PAPER *Special Section on Trust, Security and Privacy in Computing and Communication Systems*

An Authentication Framework for Wireless Sensor Networks using Identity-Based Signatures: Implementation and Evaluation*

Rehana YASMIN[†a)], **Eike RITTER**[†b)], ***and*** **Guilin WANG**[††c)],

SUMMARY In Wireless Sensor Networks (WSNs), authentication is a crucial security requirement to avoid attacks against secure communication, and to mitigate against DoS attacks exploiting the limited resources of sensor nodes. Resource constraints of sensor nodes are hurdles in applying strong public key cryptographic based mechanisms in WSNs. To address the problem of authentication in WSNs, we propose an efficient and secure framework for authenticated broadcast/multicast by sensor nodes as well as for outside user authentication, which utilizes identity based cryptography and online/offline signature (OOS) schemes. The primary goals of this framework are to enable all sensor nodes in the network, firstly, to broadcast and/or multicast an authenticated message quickly; secondly, to verify the broadcast/multicast message sender and the message contents; and finally, to verify the legitimacy of an outside user. This paper reports the implementation and experimental evaluation of the previously proposed authenticated broadcast/multicast by sensor nodes scheme using online/offline signature on TinyOS and MICA2 sensor nodes.

key words: *Wireless Sensor Network, Authentication, Online/Offline Signatures*

1. Introduction

Authentication in Wireless Sensor Networks (WSNs) can be divided into three categories, namely base station to sensor nodes, sensor nodes to other sensor nodes, and outside users to sensor nodes. The problem of authenticated broadcast by the base station has been widely addressed [**?**], [**?**], [**?**], [**?**,]. We focus on the other two categories, namely the authenticated broadcast/multicast by the sensor nodes and the outside user authentication.

To handle these two problems, we proposed an authentication framework for WSNs in [**?**,] using Identity(ID)-based Cryptography [**?**,] and Online/Offline Signature (OOS) [**?**,] schemes. This framework is comprised of two authentication schemes; quick *authenticated broadcast/multicast by sensor nodes* and *outside user authentication*. The first scheme allows every sensor node in the network to broadcast or multicast authenticated messages very quickly without the involvement of the base station. All potential receivers can verify a message sent by any sender node in the network. It also allows sensor nodes on the path from the sender node to the receivers to verify a valid message and drop false injected data. The second scheme enables all sensor nodes in the network to verify the legitimacy of any outside user without storing any user specific information. It allows a maximum possible number of legitimate users to access data from sensor nodes in a secure way. This scheme first authenticates a user and then establishes a session key for the secure exchange of user queries and sensor nodes data. The proposed framework uses an ID-based Online/Offline Signature (IBOOS) (an ID-based version of OOS) for the first scheme and an ID-based Signature (IBS) for the second scheme.

In this paper, we present our implementation and performance evaluation of a few suitable ID-based Online/Offline Signature schemes on the TinyOS operating system and MICA2 [**?**,] sensor nodes. To the best of our knowledge, the proposed framework [**?**,] was the first proposal of using IBOOS schemes in WSNs. A detailed discussion regarding security and performance of applying an IBOOS scheme in WSNs is already given in [**?**,], leaving practical implementation of IBOOS on sensor nodes as future work. We implemented and analyzed the performance of IBOOS schemes with respect to computation cost, memory usage, and signature size. The analysis confirms the suitability of our proposed authenticated broadcast by sensor nodes scheme using IBOOS for resource constrained sensor nodes.

1.1 ID-based Online/Offline Signature (IBOOS)

An Online/Offline Signature (OOS) scheme divides the process of message signing into two phases, the *Offline* phase and the *Online* phase. The *Offline* phase is performed before the message to be signed becomes available. This phase performs most of the computations of signature generation and results in a partial signature. Once the message is known, the *Online* phase starts. This phase retrieves the partial signature calculated during the *Offline* phase and performs some minor quick computations to obtain the final signature. The *Online* phase is assumed to be very fast, consisting of small computations while the *Offline* phase can be performed by any other resourceful device. IBOOS is the ID-based version of OOS, where a message signed with a signer's private key is verified using the signer's ID. In ID-based cryptography [**?**,], the signer's private key corresponding to his ID is generated by a private key generator (PKG). IBOOS enables a resource constrained sensor node to sign a message quickly, once it has some critical event to report. Moreover, some IBOOS schemes, like [**?**,], allow

[†]The author is with the University of Birmingham

[††]The author is with the University of Wollongong

*A preliminary version of this paper appeared in *Proc. Computer and Information Technology, CIT'10, pages 882-889, 2010*

a) E-mail: R.Yasmin@cs.bham.ac.uk

b) E-mail: E.Ritter@cs.bham.ac.uk

c) E-mail: guilin@uow.edu.au

DOI: 10.1587/trans.E0.??.1

to reuse the partial signature computed in the offline phase to sign more than one message, which decreases the energy consumption on sensor nodes.

