate computation results $l_1P = zP$, $l_2R = hR$, and $l_3P_0 = hcP_0$, respectively. By this mean, nodes D and E, which are the neighbors of node A, can fast verify the digital signature by performing two elliptic curve point multiplications and two elliptic curve point additions $zP - hR - hcP_0$, where hR and hcP_0 are computed by nodes D and E themselves and zP comes from the contribution of node A. Moreover, nodes F, G, H, I can also perform fast signature verification in a similar way. Hence, if some node in WSN releases its intermediate computation result, all its neighbors can fast verify the digital signature by just calculating two scalar multiplications and two elliptic curve point additions (instead of three scalar multiplications), which can improve performances by 33% as compared to the traditional vBNN-IBS signature verification procedure.

Note that in our scheme, we can assume that each node that decides to release its intermediate computation results, it sends two information among the three information to its neighboring nodes, then it can send $(l_1P - l_2R) = (zP - hR)$, $(l_1P - l_3P_0) = (zP - hcP_0)$ or $(l_2R + l_3P_0) = (hR + hcP_0)$. In this case, our solution can achieve around 66% performance improvement over the traditional vBNN-IBS signature verification procedure.

So, a question might arise: why not release the three intermediate computation results? In fact, the sensor nodes cannot use these three information

(i.e. l_1P, l_2R, and l_3P_0) to fast verify the signature with two elliptic curve point additions. The reason is that an adversary can capture a sensor node A with identifier ID_A and easily launch the following attack:

The adversary :

1. generates a bogus message m'.
2. randomly chooses, R', l'_1, l'_2 and l'_3.
3. computes $l'_1P - l'_2R' - l'_3P_0$.
4. computes $h' = H_2(ID_A, m', R', l'_1P - l'_2R' - l'_3P_0)$.
5. releases the signature (R', h', l'_1) and the three additional information l'_1P, l'_2R' and l'_3P_0.

The victim :

1. computes $c' = H_1(ID_A||R')$.
2. From the three additional information released and the signature, the victim calculates $H_2(ID_A, m', R', l'_1P - l'_2R'l'_3P_0)$ and compares it with the received h'. As a result, the victim accept m' as a valid message.

To avoid the above attack, we only allow sensor nodes to use at most two intermediate results among $\{l_1P, l_2R, l_3P_0\}$ from their neighboring nodes for signature verification. For the sake of simplicity, we assume that if some sensor nodes release their intermediate computation results they will release l_2R and l_3P_0. Thus, if a sensor node decides to release its intermediate results, it sends the following message:

$$\{M, tt, ID, Sig\{M, tt, ID\}, (l_2R + l_3P_0)\} \tag{4}$$

Where $Sig\{M, tt, ID\}$ denotes the user's vBNN-IBS signature over $\{M, tt, ID\}$.

Let MUL and ADD denote the elliptic curve scalar multiplication and the elliptic curve point addition, respectively. In our scheme, a sensor node may receive a data packet $\{M, tt, ID, (R, h, z)\}$ or $\{M, tt, ID, (R, h, z), l_2R+l_3P_0\}$. If a fresh packet $\{M, tt, ID, (R, h, z)\}$ is received, the sensor node will first compute l_1P and then wait for a very short time period α to see whether it

can obtain useful information from its neighbors for accelerating the signature verification. If it is, the node can finish the signature verification with $1MUL + 1ADD$. Otherwise, the node will complete the verification itself with $3MUL + 2ADD$ after the time period α. If the signature is verified successfully, the sensor node will continue forwarding the broadcast package to its neighbors. Otherwise, the sensor node will send a signed report to the base station. Once the base station receives enough reports from the network, it will perform appropriate security mechanisms to identify compromised nodes in WSN. Although the above basic scheme is simple and efficient, it is still vulnerable to the following attack:

The adversary :

1. chooses randomly m', z', R', Y', $M' = \{m', tt, ID_A\}$.
2. computes $h' = H_2(ID_A, M', R', Y')$.
3. computes $l_1'P = z'P$.
4. computes $Q = l_1'P - Y'$.
5. uses (R', h', z') as the signature of the message M' and releases the bogus broadcast package $\{M', (R', h', z'), Q\}$ to its neighbors.

The victim :

1. calculates $c' = H_1(ID_A||R')$
2. calculates $l_1'P = z'P$.
3. From the additional information released and the signature, the victim calculates $H_2(ID_A, M', R', z'P - Q)$ and compares it with the received h'. As a result, the victim accepts M' as a valid message.

To cope with this attack, we adopt the idea in [5] to propose an enhanced scheme as explained in the following paragraph. The enhanced scheme takes advantage of the redundancy of broadcast packets in the WSN.

5. The enhanced scheme

In the enhanced scheme, each sensor node first waits for α seconds and buffers β data packets (i.e., (R, h, z) or $\{(R, h, z), l_2R + l_3P_0\}$) received from

its neighbors, where α and β are selected such that the sensor node can receive at least one data packet from an honest neighbor. The sensor node then checks whether the cached β data packets have identical (R, h, z) and $l_2R + l_3P_0$. If the sensor node finds that the received data packets have different R, h, z or $l_2R + l_3P_0$, it will report the potential attack to the base station immediately. Otherwise, the sensor node checks whether it has received useful data packets $l_2R+l_3P_0$ for accelerating signature verification. If it is, it will calculate l_1P and then complete the signature verification with $1MUL + 1ADD$. Otherwise, the sensor node will perform the traditional signature verification with $3MUL + 2ADD$. The remaining steps after the signature verification are the same as those in the basic scheme.

5.1. Selection of α and β

We assume in our scheme that on average a sensor node A has λ neighbors and half of them will broadcast data packets to A at a certain communication round. We also assume that among A's $\lambda/2$ neighbors, v nodes can be compromised by adversaries and each of them can send at most w bogus data packets to A during that communication round. Note that all compromised nodes must collude to send identical bogus data packages to A. Otherwise, A will discard all cached data packets and report to the base station. To make our scheme resilient against collusive attacks, the threshold β should satisfy the following condition:

$$\lambda/2 \geq \beta \geq v.w + 1 \tag{5}$$

After determining the threshold β, the delay α is chosen such that β data packets can be received by the sensor node A. The delay α depends on the transmit data rate and the radio backoff of the radio transceiver used on sensor nodes. A radio backoff is a period of time where the radio pauses before attempting to transmit. Two backoff periods are to be considered, namely: *initial backoff* and *congestion backoff*. Taking into account all these factors, we suggest that the delay α should satisfy the following condition:

$$\begin{aligned} \alpha \geq (&Size_{MAX}/Rate_{MAX} + Init\_Backoff_{MAX} \\ &+ Cong\_Backoff_{MAX}) * \beta \end{aligned} \tag{6}$$

where, $Size_{MAX}$, $Rate_{MAX}$, $Init\_Backoff_{MAX}$, and $Cong\_Backoff_{MAX}$ are, respectively, the maximum allowable packet size, the maximum transmit data rate, the maximum initial backoff, and the maximum congestion backoff.

We assume that there are on average N sensor nodes working on the signature verification and P_T is the probability that T sensor nodes will release their intermediate computation results. Let E_s, E_r, and E_{MUL} be the energy consumption of sending and receiving one packet, and calculating one elliptic curve scalar multiplication on sensor nodes, respectively. Recall that we assume that each node has λ neighbors on average. Then, we can roughly estimate the additional energy consumption/saving due to the use of our fast signature verification technique as follows:

1. T sensor nodes will locally broadcast their intermediate computation results, with energy consumption of $T * E_s$.

2. About $\lambda T/2$ sensor nodes will receive the intermediate computation results, with energy consumption of $\lambda T/2 * E_r$.

