# Identity-Based Encryption for Sensor Networks

Leonardo B. Oliveira*
UNICAMP, Brazil
leob@ic.unicamp.br

Ricardo Dahab
UNICAMP, Brazil
rdahab@ic.unicamp.br

Julio López
UNICAMP, Brazil
jlopez@ic.unicamp.br

Felipe Daguano
UNICAMP, Brazil
daguano@ic.unicamp.br

Antonio A. F. Loureiro
UFMG, Brazil
loureiro@dcc.ufmg.br

## Abstract

*In spite of several years of intense research, the area of security and cryptography in Wireless Sensor Networks (WSNs) still has a number of open problems. On the other hand, the advent of Identity-Based Encryption (IBE) has enabled a wide range of new cryptographic solutions. In this work, we argue that IBE is ideal for WSNs and vice versa. We discuss the synergy between the systems, describe how IBE can solve the key agreement problem in WSNs, and present some estimates of performance.*

## 1 Introduction

Wireless sensor networks (WSNs) are ad hoc networks comprised mainly of small sensor nodes with limited resources and one or more base stations (BSs), which are much more powerful laptop-class nodes that connect the sensor nodes to the rest of the world [6]. They are used for monitoring purposes, providing information about the area being monitored to the rest of the system. Application areas range from battlefield reconnaissance and emergency rescue operations to surveillance and environmental protection.

Like any wireless ad hoc network, WSNs are vulnerable to attacks [10, 24]. Besides the well-known vulnerabilities due to wireless communication and ad hocness, WSNs face additional problems. For instance, sensor nodes are small, cheap devices that are unlikely to be made tamper-resistant or tamper-proof. Also, they are often deployed in unprotected, or even hostile areas, which makes them more vulnerable to attacks. It is therefore crucial to add security to these networks, specially those that are part of mission-critical applications.

*Supported by FAPESP grant 2005/00557-9

Until recently, security solutions for WSNs relied on symmetric encryption algorithms (e.g., RC5 [18]) to provide properties such as authentication and confidentiality since, due to their resource constraints, nodes cannot afford to use conventional algorithms of Public Key Cryptography (PKC), e.g. RSA/DSA.

Although more efficient than PKC, symmetric cryptosystems have some drawbacks. Firstly, nodes face the *key agreement* problem, i.e., they must decide on a shared key to communicate securely. This problem is even worse in WSNs due to the open and unattended environments where nodes are commonly deployed [24]. Further, the ideal level of security in these cryptosystems is achieved by using pairwise keys. However, this scheme is not scalable and thus is inadequate for WSNs which may comprise thousands of nodes. Finally, symmetric cryptosystems do not provide nonrepudiation.

To address some of theses drawbacks, a number of key predistribution schemes have been proposed (e.g., [5, 12, 27]). Although effective in trying to achieve a good trade-off between resource consumption and resiliency, these proposals eventually incur some degree of overhead.

LEAP [27], perhaps the most efficient proposal, allows a pairwise key agreement protocol between neighboring nodes using only symmetric primitives. However, LEAP has also drawbacks. Firstly, LEAP assumes that a predistributed key shared among all nodes will not be disclosed during the $t$ initial time units of the network operation. Secondly, LEAP assumes that once this key is erased, it cannot be recovered from memory. However, this is not always the case. Lastly, LEAP does not provide digital authentication and repudiation of messages is still possible.

Today, motivated by this vulnerabilities, the cryptography community in WSNs has been investigating more efficient techniques of PKC. By using Elliptic Curve Cryptography (ECC) [15, 11], for example, it has been shown (e.g., [8]) that PKC is indeed feasible in WSNs since ECC

consumes considerably less resources than conventional PKC, for a given security level.

However, in order to use effectively ECC in WSNs, it is first necessary to enable authentication of public keys. Otherwise, the network shall be vulnerable to *man-in-the-middle* attacks. Public key authentication is usually achieved by means of a Public Key Infra-structure (PKI), which issues certificates and requires users to store, exchange, and verify them. These operations, in turn, incur high overheads of storage, communication, and computation and, as a result, are inadequate for WSNs [4].

Identity-Based Encryption (e.g. [2]) (IBE) is an exception where an information that uniquely identifies users (e.g. IP or email addresses) can be used to both exchange keys and encrypt data, and thus PKI is unnecessary. Although the notion of Identity-Based Encryption dates from Shamir's original work [21], it only has become truly practical with the advent on Pairing-Based Cryptography (PBC) [19, 14].