1.2 Motivations

There are many critical situations where a sensor node needs to broadcast or multicast a quick authenticated message. Consider the military application scenario discussed in [**?**,], where a troop of soldiers needs to move through a battlefield. Sensor nodes deployed there detect the presence of the enemy and broadcast this information authentically as soon as possible throughout the network. Soldiers, passing near these sensor nodes, use this information to strategically position themselves in the battlefield. In a forest fire alarm application [**?**,], sensor nodes deployed in a forest immediately inform authorities about the event and the exact location of the event before the fire spreads uncontrollably. The receiver of these messages may be a powerful device or another sensor node in the network. Message authentication is required for all these applications otherwise an adversary can exploit the situation and send a fake message, for instance, regarding the location of the enemy or cause a fake fire alarm [**?**,].

However, the problem of authenticated broadcast or multicast by the sensor nodes has not been addressed by the existing authentication schemes for WSNs. Symmetric authentication schemes for WSNs, for instance, μTESLA [**?**,] and its variations, use Message Authentication Code (MAC) and are efficient in terms of processing time and energy consumption. However, they suffer from certain issues described in [**?**,] and do not provide a solution. They mainly focus on providing base station to sensor nodes authentication. Asymmetric schemes using digital signatures overcome the problems of symmetric schemes, nevertheless, it is more time consuming for the sensor nodes to sign and verify a message than to compute a MAC. Hence, they do not provide an ideal solution for time critical applications.

The proposed authenticated broadcast by sensor nodes scheme uses IBOOS which allows a sensor node to perform the most time consuming computations of the signature generation before the message to be signed is known. Whenever this sensor node has some critical event to report, it quickly computes the inexpensive online signature and reports the event authentically as soon as possible. Moreover, OOS allows the offline phase to be performed on some other resourceful device. So, depending on the nature of the application, it is possible for the base station to perform the complex computations of the offline phase and distribute the partial offline signature to the sensor nodes. This will reduce the computation overhead on sensor nodes. The sensor nodes then only perform the small, energy efficient computations of the online phase. However, this is a trade-off between the computation cost of offline signature and the communication cost. Some IBOOS schemes also allow to reuse the offline signature to sign more than one message. ID-based cryptography, on the other hand, handles the problem of public keys and certificates management in WSNs. For simplicity, we will use the term broadcast to represent both broadcast and multicast in the rest of the document.

Organization: Sec. 2 introduces the cryptographic primitives used, Sec. 3 reviews the proposed authenticated broadcast by sensor nodes scheme, Sec. 4 describes implementation details of IBOOS schemes and the results, Sec. 5 includes a discussion about the results, Sec. 6 compares our scheme with the existing signature based authentication schemes for WSNs, and Sec. 7 concludes the paper.

2. Cryptographic Primitives

2.1 ID-based Online/Offline Signature (IBOOS)

Definition 1. *An ID-based online/offline signature (IBOOS) scheme consists of five algorithms as follows:*

1. ***System Setup (SS):*** *Given a security parameter 1^k, outputs a master secret key SK_{PKG} and system parameters SP.*
2. ***Key Extraction (KE):*** *Given a user's identity ID_i and a master secret key SK_{PKG}, outputs a corresponding private key D_{ID_i}, i.e., $D_{ID_i} \leftarrow KE(ID_i, SK_{PKG})$.*
3. ***Offline Signing (OffSign):*** *Given a signing key D_{ID_i} [†] and system parameters SP, outputs an offline signature S, i.e., $S \leftarrow OffSign(D_{ID_i}, SP)$.*
4. ***Online Signing (OnSign):*** *Given a message m and an offline signature S, outputs an online signature σ, i.e., $\sigma \leftarrow OnSign(m, S)$.*
5. ***Signature Verification (Ver):*** *Given a message m, user's identity ID_i, signature σ and system parameters SP, returns 1 if the signature is valid and 0 if not. Namely, $0/1 \leftarrow Ver(m, ID_i, \sigma, SP)$.*

3. Authenticated Broadcast by Sensor Nodes Scheme

We now present a review of our proposed authenticated broadcast by sensor nodes scheme using IBOOS, described in [**?**,]. The first two phases of this schemes i.e., the *System Initialization* and the *Key Generation* are performed once, before the deployment of the WSN.