3. About $\lambda T/2$ sensor nodes will accelerate their signature verifications using the received intermediate computation results, with energy saving of $\lambda T/2 * 2E_{MUL} = \lambda T * E_{MUL}$.

Therefore, the expected additional energy consumption/saving will be:

$$\sum_{T=1}^{N} P_T(T * E_s + \lambda T/2 * E_r - \lambda T * E_{MUL}) \tag{7}$$

6. Security Analysis

In what follow, the security strength of the proposed broadcast authentication protocol is discussed:

6.1. Replay Attack

A replay attack is the retransmission of an outdated legitimate message as a current message. Our scheme withstand this attack by the timestamp information tt carried with broadcasted messages. However, it is important to point out that the use of timestamp to prevent a replay attack must be supported by a synchronization mechanism (e.g., [26, 27]) which requires

itself to be kept secure [28]. As an alternative, random nonces can be used instead of timestamps. A nonce is an unpredictable bit string, usually used to achieve freshness.

6.2. User Revocation

User revocation can be achieved by simply broadcasting the identity of revoked users by the base station. The revoked identities are placed into the revocation list of nodes upon their reception. To reduce the risk of user compromise due to hardware capture, we assume a password-based protection approach [9] to safeguard the private keys. The approach consists in storing a private key calculated from the original private key and a hashed password. The original private key will be recovered from the stored one only when the correct password is provided. Indeed, finding (R, s) from (R', s') without knowing the password is as hard as the ECDLP problem.

6.3. Sybil Attack

In Sybil attack [29], an attacker illegitimately claims multiple identities in order to disrupt the behavior of network's protocols and applications. The ID-based cryptography mechanism relies on the existence of a trusted authority, called a private key generator (PKG). We assume that the base station is always reliable and can play the trusted authority role. Recall that PKG generates a private key for each user's identifier, and assigns the private key to the user. The user can always sign on any message with private key, and the generated signature can be verified with the user's ID. In order to be able to fake a user's identity, the attacker has to fake a new private key which is possible only by obtaining the system secret key x. Unless the secret key x is stolen or the base station is attacked, the Sybil attack can be prevented.

6.4. Denial of Service Attack

Denial of Service (DoS) attack is any event that diminishes or eliminates a network's capacity to perform its expected function [30]. Due to their inherent resource constraints, WSNs are especially vulnerable to DoS attacks that aim to exhaust the network resources. DoS attack against the sensor node storage is avoided by the immediate authentication of broadcast messages upon their reception using signature verification. Moreover, the forged packets are dropped rather than being stored or forwarded to the next hop which prevents both storage and bandwidth resource exhaustion. The energy

consumed by sensor nodes during the signature verification process is substantial even with acceleration. Therefore, an attacker can easily force nodes in its vicinity to run out of the battery's power by verifying the signature of a large number of forged packets. However, this attack can be mitigated by setting a threshold on the number of verification failure, after which a sensor node will send a signed report to the base station. Once the base station receives enough reports from the network, it will perform appropriate security mechanisms (outside the scope of this paper) to identify compromised nodes in the network.

6.5. Scalability

The proposed broadcast authentication scheme achieves both sender and receiver scalability based on vBNN-IBS. A user can dynamically join the network by querying the base station for system parameters and a private key corresponding to his/her identity. In addition, a large number of sensor nodes can be supported, and new nodes can be added to the WSN after being preloaded with system parameters.

7. Performance Evaluation

We analyzed the performance of the scheme in the ideal case (no adversary) and in the 4×4 grid-based WSN (See Figure 2).

7.1. Case Study and Evaluation Methodology

We propose a sensor network where each node only can directly communicate with its one-hop neighbors. A user sends its signed broadcast packet to node 1 at Round 0. After six communication rounds, the broadcast packet will be received and verified by all sensor nodes. Furthermore, in our signature verification scheme, we assume that one sensor node will release the intermediate computation result $l_2R + l_3P_0$ in each communication round (the red nodes 1, 2, 6, 7, 11 and 12 on Figure 2).

The performance of the proposed scheme in terms of energy consumption and diffusion latency was first evaluated through a theoretical analysis assuming that MICAz motes are used. After that, The protocol was implemented on TinyOS and evaluated through both simulation using Avrora [31] and real-world experimentation using Crossbow MICAz motes. Note that all simulation and experimentation results are averaged over 10 independent runs with 95% confidence interval.

The performance evaluation was done in two stages:

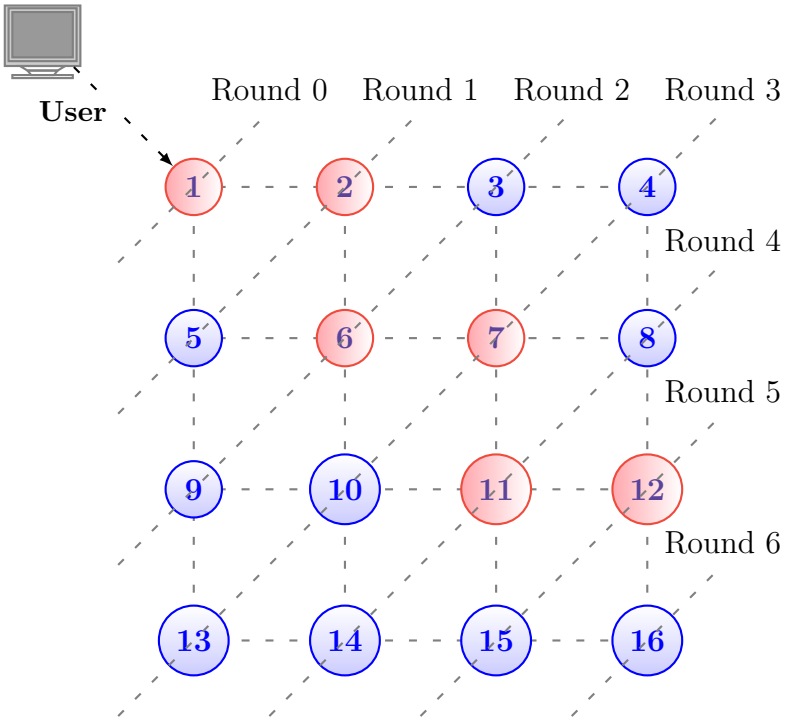


Figure 2: Broadcast Authentication in a 4×4 grid-based WSN

First stage In which only the three dominant energy consuming operations, involved in the signature verification procedure, were considered. These operations are scalar point multiplication, message transmission, and message reception. Here we assumed that the power consumption by the remaining operations and node's states is negligible.

Second stage In which all operations and node's states are taken into consideration. Thus, the performance evaluation is done over the entire run of the experiment. Sensor nodes involved in the broadcast authentication process are configured to operate in three different states, namely: *active*, *idle*, and *standby*. In the *active* state, the node is either *transmitting*, *receiving*, or *processing* a packet. The *idle* state keeps the radio transceiver *on*, listening for authentication packets. While in the *standby* state, the node is unable to transmit, receive, or process information. Note that nodes consume power not only in active state as assumed in the first stage, but also when operating in idle or standby

states. The power consumed in these two states increases over time. Hence, more the authenticated diffusion takes time; more the energy is consumed by the two states. The second stage let us see how the acceleration approach can influence the impact of the idle and standby power in the total power consumption.

The performance evaluation was conducted by setting up the following diffusion scenario: initially a node is in the idle state until receiving authentication packets. At that moment, the node moves to the active state. Once the node finishes its authenticated diffusion, it goes into standby state until the end of the authenticated broadcast in the entire network. The diffusion scenario aims to show how the energy gain achieved by the accelerated approach is affected by the presence of a power-saving sleep mode (i.e. standby state). Figure 3 illustrates the different states by which a node transit during the diffusion.