In this work, we argue that IBE is the ideal encryption scheme for WSNs. In fact, because WSNs meet the strong needs of an IBE scheme, we go further and argue that they are the ideal scenario for using IBE as well. We discuss the use and implementation of IBE in resource-constrained nodes and present some estimated results. To be concrete, we use the ATmega128 8-bit AVR processor, which is presented in nodes from the Mica *motes*' family [9]. To our knowledge, ours is the first work to discuss implementation issues and to present performance estimates on the IBE over an 8-bit platform.

The rest of this work is organized as follows. In Section 2, we introduce the PBC concepts. In Section 3, we first discuss the synergy between IBE and WSNs and then describe how IBE can be used in the context of WSNs. We present implementation issues and results in Section 4. Finally, we discuss related work and conclude in Sections 5 and 6, respectively.

## 2 Pairings: preliminaries

In what follows, let $E/\mathbb{F}_q$ be an elliptic curve over a finite field $\mathbb{F}_q$, $E(\mathbb{F}_q)$ be the group of points of this curve, and $\#E(\mathbb{F}_q)$ be the group order.

**Bilinear pairing.** Let $n$ be a positive integer. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be additively-written groups of order $n$ with identity 0, and let $\mathbb{G}_T$be a multiplicatively-written group of order $n$ with identity 1.

A *bilinear pairing* is a computable, nondegenerate function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ that satisfies the following condition:

$$\forall\, P, P' \in \mathbb{G}_1 \text{ and } \forall\, Q, Q' \in \mathbb{G}_2, \text{ we have}$$

1. $e(P + P', Q) = e(P, Q)e(P', Q)$; and
2. $e(P, Q + Q') = e(P, Q)e(P, Q')$ .

**Embedding degree.** A subgroup $G$ of $E(\mathbb{F}_q)$ is said to have *embedding degree* $k$ if its order $r$ divides $q^k - 1$, but does not divide $q^i - 1$ for all $0 < i < k$.

**The Tate pairing.** Let $E(\mathbb{F}_q)$ contain a subgroup of prime order $r$ coprime with $q$ and with embedding degree $k$. (In most applications, $r$ also is a large prime divisor of $\#E(\mathbb{F}_q)$.) The *Tate pairing* is the bilinear, nondegenerate mapping

$$\hat{e} : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^r.$$

**Bilinear Diffie-Hellman Problem.** Most of the new applications of PBC rely on the hardness of the following problem for their security [7]: given $P$, $Q$, $aP$, and $bP$ such that $e(P, Q) \neq 1$, compute

$$e(abP, Q).$$

This problem is known as the *Bilinear Diffie-Hellman Problem*. The hardness of the Bilinear Diffie-Hellman Problem depends on the hardness of the Diffie-Hellman problems both on $E(\mathbb{F}_q)$ and in $\mathbb{F}_{q^k}$. So, for most PBC applications the parameters $q$, $r$, and $k$ must satisfy the following security requirements:

1. $r$ must be large enough so that the Elliptic Curve Discrete Logarithm Problem (ECDLP) in an order-$n$ subgroup of $E(\mathbb{F}_q)$ is infeasible to be solved using Pollard's rho algorithm;
2. $k$ must be large enough so that the Discrete Logarithm Problem (DLP) in $\mathbb{F}_{q^k}$ is infeasible to be solved using the index-calculus methods.

## 3 Applying IBE to WSNs

Today, IBE schemes (e.g. [2]) seem to be the only truly practical mean of providing public key encryption in WSNs since they do not require a PKI. Instead, they employ users' identification (e.g., node IDs) as public keys.

We go further and argue that IBE is not only ideal for WSNs, but the converse is also true. For example, IBE schemes have strong requirements such as the existence of an unconditionally trusted entity, who is responsible to issue users' private keys. WSNs, however, possess intrinsically such an entity, i.e., the BS. Another requirement is that the keys must be delivered over confidential and authentic channels to users. In most of the WSN applications, however, nodes' private keys can be distributed *offline*, i.e., they

can be generated and preloaded directly into nodes prior deployment.

In spite of all its advantages, IBE still is a public key cryptosystem and thus it is orders of magnitude more complex than symmetric cryptosystems. Because of this, we envision that IBE will be used only to nodes set up pairwise symmetric keys among themselves and the rest of the communication will be protected by using those keys. In Fig. 1, we show how IBE can be used to establish pairwise keys among communicating nodes. (In WSNs, where the communication is in general multi-hop from nodes to the BS, communicating nodes are often the neighboring nodes.) The protocol works as follows.

