An Evaluation of the Energy Cost of Authenticated Key Agreement in Wireless Sensor Networks

D. Galindo, R. Roman and J. Lopez

***Abstract*—Wireless sensors are battery-powered devices which are highly constrained in terms of computational capabilities, memory, and communication bandwidth. While battery life is their main limitation, they require considerable energy to communicate data. Due to this, the energy saving of computationally inexpensive security primitives (like those using symmetric key cryptography) can be nullified by the bigger amount of data they require to be sent. In this work we study the energy cost of key agreement protocols between peers in a network using public key cryptography techniques. Our concern is to reduce the amount of data to be exchanged. Our main news is that a computationally very demanding security primitive, such as identity-based authenticated key exchange, can present energy-wise a better performance than traditional public key based key exchange in realistic scenarios such as Underwater Wireless Sensor Networks. Such a result is not to be expected in wired networks.**

***Index Terms*— identity-based cryptography, wireless sensor networks, key agreement**

I. Introduction

It is well-known that from an efficiency point of view, symmetric key cryptography outperforms public (or asymmetric) key cryptography. Indeed, public key primitives are of the order of hundred of times more computationally intensive that their symmetric key counterparts. Along this line, one would use asymmetric cryptography not for efficiency issues but for achieving specific functionalities, like easier key management or non-repudiation.

The better performance of symmetric key primitives can be even more acute in resource-constrained devices, for which frequently battery life is the main limitation, so the less computationally expensive (and hence less energy consuming) operations the better. This is the reason why in areas like wireless sensor network security, using public key crypto has been considered prohibitive from the very beginning.

Somewhat surprisingly, is precisely in the area of wireless resource-constrained devices where this common wisdom is being challenged. The main reason behind this is the fact that communicating data in these devices requires considerable power, in contrast to wired devices. Therefore, it can be the case that the energy saving of a computationally inexpensive primitive is nullified by the bigger amount of data it requires to be sent. This has already been shown by Großschädl, Szekely and Tillich in [12], where the energy cost of two standardized symmetric and asymmetric key exchange protocols has been evaluated. Specifically, the symmetric key protocol used in that study is a light-weight variant of authenticated Kerberos [14], while the asymmetric key protocol is an elliptic curve version of Menezes-Qu-Vanstone [24,25] (ECMQV). The striking result is that in standard medium-size wireless sensor networks, ECMQV consumes less power than Kerberos, due to the fact that it requires 50% less bits to be exchanged.

The aim of this paper is to illustrate that the efficiency boundaries of public key cryptography in wireless sensor networks (WSN) can be further pushed by narrowing the amount of data exchanged in Authenticated Key Exchange (AKE) protocols. We consider standard scenarios, as well as particular scenarios with specific communication features, like underwater sensors. Our approach is two-layered. The first layer consists on investigating the use of identity-based cryptography. In the second layer we consider different types of sensors, which gives different performance results and conclusions.

A. Our contributions

The case of identity-based AKE illustrates how different wireless systems can be from wired networks. In identity-based cryptography, the identities of the users act as their public keys, so certificates and public keys need not be sent. Identity-based AKE is at the time of this writing tied to a computational number-theoretic primitive called bilinear pairing (cf. Chapter 5 in [6]), which is a computationally intensive operation. In a wired system, identity-based AKE would in general only be used for its specific functionalities, but not from a computational efficiency point of view. At first sight, one would preclude its use in WSN for a similar reason. Surprisingly, we show there exist realistic WSN scenarios where identity-based AKE performs similarly or even better than standard AKE protocols. Specifically, underwater WSN consume considerably more energy than radio-enabled WSN, to such an extent that shrinking the data to be exchanged can become the primary concern.

The authors are with the Computer Science Department of the University of Malaga. {dgalindo,roman,jlm}@lcc.uma.es

II. Public Key Cryptography for Wireless Sensor Networks

The specific features of Wireless Sensor Networks make them a very useful tool for solving problems in scenarios that require the acquisition and processing of physical measurements. The principal elements of a sensor network are the sensor nodes and the base station. Sensor nodes (nodes) are wireless-enabled, battery-powered, highly constrained devices that collect the physical information from their environment using an array of sensors such as thermistors, photodiodes, and so on. The base station is a more powerful device that serves as an interface between the nodes and the user. It collects the information coming from sensor nodes, and also send control information issued by the user. There can be from dozens to thousands of sensor nodes on a deployment field, although there is usually only one or more base stations on the same field.