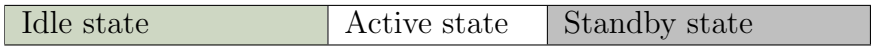


Figure 3: The sensor node's states during an authenticated diffusion

7.2. Theoretical Analysis

To give a detailed quantitative analysis, we assume that MICAz motes, which work at 8 MHZ with a 8-bit processor ATmega128L, and which adopt IEEE 802.15.4 standard, are used. The power level of a MICAz mote is $3.0V$, the current draw in active mode is $8.0mA$, the receiving current draw is $19.7mA$, the transmitting current draw is $17.4mA$, and the data rate is $250kbps$. An ATmega128L processor takes $0.81s$ to carry out a point multiplication over elliptic curve [32].

A vBNN-IBS signature comprises one point over $E(\mathbb{F}_q)$ and an integer from $\mathbb{Z}_p$. To achieve the same security strength as 1024-bit RSA, the known smallest reachable sizes of q and p are $168bits$ and $166bits$, respectively [9]. Therefore, the message of form (4) is $139bytes$, the resulted signature size is $83bytes$, assuming M is $10bytes$, ID is $2bytes$, tt is $2bytes$, and $l_2R + l_3P_0$ is $42bytes$.

Based on the formula of calculating the energy consumption on MICAz motes, we obtain the following basic facts:

- A MICAz mote needs to transmit up to $128bytes$ in the physical layer. Hence, to broadcast a packet of $139bytes$, we need to transmit two packets instead of only one transmitted in vBNN-IBS [9]. A MICAz mote consumes $E_s = 3.0 \times 17.4 \times 8/250 = 1.67\mu J$ and $E_r = 3.0 \times 19.7 \times 8/250 = 1.89\mu J$ to transmit and receive one byte respectively. So, the additional information $l_2R + l_3P_0$ requires to transmit one additional packet which consumes $E_s = 3.0 \times 17.4 \times 128 \times 8/250 = 0.214mJ$ and $E_r = 3.0 \times 19.4 \times 128 \times 8/250 = 0.466mJ$ to be transmitted and received, respectively;

- The dominant operation to verify a vBNN-IBS signature is three point multiplications, and the resulted computational energy consumption of a sensor node is $E_{MUL} = 3.0 \times 8.0 \times 0.81 = 19.44mJ$ and $E_{ver} = 58.32mJ$ to compute a scalar elliptic curve multiplication and to verify a vBNN-IBS signature, respectively.

In our scheme, some sensor nodes need to release their intermediate computation results in order to accelerate the signature verification for their neighboring nodes. Hence, our scheme consumes more energy for transmitting the intermediate computation result $l_2R+l_3P_0$, when compared to using the traditional vBNN-IBS signature verification in WSNs. In the 4×4 grid-based WSN, the six red nodes (i.e., Nodes 1, 2, 6, 7, 11, 12) will locally broadcast their intermediate computation results to their one-hop neighbors, which causes an extra energy consumption of $6 \times 0.214 = 1.284mJ$ in the network.

Note that although some sensor nodes (e.g., node 6) has four one-hop neighbors, only two of them (i.e., nodes 7 and 10) will receive the intermediate computation results since the other two (i.e., nodes 2 and 5) have finished the signature verification in the previous round (i.e., Round 1) and gone into the power-saving sleep mode. Therefore, there are totally 11 sensor nodes receiving the intermediate computation results (2, 3, 5, 6, 7, 8, 10, 11, 12, 15, 16), which causes an extra energy consumption of $11 \times 0.466 = 5.126mJ$ in the WSN. In brief, our faster signature verification incurs an extra energy consumption of $1.284 + 5.126 = 6.410mJ$ for transmitting (i.e., sending and receiving) the intermediate computation results for the WSN in question.

At the same time, the signature verification on Nodes 2, 3, 5, 6, 7, 8, 10, 11, 12, 15 and 16 will be accelerated by 66% (i.e., saving two elliptic curve scalar multiplications) due to the use of the intermediate computation results

from their neighboring nodes, which leads to a significant energy saving of $11 \times 2 \times 19.44mJ = 427.68mJ$ in the WSN, as compared to the traditional vBNN-IBS signature verification technique.

To sum up, for the broadcast authentication in the target 4×4 grid-based WSN, our faster signature verification can save the energy consumption of $427.68 - 6.410 = 421.27mJ$ in total, considering both the communication and computation overheads. Therefore, using the accelerated vBNN-IBS scheme, one can save up to $421.27mJ \times 100/(16 \times 58.32) = 45.15\%$ energy consumption for the grid-based WSN in question.

Following the same analysis procedure as above, we find that the accelerated ECDSA scheme can achieve a theoretical energy gain of about $206.99mJ \times 100/1244.16 = 16.64\%$.

7.3. Simulation Study

In this section, we will determine, by simulation, the achievable energy gain by the accelerated approach when applied on the vBNN-IBS scheme. The obtained results will be compared with those of the theoretical study.

7.3.1. First Stage

Recall that the CC2420 radio module is able to transmit up to $128bytes$ per packet. As the size of a message of form (4) is $139bytes$, two packets are needed to be transmitted. To this end, we implemented two new 802.15.4 packet structures in TinyOS; one containing the user's message and the generated signature, and the second containing the intermediate computation result $l_2R + l_3P_0$. Let SIG and $INTER$ denote the two aforementioned packets. The SIG and $INTER$ packets are $98bytes$ and $56bytes$, respectively. To measure the energy consumed during the emission and reception of the two packets, we programmed a NesC application for sending and receiving 802.15.4 packets. The application was run using two payload sizes corresponding to the sizes of SIG and $INTER$. We found that SIG packet consumes $E_s(SIG) = 491.4\mu J$ and $E_r(SIG) = 598\mu J$ to be transmitted and received, respectively. In the other hand, we found that $INTER$ packet consumes $E_s(INTER) = 387,1\mu J$ and $E_r(INTER) = 467\mu J$ for its transmission and reception, respectively. Note that the obtained values reflect the average of 10 independent runs for each payload size.

The scalar point multiplication can use several methods and optimization techniques for its calculation [33]: The Affine Coordinate System (ACS), the Projective Coordinate System (PCS), the Affine Coordinate System with

Sliding Window ($ACS\ with\ SW$), and the Projective Coordinate System with Sliding Window ($PCS\ with\ SW$). We tested and evaluated the execution time as well as the required memory for each method, in order to decide which of them will be used in our implementation. When the sliding window technique is used, the window size is fixed to $w = 15$. In fact, we chose to perform two scalar point multiplications for each method: The selected point is the generator G, while the $168bits$ scalar takes two possible values $(000\cdots001)$ and $(111\cdots111)$. By doing so, we obtain a time interval allowing us to estimate the average execution time of a scalar point multiplication. The results are summarized in Table 1.

	ACS (ms)	PCS (ms)	$ACS\ with\ SW$ (ms)	$PCS\ with\ SW$ (ms)
$(000\cdots001)$	21566	1619	21186	1482
$(111\cdots111)$	41568	3217	26027	1882

Table 1: Execution time of a scalar point multiplication

We can clearly observe from Table 1 that the $PCS\ with\ SW$ method is the fastest. However, we have also taken care to evaluate the required memory as demonstrated in Table 2. The results show that the fastest method is also the most expensive in terms of RAM consumption.