Prior deployment, each node $X$ is assigned the following information: the node's ID $id_X$, the node's IBE private key $S_X$, and a function $f$ that takes an ID (e.g., $id_X$) as input and outputs its corresponding IBE public key (e.g. $P_X$). After deployment, each node broadcasts its ID in its neighborhood (Step 1). Neighboring nodes thus use the function $f$ together with the received ID to generate its corresponding public key. After that, each of the neighbors generate a pairwise key and respond to the original node by including this key in the message (Step 2). The transmission of the message is protected by using IBE's public keys. Finally, subsequent communications among nodes are protected with MACs [1] computed using the exchanged pairwise keys (Step 3).

IDs being broadcast by nodes (e.g. $A$ and $B$):

1. $A \Rightarrow \mathcal{G}_A: \quad id_A$
   $B \Rightarrow \mathcal{G}_B: \quad id_B$
   ...

Neighboring nodes (e.g., $M$ from $A$ and $N$ from $B$) use received IDs to generate public keys (e.g. $P_A$ and $P_B$) and exchange pairwise keys:

2. $M \rightarrow A: \quad id_A, \mathsf{enc}_{P_A}(id_M \mid id_A \mid k_{M,A})$
   $N \rightarrow B: \quad id_B, \mathsf{enc}_{P_B}(id_N \mid id_B \mid k_{N,B})$
   ...

Secure exchange of information between neighboring nodes (e.g., $A$ and $M$, and $N$ and $B$)

3. $A \rightarrow M: \quad id_A, id_M, m, \mathsf{mac}_{k_{M,A}}(id_A \mid id_M \mid m)$
   $N \rightarrow B: \quad id_N, id_B, m, \mathsf{mac}_{k_{N,B}}(id_N \mid id_B \mid m)$
   ...

The various symbols denote:

| | |
|---|---|
| $id_X$: | Node $X$'s ID |
| $\mathcal{G}_X$: | Group of nodes in node $X$'s neighborhood |
| $k_{X,Y}$: | Key shared between nodes $X$ and $Y$ |
| $P_X$: | Node $X$'s public key |
| $S_X$: | Node $X$'s private key |
| $\mathsf{mac}_k()$: | MAC computed using key $k$ |
| $\mathsf{enc}_k()$: | Encryption computed using key $k$ |
| $m$: | Message information |
| $\Rightarrow, \rightarrow$: | Broadcast and unicast, respectively |

**Figure 1. Key agreement protocol.**

## 4 Implementation and Evaluation

In this section, we will describe some implementation issues (Section 4.1) and present estimated numbers (Section 4.2) on the costs of computing IBE in such a platform.

### 4.1 Implementation Issues

Recall from Section 2 that $E/\mathbb{F}_q$ is an elliptic curve defined over $\mathbb{F}_q$, $r$ is a large prime divisor of $\#E(\mathbb{F}_q)$ coprime to $q$, and $k$ is the embedding degree.

**The pairing.** The two most important pairings in ECC are the Tate and the Weil pairings. According to [7], the Tate pairing seems to be more efficient than the Weil pairing. Therefore, the Tate pairing appears to be more adequate to WSNs than the Weil pairing.

**The field.** Given a cryptosystem, the hardness of its underlying problem dictates the size of the security parameters. Namely, the harder the problem, the smaller the parameter size. The parameter size, in turn, dictates the efficiency, i.e., the smaller the parameter size, the faster the computation time. The DLP in prime fields is considered to be harder than the DLP in binary fields and thus it seems that prime fields are more adequate to WSNs.

**Curve selection.** Authors tend to choose nonsupersingular curves rather than supersingular curves because they feel that the formers have security advantages compared to the latters. We argue that until now there is no concrete evidence for that and thus it seems that supersingular curves are more adequate to WSNs since they have been shown empirically to be faster [20].

**Parameters $q$ and $r$.** The choice of the parameters $q$ and $r$ is a key factor in the efficiency of pairing computation, as curve operations are performed using arithmetic of the underlying field. In prime fields, by choosing $q$ a Mersenne prime (i.e., a number of the form $2^p - 1$) helps in computing modular reduction operations efficiently. At the same time, by choosing $r$ a Solinas prime (in practice, a prime of low Hamming weight) reduces considerably the computation of

[1] Note that MAC is often used to stand for medium access control in networking papers. Here, MAC stands for message authentication code.

pairings. Note, however, that because of the idiosyncrasies of the both types of primes, often it is not possible to find a pair $q$ and $r$ Mersenne and Solinas primes, respectively, suitable for pairings.