Security is one of the principal concerns while designing protocols and mechanisms for WSN. In fact, a WSN is inherently insecure due to the features of its nodes and the communication channel. Nodes usually are not tamper-resistant due to cost constraints, and it is easy to physically access them in most scenarios because they must be located near the physical source of the events. Furthermore, any device can access the information exchange because the communication channel is public.

As a result, it is easy for an adversary to manipulate the sensor nodes and the communication channel of an unprotected network on its own benefit. There must be some protocols and security mechanisms that guarantee the resiliency of the network against any kind of external or internal threat. The foundation of these mechanisms and protocols are the security primitives, such as Symmetric Key Cryptography (SKC), Public Key Cryptography (PKC) and Hash functions. Using these primitives, it is possible to assure the confidentiality and integrity of the communication channel, while authenticating the peers involved in the information exchange.

These primitives require the existence of some security credentials (i.e. pairwise keys) between peers in order to encrypt and protect the information flow. Key distribution is not a trivial problem in WSN because in most cases it is not possible to know in advance which nodes are going to be neighbours, that is, which nodes need to share a pairwise key. The development of an efficient key management system for creating pairwise keys between neighbours is a hot research topic, with many complex SKC-based frameworks [2].

A. *Symmetric vs. Asymmetric cryptography in WSN*

Due to its energy efficiency and fast speed, Symmetric Cryptography becomes an interesting choice for securing the foundations of a sensor network. There are many optimal SKC algorithms implemented on sensor networks (such as Skipjack), that have small requirements in terms of memory usage and encryption speed. Moreover, some sensor nodes have transceivers that implement the IEEE 802.15.4 standard, which include a hardware implementation of the AES-128 algorithm.

However, as aforementioned, it is necessary to have certain security credentials in order to open a secure channel between two peers. As a result, if a sensor network relies only on SKC, it is necessary to implement certain key management systems that distribute the pairwise keys over the nodes of the network before or after its deployment. The underlying problem here is the typical key management shortcomings of symmetric-key algorithms. Asymmetric cryptography (i.e. Public Key Cryptography) can be really useful in this context. By using authenticated key exchange protocols such as Menezes-Qu-Vanstone (ECMQV), the process of negotiating pairwise keys between previously unknown peers can be greatly simplified.

Asymmetric Cryptography has been usually considered expensive and impractical because of the amount of computation required in contrast with the very limited memory and power that sensors offer. This is not necessarily true, as previously mentioned and reported in [12]. However, it would be interesting to find better schemes with less energy consumption than existing ECC-based key exchange protocols such as ECMQV. Due to the cost of sending and receiving one bit through the wireless channel, a possible solution can consist on finding schemes that reduce the volume of information that the nodes exchange during the negotiations. Possible candidates are Identity-based cryptography (IBC) and Self-Certified cryptography.

B. *Identity-based cryptography*

Certificates are needed to establish a trusted link between a public key and the identity of its owner (in our case a sensor node) in order to prevent man-in-the-middle attacks. In a WSN, nodes are supposed to establish pair-wise keys with nodes that belong to the same network, and forbidden to do so with nodes or devices outside the network. Therefore, in key establishment protocols like ECMQV, the nodes must at the beginning exchange their public keys and certificates. It is natural to assume these certificates take the form of a signature by the base station on the identity and public key of the node. In general, nodes public and secret keys are set up by the base station. Such a setting can be viewed as a key-escrowed system, that is, there exists a trusted party who computes the secret keys of the users. As a consequence one is tempted to use different forms of key-escrowed public key paradigms, like identity-based cryptography.

The concept of identity-based cryptography was proposed by Shamir in [19], aimed at simplifying certificate management inherent to the deployment of public key cryptography. The idea is that an arbitrary string uniquely identifying a user (such as an e-mail address or a telephone number) can serve as a public key for a cryptographic scheme. The user can not compute the corresponding secret key anymore, but instead it must authenticate itself to a Key Generation Center from which it obtains the corresponding private key via a secret channel.