	ACS	PCS	$ACS\ with\ SW$	$PCS\ with\ SW$
Required memory size (Octets)	64	104	667	707

Table 2: Required memory for a scalar point multiplication

Given the limited size of MICAz sensor nodes' memory, we decided to use the $PCS\ with\ SW$ method but reducing the sliding window size to $w = 3$. In this way, the required memory will be reduced to $341octets$. The execution time of the scalar point multiplication becomes (See Table 3):

The obtained results show that using the $PCS\ with\ SW$ method while reducing the SW size from 15 to 3 leads to a scalar point multiplication slowdown of about 16.4% but with a space saving of 51.71%. Hence, the method ensures a satisfactory compromise between execution time and required memory size. From the simulation results, the average execution time of a scalar point multiplication is estimated to be $1958ms$.

	ACS with SW (ms)	PCS with SW (ms)
$(000\cdots001)$	21442	1558
$(111\cdots111)$	31270	2358

Table 3: Execution time of a scalar point multiplication using SW with $w = 3$

To measure the energy consumed by the calculation of an elliptic point multiplication, we programmed a NesC application to calculate k consecutive scalar point multiplications. The application was run by varying the value of k from 5 to 20, and then the power consumed by one scalar point multiplication was estimated. The simulation results reflect the average of 10 executions for each value of k. We found that a scalar point multiplication consumes $51.795mJ$.

To sum up, the simulation study revealed that the emission of SIG (resp. $INTER$) packet consumes $491,4\mu J$ (resp. $387.1\mu J$), its reception consumes $598\mu J$ (resp. $467\mu J$), and that a scalar point multiplication takes an average execution time of $1958ms$ and consumes $51.795mJ$ using a $PCS\ with\ SW$ ($w = 3$) method.

Based on the obtained values, we conducted the same diffusion scenario as in the theoretical study in order to estimate the energy gain achieved by the acceleration approach when applied on both vBNN-IBS and ECDSA schemes. Table 4 reports the total energy consumed by the different schemes to perform an authenticated diffusion in the target 4×4 grid-based WSN.

	vBNN-IBS	Acc. vBNN-IBS	ECDSA	Acc. ECDSA
Energy (mJ)	2503	1371	3343.52	2783.3

Table 4: The total energy consumed by the network in the first stage (simulation results)

Accelerated vs. traditional vBNN-IBS We found that the acceleration approach allows the vBNN-IBS scheme to save up to 45.53% of the total energy required for the broadcast authentication in the target 4×4 grid-based WSN. Recall that in the theoretical study, the accelerated scheme allows a node to accelerate its signature verification by 66% compared to the classical scheme. The simulation results show that

the signature verification is achieved in $6699ms$ using the traditional vBNN-IBS scheme and in $2228ms$ using the accelerated vBNN-IBS scheme. Thus, we deduce that the accelerated scheme allows some nodes to perform the signature verification 66.74% faster than in the traditional scheme.

Accelerated vs traditional ECDSA We found that the acceleration approach allows saving up to 16.90% of the total energy, required by the traditional ECDSA scheme. The simulation results show that the signature verification is achieved in $6863ms$ using the traditional ECDSA scheme and in $4797ms$ using the accelerated ECDSA scheme.

Analysis We notice that the energy gains achieved by vBNN-IBS and ECDSA in both theoretical and simulation studies differ slightly. This is mainly due to the energy consumed by the multiplication operation, considered in the theoretical study to be $19.44mJ$, given an execution time of $0.81seconds$. However, we demonstrated, by simulation, that the execution time of such an operation is in average equal to $1958ms$ and cannot go below $1400ms$ (with a SW of size 15). So, if we consider the average execution time obtained by simulation in the theoretical study, we will obtain a theoretical energy gain of 45.55% for vBNN-IBS and 16.97% for ECDSA. Now, the theoretical and simulation results become fairly close.

The simulation study demonstrates that the impact of the acceleration approach on the vBNN-IBS scheme is better than on the ECDSA scheme. The obtained results showed an energy gain of 45.53% using the accelerated vBNN-IBS scheme compared to only 16.90% when the accelerated ECDSA scheme is applied. This difference can be explained by the fact that in the ECDSA scheme, the acceleration is applied only on the signature and not on the certificate. Moreover, the accelerated vBNN-IBS scheme saves two elliptic curve scalar multiplications, while the accelerated ECDSA scheme saves only one elliptic curve scalar multiplication.

7.3.2. Second Stage

In the complete simulation, all operations and node's states are taken into account; this is the simulation of the entire program. The aforementioned diffusion scenario was performed using vBNN-IBS, ECDSA, and their accelerated versions. The energy consumed by each node was recorded, and then

the total energy consumed within the network was computed. The diffusion time was also recorded. Table 5 reports the obtained results. Note that the results were averaged over 10 independent runs.

	vBNN-IBS	Acc. vBNN-IBS	ECDSA	Acc. ECDSA
Energy (J)	42.79	27.26	43.46	36.49
Time (s)	50.30	25.17	53.87	39.60

Table 5: The total energy consumed by the network in the second stage (simulation results)

According to Table 5, the simulation results show that the acceleration approach speeds up the diffusion time by 49.96% in the case of vBNN-IBS and by 26.49% in the case of ECDSA. As a result, the accelerated version of vBNN-IBS (resp. ECDSA) is capable of saving up to 36.29% (resp. 16.04%) of the total power required by its traditional version. Comparing these results with those of the first stage, we can see that lower energy savings are gained when the remaining operations and node's states are taken into account. An interesting observation is that the remaining operations and node's states constitutes a significant fraction (more than 92%) of the total energy dissipated within the network. Therefore, neglecting this important fraction of consumed energy has resulted in an over-estimated energy gain in the first stage. Previous studies [34, 35, 36, 37] have demonstrated that the idle state dominates the energy usage in WSNs. Even the standby mode tries to reduce energy wastage by turning off radio and putting CPU into sleep mode, a node operating in this mode still consume energy. The power consumed in idle and standby states increases over time. Thus, more the authenticated diffusion takes time; more the energy will be drained.

To better assess the impact of both standby and idle states on performance, the energy wasted by a node operating in these two states was determined. Table 6 presents the energy breakdown, by components, of a MICAz node. We found that a node consumes an average of $66.49mJ/s$ when operating in idle mode and $4.44mJ/s$ when operating in standby mode. The results show that the energy consumption in the idle state is dominated by the radio.

Using the energy values determined by the complete simulation, we estimated the network lifetime as follow:

If we assume that nodes are powered by two AA batteries having a ca-

	CPU	Radio	Sensor board	Flash Memory	Total
Idle (mJ/s)	10.62	53.76	2.1	0.006	66.49
Standby (mJ/s)	2.33	0.0006	2.1	0.006	4.44

Table 6: Simulated energy breakdown of a MICAz mote operating in idle and standby modes

pacity of $2450mAh$, the considered 4×4 grid-based network will initially have an energy equal to $16 * ((2450 * 3 * 3600)/1000) = 423360J$. During experiments, we noticed that MICAz motes stop operating correctly when their power source drops below $2.1Volt$. In other words, when the power source reaches $18522J$, the node cannot function properly. Consequently, we have considered that the network is exhausted when its energy reaches the $296352J$.