System Initialization: In our scheme, the base station plays the role of PKG, a trustworthy entity, and initializes the system in this phase. Let SK_{BS} be the secret key of the base station. The base station computes the corresponding public key PK_{BS} and sets up the public system parameters SP which include PK_{BS}. The master secret key SK_{BS} is only kept by the base station while SP is made public.

Key Generation: In this phase, the base station computes the secret keys of all sensor nodes corresponding to their IDs using the master secret key SK_{BS}. For a sensor node i with identity ID_i, the corresponding secret key is D_{ID_i} computed as $D_{ID_i} \leftarrow KE(ID_i, SK_{BS})$. IDs, corresponding private keys and system parameters are stored on sensor

[†]For some IBOOS schemes, the signing key is used in online phase rather than in offline phase

nodes before deployment. Hence, every sensor node i stores $\{ID_i, D_{ID_i}, SP\}$.

Message Broadcast and Authentication: In this phase, the sensor nodes broadcast authenticated messages which are verified using their Identity information, for instance, ID. The signature generation of a broadcast message is divided into *Offline* and *Online* phases:

Offline phase: The offline phase is performed before the message to broadcast becomes available. This phase can also be carried out by the base station. The offline signature algorithm runs in this phase and performs most of the signature computations to calculate the partial signature S as $S \leftarrow OffSign(D_{ID_i}, SP)$. The resulting offline signature S is stored on the sensor node i.

Online phase: Whenever the sensor node i senses an event which requires quick reporting, the online phase starts. In this phase, the sensor node i retrieves the offline signature S calculated during the offline phase. The online signature algorithm runs in this phase on sensor node i, and performs very minor and fast computations to obtain the final signature σ over message m as $\sigma \leftarrow OnSign(m, TS, ID_i, S)$, where TS is the current time stamp. The final broadcast message then contains the message m, time stamp TS, identity of the sensor node ID_i and the signature σ i.e., $\{m, TS, ID_i, \sigma\}$. Here TS is included to defeat a message replay attack.

Authentication: On receiving a broadcast message, the receiver first checks the time stamp TS to avoid the verification of a replayed message. If it is a fresh one, the receiver further proceeds with signature verification; otherwise it discards the message. The receiver verifies the signature σ using the identity information ID_i of the sender node and other system parameters as $0/1 \leftarrow Ver(m, TS, ID_i, \sigma, SP)$. If verification succeeds, the receiver accepts the message; otherwise it discards it. If necessary, it rebroadcasts the message to sensor nodes belonging to the next hop.

Sender Revocation: To revoke a compromised sensor node i, the base station broadcasts its identity ID_i to all other nodes in the network, who store ID_i. If in the future a sensor node receives a message containing ID_i, it simply rejects the message without going through the authentication process. An adversary is assumed to compromise only a few sensor nodes in the network. If the adversary compromises a majority of the sensor nodes, all the security mechanisms will fail. Therefore, storing the IDs of a few compromised nodes would incur a reasonable storage overhead for sensor nodes. Moreover, the base station can periodically update system parameters and secret keys of all legitimate sensor nodes excluding malicious nodes. However, this update might be costly. Another possible solution is to manually detach these compromised sensor nodes from the sensor network.

4. Implementation and Evaluation

4.1 Choice of IBOOS Schemes

There are many IBOOS schemes available, for example, based on Elliptic Curve Cryptography (ECC) and RSA signatures. Keeping in mind the security and efficiency requirements, the two different ECC based IBOOS schemes given in [**?**,] and [**?**,] were selected (reasons behind this selection are described in [**?**,]) for implementation and evaluation purposes. The IBOOS scheme in [**?**,] proposed by Ren et al. (say R-IBOOS) presents a method to convert an underlying signature scheme into an online/offline signature scheme to mitigate phishing attacks. The offline signature in this scheme can be securely reused to sign more than one message. This signature scheme is proved to be existentially unforgeable. Its security depends on the *Discrete Logarithm Problem*. Unlike R-IBOOS, the IBOOS scheme presented in [**?**,] by Xu et al. (say X-IBOOS) provides a direct online/offline signature scheme for authentication in mobile ad-hoc networks, which does not require another underlying signature scheme. This signature scheme is existentially unforgeable under adaptive chosen message attacks. To see how efficient these IBOOS schemes would be on sensor nodes, we went for the implementation of these IBOOS schemes on actual sensor nodes. However, we only implemented X-IBOOS scheme and based on the implementation results (discussed later in the paper), we decided to skip R-IBOOS. For the sake of interest, the details of X-IBOOS scheme are given in appendix A.