The interest of IBC for WSN is that in IBC systems only the identity of the sensors must be exchanged, and thus public keys and certificates need not be sent. This results in an energy saving for the point of view of the communication between sensors, which can be very considerable depending on the sensor's transmitter. Additionally, in WSN the base station can naturally play the role of the Key Generation Center in an IBC system. The base station embeds the secret key   prior to its use in the field, and no authentic nor secret channel is needed for key setup.

III. Preliminaries and Energy Consumption Estimations

Before presenting our contribution, we need to fix some notation and recall basic concepts. In addition, we collect energy consumption estimations for elliptic curve computations and for the transmission and reception of information.

A. *Elliptic curves*

Let $GF(p)$ be a finite field. An elliptic curve over the field $GF(p)$ can be defined by an equation of the form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where $a_1, a_2, a_3, a_4, a_6 \in GF(p)$ and satisfy certain restrictions (see [10]). A point of the curve is specified by a pair (x, y) satisfying the above equation. It is possible to define an addition law on the points of the elliptic curve together with a special point called the "point at infinity", obtaining a finite order abelian group. Let G be the subgroup of order a certain prime integer n. We use multiplicative notation for the group G and assume it is finitely generated by an element g, i.e. every element $h \in G$ can be uniquely written as $h = g^{\alpha}$ for some $\alpha \in \mathbf{Z}_n$. An exponentiation refers to the operation g^r for a randomly taken $r \in \mathbf{Z}_n$. A multi-exponentiation $mexp(l)$ refers to computing $g_1^{r_1} \cdots g_l^{r_l}$, where $g_1, \ldots, g_l \in G$ and $r_1, \ldots, r_l \in \mathbf{Z}_n$, an operation that can be computed more efficiently than just computing l single exponentiations due to an algorithm by Strauss [22], which is sometimes referred to as Shamir's trick in the literature.

.

B. *Pairings*

We start by defining the concept of bilinear map. Let $\mathbf{G} = \langle \mathbf{g} \rangle$ and $\mathbf{G}_T$ be cyclic groups of order q for a prime $q > 3$. A map $e : \mathbf{G} \times \mathbf{G} \to \mathbf{G}_T$ is called a bilinear map, if it satisfies the following two properties:

Bilinearity: $e(\mathbf{g}^a, \mathbf{g}^b) = e(\mathbf{g}, \mathbf{g})^{ab}$ for all integers a, b

Non-trivial: $e(\mathbf{g}, \mathbf{g}) \neq 1 \in \mathbf{G}_T$.

and if it is efficiently computable.

Bilinear maps are usually implemented using the Weil or modified Tate pairings on an elliptic curve. Let $\mathbf{G} = E(GF(r))$ denote the group of points of an elliptic curve E over the finite field $GF(r)$ with order divisible by q such that q also divides $r^{\alpha} - 1$, where α is the order of r in $\mathbf{Z}_q^*$ and is called the MOV embedding degree. The modified Tate pairing $t(\cdot,\cdot)$, which is the bilinear map usually recommended, takes values in the subgroup $\mathbf{G}_T$ of $GF(r^{\alpha})^*$ of order q.

These curves correspond to the Type 1 category defined in [8,11], and there exist efficient hash functions $H : \{0,1\}^* \to \mathbf{G}$ mapping bit strings to elements in $\mathbf{G}$. Additionally, the equality $e(u, v) = e(v, u)$ holds for any $u, v \in \mathbf{G}$

C. *Evaluation of computation energy*

Our energy figures for elliptic curve computations are based on the work [21]. The elliptic curve used is $E_1 : y^2 = x^3 - 3x + 157$ over the field $GF(p)$ where $p = 2^{160} - 2^{112} + 2^{64} + 1$. $GF(p)$ allows for several computational efficiency improvements, such as improved squaring in the group of points G of the elliptic curve and efficient square roots in the finite field $GF(p)$. The latter is specially relevant in our context, since thanks to the technique of point compression (cf. § IV.4 [5]), a point $h = (x, y) \in G$ can be represented only by its x-coordinate together with a so-called compression bit. With the x-coordinate in hand, one can find y satisfying $y^2 = x^3 - 3x + 157$ by computing one square root in $GF(p)$. This results in at most two square roots, and the compression bit determines the right solution.