In addition, we assume that an authenticated diffusion is performed *once every minute*. At the beginning of each diffusion interval, all nodes wake-up from standby state and move to idle state waiting for incoming authentication packets, as shown in Figure 4:

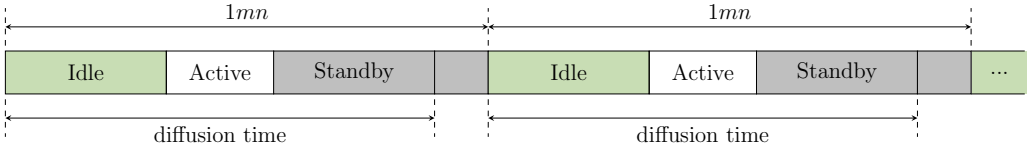


Figure 4: The sensor node's states when an authenticated diffusion is performed once every minute

By knowing the duration of an authenticated diffusion (i.e. *diffusion time*), the total energy spent to perform this diffusion within the network as well as the energy wasted per second in standby state, the total energy spent per minute can be deduced. For example, in the traditional vBNN-IBS scheme, the network consumes $42.79Joule$ during $50.3s$ and $0.6896J$ during the remaining $9.7s$; thus, a total of $43.48J/mn$. The total energy spent per minute for vBNN-IBS, ECDSA, and their accelerated versions is reported in Table 7.

Using the above results, the network lifetime according to the number of authenticated diffusions performed, before the network is exhausted, is depicted in Figure 5. The figure shows that the accelerated schemes extend

	vBNN-IBS	Acc. vBNN-IBS	ECDSA	Acc. ECDSA
Energy (J/mn)	43.48	29.66	43.89	37.93

Table 7: The total energy consumed by the network to perform one authenticated diffusion per minute

more the network lifetime compared to their traditional versions. Furthermore, the accelerated vBNN-IBS scheme achieves the longest lifetime by allowing for 4282 diffusions before the exhaustion of the network.

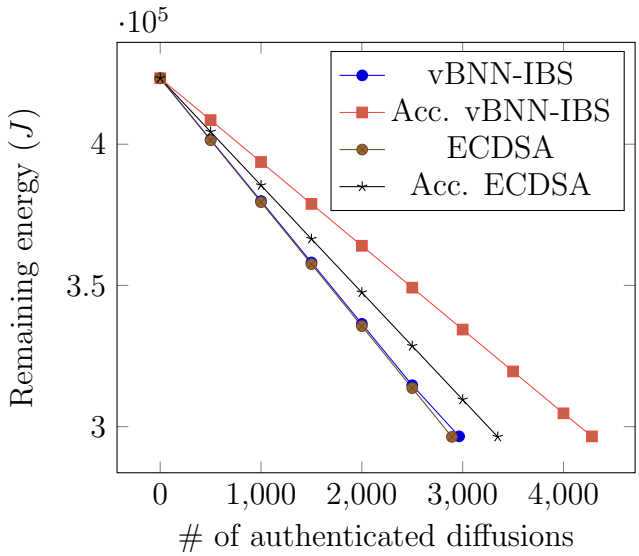


Figure 5: Network lifetime vs. the number of authenticated diffusions

7.4. Experimental Study

In this section, we will show the achievable energy gain by the vBNN-IBS and ECDSA schemes using the real-world experimentation. The obtained results will be compared with those obtained by both the theoretical analysis and simulation.

7.4.1. First Stage

We first determined the power draw for the signature verification, transmission, as well as reception operations. The measurement setup is schematically depicted in Figure 6. It consists of a MICAz mote connected in series

to a 10Ω resistor. The circuit is powered by two AA rechargeable batteries with a supply voltage of $3V$. Using a $GDS-3352$ digital storage oscilloscope, the voltage drop over the resistor was measured and used to calculate the current flow based on Ohm's law. We found that the current draw when a node is performing a signature verification is $27.6mA$, the receiving current draw is $25.53mA$, and the transmitting current draw is $26.40mA$.

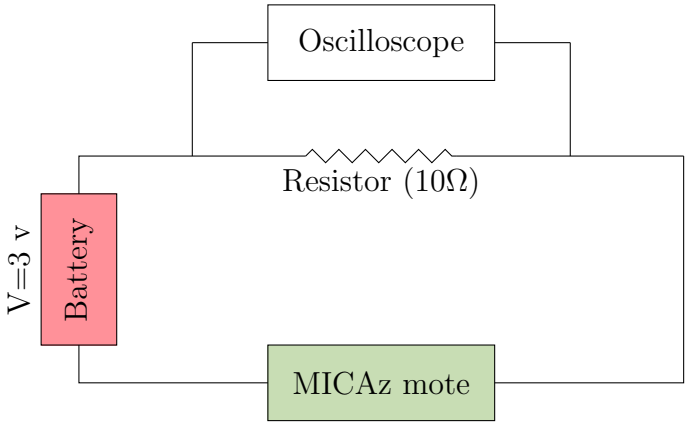


Figure 6: MICAz power collection setup

To compute the electrical energy consumed by a sensor node during t seconds, we apply the Joule's law as shown in equation 8.

$$E_{(Joule)} = V_{(Volt)} * I_{(Ampere)} * t_{(seconds)} \tag{8}$$

Where E, V, and I are respectively the electrical energy, the supply voltage, and the current intensity. The oscilloscope allowed us to determine the time taken to transmit SIG, $INTER$, and $CERTIF$ packets, and to execute a scalar point multiplication. it took for the different algorithms to execute. The measured times are presented in Table 8.

Based on the obtained values, we conducted the same diffusion scenario as the in the theoretical and simulation studies in order to estimate the energy gain achieved by the acceleration approach when applied on both vBNN-IBS and ECDSA schemes. Table 9 summarizes the total energy consumed by the different schemes to perform an authenticated diffusion in the target 4×4 grid-based WSN.

	time
Tx of SIG (ms)	3.552
Tx of $INTER$ (ms)	2.208
Tx of $CERTIF$ (ms)	3.232
Point multiplication (s)	1.95

Table 8: Duration of transmitting SIG, $INTER$, and $CERTIF$ packets and executing a scalar point multiplication (Experimental results)

	vBNN-IBS	Acc. vBNN-IBS	ECDSA	Acc. ECDSA
Energy (J)	7.75	4.202	10.35	8.57

Table 9: The total energy consumed by the network in the first stage (experimental results)

Accelerated vs. traditional vBNN-IBS We found that the acceleration approach allows the vBNN-IBS scheme to save up to 45.78% of the total energy required for the broadcast authentication in the target 4×4 grid-based WSN. The experimental results show also that the signature verification is achieved in $6.69s$ using the traditional vBNN-IBS scheme and in $2.2s$ using the accelerated vBNN-IBS scheme. Thus, we deduce that the accelerated scheme allows some nodes to perform the signature verification 67.11% faster than in the traditional scheme.

Accelerated vs. traditional ECDSA We found that the acceleration approach allows saving up to 17.20% of the total energy, required by the traditional ECDSA scheme. The experimental results show that the signature verification is achieved in $6.869s$ using the traditional ECDSA scheme and in $4.72ms$ using the accelerated ECDSA scheme. Thus, we deduce that the accelerated scheme allows some nodes to perform the signature verification 31.28% faster than in the traditional scheme.

Analysis The comparison of results presented in Tables 4 and 9 shows that the total consumed energy measured by the real-world experimentation is roughly 3 times bigger than that measured by simulation. This difference is mainly due to the current draw recorded during the signature verification. The value measured by simulation was that of the CPU component only, however the value found by the real-word experimen-

tation was that of all mote's components including the radio which was turned on during the signature verification.

We notice that the experimental results in terms of energy gain are consistent with both theoretical and simulation results. The experimental results promote once again the application of the acceleration approach on vBNN-IBS rather than ECDSA.

7.4.2. Second Stage

In this stage, the energy gain achieved by the execution of the entire program was evaluated. We first determined the power draw for idle and standby states. Using the afore-described measurement setup, we found that the current draw of a MICAz mote in idle state is $22mA$. Regarding the standby state, the current draw was very low to be shown on the oscilloscope. So, we used a Fluke 15B digital multimeter which reports a current draw of $2\mu A$ for a MICAz mote in standby mode.

