TinyTate: Identity-Based Encryption for Sensor Networks

*Leonardo B. Oliveira, Diego Aranha, Eduardo Morais, Felipe Daguano, Julio López, and Ricardo Dahab
Email {leob,diego.aranha,eduardo.morais,daguano,jlopez,rdahab}@ic.unicamp.br
University of Campinas, Brazil

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 WSNs can take advantage of IBE, and present results for computation of the Tate pairing over resource constrained nodes.

keywords: *identity-based encryption, bilinear pairings, key management, sensor networks*

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) [8, 28]. They are used for monitoring purposes, providing information about the area being monitored to the rest of the system. Aside from the well known vulnerabilities due to wireless communication, WSNs lack physical protection and are usually deployed in open, unattended environments, which makes them vulnerable to attacks [15, 34]. It is thus crucial to devise security solutions to these networks.

Until recently, proposals for securing WSNs relied on symmetric cryptosystems (e.g., RC5 [26] and SkipJack [14]) to provide properties such as authentication and confidentiality since, due to their resource constraints, nodes cannot afford to run [2] conventional Public Key Cryptography (PKC), e.g. RSA/DSA. Although more efficient than PKC, symmetric cryptosystems face the *key distribution* problem, i.e., they must decide on a shared key to communicate securely.

Motivated by that, the research community has been investigating more efficient techniques for the deployment of PKC. By using Elliptic Curve Cryptography (ECC) [24, 16], for example, it has been shown (e.g., [10, 20]) 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 counter *man-in-the-middle* attacks by using public key authentication. Public key authentication is typically achieved by means of a

[*] Supported by The State of São Paulo Research Foundation under grant 2005/00557-9

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 [6].

Identity-Based Encryption [1, 3] (IBE) is an exception where a known information that uniquely identifies users (e.g. IP or email address) can be used as a public key and thus PKI is unnecessary. Although the notion of IBE dates from Shamir's original work [32], it only has become truly practical with the advent on Pairing-Based Cryptography (PBC) [29, 13, 22].

In this work, we argue that IBE is the ideal cryptographic scheme for WSNs. In fact, because WSNs meet the strong needs of an IBE scheme, we go further and argue that they are an ideal scenario for using IBE as well. We also discuss the use and implementation of IBE in resource-constrained nodes and present some results. Specifically, we evaluate pairings, the most significant operation of IBE, over the MICAz – the new generation of MICA *mote* nodes [12].

The rest of this work is organized as follows. In Section 2, we introduce 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: concepts

Bilinear pairings – or pairings for short – were first used in the context of cryptanalysis [22], but their pioneering use in cryptosystems is due the works of Sakai [29] *et al.* and Joux [13]. In this section we first present some paring concepts and then define the Tate pairing. (For more on these definitions, see for instance Galbraith [9].) 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 $\mathcal{O}$, and let $\mathbb{G}_T$ be a multiplicatively-written group of order n with identity 1.

A *bilinear pairing* is a computable, non-degenerate function $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The most important property of pairings in cryptographic constructions is the bilinearity, namely:

$$\forall\, P \in \mathbb{G}_1, \forall\, Q \in \mathbb{G}_2 \text{ and } \forall\, a, b \in \mathbb{Z}^*, \text{ we have } e([a]P, [b]Q) = e(P, Q)^{ab}.$$

Embedding degree. A subgroup $\mathbb{G}$ of $E(\mathbb{F}_q)$ is said to have an *embedding degree* k with respect to ℓ if k is the smallest integer such that $\ell \mid q^k - 1$.

Bilinear Diffie-Hellman Problem. Most of the PBC applications rely on the hardness of the following problem for their security [9]: Given P, $[a]P$, $[b]P$, and $[c]P$ for some $a, b \in \mathbb{Z}^*$, compute $e(P, P)^{abc}$.

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, ℓ, and k must satisfy the following security requirements:

1. ℓ must be large enough so that solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) in an order-n subgroup of $E(\mathbb{F}_q)$ is infeasible (e.g. using Pollard's rho algorithm);

2. k must be large enough so that solving the Discrete Logarithm Problem (DLP) in $\mathbb{F}_{q^k}$ is infeasible (e.g., using the index-calculus method).

