ABSTRACT

KAMPANAKIS, PANAGIOTIS T. Identity-Based Cryptography: Feasibility & Applications in Next Generation Sensor Networks. (Under the direction of Associate Professor Peng Ning).

Elliptic Curve Cryptography (ECC) has been a very interesting research field for many applications especially in sensor networks. The main reason for this is the restrictions on resources posed by sensor motes. ECC alleviates these restrictions by using small prime fields. At the same time, the evolution of hardware technology has built new, sophisticated motes with more capabilities which open new ways for sensor network applications.

In our work, we are extending the TinyECC package, initially written by An Liu, which provides elliptic curve cryptography functionality on sensor motes running TinyOS. We extend this package by porting it to new powerful motes, called Imote2 by Intel. We also further implement Bilinear Pairing which is the most important part of Identity-Based Cryptography. Furthermore, we evaluate and prove the feasibility of using pairing on next generation sensor motes. There are various ways that Bilinear Pairing could be used in Identity-Based cryptosystems, so in this way we insert Identity-Based Cryptography in the area of security for sensor networks. Additionally, we propose a scheme that can establish keys on sensor networks using Identity-Based Cryptography. We believe that our study will open the way and motivate further research in the field and contribute in providing more robust and secure sensor networks.

Identity-Based Cryptography:
Feasibility & Applications in Next Generation Sensor Networks

by

Panagiotis T. Kampanakis

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Engineering

Raleigh, NC

2007

Approved By:

Mihail L. Sichitiu

Michael Devetsikiotis
Chair of Advisory Committee

Peng Ning
Co-chair of Advisory Committee

Dedication

Σ΄ αυτούς που προσπαθούν να
κάνουν ένα βήμα μακρύτερα...

Translation: To those who are trying to
make one step further...

Biography

Panagiotis (Panos for short) Kampanakis was born in July of 1982 in Athens, Greece. He grew up in Athens and earned his first degree from the ECE Department of National Technical University of Athens (NTUA), in 2005. He then decided to pursue graduate studies in the US and North Carolina State University. During his academic years he has worked for companies like the National Company of Electricity, the National Company of Telecommunications, Fine & Com-ToNet in Greece and LSI Corporation in Austin, Texas in USA. Also during his graduate studies he has been a lab instructor and Teaching Assistant for undergraduate courses and conducted research in the team of the Cyber Defense Laboratory of NCSU. His current interests focus on Computer Networks, Sensor Network Security, Network Security and Cryptography. For more details about his biography you can visit his personal webpage.

Acknowledgements

I would like to thank Dr. Peng Ning for supporting me financially along with providing the necessary resources and giving me the opportunity to work in such a successful team. Also, I would like to thank Dr. Michael Devetsikiotis for his valuable advice and contribution in my graduate studies in North Carolina State University.

In addition, I have to thank our team in Cyber Defense Laboratory and especially An Liu whose ECC implementation and contribution was very important for this work. What's more, Dr. Geraldo Pelosi from Politecnico di Milano and Dr. Michael Scott from Dublin City University have been very helpful, provided me with valuable resources and answered technical questions.

Finally, I would like to thank my family and a couple of friends for supporting me psychologically and thus contributing in the successful completion of this work...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Elliptic Curve Cryptography (ECC) has gained great attention from the cryptography research community in recent years. The main reason is the small key sizes it uses that can lead to efficient cryptographic calculations. In our case, Cyber Defense Lab of NC State University and especially An Liu have worked in this field on sensor networks by producing the TinyECC package.

In the work that follows we are trying to extend his implementation and try to evaluate the feasibility of using some new computationally expensive cryptographic techniques on next generation sensor networks. The new implementation of TinyECC can now be used on Imote2, the new high-end mote by Intel. What is more, Identity-Based Cryptography functionality is also added in. More specifically, TinyECC now supports Tate Pairing which can be used for Identity-Based encryption or signatures, key establishment and more, which will be addressed below. On the other hand, we try to evaluate and study the feasibility of using such techniques on sensor motes. Thus, we study the performance; code size, timing and energy consumption for the cryptographic techniques implemented and we are proving that they can efficiently be used on new technology motes. What is more we are proving that the above techniques are very hard to be efficiently used on low-end, traditional motes because of their computational restrictions.

As far as some applications of the above tools are concerned, we also describe a scheme that could be used to establish keys on a sensor networks. Except that, we are also addressing the problem of using expensive computational techniques on networks consisting of both high and low-end motes. Such networks would suffer from the low-end motes being the computational bottleneck. To overcome this obstacle, we are trying to achieve workload

delegation to the more powerful motes.

Chapter 2 describes the motivation and states the research problem, Chapter 3 goes through some important research work in the field of Bilinear pairing and describes some significant security schemes. Then, Chapter 4 presents the background theory behind bilinear pairing and Chapter 5 describes our scheme for key establishment on sensor networks. After that, Chapter 6 shows the curve parameters we use for our evaluations and we proceed with Chapter 7 which evaluates our implementation work on real motes running TinyOS. Finally Chapter 8 concludes this thesis, it summarizes the accomplishments and mentions future research work that could be derived from it.

Overall, we believe that our work is opening the way to the new direction of Identity and Pairing-Based Cryptography by proving that it can be used on sensors. It also opens the way for next generation motes and proves that their robustness can greatly contribute to the security of our Wireless Sensor Networks (WSNs).

Chapter 2

Problem - Motivation

Before proceeding with the details of our work, it is worth presenting our motivations and the problem statement. After going through the definition and basics of Discrete Logarithm problem in Section 2.1 we proceed with the Elliptic Curves in Section 2.2 and the Discrete Logarithm on Elliptic Curves in section 2.2.1. Finally, Section 2.3 explains our intension of studying Bilinear Discrete Logarithm problem on next generation sensor networks.

2.1 Discrete Logarithm problem

Many cryptosystems in the past relied on the security provided by the hardness of the Discrete Logarithm (DL) problem. First, we have to define when a problem is considered to be hard. In an intuitive manner, a problem is considered to be hard when there is no algorithm that solves it in polynomial, in size of the input, time. Furthermore, hard problems can in turn be divided in subexponential and exponential time problems according to the time it takes an algorithm to solve them at best. We say that exponential time problems are harder than subexponential ones.

Thus, we can proceed with defining the DL problems as follows:
We regard these problems on the finite multiplicative group (G,*) of order m. Without loss of generality we can assume m of G to be prime, meaning G is cyclic and has base, say g. Let $h \in G$, such that $h = g^x \ mod \ q$ for some $x \in Z_q^*$. Given g and h, the DL problem is to find x. Notationwise it is $DL_g(h) = x$.

Also, a closely related problem is Computational Diffie-Hellman (CDH) problem:

Let $a, b \in \mathbb{Z}_q^*$. Given g, g^a, g^b, the CDH problem is to find $h \in G$ such that $g^{ab} = h \; mod \; q$. Notationwise $CDH_g(g^a, g^b) = h$.

Except from the problems presented above there are many relevant to DL problems which have been used in security protocols. Some of them include Decisional Diffie-Hellman (DDH), Weak Deffie-Hellman (WDH), Reversion of CDH (RCDH) k-strong Diffie-Hellman (k-SDH) and more. As far as the security of DL problems is concerned, there have been several attacks on the DL. They can be divided in *generic* and *Index Calculus* algorithms. Generic ones are also called square root methods and take exponential time to solve the DL problem. Index Calculus algorithms and its extensions (Number Field Sieve) take advantage of the properties of the multiplicative group and take subexponential time. These are the best tools at present to solve the DL problem. Thus, choosing the group parameters appropriately make the DL problem to be generally believed subexponential.

2.2 Elliptic Curve Cryptography

Elliptic curve cryptography [21] has a history of almost a hundred years. It has its origins in mathematics and number theory. Though, by considering the discrete logarithm problem and mapping it to discrete logarithms on groups of points of an elliptic curve we can realize the alternative use of elliptic curves in elliptic curve cryptography.

In the mid-nineties, researchers like Neal Koblitz and Victor Miller proposed using elliptic curves for public-key cryptosystems. Since then, elliptic curve cryptography has been widely studied and there are a significant number of systems and protocols where it is being used. Many of these systems are of commercial acceptance and this is only the beginning as the research community is constantly accepting and extending the capabilities and uses of elliptic curves.

2.2.1 Discrete Logarithm on Elliptic Curves

Although the DL problem is defined on a multiplicative group G of a finite field $\mathbb{Z}^*{}_q$, it can actually be defined on any group. And fortunately there exist groups in which solving the discrete logarithm problem is harder than in $\mathbb{Z}^*{}_q$. These, for example can be groups of points on an elliptic curve.

The main reason elliptic curves were chosen as an alternative for the discrete logarithm problem is that the Index Calculus method doesn't have a natural analogue in

this particular group of points. There have been a number of attempts to extend the Index Calculus method to elliptic curves but without success. So, it seems that there isn't a subexponential time algorithm to solve the discrete logarithm problem. As a result, elliptic curves are very appealing and can be used to achieve the same level of security with shorter keys, which is one of their main advantages.

We next give the definition of symmetric pairing. Note that it is not the formal definition of pairing which will be presented in Section 4.4. Let G_1 be an additive cyclic group of prime order m. Also, group G_2 is a cyclic multiplicative group of order m. Then the pairing is defined as follows: A map : $e : G_1 \times G_1 \rightarrow G_2$, called a symmetric pairing (i.e. pairing of points P, Q is $e\langle P, Q\rangle$). As we will also see later, one of the properties of pairing is bilinearity which means that if $a, b \in \mathbb{Z}^*{}_m$ then $e\langle aP, bQ\rangle = e\langle aP, Q\rangle^b = e\langle P, bQ\rangle^a = e\langle P, Q\rangle^{ab}$.

Let $(G_1, +)$ be an additive cyclic group and $(G_2, *)$ a multiplicative one, both of prime order m, and $e : G_1 \times G_1 \rightarrow G_2$ be a symmetric pairing. Then $h = e\langle P, P\rangle$ is the generator of G_2 and the Bilinear Diffie-Hellman (BDH) problem is defined as follows:
Let $a, b, c \in \mathbb{Z}^*{}_m$. Given P, aP, bP, cP the BDH is to compute $e\langle P, P\rangle^{abc}$. Notationwise $BDH_P(aP, bP, cP) = e\langle P, P\rangle^{abc}$

As for the hardness of BDH, so far there is no known method of solving BDH without first solving CDH. Thus, even though it isn't proved, the hardness of BDH is equivalent to hardness of CDH. Finally, many relevant problems have been studied and used in some protocols which include Decisional Bilinear Diffie-Hellman (DBDH), Decisional Hash Bilinear Diffie-Hellman (DHBDH) and others. These problems are beyond the scope of this work.

2.3 Sensor networks

It is well-known that sensor networks have been very popular in the research community for the passed years. A number of different reasons and its multiple uses have triggered great research activity to this promising field. Some of their applications involve military, forest monitoring, wildlife monitoring, surveillance, assembly line monitoring and many more. Now that this is gradually becoming more mature, the problems become even more interesting.

Sensor motes have had many restrictions which pose great barriers and challenges for the sensor network capabilities. Power consumption, energy and storage space restric-

tions, computational weakness, transmission inaccuracies are some examples of the restrictions in sensor networks. On the other hand, given the open environment where such networks are deployed, in the clear transmission and the challenging applications (i.e. military, monitoring), it is becoming obvious that there are also great security issues against the viability of such networks. These problems have been studied for the passed years and a number of different solutions have been proposed for many of them.

The Cyber Defense Lab of NC State Univ. has focused on cryptography and its efficient implementation on sensor motes. In more detail, given the security advantages of elliptic curves and the short keys being used, ECC seemed as a very interesting topic to study on sensor networks. In this way, short keys would save a lot of storage space and the arithmetic operations on smaller numbers would make public key cryptography on devices like motes more efficient than in traditional cryptography. An Liu has been working on ECC for almost 2 years and his work has given some very important results [27].