After that, we performed a lab experiment where the behavior of each mote were analyzed during the diffusion time. We distinguish two types of behavior depending on whether the signature verification is accelerated or not. The measurement setup described in the first stage was used to measure the execution time of each operation (i.e. transmission, reception, and signature verification) and state (i.e. idle and standby) for a given mote during the diffusion time in the 4×4 grid-based network. Table 10 reports the total execution time taken by the different operations and states in the entire network (i.e. by the 16 motes).

Operation or mode	vBNN-IBS (s)	Acc. vBNN-IBS (s)	ECDSA (s)	Acc. ECDSA (s)
Standby mode	373.02	162.355	417.03	284.6
Idle mode	324.58	182.72	335.13	263.09
Emission	$53.5 * 10^{-3}$	$48.76 * 10^{-3}$	$54.38 * 10^{-3}$	$75.68 * 10^{-3}$
Reception	$71.03 * 10^{-3}$	$95.31 * 10^{-3}$	$108.79 * 10^{-3}$	$147.8 * 10^{-3}$
Signature verification	107.04	57.65	109.904	75.68

Table 10: The total execution time taken by each operation and state during an authenticated diffusion in the target 4×4 grid-based network

Knowing the current draw and the total execution time for a each operation and state, the total energy consumed in the entire network can be computed by applying the Joule's law. The Table 11 summarizes the obtained results.

Operation or mode	vBNN-IBS (J)	Acc. vBNN-IBS (J)	ECDSA (J)	Acc. ECDSA (J)
Standby mode	$2.238 * 10^{-3}$	$0.974 * 10^{-3}$	$2.502 * 10^{-3}$	$1.708 * 10^{-3}$
Idle mode	21.42	12.06	22.12	17.36
Emission	$2.813 * 10^{-3}$	$3.862 * 10^{-3}$	$4.307 * 10^{-3}$	$5.994 * 10^{-3}$
Reception	$5.44 * 10^{-3}$	$7.30 * 10^{-3}$	$8.332 * 10^{-3}$	$11.32 * 10^{-3}$
Signature verification	8.863	4.773	9.100	7.124
Total	30.29	16.84	31.23	24.50

Table 11: The total energy consumed by the network in the second stage (Experimental results)

The comparison of results presented in Tables 11 and 5 shows that the total consumed energy measured by simulation is roughly 1.48 bigger than that measured by the real-world experimentation. This difference is due to several considerations: First, the startup and shutdown energy of the radio as well as the energy wasted in transition between the different states were not accounted in the total energy found by the experimental study. Second, the standby current draw recorded by simulation was $1.48mA$ against $2\mu A$ recorded by the real-world experimentation.

According to Table 11, the real-world experimentation results show an energy gain of 44.40% (resp. 21.55%) when the acceleration approach is applied on vBNN-IBS (resp. ECDSA). The experimental results promote once again the application of the acceleration approach on vBNN-IBS rather than ECDSA.

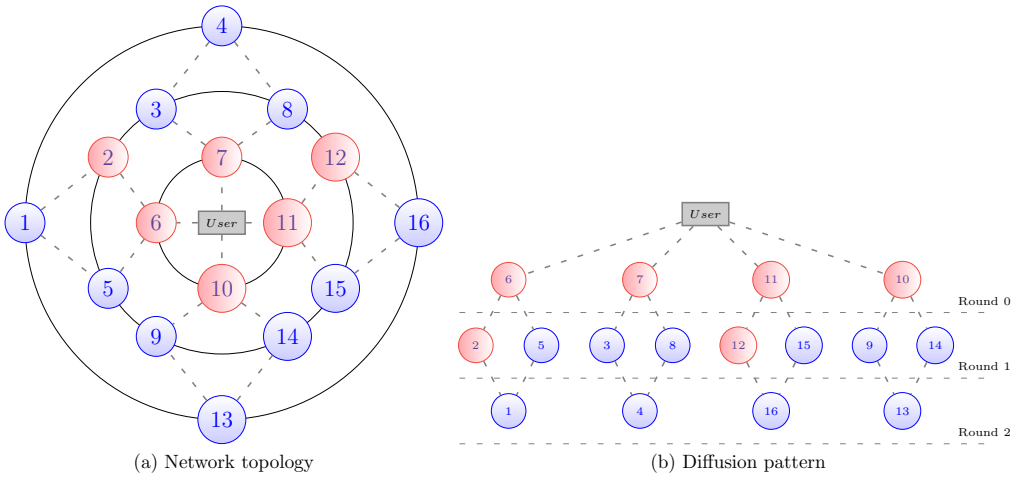
8. Impact of Network Topology on Performance

In this section, we investigate the impact that a network topology can have on the performance of our scheme in terms of energy saving and diffusion latency. In order to compare with the results obtained above, the network size and the number of nodes selected to release the intermediate values are kept the same as in the 4×4 grid-based topology. Two topologies are defined, namely:

Circular topology in which the user is at the center of the topology and the 16 sensor nodes are placed around the user as shown in Figure 7(a). A user broadcasts its signed packet to nodes 6, 7, 10, and 10 at Round 0. After two other communication rounds according to the diffusion pattern depicted in Figure 7(b), the broadcast packet will be received and verified by all sensor nodes. We assume that nodes 2, 6, 7,10,

11, and 12 are selected to release the intermediate computation result $l_2R + l_3P_0$ in order to accelerate the signature verification.

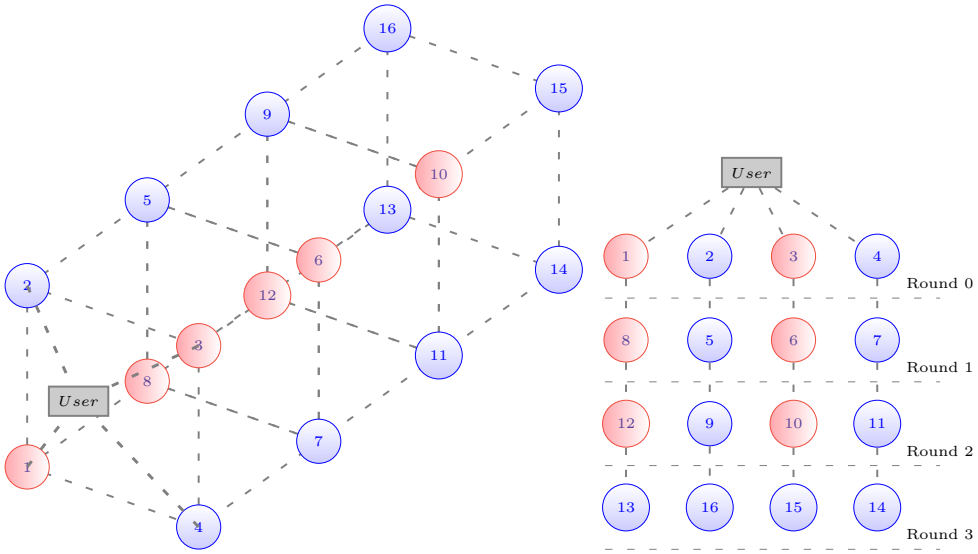
3D grid-based topology where the 16 nodes are placed on the vertexes of three consecutive cubes as shown is Figure 8(a). A user broadcasts its signed packet to nodes 1, 2, 3, and 4 at Round 0. After three other communication rounds following the diffusion pattern depicted in Figure 8(b), the broadcast packet will be received and verified by all sensor nodes in the network. Now, we assume that the six nodes selected to release the intermediate result $l_2R + l_3P_0$ are nodes 1, 3, 6, 8, 10, and 12.