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

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

3 Applying IBE to WSNs

Today, IBE (e.g. [1, 3]) seems to be the only truly practical mean of providing public key encryption in WSNs. IBE would employ nodes' identification (e.g., node IDs) as public keys and PKI's expensive operations would be thus unnecessary.

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, that is responsible for issuing users' private keys. WSNs, however, possess intrinsically such an entity, namely 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 to 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, as usual, IBE would only be used for setting up pairwise secret keys among nodes.

In Fig. 1, we show how IBE can be used to establish secret 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 to 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 ϕ that takes an ID (e.g., id_Y) as input and outputs the corresponding IBE public key to the ID (e.g. P_Y).

After deployment, each node broadcasts its ID and a nonce (Step 1). Neighboring nodes thus use the function ϕ together with the received ID to derive the corresponding public key. After that, neighboring nodes generate a secret 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 and private keys. To prevent replay attacks, the nonce from the original node's broadcast in Step 1 is also included in the message. Finally, subsequent communications among nodes are protected with MACs computed using the secret keys (Step 3). A value computed from the nonce (nonce') is also included as input to the MAC to prevent replay – in fact, the value of the "freshness token" nonce' needs to be updated in each interaction between nodes (Step 3).

4 Implementation and Evaluation

The time consuming part while evaluating IBE is the pairing computation. In this section, we describe implementation issues (Section 4.1) and present results (Section 4.2) on computing pairings over MICAz, the new generation of MICA mote node [12]. MICAz is powered with the ATmega128 microcontroller (8-bit/7.38 MHz processor, 4KB SRAM, 128KB flash memory).

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

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

Neighboring nodes (e.g., M from $\mathcal{G}_A$ and N from $\mathcal{G}_B$) use received IDs to derive public keys (e.g. P_A and P_B) and distribute secret keys:

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

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

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

The various symbols denote:

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

Figure 1. Key distribution protocol.

4.1 Implementation Issues

Recall from Section 2 that $E/\mathbb{F}_q$ is an elliptic curve defined over $\mathbb{F}_q$, ℓ 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 [9], 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. Supersingular curves have been shown empirically to be faster [31] than nonsupersingular curves. Authors, however, tend to choose nonsupersingular curves rather than supersingular curves because they feel that the latter have security advantages compared to the formers. Since until now no concrete evidence for that has appeared [31], supersingular curves seem to be more adequate to WSNs.

Parameters q and ℓ. The choice of the parameters q and ℓ 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. However, it has been shown recently that such technique also decreases the hardness of the DLP in $\mathbb{F}_q$ (e.g., [30]) and is potentially unsafe in the context of PBC. For ℓ, on the other hand, it is possible to choose a Solinas prime, which decreases the number of point additions and makes the pairing computation faster.

Embedding degree k. We have chosen $k = 2$ since it provides a number of benefits while computing pairings [31]. For example, $k = 2$ allows the denominator elimination optimization and makes $\mathbb{F}_{q^k}$ arithmetic easier to implement.

Parameter sizes. Parameter sizes often pose a tradeoff between security level and efficiency. For most PBC schemes (including IBE), the security requirements described in Section 2 can be satisfied by choosing $\ell > 2^{160}$ and $q^k > 2^{1024}$. However, security requirements in WSNs are often relaxed [26] to meet their needs for efficiency. This is possible 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 in prime fields are known to be solved are 2^{109} [17] and 2^{448} [4], respectively. Therefore, it seems that $\ell \geq 2^{128}$ and $q^k \geq 2^{512}$ are able to meet the current security requirements of WSNs.

Point coordinates. The two most common coordinate systems are the *projective* system (x, y, z) and the *affine* (x, y) system . The affine system requires inversions while performing point addition or doubling operations. The inverse operation, in turn, is commonly expensive. The projective system, on the other hand, reduces the need for inverse and thus seems to be more adequate to our target processor.