4.2 Implementation Details

For implementation purposes, the hardware platform selected was the standard MICA2 sensor node, a popular choice among the research community. MICA2 has an integrated ATMEGA 128L micro-controller from the AVR family having 8-bit processor, 4KB of SRAM, 128KB of flash memory (ROM) with a clock speed of 7.3828MHz. Our implementation involves a base station (a laptop with TinyOS installed) and two MICA2 sensor nodes; one acting as a signer while the other acting as a verifier. However, both nodes have the ability to sign as well as verify the messages. To perform cryptographic operations, we used RELIC [**?**,], a publicly available highly efficient library to implement cryptographic operations on sensor nodes, particularly pairing computation. As both above mentioned IBOOS schemes required pairing computations, we decided to go for this library. The security level of ~80-bit, considered adequate for resource constrained sensor nodes, is adopted. η_T pairing is chosen to compute pairing operation as it is the fastest one to compute on resource constrained sensor nodes and is a best choice at this security level [**?**,]. The Setup and the Extract phases were performed at the base station. The system parameters and other secret information were stored on the sensor nodes via the base station. All the software programs (including online/offline signature code) running on the sensor nodes for evaluation have been implemented in the NesC language installed on the TinyOS operating system. We developed and tested our programs first on TOSSIM and Avrora, popular simulation and analysis tools for MICA micro-controllers. These tools together give the strength of code development and debugging. These pro-

grams were then installed on MICA2 sensor nodes. The implementation results are the average of running the code 50 times. Due to the space constraints, we skip the in depth implementation details here.

4.3 Performance Matrices

The primary goal of these experiments was to gather the actual statistics about the resource consumption of an IBOOS scheme on real sensor nodes and study its performance. In case of a signature scheme, the primary factors to affect a sensor node's resources are signature generation and verification costs (computation cost) and signature size (transmission cost). Therefore, the IBOOS schemes are evaluated for the following performance matrices: computation cost (time and energy consumption), signature size, and memory consumption (ROM/RAM). The transmission cost is proportional to the signature size, thus, we only count the signature size.

4.4 Implementation Results of X-IBOOS Scheme

4.4.1 Computation Cost

X-IBOOS [**?**,] requires two pairing computations in signature verification as the most expensive cryptographic operations. Table 1 shows the time and energy consumption of this scheme. It took about 1.697s to compute the offline signature while only 0.018s to compute the online part. Thus, this scheme enables a sensor node to generate a final signature over a real time message in 0.018s only which is quite fast considering the resource constraints of a sensor node. However, the verification part of X-IBOOS is very expensive, consuming considerable time and energy[†] because of the two expensive pairing computations.

	Time (s)	**Energy** (mW)
Offline Sign	1.697	50.92
Online Sign	0.018	0.54
Verify	5.099	177.01

Table 1 Time and Energy Consumption

4.4.2 Signature Size

A signature in the X-IBOOS scheme is comprised of two group elements of the form (x, y) and one number. Based on our selection of η_T pairing and ~80-bit security level, a random number takes about 271 bits and a group element is about 2*271 bits. Therefore, the resulting signature size is 1355 bits or 170 bytes. This signature size can be reduced up to 102 bytes by applying compression and including only one co-ordinate (x) of the group element, as is usually done. Given x and a single bit of y, the receiver can regenerate y. However, this is a trade-off between the transmission cost and the computation cost of deriving y for the receiver.

[†]Energy consumption is computed using the MICA2 data sheet [**?**,] and the computed number of clock cycles for each stage. The power consumption is calculated at 3V power supply and 7.3728MHZ clock frequency.

4.4.3 Memory Consumption

Table 2 summarizes the memory requirement of the X-IBOOS scheme including the size of both signature generation and verification codes. It also includes the code size of RELIC, TinyOS code, node's ID (16 bits) and private key (2∗271 bits), master public key (2∗271 bits) and other system parameters. The memory consumption of ROM, Global RAM and Stack RAM is 63,972, 1,933 and 1911 bytes respectively. This memory consumption can be reduced by storing only one co-ordinate x of the group elements on sensor node. Given x and a single bit of y, the node can derive y when it needs. This will reduce the storage consumption per one group element stored on the sensor node by 270 bits. The stack memory is consumed only during the execution of the program, i.e., during the signature generation and verification. Once the program stops execution, this memory is available for other operations. Note that this is the total storage consumption on a sensor node when a sensor node acts as both a signer and a verifier. In our proposed broadcast authentication scheme, a sensor node can act as sender as well as receiver of broadcast messages.