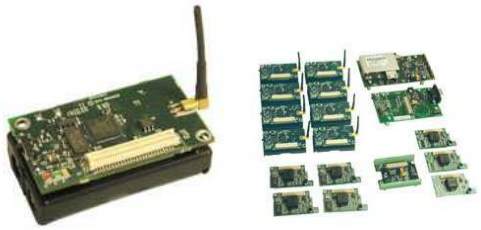


Figure 2.1: MicaZ and Wireless Sensor Network

Thus, specifically for this work, Bilinear Pairing's popularity has intrigued our interest for studying the feasibility of using it on sensor networks. If this was possible, then a lot of different protocols could be used (i.e. identity-based cryptography) to solve many problems (i.e. key exchange). Though, pairing is very expensive computationwise, even for traditional computers, as it is also proved in Shamus Software project. Thus, it would be a very challenging task to implement pairing on traditional motes. For example, the traditional MicaZ (see Figure 2.1) motes have a 6MHz 8-bit processor with 4KB of RAM and TelosB ones have a 16-bit processor with 10KB of RAM. Such devices would make efficient pairing very challenging.

Fortunately, as technology is getting more mature and Moore's law continues to hold companies like Intel and Sun have come up with new, modern, much more powerful

motes. For example, Imote2(see Figure 2.2) from Intel has a 32-bit XScale processor which can support 13-416MHz frequencies. It has 32MB flash memory, 256KB SRAM and 32MB of SDRAM. In addition, Sun has created Sun Spot which is a 190MHz processor with 512KB of RAM and a JVM built in. As we can easily understand, such devices seem more capable of performing expensive pairing operations and this is the task we concentrated on first. After working on the efficient implementation of pairing on Imote2, the results are presented in Chapter 7.

Figure 2.2: Imote2

What is more, it is easy to realize that the modern motes are much more expensive than the weak traditional ones. Thus having a network with only modern motes is an expensive ambition to follow. On the other hand, having a hybrid network which will include weak and cheap motes and some powerful ones seems like the direction that will widely be followed for cost efficiency reasons. In this way, powerful motes will have the more important (and expensive) roles and cheap motes will be the nodes performing simple operations. So, our next goal was to try to see how pairing could work in such networks and solve a key establishment problem. As we mentioned above, traditional motes cannot efficiently perform pairing operations, so we developed a scheme which would successfully delegate the hard part of the computational work to the powerful motes and would then require a reasonable amount of computation from the weak node for the scheme to complete. Thus, sensors would be able to perform key establishment securely without revealing any information to intruders that might be overhearing their communication. The scheme is described in detail in Chapter 5.

Chapter 3

Related Work

Lately, as already mentioned, the research community has been widely investigating Elliptic Curve Cryptography (ECC) and especially Bilinear Pairing. They have been focusing mostly on potential uses of these mechanisms because of their advantages against more traditional cryptographic techniques. Many schemes (some of them very successful) have been developed that can provide encryption, digital signatures, threshold decryption or signatures, key exchange, identity-based encryption and many more. Below, we will present some of the basic and most popular uses of the discrete logarithm problem and bilinear pairing. These can be used as proof of concept of the different tools bilinear pairing can provide for the security field. Also, there is much work done on the computation and optimizations of the bilinear pairing procedure but in this chapter we will focus mostly on its uses.

3.1 Diffie - Hellman

For many years, the discrete logarithm problem has been providing useful tools for key exchange, signatures and encryption. Diffie-Hellman key agreement [34] is describing a scheme that is widely used for key exchange between two parties. It has also been used in many cases and extensions of some security standards. The idea behind it is based on the DL problem. Each of the two parties A, B picks a secret number a, $b \in \mathbb{Z}^*_q$. By having a public key g, they can exchange their $g^a \ mod \ q$, $g^b \ mod \ q$ without revealing their secrets and thus establish a shared key $g^{ab} \ mod \ q$ unknown to the public.

Another traditionally famous scheme based on DL is the ElGamal digital signa-

ture [17]. According to ElGamal, a party having a long-term public/private key pair can use a temporary key to sign his messages which will enable others to verify them by just using the temporary public key. Again, the underlying property that is exploited is the DL relation between the public/private, long-term/short-term keys.

In addition, a variety of publications have been based on DL problem and some of them try to come up with complete solutions to certain problems. An example would be [3] which attempts to provide signatures by using symmetric encryption, hash functions and the DL problem. What is more, one can find numerous publications (among which the famous RSA standard) on this area that contribute to the realization of how important the DL problem has been for cryptography for the passed years.

3.2 Bilinear Pairing

As mentioned above, Bilinear Pairing has been of great attention within the cryptography and security research community lately. The advantages of ECC along with the unique characteristics of pairing itself make it very appealing to achieve traditional cryptography goals with less overhead. Only in the last 5-7 years, more than 250 papers have been published in the area and some of them can be found in The Pairing-Based Crypto Lounge webpage maintained by P. Barreto. In these papers, many schemes have been proposed that claim to achieve better security with more useful attributes. Some of them are presented below and show the significant role of bilinear pairing in cryptography. Section 3.2.1 presents some pairing based encryption schemes whereas section 3.2.2 presents signature ones and section 3.2.3 some key-sharing techniques. Finally, section 3.2.4 describes some other proposed tools that are based on pairing.

3.2.1 Encryption schemes

In 2001 Boneh and Franklin [13] proposed a scheme that uses bilinear pairing to provide signatures for a message M using the identity of the signing party. Their technique uses functions that map identity names to curve points that belong to a certain group. Then, by having a global public/secret key pair and picking a secret, private one, a signature can be generated by using the global public key, the private picked key and by XOR-ing M with the result of pairing between the mapped point and the public key. Signature generation

is more computationally expensive than verification but both require an expensive pairing computation.

Gentry and Silverberg in 2002 [20] suggested a method that can achieve hierarchical identity based encryption (HIDE) and dual hierarchical identity based encryption (Dual-HIDE). In other words, it enables a single PKG to distribute keys to a large number of network nodes. This is a heavy job, but with HIDE a root can delegate private key generation to lower level PKGs. Authentication and private key generation can be done locally. So, for example if A wants to speak to B, he only has to obtain his parent PKG and identity. On the other hand, in Dual-HIDE even if A and B are in different hierarchy levels, if they have a common ancestor they can send encrypted information between each other. HIDE and Dual-HIDE have the advantage of damage control. Though, they also require multiple pairing computations by the encrypter/decrypter, which is expensive.

Similar schemes have been investigated in [11] by Boneh and Boyen. There, they describe methods for Identity based encryption (IDE) and Hierarchical ID-Based encryption (HIBE). One of their scheme advantages is that it provides identity based encryption without the use of a Random Oracle and is resilient against selective-ID chosen ciphertext attack.

3.2.2 Signature schemes

Boneh, Lynn and Shacham in 2001 [15] proposed the Blind Short Signature Scheme. According to it, a private key distributed to different parties along with functions that map identities to points and hash functions that also map messages into points are used to perform a signature generation and verification. This technique provides an elegant way for digital signatures and requires a pairing computation for both the signer and verifier.

Also Boldyreva in [10] describes how bilinear pairing can be used to ask a signer to sign a message without revealing information about the message. It also mentions how to allow a group of users to jointly sign a message which will be used to convince the verifier that each group member participated in the signature. What is more, Zhang and Kim in [40] worked in the same context. Actually, they performed similar operations to ask a signer to blindly sign a message with the differentiation of using the signer's identity to perform this operation. Though, these schemes require more computational effort from the verifier than from the signer which would degrade their performance.

In [14] the authors propose how multiple signatures of different messages from different signers can be aggregated in one single signature from which it can be inferred that indeed a signer has signed a specific message. But again the cost of verification is very high for this scheme. Another very useful signature technique is presented in [41]. In this paper, a so called ZSS signature is generated by using only a secret key and a hash function on a message. In 2002, Hess [22] presents an alternative of identity based signatures with the advantages of allowing some pre-computation to alleviate the computational effort and the efficiency in terms of communication requirements. Finally, [12] describes how short signatures can be generated without the use of a random oracle and allow for some pre-computation to alleviate the computational burden.

3.2.3 Key sharing-agreement schemes

Bilinear pairing has also been useful for performing key exchange and establishment. This is what we attempt to achieve for sensor networks in our work in Chapter 5. In the previous years Sakai, Ohgishi and Kasahara in 2000 [35] came up with a simple protocol that achieves key exchange between two parties. Both parties must have a common secret embedded in their public key which corresponds to their identity. Thus, by using functions that map identities to points and without exchanging any messages the two parties can share a secret key for their communication. The protocol requires both parties to perform a pairing computation which is expensive especially for low power nodes. This problem for hybrid sensor networks with both low and high power motes is addressed in Chapter 5.

Joux, on the other hand, extends the above protocol in [24] to develop a protocol for a three party agreement in only one round of interaction. According to this protocol the parties exchange public keys in one round and they can compute their shared communication key using bilinear pairing. Even though Joux's technique offers a very simple and communication-wise cheap way of establishing keys between three parties it is vulnerable against man-in the-middle attacks. Al-Riyami and Paterson tried to overcome this drawback by proposing their authenticated key agreement in [4]. Moreover, Joux's protocol has been extended to perform key exchange between multiple parties in [9].

3.2.4 Other schemes

Except from the basic uses that have been mentioned above, bilinear pairing has been used in many other schemes and techniques. Some of them are mentioned here. In [10] Boldyreva describes how a private key shared among multiple servers using Shamir's secret sharing scheme can be used by each server to sign a message M (partial signature). Then from the partial signature a verifier can verify if a server honestly participated in the signature and with $(t + 1)$-honest servers he can reconstruct the signature (where t is a threshold value). Such schemes involve many computations as the verifier has to verify shares from multiple servers and reconstruct the signature. Another similar scheme that does decryption was proposed by Libert and Quisquater in [26]. There, given an encrypted message, they extend [13] in such a way that a fixed PKG plays the role of a trusted dealer that distributes shares to users which they, in turn, can decrypt and send to a recombiner that in the end will recover the decrypted message. Finally, Kim and Kim in [25] present a way for a prover to prove its identity to a verifier. This is a multi-round protocol and requires more computations from the verifier.

3.3 ECC on sensor networks

In order to study the performance of Bilinear pairing on sensor networks we had to base on the previous work done in the field. More specifically, An Liu's TinyECC package [27] was extended to support Bilinear pairing and it's operations. TinyECC previously implemented ECC and ECDSA [1] signatures on all the Certicom SECG curves [33] for MicaZ and TelosB motes running TinyOS. Also to generate the appropriate parameters for Tate pairing we used the MIRACL library. Moreover, to study our hybrid networks workload conserving scheme we used the exponentiation using Lucas functions described in [36].

Chapter 4

Theory - Preliminaries

ECC and bilinear pairing, as we already mentioned, are widely studied in various ways and there have been many publications in the area during the passed years. [18, 28, 16, 32] summarize this work and present the basic considerations and research activity. In this chapter we will present the background for the implementation on sensor networks and the research work that will follow. Sections 4.1, 4.2, 4.3 and 4.4 describe the theory preliminaries, Section 4.5 goes through the implementation details and optimizations and Section 4.6 depicts some of the security considerations lying into pairing.

4.1 Elliptic Curves

We start by going through some background knowledge on Elliptic Curve Cryptography [2]. First of all, an elliptic curve by the (affine) Weistrass equation is

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{4.1}$$

The elliptic curve is the set of points (x,y) that satisfy Equation 4.1 along with point $\mathcal{O}$ which doesn't exist in reality and is called point of infinity. E is said to be defined over K, denoted E/K, where K is a field and $a_1, a_2, a_3, a_4, a_6 \in K$. We denote E(K) the set of K-rational points (i.e. points with both coordinates in K and the point of infinity). Observe that by definition, we can write $E = E(\overline{K})$ where $\overline{K}$ is the algebraic closure of K.