Twists. Let d be a quadratic non-residue in $\mathbb{F}_q$. The *twist* of an elliptic curve $E/\mathbb{F}_q : y^2 = x^3 + ax + b$ is given by $E^t/\mathbb{F}_q : y^2 = x^3 + d^2ax + d^3b$. For $k = 2$, there exists an isomorphism $\phi : E(\mathbb{F}_{q^2}) \rightarrow E^t(\mathbb{F}_q)$ such that $\phi[(a, 0), (0, d)] \rightarrow (-a, d)$, and arithmetic in $E(\mathbb{F}_{q^2})$ can be thus carried out faster in the group $E^t(\mathbb{F}_q)$.

4.2 Results

In this section, we describe the results of TinyTate, an implementation of the Tate pairing for resource constrained nodes. Our implementation is based on Barreto *et al.*'s work, takes into consideration the discussion in Section 4.1, and uses the Miller's algorithm [23] for pairing computation.

We use the following parameters: (i) the Tate Pairing on elliptic curves defined over fields with a large prime characteristic; (ii) the embedding degree $k = 2$, q is a 256-bit prime, and ℓ a 128-bit Solinas prime; (iii) group field arithmetic uses projective coordinates. To be concrete, we use the curve $E/\mathbb{F}_q : y^2 = x^3 + x$ with the parameters:

$$q = 3778160688959823585674557647265839472148162507153330298395747614203820774616 3;$$
$$\ell = 1701411885310716326446049097026969272 33;$$
$$h = 2220603207006424499438127477911456851 08;$$

where h stands for the cofactor of the curve order $\#E(\mathbb{F}_q)$[1].

Results in Table 1 were measured on a MICAz node running TinyOS [18]. The average execution time to compute a pairing is 30.21s. The costs concerning RAM and ROM (flash) memory are 1,831 and 18,384 bytes, respectively.

Tate Pairing		
Time (seconds)	**RAM (bytes)**	**ROM (bytes)**
30.21	1,831	18,384

Table 1. Costs to evaluate the Tate Pairing on MICAz.

Since we use IBE only to distribute secret keys among neighboring nodes (Section 3), the costs above are not a heavy burden to the whole system. ï»¿

5 Related Work

The number of studies specifically targeted to secure WSNs has grown significantly. Due to space constraints, we provide a sample of studies based on cryptographic methods, and then focus on those targeted to PKC.

A considerable number of works (e.g., [7, 26, 36, 19, 27]) have focused on efficient key management of symmetric cryptosystems. Perrig *et al.* [26] proposed SPINS, a suite of efficient symmetric key based security building blocks. Eschenauer *et al.* [7] looked at random key predistribution schemes, and originated a large number of follow-on studies which we do not list here. And Zhu *et al.* [36] proposed LEAP, a rather efficient scheme based on local distribution of secret keys among neighboring nodes.

The studies specifically targeted to PKC have tried either to adequate conventional algorithms (e.g. RSA) to sensor nodes, or to employ more efficient techniques (e.g. ECC). Watro *et al.* [33] proposed TinyPK. To perform key distribution, TinyPK assigns RSA efficient public operations to nodes and RSA expensive private operations to better suited external parties. To perform key distribution, TinyPK assigns RSA efficient public operations to nodes and expensive private operations to better suited external parties. Gura *et al.* [10] reported results for ECC and RSA on the ATmega128 and demonstrated that the first outperforms the latter. Their ECC implementation uses prime fields. Malan *et al.* [20] implemented ECC using binary fields and polynomial basis and presented results for the Diffie- Hellman protocol based on the ECDLP.

The above works have shown that nodes are able to compute PKC operations, but public key authentication has not been their focus of research. Motivated by that, proposals (e.g. [6, 35, 35, 25, 5, 21]) have been made to address this issue. Du *et al.* [6] proposed a scheme based in Merkle trees which is able

[1] Note that in this particular case there is a twist with same equation of the original curve.

to authenticate public keys using only symmetric operations. Zhang *et al.* [35] have made use of IBE for key distribution in WSNs. They hoped that pairings would be soon feasible in resource-constrained nodes and were not concerned with implementation issues. Oliveira and Dahab [25] have envisioned the use of cryptography from pairings, including IBE, in sensor networks. In the same line of reasoning, Doyle *et al.* [5] have presented simulation results on pairings. The work, however, has considered a class of nodes more powerful than those found in resource-constrained nodes. And finally, McCusker *et al.* [21] have worked on low-energy hardware implementation of IBE able to meet sensor nodes' constraints.