ROM	**Global RAM**	**Stack RAM**
63,972	1,933	1,911

Table 2 Memory Consumption in Bytes

4.5 Optimization

We proposed to evaluate two IBOOS schemes, X-IBOOS and R-IBOOS, as mentioned above. However, the evaluation results of X-IBOOS scheme depict the fact that pairing based schemes are expensive for sensor nodes in terms of resource consumptions. Pairing computation consumes considerable resources on sensor nodes including processing time, battery power and memory. As R-IBOOS also requires pairing computation, we can expect similar results. The computation of a single pairing operation using RELIC takes about 1.9s [**?**,] and, hence, consumes considerable battery power. To the best of our knowledge, this is the most efficient implementation of pairing operation for MICA2 sensor nodes. Although the application of online/offline signatures itself brought the benefit of quick authenticated broadcast by resource constrained sensor nodes, the existing IBOOS schemes being pairing based proved expensive for the sensor nodes. To the best of our knowledge, there are a few ECC based IBS schemes without pairing but no ECC based IBOOS scheme without pairing.

4.5.1 Bellare et al.'s IBS as IBOOS

The implementation results in Sec. 4.4 proved that by having a pairing-free IBOOS scheme, we could get better results for IBOOS schemes for WSNs. We realized that the IBS scheme (BNN-IBS) proposed by Bellare et al. [**?**,] and improved by Cao et al. [**?**,] could be used as an IBOOS scheme (say B-IBOOS). For the sake of interest, the details of B-IBOOS are given in appendix B. BNN-IBS is an ECC based pairing-free IBS scheme having only one point multiplication as the expensive operation in signature generation. This point multiplication computation, resulting in a partial signature, is independent of the message to be signed. Thus, it can be computed as an offline signature before the message to be signed is known. The rest of the signature generation uses this offline signature and the message and only performs integer arithmetics to get the final signature of the message. Integer arithmetics is very efficient for sensor nodes in terms of time and energy consumption. Operations using integer arithmetics, performed when the message to be signed is known, form the online phase of the signature generation. Thus, BNN-IBS can be computed as an online/offline signature into two phases; offline phase and online phase. The signature verification in BNN-IBS requires three point multiplications which although are expensive but far less expensive than a single pairing computation for sensor nodes. Thus, we decided to implement and evaluate BNN-IBS [**?**,] as IBOOS scheme, named as B-IBOOS.

Security of B-IBOOS: The security of BNN-IBS depends on the *Elliptic Curve Discrete Logarithm Problem*. Using BNN-IBS as IBOOS does not affect the security of this signature scheme. It is secure to compute the offline part before the message is known and store it. If an attacker compromises the sensor node and gets both the offline signature Y and the random number y used to generate Y, he still would not be able to get any extra benefits other than the ones obtained by compromising a sensor node. After computing the final signature, the sensor node deletes both Y and y.

4.6 Implementation Results of B-IBOOS Scheme

The IBS scheme in [**?**,] is actually an improvement over the BNN-IBS [**?**,] scheme to reduce the signature size. This improved version of the scheme has already been proposed to use in WSNs as IBS to provide outside user authentication where sensor nodes are the verifiers. We implemented this scheme as B-IBOOS scheme for ∼80-bit security level and selected the curve parameters accordingly.

4.6.1 Computation Cost

B-IBOOS [**?**,] computes three point multiplication operations in signature verification as expensive cryptographic operations for sensor nodes. The offline phase is comprised of only one point multiplication. One point multiplication took 0.295s in our implementation which confirmed the point multiplication cost obtained by [**?**,], using the same RELIC library. Table 3 shows the time and energy consumption of this scheme. It took about 0.295s to compute the offline signature while only 0.025s to compute the online part. The computation cost of the online phase is almost the same for both B-IBOOS (Table 3) and X-IBOOS (Table 1) schemes. Nevertheless, in the offline phase X-IBOOS took more time and, thus, consumed more energy than B-IBOOS. The same is the case with the verification phase. B-IBOOS, being pairing-free, verifies the signature in 1.044s only as compared to the verification time of 5.099s of X-IBOOS. A comparison of Table 1 and Table 3 highlights the fact that B-IBOOS is very efficient for resource constrained sensor nodes in terms of computation cost when compared with X-IBOOS.