The amount of energy consumed by a primitive on a given processor is proportional to the time needed to execute the primitive. Using the ATmega128L [3] microcontroller, the energy spent to compute an exponentiation on the above curve is 30.02mJ, while computing a multiplication in $GF(p)$ requires $12 \cdot 10^{-3}$ mJ [21]. On the other hand, theoretical and experimental figures suggest that computing multi-exponentiations $mexp(2)$ and $mexp(3)$ take about 22% and 44% more energy respectively than that of a single exponentiation [7].

Since $p = 3 \, mod \, 4$, computing a square root modulo p requires 160 finite field products and 1 division by virtue of Algorithm 3.36 in [17]. One field division can be computed by one field inversion and one field product. Field inversion can be done trough the Greatest Common Divisor (GCD) algorithm. By using Lehmer's [15] GCD computation method, the cost of 1 division equals 9 field multiplications [4]. Hence, a square root modulo p consumes around 2 mJ.

Regarding pairings, [21] measures that computing the Tate pairing in the (supersingular) elliptic curve $E_2 : y^2 + y = x^3 + x^2$ over $GF(2^{271})$ and with MOV embedding degree $\alpha = 4$ consumes circa 258.44 mJ. Additionally, computing an exponentiation in $\mathbf{G} = E_2(GF(2^{271}))$ can be done within 50.93 mJ. A hash evaluation $H : \{0,1\}^* \to \mathbf{G}$ requires roughly the same amount of processing time/energy than an exponentiation [18].

D. Evaluation of communication energy

The cost of using the communication channel largely impacts the energy required to run any interactive protocol between sensor nodes.

The analyses in [12] were done considering a sensor node that uses the air as a transmission medium. This is the most common situation for a WSN, and most prototypes have been deployed on such conditions. However, there are some scenarios where the sensor nodes should be deployed in a lake or in the sea. For example, a sensor network can be used to monitor oil platforms or underwater construction projects. These networks have received the generic name of Underwater Sensor Networks (UWSN) [1].

In these UWSN, it is unpractical to use radio frequency transceivers, because of the severe attenuation factor presented by water. In order to open a communication channel between sensors, it is necessary to use specific underwater acoustic modems. These modems have different features than RF transceivers, and as a result, sending one bit of information carries a high energy penalty.

The differences between radio transceivers and acoustic modems in terms of the energy consumed by transmitting and receiving one single bit of data are highlighted in Table 1. It can be seen that the difference in consumption (J per bit) between acoustic modems and RF transceivers is not negligible. For the radio transceivers, we have considered the most popular sensor nodes platforms as of today, which are the Mica2 and the MicaZ [16]. The Mica2 transceivers use the 868/916 MHz ISM bands, while the MicaZ transceivers use the IEEE 802.15.4 standard. For the acoustic modems, we have considered the UWM2000 and UWM4000 modems [13], which are commonly used in research literature.

These results have been obtained using the information contained in the modem and mote datasheets, under the following assumptions: i) For the UWM2000 modem, we have used the mean of the transmission power indicated in its datasheet (2–8W). ii) For the transceivers used in the Mica2 and MicaZ motes, we have considered the most expensive transmission mode, which is theoretically able to send a bit of data to the maximum working range.

	Mica2	MicaZ	UWM2000	UWM4000
Working range	150 m	100m	1500 m	4000 m
Throughput	19.2 kbit/s	250 kbit/s	9600 bit/s	4800 bit/s
Tx. Consumption	81 mW	52,2 mW	4.000 mW	7.000 mW
Rx. consumption	30 mW	59,1 mW	800 mW	800 mW
μ J per bit (Tx)	4,12 μ J	0,204 μ J	416,66 μ J	1458,33 μ J
μ J per bit (Rx)	1,52 μ J	0,23 μ J	83.33 μ J	166.66 μ J

1. Analysis of the energy consumption of acoustic modems.

IV. Authenticated Key Exchange in Wireless Sensor Networks

A. Traditional approach: ECMQV

The elliptic curve version of the Menezes-Qu-Vanstone authenticated key exchange protocol [24,25] is described in Algorithm 4.1. This is the most standardized key exchange protocol using public key cryptography. We provide an abridged version of the scheme which suffices for our purposes.