In our case, we will be working with elliptic curves over a finite field, thus $K = \mathbb{F}_q$ and $\overline{K} = \cup_{i \geq 1} \mathbb{F}_{q^i}$. Also, the curves we will use have to have certain properties in order for us to be able to perform the optimizations described in section 4.5.2. So, the curves we will

use are supersingular of the form

$$E : y^2 = x^3 + x \tag{4.2}$$

For the points of an elliptic curve, there exists an operation called tangent-and-chord method. It is written additively and has point of infinity $\mathcal{O}$ as the zero element. Let points $U(x_1, y_1)$, $V(x_2, y_2) \in E\backslash\{\mathcal{O}\}$. The point $-P$ is given by $(x_1, -y_1)$. If $P \neq -Q$, then for the point $P + Q$ with coordinates (x_{U+V}, y_{U+V}) it is

$$\begin{cases} x_{U+V} = (\frac{y_2-y_1}{x_2-x_1})^2 - x_1 - x_2 \\ y_{U+V} = (\frac{y_2-y_1}{x_2-x_1})(x_1 - x_{U+V}) - y_1 \end{cases}$$

Figure 4.1 (reprinted from [23]) presents the point addition operation graphically.

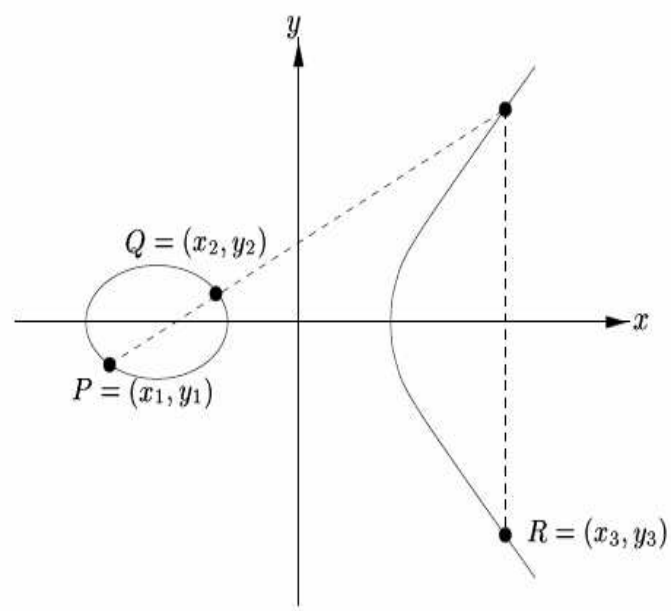


Figure 4.1: Point addition

Since addition operation exists on elliptic curves, we can also define scalar doubling and multiplication. Given a point $U(x_1, y_1)$, the double of it (x_{2U}, y_{2U}) is defined to be

$$\begin{cases} x_{2U} = (\frac{3x_1^2+a}{2y_1})^2 - 2x_1 \\ y_{2U} = \frac{3x_1^2+a}{2y_1}(x_1 - x_{2U}) - y_1 \end{cases}$$

The graphical representation of point doubling is shown in Figure 4.2 (reprinted from [23]).

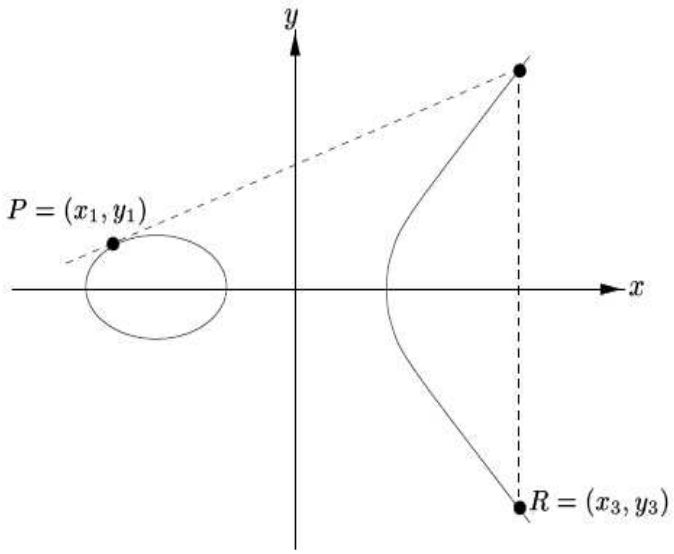


Figure 4.2: Point doubling

Thus, for $m \in \mathbb{Z}$ and $P \in E$, the multiplication is given by:

$[m]\,\mathrm{P}$=P+P+...+P (m times)
$[0]\,\mathrm{P}=\mathcal{O}$, $\qquad$ $[-m]\,\mathrm{P}=[m]\,(\text{-P})$

We say that the order of a point P is m, when m is the minimum number that satisfies $[m]P = \mathcal{O}$ (in this thesis mP will also be referred as $[m]P$). Further, we write that

$$E(K)\,[n] = \{P \in E(K) : [n]\,P = \mathcal{O}\}$$

is the subgroup of n-torsion points, where an n-torsion point is point whose order divides n.

Finally, order $\#E(\mathbb{F}_q)$ of an elliptic curve E in $\mathbb{F}_q$ is the number of points in $E(\mathbb{F}_q)$. For the supersingular curves we will be using for our work, the curve order is $\#E(\mathbb{F}_q) = q+1$.

4.2 Function Field of an Ellitpic Curve

If we have a curve E of the form in Equation 4.1 then we can define function F such that

$$F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 \tag{4.3}$$

- The coordinate ring $\mathbb{F}_q[E]$ is the integral domain $\mathbb{F}_q = \mathbb{F}_q[x,y]/(\mathbb{F}_q)$. Additionally, we can define the coordinate ring $\overline{\mathbb{F}}_q = \overline{\mathbb{F}}_q[x,y]/(\mathbb{F}_q)$ and its elements are called regular functions.
- A field of fractions of $\overline{\mathbb{F}}_q[E]$ is called a function field $\overline{\mathbb{F}}_q(E)$ of E over $\overline{\mathbb{F}}_q$. And the elements of $\overline{\mathbb{F}}_q(E)$ are called rational functions.

It is worth noticing that each rational function can be represented as $u(x)+yw(x)$ where $u(x), w(x) \in \overline{\mathbb{F}}_q[x]$. Now, if we have a regular function $v(x,y) = u(x) + yw(x)$ we define its degree as $deg(v) = max\{2deg_x(u), 3 + 2deg_x(w)\}$.

If there is a rational function f and a point of the elliptic curve $P(x_0, y_0)$ then f can be called *regular* or *defined* if there is a representation of it as g/h where g, h belong to the coordinate ring $\mathbb{F}_q[E]$ such that $h(x_0, y_0) \equiv h(P) \neq 0$. Then, we can say that if f is regular and $f(P) = 0$, it has a zero at P. If it isn't regular then it has a pole at P ($f(P) = \infty$). Also, we can define

$$f(\mathcal{O}) = \begin{cases} 0, \text{ if } deg(g) < deg(h) \\ \text{(highest degree of g)/(highest degree of h)}, \text{ if } deg(g) = deg(h) \\ \infty, \text{ if } deg(g) > deg(h) \end{cases}$$

For each point of the elliptic curve there exists a rational function $u \in \overline{\mathbb{F}}_q(E)$: $u(P) = 0$, such that $\forall f \in \overline{\mathbb{F}}_q(E)$, it can be $f = u^d s$ where $s \in \overline{\mathbb{F}}_q(E), d \in \mathbb{Z}$, $s(P) = 0$ or ∞. This u is named uniformizing parameter for P. Next the order of f at P is defined to be d, $ord_P(f) = d$. Point P is a zero of f if and only if $ord_P(f) > 0$ and the multiplicity of that zero is $ord_P(f)$. On the other hand, P is a pole of f if and only if $ord_P(f) < 0$ and the multiplicity of that pole is $-ord_P(f)$. $ord_P(f) = 0$ if and only if f is defined at P.

Finally, let $f \in \overline{\mathbb{F}}_q(E)$. Then f has finite number of zeros and poles on points of E. Furthermore,

$$\sum_{P \in E} ord_P(f) = 0 \tag{4.4}$$

4.3 Divisor Theory

The group of divisors $Div(E)$ is the free abelian group generated by a formal sum of points in E.

$$Div(E) = \left\{ \sum_{P \in E} m_P \langle P \rangle : m_P = 0 \ \forall \text{ but finitely many } P \in E \right\}$$

We can define

- The degree of a divisor D to be $deg(D) = \sum_{P \in E} m_P$.
- The support of a divisor D to be $Supp(D) = \{P \in E | m_P \neq 0\}$.
- The subgroup of $Div(E)$ of divisors of degree zero is $Div^0(E)$.
- If r is a rational function then $div(r) = \sum_{P \in E} ord_P(r) \langle P \rangle$. $div(r)$ is a divisor because a rational function has finite number of poles and zeros. And a divisor D is called principal if $D = div(r)$ for some $r \in \overline{\mathbb{F}}_q^*$.
- The subgroup of $Div(E)$ of principal divisors is $Prin(E)$.
- Equivalent are two divisors D_1 and D_2 if $D_1 - D_2 \in Prin(E)$ and equivalence is denoted as $D_1 \ D_2$.

It can be shown that for every $D \in Din^0(E)$, there is a unique point $Q \in E$: $D \ \langle Q \rangle - \mathcal{O}$.

Now let D be a divisor and f be a rational function: $Supp(D) \cap Supp(div(f)) = 0$. Then, we can define a value of f at D to be

$$f(D) = \prod_{P \in Supp(D)} f(P)^{m_P} \tag{4.5}$$

.

4.4 Bilinear Pairing

There are two techniques which have been proposed for computing bilinear pairing and are based on the Bilinear Diffie-Hellman problem. These are Weil and Tate Pairing. For our work, we will focus on the second technique but for the completeness of the theory presentation we will go through the formal definition of both.

4.4.1 Weil pairing

Let E be an elliptic curve defined over $\overline{\mathbb{F}}_q$. Let m be a positive integer coprime to q and $\mu_m \subset \overline{\mathbb{F}}_q$ be the m^{th} root of unity ($\mu_m m \subset \overline{\mathbb{F}}_{q^k} | (q^k - 1)$).

Let $P, Q \in E[m]$. Let $D_1, D_2 \in Div^0(E)$: $D_1 \ \langle P\rangle - \langle \mathcal{O}\rangle$, $D_2 \ \langle Q\rangle - \langle \mathcal{O}\rangle$ and $Supp(D_1) \cap Supp(D_1) \neq \oslash$. Then mD_1, mD_2 are principal divisors. So, if $f_{D_1}, f_{D_2} \in \overline{\mathbb{F}}_q(E)$: $div(f_{D_1}) = mD_1$, $div(f_{D_2}) = mD_2$ the Weil pairing function is

$$e_m : E[m] \times E[m] \rightarrow \mu_m \text{ defined by } e_m \langle P, Q\rangle = \frac{f_{D_1}(D_2)}{f_{D_2}(D_1)} \tag{4.6}$$

The Weil pairing has to satisfy certain properties:

1. Well-defined: $e_m \langle P, Q\rangle$ has to be independent of the choice of $D_1, D_2, f_{D_1}, f_{D_2}$.
2. Identity: $\forall P \in E[m] e_m \langle P, Q\rangle = 1$.
3. Non-Degeneracy: For a certain $P \in E[m]$, $e_m \langle P, Q\rangle = 1$, $\forall Q \in E[m]$ if and only if $P = \mathcal{O}$.
4. Bilinearity: $\forall P, Q, R \in E[m] e_m \langle P + Q, R\rangle = e_m \langle P, R\rangle \cdot e_m \langle Q, R\rangle$ and $e_m \langle P, Q + R\rangle = e_m \langle P, Q\rangle \cdot e_m \langle P, R\rangle$.
5. Alternation: $\forall P, Q \in E[m] e_m \langle P, Q\rangle = e_m \langle Q, P\rangle^{-1}$
6. If $E[m] \subset E(\mathbb{F}_{q^n})$, then $e_m \langle P, Q\rangle \in \mathbb{F}_{q^n}$, $\forall P, Q \in E[m]$.

4.4.2 Tate pairing

Tate Pairing has similarities with Weil Pairing. Though, it requires half the rational function computations which makes it more attractive for implementation on sensors.