	Time (s)	**Energy** (mW)
Offline Sign	0.295	8.85
Online Sign	0.025	0.74
Verify	1.044	31.33

Table 3 Time and Energy Consumption

4.6.2 Signature Size

The signature in B-IBOOS is comprised of one elliptic curve point of the form (x, y) and two numbers. We used 163-bit field for ECC to meet the ∼80-bit security level. For these settings, a random number takes 160 bits while one elliptic curve point takes 2∗163 bits. Therefore, the resulting signature size is 646 bits (80 bytes) without compression while 484 bits (60 bytes) with compression. This signature size is much smaller than the one in X-IBOOS scheme and, thus, results in a reduced transmission cost. IEEE Std. 802.15.4 [**?**,], the standard for the low-power sensor networks, allows a variable payload of up to 102 bytes. With this packet size, a sensor node still has 22 bytes available to include *ID* and the message *m*, other than uncompressed 80 bytes of the signature, to send them all together in a single packet. The messages exchanged to report any critical event, for instance the location of enemy, are usually short in size up to a few bytes. Therefore, 22 bytes provide enough space for both *ID* and the message *m*.

4.6.3 Memory Consumption

Table 4 shows the memory requirement of B-IBOOS scheme. Like X-IBOOS (Table 2), it also includes the memory consumed by the signature generation and verification code, RELIC code, TinyOS code, node's ID (16 bits) and private key (160 bits), master public key (2∗163 bits) and other system parameters. For B-IBOOS, ROM, Global RAM and Stack RAM consume 47,798, 1,902 and 1,821 bytes respectively. This memory consumption can also be reduced by storing only one co-ordinate x of the elliptic curve points of the form (x, y). This reduces the memory

consumption per one elliptic curve point stored on the sensor node by 162 bits. The memory consumed by the stack is returned once the program completes its execution. Like in the case of X-IBOOS, this is the total storage consumption on a sensor node when a sensor node acts as both a signer and a verifier. Compared with the memory consumption in X-IBOOS (Table 2), the ROM consumption is lower in B-IBOOS than X-IBOOS while the Global RAM consumption is almost the same in both schemes. The stack usage is slightly lower in B-IBOOS than X-IBOOS. However, the overall RAM consumption of B-IBOOS scheme is slightly smaller than the RAM consumption in X-IBOOS scheme.

ROM	Global RAM	Stack RAM
47,798	1,902	1,821

Table 4 Memory Consumption in Bytes

4.7 Optimization

In the light of the results we obtained for computation cost, signature size and memory consumption of both schemes, X-IBOOS and B-IBOOS, it is clear that B-IBOOS scheme outperforms X-IBOOS in terms of computation cost and signature size. The memory usage, however, does not make a big difference in both schemes. One factor, which is contributing towards the memory usage in B-IBOOS, is that RELIC uses a precomputed table to fasten the computation of point multiplication for ECC based schemes. This precomputed table is also stored and consumes some memory space on the sensor node. To optimize B-IBOOS scheme for the memory consumption, we decided to evaluate this scheme without the precomputed table of RELIC. It restrained us from using one efficient RELIC function used to compute the point addition of the two point multiplications, i.e., $(aP + bQ)$. Table 5 and Table 6 show the results obtained after removing the precomputed table of RELIC.

	Time (s)	**Energy** (mW)
Offline Sign	0.317	9.52
Online Sign	0.025	0.74
Verify	1.118	33.54

Table 5 Time and Energy Consumption

ROM	Global RAM	Stack RAM
45,612	1,634	1,381

Table 6 Memory Consumption in Bytes

Compared with Table 3 and Table 4, avoiding the precomputed table reduces the memory consumption of B-IBOOS scheme, particularly the RAM consumption. Although it slightly increases the time and energy consumptions of the offline phase and the signature verification phase, this increment is not a drastic change. It is an acceptable trade-off between the computation cost and the memory usage giving 10% of free memory. However, it depends on the nature of the application whether it can compromise on speed or memory.

5. Discussion

Our implementation results obtained in Sec. 4 strengthened the idea of using online/offline signatures for resource constrained sensor nodes. The X-IBOOS scheme proved expensive for the sensor nodes, consuming considerable resources. The reason behind this was not the online/offline signature itself but the expensive pairing based cryptography. The implementation results of B-IBOOS proved this argument. Hence, if we use pairing-free ECC based IBOOS schemes we can obtain better results.