(a) Network topology (b) Diffusion pattern

Figure 7: The circular topology and the corresponding diffusion pattern

Comparing these topologies to the 4×4 grid-based topology, we can draw the following observations: (1) The number of nodes accelerating their signature verification decreases. While the 4×4 grid-based topology allows to 11 nodes to accelerate their signature verification, 10 nodes (i.e., nodes 1, 2, 3, 5, 8, 9, 12, 14, 15, and 16) are able to benefit from the acceleration process in the circular topology and only 6 nodes (i.e., nodes 6, 8, 10, 12, 13, and 15) in the case of 3D grid-based topology. (2) The number of communication rounds decreases also. The diffusion in the entire network is made in 3 rounds with the circular topology and 4 rounds with the 3D grid-based topology, instead of 7 rounds in the case of 4×4 grid-based topology.



(a) Network topology (b) diffusion pattern

Figure 8: The 3D grid-based topology and the corresponding diffusion pattern

These differences, either in number of nodes accelerating their signature verification or number of communication rounds, will certainly impact both the energy consumption and the diffusion time. To assess this impact, the vBNN-IBS and its accelerated versions were simulated on both topologies using Avrora [31] and considering the two stages; that is, the first stage where only the three dominant energy consuming operations are considered and the second stage where all operations and node's states are taken into consideration. Furthermore, the same diffusion scenario as the one described in Section 7.1 is used.

8.1. Scheme's Performance under Circular Topology

In the first stage, we found that vBNN-IBS consumes a total energy of $2506.41mJ$ while the accelerated version consumes $1477.5mJ$, leading to save up to 41% of the total energy required for the broadcast authentication in the entire network. The accelerated scheme still allows some nodes to perform the signature verification 66.74% faster than in the traditional scheme.

In the second stage, where all operations and node's states are taken into account, the entire program was simulated and the total energy consumed

within the network as well as the diffusion time were recorded. Table 12 reports the obtained results.

	vBNN-IBS	Acc. vBNN-IBS
Energy (J)	23.48	20.9
Time (s)	25.36	20.96

Table 12: The total energy consumed by the network in the second stage when the circular topology is used

According to Table 12, the results show that the acceleration approach speeds up the total diffusion time by 17.33% and saves up to 11% of the total power required by its traditional version. We notice that the energy gain is lower than the one achieved in the first stage. As explained above, this difference is mainly due to the energy consumed by the idle and standby states which represents more than 92% of the total energy dissipated within the network. Indeed, the diffusion is made in only three rounds and all nodes in the second level accelerate their signature verification. Thus, the energy gain is dominated by the time passed by the leaf nodes (i.e., nodes 1, 4, 13, and 16) in the idle mode which is reduced by only $4s$ in the accelerated version. However, it should be noted that the circular topology reduced the total diffusion time by 16.73% and the energy gain by 23.33% compared to the 4×4 grid-based topology when the acceleration approach is applied.

8.2. Scheme's Performance under 3D grid-based Topology

In the first stage, we found that vBNN-IBS consumes a total energy of $2501.62mJ$ while the accelerated version consumes $1885.2mJ$, leading to save up to 24.64% of the total energy required for the broadcast authentication in the entire network. We notice that a lower energy gain is achieved compared to the two other topologies. In fact, this is due to the number of nodes accelerating their signature verification which is equal to 6 nodes only. However, the accelerated scheme still allows some nodes to perform the signature verification 66.74% faster than in the traditional scheme.

In the second stage, the results showed in Table 13 were obtained.

The results in Table 13 show that the acceleration approach speeds up the total diffusion time by only 5.8%. Indeed, the nodes placed on the left upper vertices (i.e., 2, 5, 9, and 16) and those placed on the right lower vertices (i.e, 4,7,11, and 14) have to operate a classical signature verification regardless if

	vBNN-IBS	Acc. vBNN-IBS
Energy (J)	28.9	27.3
Time (s)	31.36	29.54

Table 13: The total energy consumed by the network in the second stage when the 3D grid-based topology is used

the acceleration approach is used or not. In other words, the total diffusion time will be almost the same apart from the execution time of the code on each node. The results show also that the acceleration approach saves up to 5.54% of the total power required by its traditional version. We notice that the diffusion time energy gain is lower than the one achieved in the first stage which is mainly due to the energy consumed by the idle and standby states. As the diffusion time is almost the same and the number of nodes accelerating their signature verification is only 6, the acceleration approach will have a little impact on reducing the amount of energy wasted on idle and standby modes.

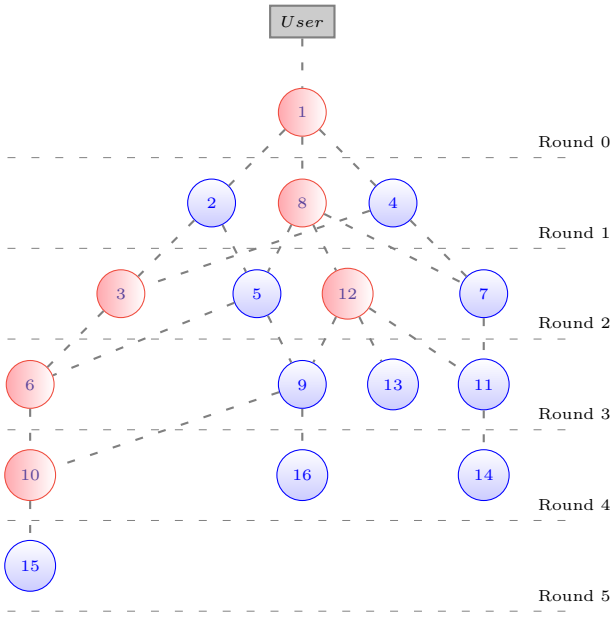


Figure 9: The new diffusion pattern under the 3D grid-based topology

It is clear that the diffusion pattern affects the way energy is wasted in the

network and consequently the achieved performance. To assess this impact, the second stage simulation of the proposed scheme under the 3D grid-based topology was redone, this time following the diffusion pattern depicted in Figure 9. The new diffusion pattern allows to twelve nodes (i.e., 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 15) to accelerate their signature verification. We assume that the user through the base station is able to communicate only with node 1. The total diffusion time and the total energy consumed in the entire network are shown in Table 14.

	vBNN-IBS	Acc. vBNN-IBS
Energy (J)	36.8	28.77
Time (s)	44.83	27.54

Table 14: The total energy consumed by the network in the second stage when the 3D grid-based topology is used and the diffusion pattern in Figure 9 is followed

The results in Table 14 demonstrate that the acceleration approach speeds up the total diffusion time by up to 38.57% and saves up 21.82% of the total power required by its traditional version. Compared to the previous diffusion pattern, the new one reduced the total diffusion time by almost 7%.

From the above results, it is easy to show that the performance is not only affected by the number of nodes releasing intermediate results, but also by the deployment topology of nodes and the diffusion pattern followed to broadcast the user packet in the entire network.