Let E be an elliptic curve defined over $\mathbb{F}_q$. Let m be a positive integer coprime to q and k be a positive integer such that $m|(q^k - 1)$. From now on k will be called the *embedding degree* of the curve with respect to m. As we will see in Section 4.6, k is a very important factor for the security of the Bilinear Diffie-Hellman problem.

Let $P \in E[m]$ and $Q \in E$. Let $D_1, D_2 \in Div^0(E)$: $D_1 \ \langle P\rangle - \langle \mathcal{O}\rangle$, $D_2 \ \langle Q\rangle - \langle \mathcal{O}\rangle$ and $Supp(D_1) \cap Supp(D_1) \neq \oslash$. Then mD_1 is a principal divisor. So, if $f_{D_1} \in \overline{\mathbb{F}}_q(E)$: $div(f_{D_1}) = mD_1$, the Tate Pairing function is

$$t_m : E(\mathbb{F}_{q^k})[m] \times E(\mathbb{F}_{q^k})/mE(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^m \text{ defined by } t_m \langle P, Q\rangle = f_{D_1}(D_2) \tag{4.7}$$

Tate pairing has to satisfy certain properties:

1. Well-defined: $t_m \langle \mathcal{O}, Q\rangle \in (\mathbb{F}^*_{q^k})^m \ \forall Q \in E(\mathbb{F}_{q^k})$ and for $P \in E(\mathbb{F}_{q^k})[m]$ $t_m \langle P, Q\rangle \in (\mathbb{F}^*_{q^k})^m$. Also the value of $t_m \langle P, Q\rangle$ is independent of the choice of D_1, D_2, f_{D_1}.

2. Non-Degeneracy: For $P \in E[m]$, $t_m \langle P, Q \rangle = 1 \; \forall Q \in E$ if and only if $P = \mathcal{O}$.

3. Bilinearity: $\forall P, Q, R \in E[m]$, $t_m \langle P + Q, R \rangle = t_m \langle P, R \rangle \cdot t_m \langle Q, R \rangle$ and $t_m \langle P, Q + R \rangle = t_m \langle P, Q \rangle \cdot t_m \langle P, R \rangle$.

It is also worth mentioning that $t_m \langle P, Q \rangle$ is not a unique value because it is an equivalence class in $\mathbb{F}^*_{q^k}/(\mathbb{F}*_{q^k})^m$. For a unique outcome we need to eliminate the $m - th$ powers. This is done by raising to the power of $(q^k - 1)/m$. Consequently, the pairing value is an $m - th$ root of unity.

4.5 Tate Pairing computation

We chose Tate Pairing as the technique to insert Identity-Based Cryptography in sensor networks. The reason was that we wanted to do it as computationally cheap as possible. Tate Pairing is the more suitable solution in contrast to Weil Pairing. In this section we will present how it is computed and the optimizations we used to improve the timing results of the algorithm.

4.5.1 Miller's algorithm

In 1986, Miller [29] proposed an algorithm for short functions on curves. About two decades later, this algorithm became the prevalent algorithm to compute the Weil and Tate Pairing (rational functions). Additionally, Baretto et al. [7] in 2002 suggested and formalized the computation and some of the optimizations. Miller's algorithm is shown in Algorithm A.1.

As we can see, the algorithm works iteratively. Each iteration involves a point doubling and/or not a point addition. Also, in each iteration we have evaluations of the rational functions at certain points and their division. The number of iterations depends on the size of the order m of point P. To make it more concise, the g functions used in the algorithm are defined as follows:

If we have two points $U(x_1, y_1), V(x_2, y_2)$ then the line defined by these points is

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \equiv \lambda \Rightarrow y - y_1 + \lambda(x_1 - x) = 0$$

Then, the function $g_{U,V}(Q)$ at point Q is the evaluation of this point coordinates on this line

$$g_{U,V}(Q) = \begin{cases} 1 & , Q \in \{\mathcal{O}, P\} \\ y_Q - y_1 + \lambda_1(x_1 - x_Q) & , \text{otherwise} \end{cases} \tag{4.8}$$

On the other hand, if we have one point $U(x_1, y_1)$ then the line between points $U(x_1, y_1), -[2]U(x_1, y_1)$ where $-[2]U(x_1, y_1) = U'(x_{2U}, -y_{2U})$, is

$$\begin{cases} x_{2U} = (\frac{3x_1^2+a}{2y_1})^2 - 2x_1 \\ y_{2U} = \frac{3x_1^2+a}{2y_1}(x_1 - x_{2U}) - y_1 \end{cases} \Rightarrow \lambda_2 = -\frac{y_1 + y_{2U}}{x_{2U} - x_1} = \frac{3x_1^2 + a}{2y_1}$$

So, in the same way function $g_{U,U}(Q)$ at point Q is

$$g_{U,U}(Q) = y_Q - y_1 + \lambda_2(x_1 - x_Q). \tag{4.9}$$

Finally, let point $U(x_1, y_1)$. Then the vertical line that crosses U is $x - x_1 = 0$. And $g_U(Q)$ is

$$g_U(Q) = \begin{cases} 1 & , U == \mathcal{O} \\ x_Q - x_1 & , \text{otherwise} \end{cases} \tag{4.10}$$

Please note that above we presented the formal definition of Miller's algorithm but in section 4.5.2 we will see the actual format of the algorithm used. There, a number of different techniques are used in order to make the computation efficiently.

4.5.2 Optimizations

Below, we present the actual implementation techniques we used for our own implementation. They were chosen in order to make the computation quicker and more efficient which is imperative for sensor motes.

In [8] and [7] M. Scott, P. Barreto et al. propose a method of optimizing Tate pairing. In more detail, they propose a twist of the original curve $E(\mathbb{F}_q)$: $y^2 = x^3 + ax + b$ over $E(\mathbb{F}_{q^d})$ to be $E'(\mathbb{F}_{q^d})$: $y^2 = x^3 + u^2ax + u^3b$ where $u \in \mathbb{F}_{q^d}$ is a quadratic non-residue. Then, modified Miller's algorithm can be simplified and point operations are performed

only in $E(\mathbb{F}_{q^d})$ which improves the performance of algorithm. In our case $d = 1$ and thus point operations remain in $E(\mathbb{F}_q)$. Alternatively, the above technique is described in the literature [18] as a distortion map or an efficient endomorphism. It is proven that if there is a distortion map $\phi(Q)$ for the point Q of curve $E(\mathbb{F}_q)$ then the denominators for the computation of $t_m \langle P, Q \rangle$ in Miller's algorithm can be discarded without affecting the result value.

Though, there is not always an endomorphism for all different elliptic curves. Actually, endomorphisms exist for supersingular curves. More importantly, for the curve $E(\mathbb{F}_q)$: $y^2 = x^3 + ax + b$ which we are using for our work with $a = 1,\ b = 0,\ q \mod 4 = 3$ the distortion map of a point $Q(Qx, Qy) \in E(\mathbb{F}_q)$ is defined to be

$$\phi(Q) \equiv \phi(Qx, Qy) = (-Qx, iQy)$$

where $i \in \mathbb{F}_{q^2},\ i^2 = -1$.

Actually, such methodology used in accordance with the Lucas functions mentioned later decrease the size of operations that do not involve pairing, like key generation and point transmission in half, whereas points in $E(\mathbb{F}_{q^2})$ are only used for the pairing computation and are then mapped again in points in $E(\mathbb{F}_q)$. More details about distortion maps and how they can be used can be found in the survey by Joux [5]. After using distortion maps, the modified algorithm for Tate pairing becomes Algorithm A.2.

Another important factor in the choice of m. In the binary expansion of m : $(m_{t-1}m_{t-2}...m_0)_2$, if the number of bits $m_i == 1$ is small then the second step (if-statement) of Miller's algorithm can be avoided in many iterations. So, choosing a prime m for which the Hamming weight is as low as possible would make our algorithm more efficient. Such primes are Solinas primes [39] of the form $m = 2^b \pm 2^a \pm 1$. For our implementation, the m of the 512-bit curves is such that we have only one point addition evaluation. What is more, int the last iteration of Miller's algorithm if $i = 0$ then the evaluation $g_{V,P}(\phi(Q)$ is avoided because $V + P = \mathcal{O}$.

Now, we can proceed with exponentiation that is performed in the end of the Algorithm A.2 and is used to make sure we have a unique pairing values (section 4.4.2). If f is number $u + i \cdot v$ then we have to perform $f^{\frac{q^k-1}{m}}$. In our case $k = 2$, thus

$$\frac{q^k - 1}{m} = (q - 1)\frac{q + 1}{m} = (q - 1)c$$

But also

$$(u + i \cdot v)^{q-1} = \frac{(u + iv)^q}{u + i \cdot v} = \frac{u - iv}{u + i \cdot v} = \frac{u^2 - v^2}{u^2 + v^2} - i\frac{2uv}{u^2 + v^2}$$

So,

$$f^{\frac{q^k-1}{m}} = (\frac{u^2 - v^2}{u^2 + v^2} - i\frac{2uv}{u^2 + v^2})^c$$

In this way, we are saving a lot of additions and multiplications that would be needed for the complete exponentiation of f.

In modified Miller's algorithm, the point operations can also be used to perform the slope calculations used in the evaluation of the g functions. That is because these values are also used in for the point operations. This trick saves calculation time and decreases the code used.

It is also worth noting that for the final exponentiation the Lucas function is used in order to get into $\mathbb{F}_{q^2}$ extension field as proposed in [39]. The Lucas function algorithm used is presented in Algorithm A.3. More information on how the exponention using Lucas functions works and the use of twists on curves can be found in [38, 8, 36].

To summarize, it can be proven that Algorithm A.2 can be changed in the final exponentiation step to be

$$f^{\frac{q^k-1}{m}} = 2^{(-1)} \cdot Lucas[2 * (u^2 - v^2)/(u^2 + v^2), (q + 1)/m]$$

and also that in the evaluation of the g-function, it is

$$g = [Vy - \lambda_j(Qx + Vx)] + i \cdot (-Qy)$$

where $f = u + i \cdot v$ and all the operations are modular q.

The overall computational advantage is considerable as we will see later in Chapter 7. Not to mention that there were many cases were we used pre-computation to avoid inversions in F_q and shifting to perform simple doublings, triplings and halvings in F_q. Also, since the pairing computation point P is usually a constant key the line computation used in the g functions is the same and thus we can importantly save a lot of time by storing the lines and reusing them. What is more, using projective coordinates can save us from a lot of divisions in the evaluation of the slope in each iteration. This optimization along with Baretto reduction can be proven to be very useful and were inserted by An Liu as an extension to this work.

However, the literature proposes some more optimizations which weren't applicable in our case for a number of different reasons. In [19] it is argued that divisions are more expensive than multiplications. So, it is suggested to compute function f as the quotient of functions f_1 and f_2 using only multiplications. In other words, $f = f_1/f_2$ and only a single division is required for the final computation of f. Though, in our case, the denominator elimination optimization makes such an attempt unnecessary. Additionally, [18] argues that if $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{p^k})$ then the computation of Tate pairing $e\langle P, Q\rangle$ is much faster than $e\langle Q, P\rangle$, but in our case the distortion map makes it redundant. Another suggestion made in [19] is to do a pre-calculation of $[n]P$ for all possible values of n inside a 'window' of a certain number of bits. Though, such scheme would cause much memory burden which we wanted to avoid given the restrictions of sensor nodes. Finally, some other further memory-wise challenging optimizations are described in [37].

4.6 Security Considerations

Recall that Elliptic Curves were proposed in the first place because they could provide the same security levels with DL-based systems using shorter keys. However, it turned out that there exist certain reductions to less secure groups on certain curves (i.e. supersingular). Thus, to avoid such cases, elliptic curves have to be chosen with great care. The ideal case would be to choose curves at random and perform some specific security checks. Though, this technique is not always capable to provide curves suitable for pairing.