Moreover, the two implementations of B-IBOOS offer a trade-off between the computation cost and the memory usage. Memory can be saved by removing the precomputed table and slightly increasing the computation time. However, this can be decided depending on the type of application. The offline signature is computed before the message to be signed is available and the online phase takes the same time in both implementations, i.e., 0.025s. Therefore, the time to compute the final signature, once the message is known, is the same in both cases. For time critical applications, it is reasonable to use the first implementation of B-IBOOS if the receiver is a sensor node, and the second implementation of B-IBOOS if the receiver is a powerful device. Broadcast of a message by a sensor node is not a very frequent event in time critical applications, for instance, forest fire alarm application. In forest fire alarm application, a message is sent by a sensor node only when a fire is set up somewhere in the forest. Signing and verifying a message occasionally only in critical situations is not very expensive for the sensor nodes. Moreover, if the offline phase is performed on the base station and the resulting offline signature is stored on the sensor node, it can further reduce the computation overhead on the sensor nodes. X-IBOOS can also be useful for such applications of WSNs where the offline signature is computed and stored by the base station on the sensor nodes and the signature verifier is a powerful device.

6. Comparison with Existing Authentication Schemes

Now we will compare our proposed broadcast authentication scheme using IBOOS with the existing signature based authentication schemes [**?**], [**?**], [**?**,] for WSNs. We could not find the exact implementation results of these schemes for WSNs. However, using the most efficient results of pairing, point multiplication and elliptic curve digital signature algorithm (ECDSA) costs on MICA2 and ignoring all other costs including hash computations, we roughly estimate the time cost of these schemes for comparison purposes. A point multiplication operation on MICA2 takes 0.30s [**?**,], pairing operation takes 1.9s [**?**,], signature generation and

verification for ECDSA take 0.36s and 0.63s respectively [**?**,]. Using these results, Table 7 gives the comparison. The message size includes both the signature size as well as the public key (20 bytes for ECDSA) for the first scheme while only the signature size for all other schemes.

Schemes	Sign. Time		Verif. Time	Message Size
	(Offline) s	(Online) s	s	bytes
Existing Broadcast Authentication Schemes				
CAS [**?**,]	0	0.36	2*0.63	60
DAS [**?**,]	0	0.36	0.63	40
IDS [**?**,]	0	2.2	5.7	84 [**?**,]
IMBAS [**?**,]	0	0.32	1.044	80
Proposed Broadcast Authentication Scheme				
Proposed	0.295	0.025	1.044	80

Table 7 Comparison of proposed broadcast authentication scheme using B-IBOOS with the existing broadcast authentication schemes for WSNs.

The first two schemes CAS and DAS [**?**,] propose to use ECDSA to sign a message. CAS requires the signer's public key and certificate to be sent with every signed message, increasing transmission overhead. The receiver verifies two ECDSA signatures for every message; one to verify the signed certificate and other to verify the signed message. DAS, on the other hand, requires all sensor nodes to store the public keys of every sender in the network increasing storage overhead. For a large scale network, it is not possible for a sensor node, with limited storage capability, to store the public key of every other sensor node in the network. Signature generation in IDS [**?**,] comprises one pairing and one point multiplication computations while signature verification involves two pairing computations and one exponentiation in G_T. The exponentiation in G_T (say E_T) is more expensive than the exponentiation in G_1 (say E_1) (E_1 is equivalent to one point multiplication). However, we only consider two pairing computations and one point multiplication (for E_T) as signature verification cost for this scheme. IMBAS [**?**,] proposes BNN-IBS for sensor networks where sensor nodes are only receivers (verifiers).

The figures in Table 7 show that the proposed authentication scheme using IBOOS scheme enables a sensor node to sign a message in 0.025s only as compared to the existing schemes which take significantly longer. In signature verification, only DAS takes less time than the proposed scheme but increases the storage overhead of storing senders public keys beyond the storage capability of the sensor nodes. Based on these results, we conclude that the proposed scheme using IBOOS is the most efficient scheme for time critical applications of WSNs when compared with the existing signature based authentication schemes. It also allows the base station to compute the offline signature on behalf of a sensor node, reducing computation overhead on them. With this implementation work, we have completed one part of our proposed framework [**?**,], i.e., the scheme of *authenticated broadcast by sensor nodes*.

7. Conclusion

In this paper, we presented the implementation evaluation of our previously proposed IBOOS signature based authentication scheme for MICA2 sensor nodes. We assessed the cost incurred by using two different IBOOS schemes for resource constrained sensor nodes. We first implemented and evaluated one pairing based IBOOS scheme named as X-IBOOS. For optimization purposes, we also converted the well-known pairing-free BNN-IBS scheme into an IBOOS scheme and implemented it on MICA2 sensor nodes. The implementation results show the suitability of IBOOS for WSNs. In future, we are going to focus on the session key establishment between the outsider user and the sensor node after successful user authentication, i.e., the second authentication scheme of the proposed framework.