KDF is a key derivation function, which can be implemented with SHA-160 for example. Node A's public key is $pk_A = g^{x_A}$, where x_A is A's secret key. Similarly for node B. In the first stage, the nodes exchange and verify certificates vouching for the fact that pk_A and pk_B are public keys from nodes belonging to the network. In a second stage, they exchange the ephemeral keys $E_A = g^{y_A}$ and $E_B = g^{y_B}$, where y_A, y_B are taken at random from the finite field $GF(p)$. We assume certificates are minimalist and take the form of ECDSA [23] signatures (r_A, s_A) and (r_B, s_B) by the owner/manufacturer of the network on the messages $id_A \| pk_A$ and $id_B \| pk_B$ respectively, where $\|$ denotes concatenation.

Algorithm 4.1 *ECMQV key derivation for entity A*

```
Input: Elliptic curve domain parameters $G, g, n$, the secret keys $x_A, y_A$ and the public elements $pk_A, pk_B, E_A, E_B$
Output: A secret key $K_{AB}$ shared with entity with public key $pk_B$
$m \leftarrow \lceil \log_2(n) \rceil / 2$    { $m$ is the half bitlength of $n$ }
$u_A \leftarrow (u_x \; mod \; 2^m) + 2^m$    { $u_x$ is the $x$-coordinate of $E_A$ }
$s_A \leftarrow (y_A + u_A x_A) \; mod \; n$
$v_A \leftarrow (v_x \; mod \; 2^m) + 2^m$    { $V_x$ is the $x$-coordinate of $E_B$ }
$z_A \leftarrow s_A v_A \; mod \; n$
$K_{AB} \leftarrow KDF(E_B^{s_A} \cdot pk_B^{z_A} \; mod \; n)$
```

Entity B runs the same algorithm by simply swapping the values $(x_A, y_A, pk_B, E_A, E_B)$ in Algorithm 4.1 with $(x_B, y_B, pk_A, E_A, E_B)$ and finally obtains the same key K_{AB} (cf. [25]).

The energy cost is dominated on the communication side by the exchange of public keys, certificates and ephemeral keys. Public keys have 161 bits (160 bits + 1 compression bit), each ECDSA certificate has 320 bits, while each ephemeral key contributes with 161 bits. Additionally, each message exchanged includes a payload consisting on communicating nodes identities, protocol ID, message ID, checksum, and low-level headers and footers, amounting to a total of 384 bits. Therefore the communication bandwith of ECQMV amounts to 1730 bits in the first stage, and 1090 bits. In total, 2820 bits are exchanged, which means every such a bit is sent and received by each node. On the computation side, each party has to verify an ECDSA signature, whose individual computational cost is dominated by one multi-exponentiation $mexp(2)$, and has to run the ECMQV protocol, whose computational cost is dominated by one exponentiation plus one multi-exponentiation $mexp(2)$. Additionally, two square roots are computed by each node to obtain the y-coordinate from the x-coordinate of pk_A, pk_B, E_A, E_B. The overall energy cost of ECMQV for a single node thus amounts to

$$2mexp(2)+1exp+2sqrt\ \text{(+transm 1410 bits+recep 1410 bits)} \qquad (1)$$

B. *Authenticated key exchange using identity-based keys*

In this section we explore the energy performance of identity-based authentication schemes in WSN. Due to the lack of any standardized identity-based key exchange protocol, we describe a non-interactive scheme due to Sakai, Ohgishi and Kasahara [20,9]. We provide an abridged version of the scheme which suffices for efficiency considerations. In order to run the key agreement protocol, the nodes only need to exchange their identities. The SOK protocol does not need any further communication between the parties for building a shared authenticated key. However this key remains unchanged for the life-time of the system (and thus this protocol does not provide forward secrecy), which can be unacceptable in some applications. SOK was the first identity-based authenticated key agreement protocol proposed in the literature..