In our case, the security of the Bilinear Diffie-Hellman problem depends on the embedding degree of the curve. That is because the Weil and Tate Pairing are bilinear maps that map a pair of points of an elliptic curve over a finite field $\mathbb{F}_q$ to a multiplicative group ${\mathbb{F}^*}_{q^k}$ of an extension field of degree k. Hence, in order to be able to efficiently evaluate the pairing, we need k to be sufficiently small, but at the same time we need it to be large to provide sufficient security. On the whole, we are interested in curves with $k < (\log q)^2$. [6] shows that there are very sparse curves satisfying this condition. Thus, choosing a curve at random will with overwhelming probability give a non-suitable for pairing curve. The curves that have been proposed as more suitable for pairing so far are supersingular with embedding degree 2 and MNT-curves with higher embedding degrees (i.e. $k = 6$). For a comparison between supersingular and MNT-curves in the Bilinear Pairing context, the reader can go through [31].

For our work, we chose supersingular curves because they can provide sufficient security with the current tools of our current TinyECC implementation. If we chose curves with higher embedding degrees then we should be able to provide operations in the extension field $\mathbb{F}_{q^{\frac{k}{2}}}$ which isn't supported in TinyECC. As far as the security of such curves is concerned, to ensure the hardness of the Bilinear Diffie-Hellman problem we have to make the CDH hard in both the $E(\mathbb{F}_q)$ and $E(\mathbb{F}_{q^k})$. Thus, the desired value of k is a trade-off between efficiency and security. Curves that satisfy these criteria are with $k \leq 6$. On the other hand, [30] argues that an embedding degree of 6 doesn't add any further security in contrast to embedding degree equal to 2. It also argues that supersingular curves with embedding degree 2 can provide sufficient security for current cryptosystems when the prime field number q is 512-bits long. So, even if in some way we break the advantage of ECC which provides same security with smaller keys, the advantages provided by Bilinear Pairing are still important (Identity-Based Cryptography) and we still have a reduction on the key sizes.

In conclusion, we have to remind the reader that even though the BDH is believed to be equivalent to CDH there is no proof of this assumption. And there also might be cases where the BDH is less secure than in general. Additionally, the pairing curve structure might provide cryptanalysts with tools to attack systems. Only the passage of time and maturing of these techniques will provide with enough information about the security, vulnerabilities and possible extra measures for these systems.

Chapter 5

Key establishment in WSNs

Now, that we have gone through the details of the implementation of Bilinear Pairing, it is important to see its uses in sensor networks. As someone might imagine there could be many applications some of which have been discussed in Chapter 3. Though, most of these applications require multiple calculations of Tate Pairing which might make them expensive if they are performed on a per message basis. For example, digital signatures using pairing on a per message basis might prove to be very expensive for both the signer and the verifier. This overhead might also cause an extra DoS vulnerability on the scheme. Not to mention the signature size, which is 64-bytes (for a 512-bit curve) and decreases the maximum possible payload size to 38-bytes (ZigBee standard).

In this chapter, we will present how Identity-Based Cryptography and Bilinear Pairing can be successfully applied in next generation sensor networks. More specifically, we will present a very useful application of Tate Pairing for sensor networks whose relatively seldom use can alleviate the high expense of Pairing. In Section 5.1 we will show how keys can be established in a network consisting of modern nodes only (i.e. Imote2). In Section 5.2 we will show how this scheme would work in a hybrid network consisting of traditional (MicaZ) and modern (Imote2) nodes.

5.1 Next generation networks

In the case where there are no financial restrictions, a sensor network can be composed by new, modern motes, like Imote2s. These nodes, as we will show in Chapter 7 can efficiently compute the Tate Pairing of two points.

If we have a certain authority that chooses a secret number $s \in \mathbb{Z}^*_q$ then it can pre-distribute a secret sP_i to every node before the network deployment. P_i is the point that corresponds to the identity i of the mote. Thus, there also has to be a well-defined function that maps identity i to a unique point P_i of the elliptic curve shared by the sensor network. Consequently, each powerful node has the ability to establish a shared session key with any of its neighbors using Tate Pairing without requiring any communication between the parties.

In a more formal manner, if node i wants to establish a key with node j then he will compute

$$t_m \langle sP_i, P_j \rangle$$

Note that only i knows sP_i and P_j can be derived easily by i using the map-to-point function on the id j of node j. On the other hand, j will compute

$$t_m \langle P_i, sP_j \rangle$$

Because of the bilinear property of pairing, the two keys computed by each of the motes will be equal and thus they can establish pair-wise session keys without exchanging any messages. Figure 5.1 demonstrates the formulation of this scheme.

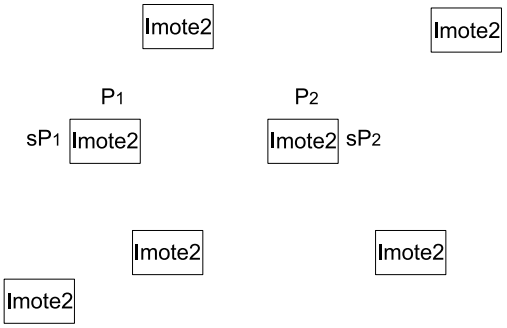


Figure 5.1: Next generation network

From a security point of view, the master key s isn't stored in any of the nodes, thus not giving an intruder the opportunity to compromise any of them and steal it. In addition, the scheme requires one pairing computation per session key. Thus the maximum number of pairings a node would have to compute would be the number of its neighbors. For the current timing results of pairing on Imote2, such a situation makes this scheme

unsusceptible to DoS attacks. Finally, providing Identity-Based Encryption between two nodes would work in a very similar way, which is beyond the scope of this work to present.

5.2 Hybrid networks

We believe that because of the high cost of Imote2s and high-end motes in general, the future sensor networks will not purely consist of them. Rather, they will consist of many low-end, cheap motes (MicaZ, TelosB) and a few more expensive ones (Imote2). In this way, the cost will be kept low. The most demanding functionality will be distributed on the powerful motes whereas the traditional ones will perform inexpensive operations. For example, the data aggregation in a network will be performed on the Imotes which will be chosen as parents in the upper levels of the aggregation tree and the low-end ones will perform sampling and forward the samples to their parents in the tree.

Such a situation would complicate things in a sensor network using Tate Pairing to perform key establishment. That is because a low-end sensor having to establish a key with another low-end sensor or with an Imote2 would be a very challenging operation. MicaZs or TelosBs cannot efficiently compute Tate Pairing, so key establishment has to be done in a different manner.

In this case, we can assume that each node will delegate an important part of the computational work to a nearby Imote2 without revealing his secret. Then he will perform the final part of the key establishment on its own. In more detail, lets assume that every low-end node has locally stored a secret $s_i \in \mathbb{Z}^*_q$ and its corresponding point s_iP_i. Once again P_i is the point that is derived from the identity of i (using map-to pount function) of each node. High-end nodes have stored their own s_iP_i only. When a low-end node i wants to exchange keys with another low-end node j, it will ask an Imote2 to compute

$$t_m \langle P_i, s_jP_j \rangle$$

by first having node j to provide his s_jP_j. Then the Imote2 will forward this value to node i. Besides, the Imote2 will be provided with s_iP_i from i and it will also compute

$$t_m \langle s_iP_i, P_j \rangle$$

and forward it to node j. Then node i will compute

$$(t_m \langle P_i, s_jP_j \rangle)^{s_i}$$

and node j

$$(t_m \langle s_i P_i, P_j \rangle)^{s_j}$$

Because of the bilinearity

$$(t_m \langle P_i, s_j P_j \rangle)^{s_i} == (t_m \langle s_i P_i, P_j \rangle)^{s_j}$$

and thus the two parties have established a shared key.

Please note that although computing Tate Pairing on low-end motes cannot be achieved efficiently, the exponentiation used for this scheme can be computed (using Lucas funtions) as we will show in Section 7.3.

In a different scenario where an Imote2 i wants to establish a key with a MicaZ j or inversely, the Imote2 would provide

$$t_m \langle s_i P_i, P_j \rangle$$

to node j and after being provided with $s_j P_j$ it would use

$$t_m \langle s_i P_i, s_j P_j \rangle$$

as the session key. Node j on the other hand would expenentiate

$$(t_m \langle s_i P_i, P_j \rangle)^{s_j}$$

and the two parties will have an established shared key. Figure 5.2 shows how such a schemes would be formulated.

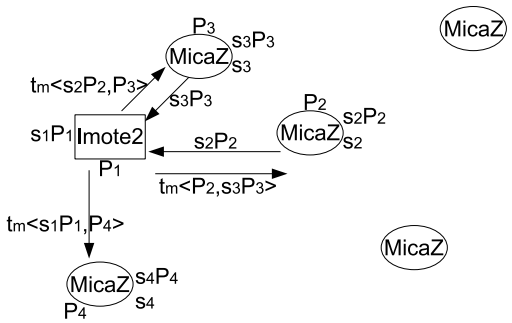


Figure 5.2: Hybrid network

The above schemes require two pairing computations from the high-end motes and one exponentiation from the low-end mote for each key establishment. According to the

results in Chapter 7 such computations can be performed on these motes without having significant impact on their power. Though, if the key establishments are performed often such computations would burden the motes performing them. It is also worth mentioning that an alternative that would alleviate some communication overhead would be to store s_iP_i on the Imote2 beforehand. This is achievable because of the high memory capacity of such sensors but would pose a restriction to the network in case sensors changed positions. In such a case, the Imote2 would then have to be updated about the new s_iP_i's of its neighbors.

For the security perspective of the schemes described above, we have to say that there are certain **vulnerabilities** that can be exploited. One would be compromising a MicaZ mote to steal s_i. This would give an intruder the opportunity to compute the session key that node i establishes with any other node. Thus, for now we have to assume that all the low-end motes are tamper resistant, which can be thought to be quite realistic as technology in this field has had good results so far. On the other hand, in this context we are assuming that the values provided by the Imote2 can be authenticated and thus the MicaZ can verify that a legitimate Imote2 is sending the correct value. But these threats have to be addressed more systematically in the future in order to have a complete key establishment scheme in hybrid sensor networks.

Chapter 6

Implementation Parameters

For our implementation we chose to use supersingular curves with embedding degree $k = 2$ over 512-bits prime field. These curves can provide security equivalent to 1024-bit DL [30]. The reason we chose them is that they have been widely studied for pairing purposes, they provide an efficient distortion map, they have small k which makes pairing more efficient and also can be supported with the existing tools in TinyECC.

The format of our curves is

$$\boxed{y^2 = x^3 + x}$$

More specifically we experimented with two supersingular curves. One is SSc_k2_192 which is over a 192-bit prime field and the other is SSc_k2_512 over 512-bit prime field. For both curves we made sure to have low Hamming-weight of the group order m and also generated points for the verification of the properties of Tate Pairing.

Note that the 192-bit curve doesn't provide sufficient security levels and thus cannot be used in real cryptosystems. It was just used for verification and evaluation purposes. For our implementation, we used the generated points on the curves to verify that

$$t_m \langle P, sQ \rangle == t_m \langle sP, Q \rangle == t_m \langle P, Q \rangle^s$$

.

The actual parameters are shown in Appendix B. After the implementation we proceeded with the evaluation of our work and the feasibility study in real sensor network environments. These will be presented in Chapter 7.

Chapter 7

Results and Evaluation

In this chapter we will present the evaluation results of our work and describe our conclusions about the feasibility of using Identity-Based Cryptography on next generation sensor networks.

First, it is essential to go through some background benchmarks on ECC and Bilinear Pairing produced by other projects. Such information can be found in MIRACL benchmarks and MIRACL EC point multiplication benchmark and is summarized in the table below.

Table 7.1: 1GHz Pentium III MIRACL timing results (ms)

	ECDSA		Tate Pairing
	signature	verification	
160-bit curve	1.52	2.07	-
192-bit curve	2.19	3.06	-
512-bit curve	-	-	20

ECDSA is an ECC based, digital signature technique. ECDSA was already implemented in TinyECC by An Liu and it is beyond the scope of our work to go through the details of it. But it is worth noting that it is also computationally expensive. Table 7.1 shows that according to the Shamus (MIRACL) Software, Tate Pairing is much more expensive than ECDSA. We can notice that with an 160-bit curve, that in ECDSA gives us 1024-bit security, the signature verification time is 2.07 ms whereas with a 512-bit curve, that in Tate Pairing gives us the same security level, the time is 10-fold. In addition, in the Shamus Software's webpage we can read that the Tate Pairing is by far the most expen-

sive application of Identity-Based Encryption (IDE). Thus, one can easily distinguish how expensive pairing is and the significance of implementing it as efficiently as possible.