Acknowledgment

We would like to thank Diego F. Aranha, the lead developer of the RELIC library, for his generous help in using and modifying the library for implementation purposes.

Appendix A: X-IBOOS Scheme

Setup. The system parameters generated are (G_1, G_2, P, P_{pub}, H_0, H_1). Here G_1 is a cyclic additive group generated by P with a prime order q. G_2 is a cyclic multiplicative group with same order q. Let $\hat{e} : G_1 \times G_1 \rightarrow G_2$ be a bilinear mapping with the following properties:

1. *Bilinearity*: $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$ for all $P, Q \in G_1$, $a, b \in \mathbb{Z}_q$.
2. *Non-degeneracy*: There exists P, $Q \in G_1$ such that $\hat{e}(P,Q) \neq 1$.
3. *Computability*: There exists an efficient algorithm to compute $\hat{e}(P,Q)$ for all P, $Q \in G_1$.

For a random number $s \in Z_q^*$, the master public key is $P_{pub} = sP$, H_0: $\{0,1\}^* \rightarrow G_1$ and H_1: $\{0,1\}^* \times G_1 \rightarrow Z_q^*$. The master secret key is s which is kept secret.

Extract. Given an identity ID, the corresponding private key D_{ID} is computed as $D_{ID} = sH_0(ID)$ and $Q_{ID} = H_0(ID)$.

OffSign. Pick two random numbers $r, x \in Z_q^*$, output the offline signature pair (S, R), where $S = \frac{1}{r} D_{ID}$ and $R = xP$.

OnSign. Given a message m, compute the online signature as $\sigma = H_1(m,R)x + r$. The resulting signature is a triple $\langle \sigma, S, R \rangle$.

Verify. Check whether (P_{pub}, $\sigma P - H_1(m, R)R$, S, Q_{ID}) is a valid Diffie-Hellman tuple.

Appendix B: B-IBOOS Scheme

Setup. The system parameters are (E/F_q, P, p, P_0, H_1, H_2). Here E is an elliptic curve over a prime finite field F_q. The order of $E(F_q)$ is m. p is a prime number with $p^2 \nmid m$, $P \in E(F_q)$ is a point of order p and G is a group generated by P. For a master secret key $x \in \mathbb{Z}_p$, the master public key is $P_0 = xP$. $H_1 = \{0,1\} \times G^* \rightarrow \mathbb{Z}_p$ and $H_2 = \{0,1\}^* \rightarrow \mathbb{Z}_p$.

Key Extraction. Given and identity ID_i of a user I, the corresponding private key is generated as

- Choose at random $r \in \mathbb{Z}_p$ and compute $R_i = rP$.
- Compute $c = H_1(ID_i \| R_i)$.
- Compute I's private key as $s_i = r + cx$.

I gets (R_i, s_i).

Signature Generation. I with identity ID_i signs a message m in two phases as follows:

OffSign. The Offline phase is performed before the message to be signed is known.

- Choose at random $y \in \mathbb{Z}_p$ and compute $Y = yP$.

The offline signature is (y, Y).

OnSign. The Online phase is performed after the message becomes available.

- Compute $h = H_2(ID_i, m, R_i, Y)$
- Compute $z = y + hs_i$.

The tuple $\langle R_i, h, z \rangle$ is I's signature on message m.

Signature Verification. Given $\langle R_i, h, z \rangle$, ID_i and the message m, the receiver verifies the signature as:

- Compute $c = H_1(ID_i \| R_i)$
- Check whether the following equation holds.

$$h = H_2(ID_i, m, R_i, zP - h(R_i + cP_0))$$

Rehana Yasmin is currently a PhD student at the University of Birmingham, UK under the supervision of Eike Ritter and Guilin Wang. Her area of research includes security and privacy issues in wireless sensor networks and application of cryptography in wireless sensor networks.

Eike Ritter is a lecturer at the University of Birmingham, UK. His research interests include security, applications of mathematical logic and category theory to computer science, type theory and its applications to functional programming, proof theory and automated theorem proving.

Guilin Wang is a senior lecturer at the University of Wollongong, Australia. His research areas include applied cryptography, information security, and electronic commerce, especially, the analysis, design, and application of digital signatures, secret sharing schemes, and security protocols.