In the SOK protocol, as described in Algorithm 4.2, a hash function $H:\{0,1\}^* \rightarrow \mathbf{G}$ is included in the domain parameters of the system, together with $\mathbf{g}^z$, where the master secret key z is only known to the base station. Node A's secret key is $sk_A = H(id_A)^z$, while node B's secret key is defined as $sk_B = H(id_B)^z$.

Algorithm 4.2 *SOK non-interactive ID-based key derivation for entity A*

```
Input: Bilinear map domain parameters $\mathbf{G}, \mathbf{G}_1, e, \mathbf{g}^z, n$, hash function $H$, identities $id_A, id_B$ and the secret key $sk_A$
Output: A secret key $K_{AB}$ shared with entity with identity $id_B$
1: $K_{AB} \leftarrow KDF(e(sk_A, H(id_B)))$
```

Entity B on inputs id_A, id_B, sk_B computes the same key $K_{AB} \leftarrow KDF(e(sk_B, H(id_A))$ thanks to the bilinearity of the pairing,

$$e(sk_A, H(id_B)) = e(H(id_A)^z, H(id_B)) = e(H(id_A), H(id_B)^z) =$$
$$= e(H(id_B)^z, H(id_A)) = e(H(id_A), sk_B)$$

The energy cost on the communication side consists on exchanging messages containing communicating nodes identities, protocol ID, message ID, checksum, and low-level headers and footers, amounting to a total of $2 \cdot 384 = 768$ bits. This is a huge communication saving with respect to the 2820 bits required by ECMQV. On the computation side, each party has to perform one hash operation, which is roughly equivalent to 1 exponentiation in $\mathbf{G}$, plus 1 pairing computation. Thus, the energy cost of SOK key agreement for a single node amounts to around

$$1exp_{\mathbf{G}} + 1pairing\text{(+trans 384 bits+recep 384 bits)} \quad (2)$$

V. Total energy cost

Using Equations (1), (2) and (3), along with the energy figures from Sections 3.3 and 3.4, it is possible to calculate the energy consumption of a node engaged in each of the key exchange protocols in “normal" and underwater sensor networks, expressed in terms of mJ. The results are shown in Table 2. An unexpected energy performance is to be found with the underwater sensor nodes, for which identity-based AKE has better performance that standard AKE for the UWM2000 and UWM4000 nodes. In fact, identity-based AKE presents the smallest energy cost in the UWM4000 case. Such a (energy-wise) efficiency result for identity-based cryptography primitives is unknown in traditional wired systems. The reason is quite simple: identity-based key exchange protocols exchange less data than the other protocols, and the more energy is required to send a bit of data, the more optimal identity-based key exchange becomes.

Mica2	Computation	Communication	
ECMQV	107,26	7,95	*115,21*
SOK	309,39	2,16	311,55

MicaZ	Computation	Communication	
ECMQV	107,26	0,61	*107,87*
SOK	309,39	0,166	309,55

UWM2000	Computation	Communication	
ECMQV	107,26	704,98	812,24
SOK	309,39	191,99	*501,38*

UWM4000	Computation	Communication	
ECMQV	107,26	2291,23	2398,49
SOK	309,39	623,99	*933,38*

2. Energy cost of authenticated key exchange for each node (in mJ).

VI. CONCLUSIONS

In this work we have focused on the fact that wireless sensor networks consume considerable energy in sending and receiving data. We have studied how identity-based cryptography can help to improve the energy cost of cryptographic key agreement between peers in a network. If previous work brought the novelty that the energy penalty of transmitting data made an asymmetric key agreement protocol energy-wise more efficient than a symmetric key protocol like Kerberos, our results bring the news that a computationally intensive primitive like identity-based key agreement outperforms traditional public key exchange protocols in specialized environments like underwater wireless sensor networks.

REFERENCES

[1] I. Akyildiz, D. Pompili and T. Melodia. *Underwater acoustic sensor networks: Research challenges*. Ad Hoc Networks Jounal (Elsevier), 3(3):257– 279, 2005.