Please note that we aimed at producing open source software that can be used on many platforms using TinyOS. Thus, our work can be used on sensor motes (MicaZ, TelosB, Imote2) running TinyOS.

7.1 ECDSA on Imote2

First, it is worth looking at the performance of the new technology sensor motes from Intel and comparing it with the traditional motes (MicaZ, TelosB). In order to do so, we extended An Liu's work on ECDSA and ported ECDSA to Imote2. Before we proceed with the results we have to mention that for ECDSA we use the SecG curves with 128, 160, and 192-bit prime fields.

Table 7.2: ECDSA MicaZ - TelosB

Curve	init	sign	verify
secp128r1	2.522	1.923	2.418
secp128r2	2.518	2.069	2.674
secp160k1	3.553	2.059	2.441
secp160r1	3.548	1.925	2.433
secp160r2	3.543	2.066	2.615
secp192k1	4.992	3.070	3.612
secp192r1	4.992	2.991	3.776

(a) MicaZ ECDSA timing results (secs) for W=4

Curve	init	sign	verify
secp128r1	3.861	4.059	5.056
secp128r2	3.847	4.325	5.618
secp160k1	5.208	4.433	5.209
secp160r1	5.225	4.361	5.448
secp160r2	5.197	4.457	5.609
secp192k1	7.190	6.695	7.840
secp192r1	7.204	6.651	8.331

(b) TelosB ECDSA timing results (secs) for W=4

For the MicaZ, the timing results when using hybrid window method for modular multiplication with window $w = 4$ are in Table 7.3a. For TelosB, the ECDSA timing results are in Table 7.3b. The results for the Imote2 are in Table 7.4a. The reader should be reminded that the Imote2 processor frequencies range from 13MHz to 416MHz. Thus, the processor frequencies are much higher than these of the previous motes which urges us to expect much better performance. Not to mention the 32-bit technology that certainly speeds up operations on large numbers.

As we can observe from the tables Imote2 performance is by far better than the other motes. The fact that the technology is increasing the capabilities of sensors, which were very restricted in the past, increases the uses and possible security measures on sensor networks in the future. And this is the underlying motivation of our work too.

Table 7.3: ECDSA Imote2 - energy consumption

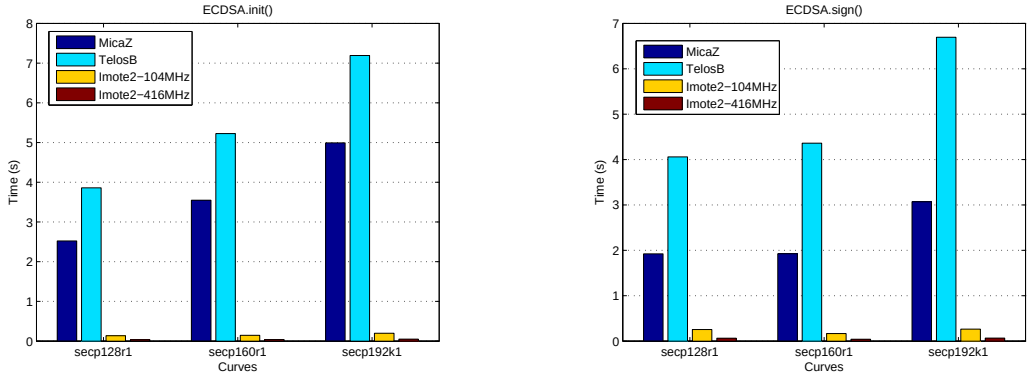
	104 MHz			**416 MHz**		
Curve	init	sign	verify	init	sign	verify
secp128r1	0.136	0.255	0.317	0.035	0.065	0.083
secp128r2	0.136	0.275	0.360	0.035	0.069	0.095
secp160k1	0.151	0.186	0.219	0.038	0.049	0.060
secp160r1	0.148	0.167	0.208	0.037	0.042	0.054
secp160r2	0.151	0.187	0.233	0.038	0.047	0.060
secp192k1	0.199	0.265	0.308	0.050	0.067	0.079
secp192r1	0.200	0.265	0.325	0.050	0.068	0.084

(a) Imote2 ECDSA timing results (secs) for W=4

	MICAZ		**TELOSB**		**IMOTE2**					
					13 MHz		**104 MHz**		**416 MHz**	
W	sig	ver	sig	ver	sig	ver	sig	ver	sig	ver
2	52.9	58.4	27.5	29.4	2.56	2.72	0.32	0.34	0.08	0.10
4	46.2	58.4	23.5	29.4	2.19	2.72	0.28	0.34	0.07	0.09
8	-	-	-	-	-	-	0.24	0.34	0.06	0.09

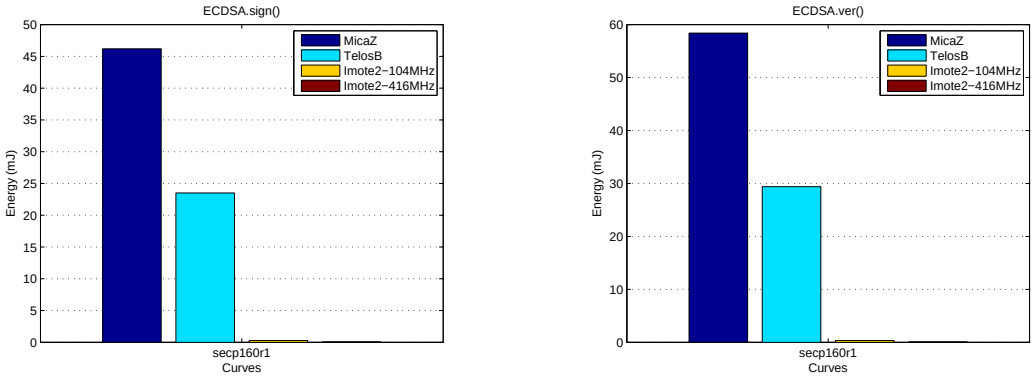
(b) ECDSA energy consumption (mJ) fow secp160r1

One of the extra advantages of Imote2 is the low power consumption. Currently it supports only deep sleep mode where the current draw is 390 μA. Also the voltage is captured on TinyOS to be 4.2V. So then we can compute the energy consumption for performing ECDSA on Imote2. Table 7.4b summarizes the power consumption for all the



(a) ECDSA.init() (b) ECDSA sign

Figure 7.1: ECDSA.init() - sign



(a) ECDSA sign energy consumption (b) ECDSA verify energy consumption

Figure 7.2: ECDSA energy consumption

motes. As we can see, the power that Imote2 uses is much less than the power used from the other motes. Especially at higher frequencies the magnitude difference is incredible. Figures 7.1, 7.2 graphically show the above results. We can easily distinguish the magnitude difference and the performance improvement of Imote2 for ECDSA.

The conclusion we can draw from the above is that the Imote2 adds a whole new direction to sensor networks' cryptography. As far as ECDSA is concerned the DoS possibility is significantly decreased as the time and energy consumption are greatly decreased too. Not only that, but at the same time it becomes clear that the Imote2 capabilities give us the opportunity to try to see how other security tools, which were thought forbidden because of the mote restrictions, could be applied in next generation sensor networks. This

doesn't just hold for Imote2 but possibly for Sun Spots, which are produced by Sun and have similar capabilities. Though, Sun Spots use Java which might make their operation slower. We have no evaluation for Sun Spots and we cannot securely be able to tell if they would perform that good for our experiments.

7.2 Tate Pairing on Imote2

Now, we can proceed with evaluating the feasibility of using Tate Pairing for Identity-Based Cryptography purposes on next generation sensor motes. As mentioned above, bilinear pairing is a very expensive operation and very challenging for the capabilities of sensor motes. For this reason, it is worth noting that MicaZ and TelosB nodes are unable to compute pairing efficiently. While experimenting on these motes we found out that they hung, probably because their task queue was overloaded.

The next step would be to evaluate Tate Pairing computation on next generation sensor motes, meaning Imote2s. We implemented the optimizations described in section 4.5.2 and then we proceeded with measuring the code size, time and energy consumption. As we have already said, we used two supersingular curves for our evaluation, one over 192-bit prime field and another over 512-bit prime field. Again, the 192-bit curve cannot be considered secure enough but was used for evaluation and verification purposes. At this point, we also have to mention the Barret Reduction which was implemented by An Liu in order to speed up the Mod operations and further improved the results of pairing. Table 7.4 shows the code size for Tate Pairing.

Table 7.4: Imote2 Tate Pairing code size

Curve	ROM	RAM
ss192k2	13,512	434
ss512k2	13,844	1,034

As we can observe the RAM size isn't very high for the capabilities of the Imote2 (32MB) for both the 192-bit and 512-bit curves. The same holds for the ROM requirements. Thus, the code size for pairing can be considered low for the Imote2.

On the other hand, we should pay careful attention to the timing results of pairing on the Imote2 which are presented in Table 7.5.

Table 7.5: Imote2 Tate Pairing timing results (secs)

Curve	**104 MHz**			**416 MHz**		
	Miller	Final exp	total	Miller	Final exp	total
ss192k2	0.459	0.032	0.491	0.115	0.008	0.123
ss512k2	4.405	0.154	4.559	1.575	0.055	1.629

We can easily see that Miller's algorithm is the most expensive part of pairing. Though, once again the timing results are acceptable for the 512-bit curve that provides sufficient security level. Both in 104MHz and 416MHz the time it takes to compute Tate Pairing is relatively low and can be used in real sensor networks. Especially if the operations used aren't performed so often (i.e. Identity Based Key exchange). One disadvantage would be the possibility of overloading an Imote2 with many computations in order to achieve DoS attacks. In the future, we believe that the use of projective coordinates will overcome this obstacle and will make Tate Pairing more DoS-resilient on Imote2s (10 times less). And also, the reader should bear in mind that these timing results are for the worst case scenario. In case the neighbors of the motes are static, then we are able to store and reuse the lines used for the g-functions evaluation the time will improve further.

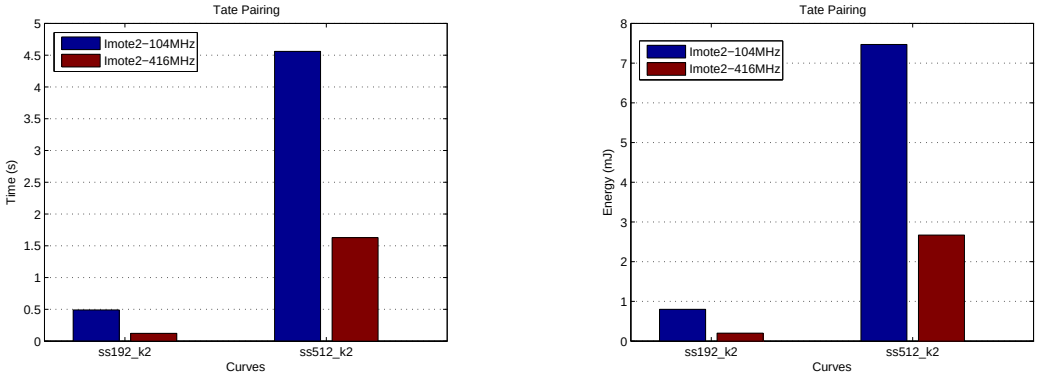
Finally, Table 7.6 shows the energy consumption of Tate Pairing on Imote2. The mode, current draw and voltage remain the same as described in the previous section.

Table 7.6: Tate Pairing Imote2 energy consumption (mJ)

Curve	104 MHz	416 MHz
ss192k2	0.80	0.20
ss512k2	7.47	2.67

The energy consumed by Imote2 is much less than the energy consumed from a MicaZ or TelosB in order to perform ECDSA verification. Thus, the energy consumption is acceptably low for Tate Pairing on Imote2. Figure 7.3 shows the above results.