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 distribution problem in the context of WSNs. Finally, we discussed implementation issues and present results on computing the Tate pairing over resource-constrained nodes.

For future work, we will consider other efficient pairings, e.g. the Ate pairing [11].

References

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

[2] David W. Carman, Peter S. Kruus, and Brian J. Matt. Constraints and approaches for distributed sensor network security. Technical report, NAI Labs, The Security Research Division, Network Associates, Inc., 2000.

[3] Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, London, UK, 2001. Springer-Verlag.

[4] Andrey Dorofeev, Denis Dygin, and Dmitry Matyukhin. Nabble forums – number theory. `http://www.nabble.com/Discrete-logarithm-in-GF(p)-----135-digits-t2870677.html`.

[5] Barry Doyle, Stuart Bell, Alan F. Smeaton, Kealan McCusker, and Noel O'Connor. Security considerations and key negotiation techniques for power constrained sensor networks. *The Computer Journal (Oxford University Press)*, 49(4):443–453, 2006.

[6] Wenliang Du, Ronghua Wang, and Peng 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.

[7] Laurent Eschenauer and Virgil 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.

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

[9] S. Galbraith. Pairings. In IanF. Blake, Gadiel Seroussi, and Nigel Smart, editors, *Advances in Elliptic Curve Cryptography*, London Mathematical Society Lecture Notes, chapter IX, pages 183–213. Cambridge University Press, 2005.

[10] 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.

[11] Florian Hess, Nigel Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, October 2006.

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

[13] Antoine Joux. A one round protocol for tripartite diffie-hellman. *J. Cryptology*, 17(4):263–276, 2004. Proceedings of ANTS-IV, 2000.

[14] C. Karlof, N. Sastry, and D. Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *2nd ACM SensSys*, pages 162–175, Nov 2004.

[15] 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.

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

[17] R. Lercier. Home page: Computations - discrete logarithms. `http://medicis.polytechnique.fr/~lercier/?lng=en`.

[18] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, and David Culler. TinyOS: An operating system for wireless sensor networks. In W. Weber, J. Rabaey, and E. Aarts, editors, *Ambient Intelligence*. Springer-Verlag, New York, NY, 2004.

[19] Donggang Liu, Peng Ning, and Rongfang 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.

[20] David J. Malan, Matt Welsh, and Michael 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.

[21] Kealan McCusker, Noel O'Connor, and Dermot Diamond. Low-energy finite field arithmetic primitives for implementing security in wireless sensor networks. In *2006 International Conference on Communications, CircuiTS aND sYstems*, volume III - Computer, Optical and Broadband; Communications; Computational Intelligence, pages 1537–1541, June 2006.

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

[23] V. Miller. Short program for functions on curves, 1986. unpublished manuscript.

[24] 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.

[25] Leonardo B. Oliveira and Ricardo Dahab. Pairing-based cryptography for sensor networks. In *5th IEEE International Symposium on Network Computing and Applications*, Cambridge,MA, July 2006. fast abstract.

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

[27] R. Di Pietro, L. V. Mancini, and A. Mei. Random key-assignment for secure wireless sensor networks. In *1st ACM workshop on Security of ad hoc and sensor networks (SASN'03)*, pages 62–71, 2003.

[28] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, 2000.

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

[30] Oliver Schirokauer. The number field sieve for integers of low weight. Cryptology ePrint Archive, Report 2006/107, 2006. `http://eprint.iacr.org/`.

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

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

[33] Ronald J. Watro, Derrick Kong, Sue fen Cuti, Charles Gardiner, Charles Lynn, and Peter 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.

[34] Anthony D. Wood and John A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, October 2002.

[35] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):247–260, 2006. Also appear in IEEE WCNC'05.

[36] Sencun Zhu, Sanjeev Setia, and Sushil 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.