9. Conclusions and Future Work

This paper presented an accelerated verification of vBNN-IBS, a pairing-free identity-based signature with reduced signature size. The acceleration technique aims to reduce the energy consumption and thus extend the network lifetime by decreasing the computation overhead due to signature verification. The proposed technique exploits the cooperation among nodes to speed up the signature verification of nodes receiving a partially-calculated signature which leads to lower energy consumption during the verification process. The performance of the proposed scheme in terms of energy consumption and diffusion latency was evaluated through a theoretical analysis,

simulation, and real-world experimentation using a MICAz platform. The obtained results showed that the acceleration technique allows an average energy gain during the verification phase of around 45% when applied on vBNN-IBS signatures against only 16% when applied on ECDSA signatures. The conducted evaluation studies demonstrated that the acceleration approach reduces the idle energy by speeding up the total diffusion time in the network. In a 4×4 grid-based network, the accelerated vBNN-IBS speeds up the total diffusion time by 49.96% allowing an overall energy gain of about 36% and then a longer network lifetime. Moreover, the total energy cost of the accelerated vBNN-IBS is reduced by more than 25% compared to the accelerated ECDSA scheme which promotes the application of the acceleration approach on ID-based signature schemes.

A performance evaluation under collusive attacks forms the basis of our current work. A future work consists in devising a load-balanced technique to share the burden of releasing intermediate results among all nodes and in finding the adequate diffusion pattern to achieve higher energy saving.

References

[1] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, D. E. Culler, Spins: Security protocols for sensor networks, ACM Wirel. Netw. 8 (5) (2002) 521–534.

[2] D. Liu, P. Ning, Multi-level μtesla: broadcast authentication for distributed sensor networks, ACM Trans. on Embed. Comput. Syst. 3 (4) (2004) 800–836.

[3] P. Ning, A. Liu, W. Du, Mitigating dos attacks against broadcast authentication in wireless sensor networks, ACM Trans. Sen. Netw. 4 (1) (2008) 1–35.

[4] P. Chuchaisri, R. Newman, Fast response pkc-based broadcast authentication in wireless sensor networks, Mobile Netw. and Applicat. 17 (4) (2012) 508–525.

[5] X. Fan, G. Gong, Accelerating signature-based broadcast authentication for wireless sensor networks, Ad Hoc Netw. 10 (2012) 723–736.

[6] A. Wander, N. Gura, H. Eberle, V. Gupta, S. Shantz, Energy analysis of public-key cryptography on small wireless devices, In Proc. of IEEE PerCom (2005) 324–328.

[7] K. Ren, K. Zeng, W. Lou, P. Moran, On broadcast authentication in wireless sensor networks, IEEE Trans. Wirel. Commun. 6 (11) (2007) 4136–4144.

[8] K. Ren, S. Y, W. Lou, Y. Zhang, Multi-user broadcast authentication in wireless sensor networks, IEEE Trans. on Veh. Technol. 58 (8) (2009) 4554–4564.

[9] X. Cao, W. Kou, L. Dang, B. Zhao, Imbas: Identity-based multi-user broadcast authentication in wireless sensor networks, Comput. Commun. 31 (2008) 659–667.

[10] K.-A. Shim, Y.-R. Lee, C.-M. Park, $\mathcal{EIBAS}$: An efficient identity-based broadcast authentication scheme in wireless sensor networks, Ad Hoc Netw. 11 (1) (2013) 182–189.

[11] R. C. Merkle, Protocols for public key cryptosystems, In Proc. of IEEE Symp. on Research in Security and Privacy (1980) 122–134.

[12] D. J. Johnson, A. J. Menezes, S. A. Vanstone, The elliptic curve digital signature algorithm (ecdsa), Int. J. of Inform. Security 1 (1) (2001) 36–63.

[13] A. Shamir, Identity-based cryptosystems and signature schemes, In Proc. of CRYPTO 84 on Advances in Cryptology (1985) 47–53.

[14] D. Liu, P. Ning, S. Zhu, S. Jajodia, Practical broadcast authentication in sensor networks, In Proc. of the 2nd Annu. Int. Conf. on Mobile and Ubiquitous Syst.: Networking and Services (2005) 118–129.

[15] A. Perrig, J. D. Tygar, D. Song, R. Canetti, Efficient authentication and signing of multicast streams over lossy channels, In Proc. of the IEEE Symposium on Security and Privacy (SP'00) (2000) 56–73.

[16] A. Perrig, The biba one-time signature and broadcast authentication protocol, In Proc. of the 8th ACM Conference on Computer and Communications Security (CCS'01) (2001) 28–37.

[17] M. Mitzenmacher, A. Perrig, Bounds and improvements for biba signature schemes, Technical Report TR-02-02, Computer Science Group, Harvard University.

[18] L. Reyzin, N. Reyzin, Better than biba: Short one-time signatures with fast signing and verifying, In Proc. of the 7th Australian Conference on Information Security and Privacy (ACISP'02) (2002) 114–153.

[19] S.-M. Chang, S. Shieh, W. W. Lin, C.-M. Hsieh, An efficient broadcast authentication scheme in wireless sensor networks, In Proc. of the 2006 ACM Symposium on Information, Computer and Communications Security (ASIACCS'06) (2006) 311–320.

[20] J. Lee, S. Kim, Y. Cho, Y. Chung, Y. Park, Horsic: An efficient one-time signature scheme for wireless sensor networks, Information Processing Letters 112 (20) (2012) 783–787.

[21] B. H. Bloom, Space/time trade-offs in hash coding with allowable errors, Communications of the ACM 13 (7) (1970) 422–426.

[22] D. Naccache, J. Stern, Signing on a postcard, In Proc. of the 4th International Conference on Financial Cryptography (2000) 121–135.

[23] F. Hess, Efficient identity based signature schemes based on pairings, In Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography (SAC'02) (2002) 310–324.

[24] M. Bellare, C. Namprempre, G. Neven, Security proofs for identity-based identification and signature schemes, In Advances in Cryptology-EUROCRYPT'2004, LNCS 3027 (2004) 268286.

[25] D. Hankerson, A. Menezes, S. Vanstone, Guide to elliptic curve cryptography, Springer.

[26] J. Elson, D. EstrinBirman, Fine-grained network time synchronization using reference broadcast, In Proc. of the 5th Symp. on Oper. Syst. Design and Implementation (OSDI) (2002) 147–163.

[27] M. Maroti, B. Kusy, G. Simon, A. Ledezi, The flooding synchronization protocol, In Proc. of the 2nd ACM Conf. on Embedded Networked Sensor Syst. (SenSys).

[28] C. Benzaid, A. Saiah, N. Badache, An enhanced secure pairwise broadcast time synchronization protocol in wireless sensor networks, In Proc. of the 22nd Euromicro International Conference on Parallel, Distributed, and Network-based Processing (PDP'2014) (2014) 569–573.

[29] J. R. Douceur, The sybil attack, In Revised Papers from the First International Workshop on Peer-to-Perr Systems (IPTPS'01) (2002) 251–260.

[30] A. D. Wood, J. A. Stankovic, Denial of service in sensor networks, Computer 35 (10) (2002) 54–62.

[31] B. L. Titzer, D. K. Lee, J. Palsberg, Avrora: Scalable sensor network simulation with precise timing, In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN'05) (2005) Article 67.

[32] N. Gura, A. Patel, A. Wander, H. Eberle, S. C. Shantz, Comparing elliptic curve cryptography and rsa on 8-bit cpus, In Proc. of the 6th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2004) (2004) 119–132.

[33] A. Liu, P. Ning, Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks, In Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08) (2008) 245–256.

[34] M. Stemm, R. H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, IEICE Transactions on Telecommunications E80-B (8) (1997) 1125–1131.

[35] T. Dam, K. Langendoen, An adaptive energy-efficient mac protocol for wireless sensor networks, In Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys) (2003) 171–180.

[36] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking 12 (3) (2004) 493–506.

[37] G. Anastasi, A. Falchi, A. Passarella, M. Conti, E. Gregori, Performance measurements of motes sensor networks, In Proc. of the 7th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (2004) 174–181.