Concluding, we can say that Tate Pairing can efficiently be used for Identity-Based Cryptographic techniques on next generation motes. Networks consisting of these nodes can use such techniques in order to be more secure in an efficient way. Though, there might be challenging problems when networks consist of both Imote2s and traditional MicaZs or



(a) Tate Pairing time

(b) Tate Pairing energy consumption

Figure 7.3: Tate Pairing results

TelosBs. Such hybrid networks would have to integrate some intelligence to be able to take advantage of the extra power of Imote2s to overcome the restrictions of the traditional nodes (Section 5.2).

7.3 Work conserving scheme

In Section 5.2 we described a scheme that aims at establishing keys in a hybrid network with low and high-end nodes. In order for this scheme to be effient the high-end nodes have to compute the Tate Pairing of two points whereas the low-end nodes have to perform an exponentiation. This exponentiation is performed using the Lucas function that we presented in section 4.5.2.

Before going into the evaluation details of the exponentiation on a low-end MicaZ node it is worth evaluating the feasibility of computing Tate Pairing on a MicaZ. When trying to compute it we found out that the mote hung. This was probably owed to the overflow of the TinyOS task queue as it took a long time to complete the posted tasks. In order to calculate the number of operations required to provide a time estimation for pairing on MicaZ we wrote a Java tool that computes Tate Pairing using the distortion map-Lucas function or without using the Lucas function. Table 7.7 shows the results for the necessary computations required.

Then, by using the above table and taking into consideration the operations in a Lucas function, a point addition, a point doubling and a multiplication in F_{q^2} we can

Table 7.7: Number and kind of operations for Tate Pairing

add: 125, sub: 2, mult: 125, div: 2 mod 192 bits add: 61, mult: 121 $F_{192^2 bits}$ point additions: 1, point doubleings: 60 in in $F_{192 bits}$

(a) SSc_k2_192, regular

add: 62, sub: 63, mult: 63, div: 1 mod 192 bits add: 0, mult: 121 in $F_{192^2 bits}$ point additions: 1, point doubleings: 60 in $F_{192 bits}$

(b) SSc_k2_192, Lucas sequences

add: 323, sub: 2, mult: 323, div: 2 mod 512 bits add: 160, mult: 319 in in $F_{512^2 bits}$ point additions: 1, point doubleings: 159 in $F_{512 bits}$

(c) SSc_k2_512, regular

add: 161, sub: 162, mult: 162, div: 1 mod 512 bits add: 0, mult: 319 in $F_{512^2 bits}$ point additions: 1, point doubleings: 159 in $F_{512 bits}$

(d) SSc_k2_512, Lucas sequences

construct Table 7.8. This table includes the total number of operations used for the implementation of Tate Pairing using our 512-bit curve.

Table 7.8: Total number of operations for Tate Pairing

	Miller's algor, mod				**Point**		**Expon, mod**	
Curve	add	sub	mult	div	add	doub	sub	mult
ss512k2	480	481	1438	1	1	159	702	703

So, now we can compare the operations for the calculation of pairing and exponentiation on a MicaZ in order to see how much the scheme in Section 5.2 improves the performance. Table 7.9 and Figure 7.4 shows these results. x in the table is the number of bits of the exponent.

As we can see, the Lucas exponentiation scheme highly decreases the number

Table 7.9: Operations comparison

Modular	Tate Pairing	Exponentiation
add	528	-
sub	1666	2x
mult	2620	2x+1
div	162	1

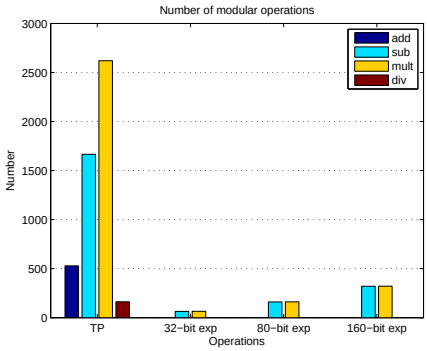


Figure 7.4: Operations comparison

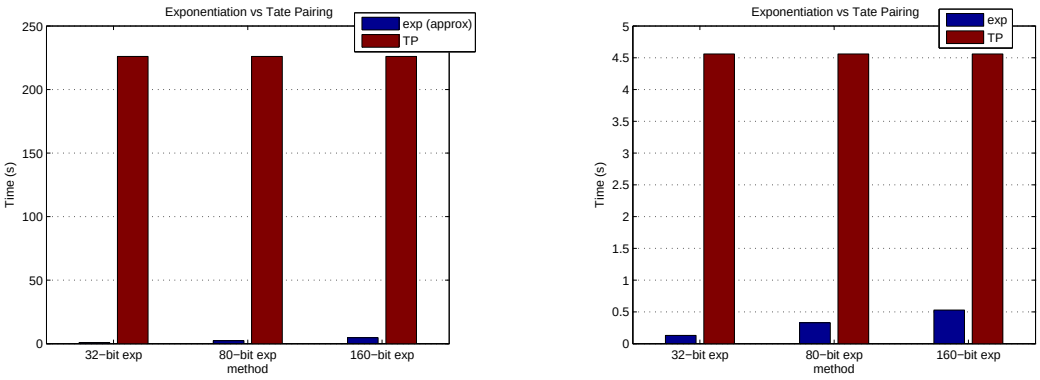
of operations. More importantly, the cost depends on the bit size of the exponent (the maximum bit size is the bit size of the order m). Then, by measuring the average time of addition, subtraction, multiplication and division on a 512-bit prime field on a MicaZ we can compare the Tate Pairing time estimation and the exponentiation time on a MicaZ or Imote2 (Table 7.10[1], Figure 7.5). It is easy to notice that the exponentiation time varies according to the exponent bit size and is much lower (1/10-1/100-fold) than in the Tate Pairing. In fact pairing time makes such a pairing scenarion on a MicaZ unrealistic and impossible. Thus, the scheme proposed in Section 5.2 makes Tate Pairing in hybrid networks more realistic and achievable by delegating part of the computationally expensive work to the powerful motes and a more practical computation to the low-end mote.

Eventually, it is worth noticing Table 7.11 which shows the code size for the exponentiation on a MicaZ. Note that the ROM code size is approximately 2KB for one simple exponentiation whereas the EEPROM memory of the MicaZ is only 4KB.

[1]Division time cannot be calculated accurately as it depends on the numbers that is performed on each time, so the division time might vary a little. The magnitude difference in the timing results would remain the same regardless of the specific numbers, though.

Table 7.10: Timing results, TP vs Exp (secs)

	MicaZ		**Imote2 - 104MHz**	
Secret number	Expon	T. Pair. (approx)	Expon	T. Pair.
32-bit	0.94	226	0.13	4.56
64-bit	1.90	226	0.26	4.56
80-bit	2.38	226	0.33	4.56
160-bit	4.78	226	0.53	4.56



(a) MicaZ

(b) Imote2

Figure 7.5: Tate Pairing vs Exponentiation

Table 7.11: MicaZ exponentiation code size

Curve	ROM (Bytes)	RAM (Bytes)
ss512k2	1,996	133

Chapter 8

Conclusion

In conclusion of our work, we can make some interesting remarks. First of all, we have proven that as Moore's law continues to hold and we get more elaborate embedded devices with better capabilities we can use more cryptographic techniques that can solve important security considerations in sensor networks. We extended the already famous TinyECC package by porting it to Imote2, adding significant functionality and leading the way to the Identity-Based Cryptography field.

One of our main contributions is porting ECDSA and efficiently implementing Tate Pairing and its optimizations on Imote2. With this work, except ECDSA which is performed very efficiently, we can also relatively efficiently compute Tate Pairing. What is more, the evaluation results establish that pairing is feasible on next generation motes from a code size, time and energy point of view. Thus, we can now use pairing in any security scheme on sensor networks. Not to mention, that it has become so efficient that it is almost resilient against DoS attacks. For example, if someone tried to wear out the battery of an Imote2 by getting it to compute the Tate Pairing value many times, he probably wouldn't be able to achieve much as Tate Pairing is computed without severe effort for the mote (especially after An Liu's contribution).

Our evaluation shows that Imote2 is a very promising platform as it performs very well; it is robust and has low power consumption. This will probably be the case with other embedded devices in the future to come which permits us to envision more powerful and secure sensor networks.

As far as the application of Identity-Based Cryptography on sensor networks are concerned, we believe that they can be numerous. For this thesis, we proposed a scheme

that can solve the problem of key establishment on a sensor network. This scheme can be successfully applied on networks consisting of just Imote2's. But it can also be used in hybrid networks consisting of low and high-end motes. Even though the latter has some disadvantages and needs further improvements in order to be considered a viable solution, we believe that our contribution is important because it opens the way for the study of schemes that exploit the bilinearity of Tate Pairing. Especially as current trends show that we are heading toward cost efficient hybrid networks, schemes facing similar challenges like the one we are proposing will be widely studied in the future.

Finally, we hope that our work will inspire others for continuing research on the field of cryptography and security on sensor networks. All this work is part of the attempt of many researchers to provide and mature the technology for secure sensor networks that can serve us in many different ways. I am sure that the group in which this thesis was produced will continue wholeheartedly in this direction.

8.1 Future Work

TinyECC was started about 2.5 years ago and it seems that this work will continue being extended in the future too. An Liu started by implementing ECC and ECDSA on MicaZ and TelosB sensor motes. For the past year he kept improving the performance and optimizing these techniques. The research community and industry showed great interest on TinyECC. Now, it has been ported to Imote2 and been extended even more. In the time to come, there are more things that can be added to it.

First of all, for the performance improvement of Tate Pairing there are a couple more methods that, we believe, can significantly bring down the timing results. One of them is to perform the point operations and line calculations in projective coordinates; this would eliminate the divisions of the slope calculations and would improve the performance. Another optimization is to pre-compute and store the line coefficients when the point P is constant which would save some calculations in optimized Miller's algorithm. After these optimizations we believe that the DoS resilience of pairing will be achieved.

Now, after having an efficient pairing implementation, we can apply it on sensor networks in many ways. We have already described the disadvantages of the scheme in Section 5.2. Thus, it is worth studying ways of authenticating the pairing values provided from Imote2's to MicaZ motes. To our knowledge, there are ways to weakly or self-authenticate

the Identity-Based values provided by other parties [4]. But their mapping to the above case would need careful attention. This way, the scheme will become resilient against DoS attacks on the MicaZ motes. What would also be very interesting is to try to see how we could remove the tamper-resilient assumption on the MicaZ's. If the above are achieved then we would have successful ways to establish keys between motes in any kind of hybrid network which is of great importance.

Overall, we consider the field of Identity-Based Cryptography is a very interesting for sensor networks because it provides cryptographic ways of overcoming traditional problems of them. Some applications that we want to focus on and could solve important problems are identity-based signatures and threshold cryptography. The reason is that many times sensor motes form groups (clusters) that might need to be authenticated all together for load balancing reasons. In the same context, aggregate signatures could also be used. Other schemes that could be applied on sensor networks are blind signatures especially for hybrid nets where Imote2's can blindly sign data the MicaZ cannot sing due to power restrictions. And of course, multiparty key agreement protocols [9] and signature schemes can serve for different reasons on sensors too. Though, our future research will always have to cope with computation expenses and power consumption on motes, especially since most of the above schemes require more than one Tate Pairing computation. What is more, a great challenge would again be the hybrid network case where the low-end motes are the bottleneck of the network. In such cases, the research community would always have to address problems like load balancing in a secure way and without revealing private information between the low and the high end sensors (zero knowledge).

On the other hand, TinyECC will continue to be further optimized and extended in the future. Potentially, other IBE protocols or cryptographic techniques like ECDH will be added to it. Overall, we anticipate that this package will continue to provide good background and resources for ECC on sensor networks.

Bibliography

[1] Working draft American National Standard x9.62-1998 public key cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), September 1998.

[2] IEEE Standard Specifications for Public-Key Cryptography, January 2000.

[3] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman problem. Technical Report 99–07, 1999.