[2] C. Alcaraz and R. Roman. *Applying key infrastructures for sensor networks in cip/ciip scenarios*. In 1st International Workshop on Critical Information Infrastructures Security (CRITIS 2006), 2006.

[3] Atmega128l. http://www.atmel.com/dyn/resources/prod_documents/2467S.pdf.

[4] R. Brent. *Some integer factorization algorithms using elliptic curves.* Australian Computer Science Communications, 8:149–163, 1986.

[5] I.F. Blake, G. Seroussi and N. Smart. *Elliptic Curves in Cryptography*, vol. 265 of London Mathematical Society Lecture Note Series. Cambridge University Press, 1999.

[6] I.F. Blake, G. Seroussi and N. Smart. *Advances in Elliptic Curve Cryptography*, vol. 317 of London Mathematical Society Lecture Note Series. Cambridge University Press, 2005.

[7] Billy Bob Brumley. *Efficient three-term simultaneous elliptic scalar multiplication with applications*. In Viiveke Fak, editor, Proceedings of the 11th Nordic Workshop on Secure IT Systems—NordSec ’06, pp. 105–116, Linköping, Sweden, October 2006.

[8] L. Chen, Z. Cheng and Nigel P. Smart. *Identity-based key agreement protocols from pairings*. Int. J. Inf. Sec., 6(4):213–241, 2007.

[9] Régis Dupont and Andreas Enge. *Provably secure non-interactive key distribution based on pairings*. Discrete Applied Mathematics, 154(2):270–276, 2006.

[10] Alfred Menezes Darrel Hankerson and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2004.

[11] S.D. Galbraith, K.G. Paterson and N.P. Smart. *Pairings for cryptographers*. Cryptology ePrint Archive, Report 2006/165, 2006. http://eprint.iacr.org/.

[12] Johann Großsch¨adl, Alexander Szekely and Stefan Tillich. *The energy cost of cryptographic key establishment in wireless sensor networks*. In ASIACCS, pp. 380–382. ACM, 2007.

[13] LinkQuest Inc. *Underwater acoustic modems.* http://www.link-quest.com/, 2007.

[14] John T. Kohl and B. Clifford Neuman. *The Kerberos network authentication service (V5)*. Technical Report 1510, 1993.

[15] D. H. Lehmer. *Euclids algorithm for large numbers*. Amer. Math. Monthly, 45:227–233, 1938.

[16] Crossbow Technology, Inc. *Mica2 / MicaZ Datasheet*. http://www.xbow.com

[17] A. Menezes, , P. van Oorschot and S. Vanstone. *Handbook of Applied Cryptography*. Discrete Mathematics and its Applications. CRC Press, 1997. Available at http://www.cacr.math.uwaterloo.ca/hac/.

[18] *Deterministic hashing to points on ibe-friendly elliptic curves*, 2005.

[19] A. Shamir. *Identity-based cryptosystems and signature schemes*. In CRYPTO 1984, vol. 196 of LNCS, pp. 47–53, 1985.

[20] R. Sakai, K. Ohgishi and M. Kasahara. *Cryptosystems based on pairing over elliptic curve* (in japanese). In The 2001 Symposium on Cryptography and Information Security, 2001. Oiso, Japan.

[21] Piotr Szczechowiak, Leonardo B. Oliveira, Michael Scott, Martin Collier and Ricardo Dahab. *Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks*. In European conference on Wireless Sensor Networks (EWSN’08), 2008.

[22] Strauss. *Addition chains of vectors*. American Mathematical Monthly, 71(7):806–808, Aug.-Sept. 1964.

[23] Accredited Standards Committee X9. *American national standard x9.62-2005, public key cryptography for the financial services industry, the elliptic curve digital signature algorithm (ecdsa)*, 2005.

[24] A. Menezes, M. Qu and S. Vanstone. *Some new key agreement protocols providing mutual implicit authentication*. Proceedings of the 2nd Workshop on Selected Areas in Cryptography (SAC 95), 1995.

[25] L. Law, A. Menezes, M. Qu, J. Solinas and S.A. Vanstone. *An efficient protocol for authenticated key agreement*. Des. Codes Cryptography, vol. 28, no. 2, pp. 119-134, 2003.