**Embedding degree $k$.** We have chosen $k = 2$ since it provides a number of benefits while computing pairings [20]. For example, $k = 2$: 1) allows the important denominator elimination optimization; 2) helps in finding a $r$ of low Hamming weight; 3) makes $\mathbb{F}_{q^k}$ arithmetic relatively easy to implement; 4) has been shown empirically to be efficient;

**Parameter sizes.** Parameter sizes often pose a tradeoff between security level and efficiency. This issue is especially important when dealing with resource-constrained nodes.

For most PBC schemes (including IBE), the security requirements described in Section 2 can be satisfied by choosing $r > 2^{160}$ and $q^k > 2^{1024}$. However, security requirements in WSNs are often relaxed [18]. This is because of their short lifetimes and because the goal is not to protect each node individually, but the network operation as a whole. Until now, the larger parameters sizes for which the ECDLP and the DLP are known to be solved are $2^{109}$ and $2^{431}$, respectively. Therefore, it seems that $r \geq 2^{128}$ and $q^k \geq 2^{512}$ are able to meet the current security requirements of WSNs.

**Point coordinates.** It has been shown by empirical work that, if precomputation is not allowed, to represent curve points as *projective* coordinates $(x, y, z)$ rather than in *affine* coordinates $(x, y)$ is faster [20]. On the other hand, Barreto *et al.* [1] have shown that affine coordinates are the most efficient coordinate system in some cases where precomputation of intermediate results is possible. This indicates that the coordinate system to be used will depend on the amount of free space available in nodes' memory, i.e., it will depend on the nodes' capacity for storing intermediate results in memory.

### 4.2 Results

The time consuming part while evaluating IBE is the pairing computation. The work of Barreto *et al.* [1] gives estimates of computing pairings by means of the number of modular multiplications in $\mathbb{F}_q$. According to their work, assuming $k = 2$ and $q$ a large prime, the costs for computing the Tate pairing using projective coordinates without precomputation, and projective and affine coordinates with precomputation are equivalent to 4153.2, 2997.6, and 1899.6 modular multiplications, respectively.

In what follows, we have measured the costs for modular multiplications in ATmega128, and estimated times for pairing computation based on the work of Barreto *et al.* [1]. We have considered a security level equivalent of $q^k = 2^{512}$, i.e., $k = 2$ and $q$ a 256-bit prime. In fact, we generated results for $q$ both a random 256-bit prime and a generalized Mersenne 256-bit prime (e.g. secp256r1 [22]). The results are shown in Table 1. They range from 4.33s (Mersenne prime using affine coordinates with precomputation) to 13.93s (random prime using projective coordinates without precomputation) thus indicating that pairing computation in the ATmega128 processor is feasible.

| | Coordinate System | | |
|---|---|---|---|
| | Projective | | Affine |
| **Prime** | w/o precomp. | precomp. | precomp. |
| **Random** | 13.93s | 10.05s | 6.37s |
| **Mersenne** | 9.45s | 6.82s | 4.33s |

**Table 1. Time estimates of the Tate Pairing (in seconds).**

In the context of WSNs, recall from Section 3 that we envision that nodes will use IBE only to exchange pairwise keys with neighboring nodes and most of the time communication will be protected through symmetric primitives. Therefore, the costs of computing pairings indeed will not impact nodes' normal functioning.

## 5 Related Work

WSNs are a subclass of MANETs, and much work (e.g., [26]) has been proposed for securing MANETs in general. These studies are not applicable to WSNs because they assume laptop- or palmtop-level resources, which are orders of magnitude larger than those available in WSNs. Conventional public key based solutions are such an example.

Among the studies specifically targeted to resource-constrained WSNs, some [24] have focused on attacks and vulnerabilities. Wood and Stankovic [24] surveyed a number of denial of service attacks against WSNs, and discussed some possible countermeasures. Karlof and Wagner [10] focused on routing layer attacks, and showed how some of the existing WSN protocols are vulnerable to these attacks.

Of those offering cryptographic solutions, a considerable number (e.g., [5, 27, 12, 16, 17]) have focused on efficient key management of symmetric cryptosystems. Others (e.g., [23, 8, 13]) have been investigating more efficient techniques of PKC. By using ECC, for example, it has been shown (e.g., [8, 13]) that resource-constrained nodes are indeed able to compute public key operations.