[4] S. Al-Riyami and K. Paterson. Authenticated three-party Agreement protocol from pairing, 2002.

[5] Joux Antoine, Fieker Claus, and Kohel David R. The weil and tate pairings as building blocks for public key cryptosystems (survey). In *International symposium on algorithmic number theory No5*, volume 2369 of *Lecture Notes in Computer Science*, pages 20–32. Springer, 2002.

[6] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm, 1998.

[7] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Advances in Cryptology, Crypto'2002*, Lecture Notes in Computer Science 2442, pages 354–368. Springer-Verlag, 2002.

[8] P. S. L. M. Barreto, B. Lynn, and M. Scott. On the selection of pairing-friendly groups. In *Selected Areas in Cryptography (SAC'2003)*, Lecture Notes in Computer Science 3006, pages 17–25. Springer-Verlag, 2003.

[9] R. Barua, R. Dutta, and P. Sarkar. Extending Joux protocol to Multi Party Key Agreement.

[10] A. Boldyreva. Efficient Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-Group Signature Scheme. PKC 2003, LNCS 2139, pages 31–46. Springer-Verlag, 2003.

[11] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In Proceedings of Eurocrypt 2004, 2004.

[12] D. Boneh and X. Boyen. Short Signatures Without Random Oracles. In *C. Cachin and J. Camenisch, editors*, Proceedings of Eurocrypt 2004, LNCS. Springer-Verlag, 2004.

[13] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. In *SIAM J. of Computing No3*, volume 32 of *Extended Abstract in Crypto 2001*, pages 586–615, 2003.

[14] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and Verifiably Encrypted Signature from Bilinear Maps. Eurocrypt 2003, LNCS 2248, pages 514–532. Springer-Verlag, 2003.

[15] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. Asiacrypt 2001, 2001.

[16] R. Dutta, R. Barua, and P. Sarkar. Pairing-based cryptographic protocols: A survey, 2004.

[17] T. ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. volume IT-31 of *IEEE Transactions on Information Theory*, page 469•472. Springer-Verlag, 1985.

[18] M. Gagne. Applications of Bilinear Maps in Cryptography. Master's thesis, University of Waterloo, 2002.

[19] S.D. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate Pairing. In *Algorithmic Number Theory 5th International Symposium*, ANTS-V, LNCS 2369, pages 324•–337. Springer-Verlag, 2002.

[20] C. Gentry and A. Silverberg. Hierarchical ID-Based Cryptography. In *In Y. Zheng, editor*, Proceedings of Asiacrypt 2002, pages 586–615. Springer-Verlag, 2002.

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

[22] F. Hess. Efficient Identity Based Signature Schemes Based on Pairings. SAC 2002, LNCS 2595, pages 310–324. Springer-Verlag, 2002.

[23] D. Johnson and A. Menezes. The elliptic curve digital signature algorithm (ecdsa), 1999.

[24] A. Joux. A One Round Protocol for Tripartite Diffie-Hellman. Proceedings of ANTS 4, LNCS 1838, pages 385–394, 2000.

[25] M. Kim and K. Kim. A New Identification Scheme Based on the Gap Diffie-Hellman Problem. SCIS 2002, 2002.

[26] B. Libert and J. J. Quisquater. Efficient Revocation and Threshold Pairing Based Cryptosystems. PODC 2003, pages 163–171, 2003.

[27] An Liu, Panos Kampanakis, and Peng Ning. TinyECC: Elliptic curve cryptography for sensor networks (version 0.3), February 2007, http://discovery.csc.ncsu.edu/software/tinyecc/.

[28] M. Maas. Pairing-Based Cryptography. Master's thesis, Technische Universiteit Eindhoven, 2004.

[29] V. Miller. Short program for functions on curves, 1986.

[30] Koblitz Neal and Menezes Alfred. Pairing-based cryptography at high security levels. In *Cryptography and coding (10th IMA International Conference)*, 2005.

[31] D. Page, N. Smart, and F. Vercauteren. A comparison of MNT curves and supersingular curves, 2004.

[32] K.G. Paterson. Cryptography from pairings: a snapshot of current research. In *Information Security Technical Report*, volume 7(3), pages 41–54, 2002.

[33] Certicom Research. Standards for efficient cryptography, sec 1: Elliptic curve cryptography, September 2005.

[34] RSA Data Security, Inc. *PKCS #3: Diffie-Hellman Key Agreement Standard*, June 1991.

[35] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairing. SCIS 2000-c20, Japan 2000.

[36] Michael Scott. Key Exchange using Lucas exponentiation.

[37] Michael Scott. Refinements of Miller"s Algorithm for Computing Weil/Tate Pairing, October 2003.

[38] Michael Scott. Faster pairings using an elliptic curve with an efficient endomorphism. In *INDOCRYPT 2005*, volume 3797 of *Lecture Notes in Computer Science*, pages 258–269. Springer-Verlag, 2005.

[39] J. Solinas. Generalized Marsenne numbers. technical report CORR-39, Department of C&O, University of Waterloo, 1999.

[40] F. Zhang and K. Kim. ID-Based Blind Signature and Ring Signature from Pairings. volume 2510 of *Asiacrypt 2002, LNCS*. Springer-Verlag, 2002.

[41] F. Zhang, R. Safavi-Naini, and W. Susilo. An Efficient Signature Scheme from Bilinear Pairings and it"s Applications. PKC 2004, to appear, 2004.

Appendices

Appendix A

Algorithms

Here we present the algorithms used for the Tate Pairing computation along with exponentiation using Lucas functions, all mentioned in Chapter 4. First we will present the Modified Tate Pairing (note that it is Miller's algorithm slightly altered with an extra exponentiation in the end) and then the algorithm that is actually used for our implementation (with the eliminated denominators optimization). Finally, we give the Lucas functions as we use them to exponentiate numbers in $\mathbb{F}_{q^2}$) by using only half the bandwidth and reducing the number of operations. The aforementioned algorithms are following below.

Algorithm A.1 Modified Tate-pairing

Input: $P \in E(\mathbb{F}_q)[m],\ Q \in E(\mathbb{F}_{p^k})[m],\ m = (m_{t-1}m_{t-2}...m_0)_2$

where m_{t-1} is the MSB of m

Output: $e\langle P, Q\rangle = f_P(D_Q)^{(q^k-1)/m}$

begin

$f \leftarrow 1$, $V \leftarrow P$

for $i = t-2$ down to 0 **do**

$f \leftarrow f^2 \cdot \frac{g_{V,V}(D_Q)}{g_{2V}(D_Q)}$

$V \leftarrow 2V$

if $m_i == 1$ **then**

$f \leftarrow f \cdot \frac{g_{V,P}(D_Q)}{g_{V+P}(D_Q)}$

$V \leftarrow V + P$

end if

end for

$f \leftarrow f^{(q^k-1)/m}$

return f

end

Algorithm A.2 Optimized Modified Tate-pairing

Input: $P \in E(\mathbb{F}_q)[m],\ Q \in E(\mathbb{F}_{p^k})[m],\ m = (m_{t-1}m_{t-2}...m_0)_2$
where m_{t-1} is the MSB of m

Output: $e\langle P, Q\rangle = f_P(D_Q)^{(q^k-1)/m}$

begin
$f \leftarrow 1$, $V \leftarrow P$
for $i = t-2$ down to 0 **do**
$f \leftarrow f^2 \cdot g_{V,V}(\phi(Q))$
$V \leftarrow 2V$
if ($m_i == 1$ and $t! = 0$) **then**
$f \leftarrow f \cdot g_{V,P}(\phi(Q))$
$V \leftarrow V + P$
end if
end for
$f \leftarrow f^{(q^k-1)/m}$
return f
end

Algorithm A.3 Lucas exponentiation

Input: x, k, q

Output: $v_k(x)$ mod q

begin

$k \leftarrow k-1$, $a \leftarrow 2$, $b \leftarrow x$

$k = (k_{t-1}k_{t-2}...k_0)_2$

for $i = t-1$ down to 0 **do**

if $x_i == 1$ **then**

$a \leftarrow a \cdot b - x \ mod \ q$

$b \leftarrow b \cdot b - 2 \ mod \ q$

else

$b \leftarrow a \cdot b - x \ mod \ q$

$a \leftarrow a \cdot a - 2 \ mod \ q$

end if

end for

return b

end

Appendix B

Curves

Below follow the actual curve paramenters we used for our work.

Table B.1: SSc_k2_192

q= F769064B09938DE6AE1939DECBC9775F904227C7A95E9D63 o= F769064B09938DE6AE1939DECBC9775F904227C7A95E9D64 m= 1000000000000021 c= $\frac{q^k-1}{m}$= 0F769064B09938BE8657C3F1B0B78E80E4
points P(Px,Py), Q(Qx,Qy) Px= F6A71262C31820555C5686DF49CF25BEDEC3E31F118AA0D6 Py= B004EA468AF34BABCA2F4C0103D2B57EED02C81F14D20CB1 Qx= B2421FDA744BB179202D4D5BB09B5A1F8B184EB7E76167C6 Qy= DDA6A93031C53341ECF6EE196A6FDDA981A1DBD3CCA86596
multiples s, sP(sPx,sPy), sQ(sQx,sQy) where s=47348E68 sPx= 12AA8BA24F8335B6706DC282C7206BF22BFA68BE23336496 sPy= AB409D39FFC254C13EB9A426051BF3C43928BCD9B82B1D36 sQx= 020CC3967AF2BE10AE5129391CB88F7397641FA15F728FF3 sQy= 82B017D22C85A8C9BC9B8EAD2997D3D2AA8A9FFB5A0E9521

Table B.2: SSc_k2_512

q= 8000000000000000000000000000000020001400000000 0000000000000000000000000010000800000020000000000000 00000000000000000000008000 o= 8000000000000000000000000000000020001400000000000 00000000000000000000000100008000000200000000000000000 000000000000000000080004 m= 80000000000000000000000000000000020001 c= $\frac{q^k-1}{m}$= 0100000000000000000000000000000000000000008000 00000000000000000000000000000000000000000004
points P(Px,Py), Q(Qx,Qy) Px= 36094B252EF2FE159881435E65E843E222396347DB6 C5D9B36D4ADE8BA7919F46233BA3E39331EF8404BE68 38ABF12D74B90462298D38C1750398B01AE7C68DF Py= 64673F0F2E8F6DD2E3206558FBBA8C1E803111FEA00 EAEB7552C7C8AEC37E7971E267EC54555E8E3FEA750B 4B8F1561F36366EAEF5A614285C9022537574AEEF Qx= 0B534839957AD90C6CD3A71A364A031358A13D0B4F2 6DD26F5BAB6A852F4CE8C92D8C84EEFE0A9F0DDBEE2E ABE70EF0E21E902FE27595F56B7FA10B8FB97769D Qy= 7032E262B06812935CFE573FC85A98E2BCA9C17F857 7973DAACEA5F2E7CAADB2A3CC4294A4109A41250E2E7 2251890AB2ECC9D35D4FF887D440141FDD08F1B61
multiples s, sP(sPx,sPy), sQ(sQx,sQy) where s=47348E68 sPx= 187BDC8133BBE39C88DC3E91E7D474EB2A63422C24C C1A0DCD79D659AE154C0788F135E6E57ED851C30767C 7C332A8E2715A7293D5DC13E1629D73C1D65CE363 sPy= 63A46B41F6B86DB1FFF56CC5EA70FD0504BD2873BEA 3DDC132FCA2FD628FB33C8BC61DC983E6DA28DAAC4C3 CBDFD8F20E75A34075092AE3FA544E2DD9B3FF064 sQx= 150587C91B11F9DF0B86C7F77F1F2C3DFAD6EB0AAC0 53708F17BD9DF99C8EB032E10E58DFCC01BBE9E5C7CA 5CF6273523A076D8862E0F4E35C8C2FBCB779BFD2 sQy= 0D28657C79196094AE9511D35715620FB90CF44C926 D3FA1AB0E909AFC1AC878CFA1CAA6E3400F3FD214017 9F29A6200B7F9AC8011B0C5DCFC2958A2ECB3CDFC