However, public key authentication in the context of WSNs was still an open problem, as these type of networks

cannot afford a conventional PKI and the proposed alternatives (e.g. [4]) are not applicable to all contexts. Motivated by that, Zhang *et al.* [25] and Doyle *et al.* [3] have used IBE for key distribution. In spite of this, none of the works show the feasibility of computing IBE primitives in resource constrained nodes. The former has assumed that this will be soon feasible. And the latter has considered a platform more powerful than those found in resource-constrained nodes.

## 6 Conclusion

Despite of several years of intense research, the area of security and cryptography in WSNs still has a number of open problems. On the other hand, the advent of IBE has enabled a wide range of new cryptographic solutions. In this work, we first argued that IBE and WSNs are complementary systems. After that, we described how IBE can be used to solve the key agreement problem in the context of WSNs. Finally, we discussed implementation issues and showed some performance estimates. Our results indicates that pairing computation is feasible even in resource-constrained nodes.

## References


[1] P. S. L. M. Barreto, B. Lynn, and M. Scott. On the selection of pairing-friendly groups. In *Selected Areas in Cryptography — SAC'03*, LNCS. Springer-Verlag, 2003.

[2] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. Also appeared in CRYPTO '01.

[3] B. Doyle, S. Bell, A. F. Smeaton, K. McCusker, and N. O'Connor. Security considerations and key negotiation techniques for power constrained sensor networks. *The Computer Journal (Oxford University Press)*, 49(4):443–453, 2006.

[4] W. Du, R. Wang, and P. Ning. An efficient scheme for authenticating public keys in sensor networks. In *6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '05)*, pages 58–67, New York, 2005.

[5] L. Eschenauer and V. D. Gligor. A key management scheme for distributed sensor networks. In *9th ACM conf. on Computer and Communications security (CCS'02)*, pages 41–47, 2002.

[6] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking (MobiCom'99)*, pages 263–270, Seattle, WA USA, 1999.

[7] S. Galbraith. *Pairings*, chapter IX, pages 183–213. Advances in Elliptic Curve Cryptography. Cambridge University Press, 2005. I. Blake and G. Seroussi and N. Smart.

[8] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES'04)*, pages 119–132, 2004.

[9] J. L. Hill and D. E. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6):12–24, 2002.

[10] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, 1(2–3):293–315, 2003. Also apeared in 1st IEEE International Workshop on Sensor Network Protocols and Applications.

[11] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48:203–209, 1987.

[12] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(1):41–77, 2005. Also appeared in ACM CCS'03.

[13] D. J. Malan, M. Welsh, and M. D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, Santa Clara, California, October 2004.

[14] A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[15] V. Miller. Uses of elliptic curves in cryptography, advances in cryptology. In *Crypto'85, Lecture Notes in Computer Science*, volume 218, pages 417–426. Springer-Verlag, 1986.

[16] L. B. Oliveira, H. C. Wong, M. Bern, R. Dahab, and A. A. F. Loureiro. SecLEACH – a random key distribution solution for securing clustered sensor networks. In *5th IEEE International Symposium on Network Computing and Applications (NCA'06)*, Cambridge,MA, July 2006. p. 145-154.

[17] L. B. Oliveira, H. C. Wong, R. Dahab, and A. A. F. Loureiro. On the design of secure protocols for hierarchical sensor networks. *International Journal of Networks and Security (IJSN)*, 1(2):–, 2006. Special Issue on Cryptography in Networks, to appear.

[18] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, Sept. 2002. Also appeared in MobiCom'01.

[19] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Symposium on Cryptography and Information Security (SCIS2000)*, pages 26–28, Jan 2000.

[20] M. Scott. Computing the tate pairing. In *Topics in Cryptology - CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2005.

[21] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84: on Advances in cryptology*, pages 47–53. Springer-Verlag, 1984.

[22] Standards for Efficient Cryptorphy Group. Sec 2: Recommended elliptic curve domain parameters. SECG2, 2000.

[23] R. J. Watro, D. Kong, S. fen Cuti, C. Gardiner, C. Lynn, and P. Kruus. Tinypk: securing sensor networks with public key technology. In *2nd ACM Workshop on Security of ad hoc and Sensor Networks (SASN'04)*, pages 59–64, Washington, DC, October 2004.

[24] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, Oct. 2002.

[25] W. L. Zhang, W. Lou, and Y. Fang. Securing sensor networks with location-based keys. In *IEEE Wireless Communications and Networking Conference (WCNC'05)*, 2005.

[26] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

[27] S. Zhu, S. Setia, and S. Jajodia. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *10th ACM conference on Computer and communication security (CCS'03)*, pages 62–72. ACM Press, 2003.