Comparative Study of Elliptic Curve Cryptography Hardware Implementations in Wireless Sensor Networks

Hilal Houssain, Mohamad Badra
LIMOS Laboratory
CNRS France

Turki F. Al-Somani, *Senior Member, IEEE*
Computer Engineering Department
Umm Al-Qura University
Makkah, Saudi Arabia

Abstract

A comparative study of hardware implementations of Elliptic Curve Cryptography (ECC) in Wireless Sensor Networks (WSN) is presented in this paper. The study covers important parameters like the underlying finite field, representation basis, occupied chip area, consumed power, and time performances of these implementations. Additionally, most of the reviewed implementations were implemented on Application Specific Integrated Circuits (ASIC) and only one was implemented on Field Programmable Gate Array (FPGA). Likewise, most of these implementations were implemented over the binary fields GF(2^m) and using polynomial basis representation.

1. Introduction

Wireless sensor networks (WSNs) are ad hoc networks comprised of a large number of low-cost, low-power, and multi-functional sensor nodes and one or more base stations. A base station is a much more powerful laptop-class node that connects the sensor nodes to the rest of the world [1, 2] using radio interfaces. There exist a wide range of applications for WSN, such as health monitoring, industrial control, environment observation, as well as office and even military operations. In most of these scenarios, critical information is frequently exchanged among sensor nodes through insecure wireless channels. It is therefore crucial to add security measures to WSNs for protecting its data against threats in a way so integrity, authenticity or confidentiality can be guaranteed.

Efficient computation of Public Key Cryptography (PKC) in sensor nodes (e.g., [3, 10, 5, 4]) has been intensively investigated by researchers. Major problem with the sensor nodes as soon as it comes to cryptographic operations is their extreme constrained resources in terms of power, space, and time, which limit the sensor capability to handle the additional computations required by cryptographic operations. Nevertheless, PKC is indeed shown to be feasible in WSNs (e.g., [10, 5]) by using Elliptic Curve Cryptography (ECC). This is because, in comparison to traditional cryptosystems like RSA, ECC offers equivalent security with smaller key sizes, faster computation, lower power consumption, as well as memory and bandwidth savings.

Several software implementations of ECC in WSN have been reported [5 - 12]. The advantages of software implementations include ease of use, ease of upgrade, portability, low development cost and flexibility. Their main disadvantages, on the other hand, are their lower performance and limited ability to protect private keys from disclosure compared to hardware implementations. These disadvantages have motivated many researchers to investigate efficient architectures for hardware implementations of ECC in WSN. Many hardware implementations of ECC in WSN have been reported [13 - 19]. Most of these implementations were for ECC defined over GF(2^m) [15 - 19], and only implementations in [13 - 15] were defined over GF(p).

In this paper, we present a comparative study of hardware implementations of Elliptic Curve Cryptography (ECC) in Wireless Sensor Networks (WSN). The presented study covers important parameters like the underlying finite field, representation basis, occupied chip area, consumed power, and time performances of these implementations. The rest of the paper is organized as follows. In section 2, we present a brief introduction to WSNs. Section 3 presents ECC and its use in various fields. Hardware implementations of ECC in WSN are then reviewed in section 4. Discussion of the different aspects for the surveyed hardware implementation of ECC in WSN is illustrated in section 5. Section 6 concludes the presented study.

2. Wireless Sensor Networks

WSNs [1, 2] comprise mainly of a large number of small sensor nodes with limited resources and are based around a battery powered microcontroller. Wireless sensors are equipped with a radio transceiver and a set of transducers through which they acquire data about the surrounding environment. WSN form an ad-hoc multi-hop network, where nodes communicate with each other and with one or more sink nodes that interact with the outside world. Sensors in the WSN can receive commands via the sink to execute tasks such as data collection, processing and transfer. The number of nodes participating in a sensor network is mainly defined by several requirements such as the network connectivity and coverage, and the size of the area of interest.

There exist a large number of different application's scenarios for WSN: examples are health monitoring, industrial control, environment observation, as well as office and even military applications. For example, in the health monitoring applications, WSN can be used to remotely monitor physiological parameters, such as heartbeat or blood pressure of patients, and send a trigger alert to the concerned doctor according to a predefined threshold. In addition, sensor nodes may be deployed in several forms: at random, or installed at deliberately chosen spots.

3. Elliptic Curve Cryptography

ECC, which was originally proposed by Niel Koblitz and Victor Miller in 1985 [23, 24] is seen as a serious alternative to RSA [25] with much shorter key size. ECC with key size of 128-256 bits is shown to offer equal security to that of RSA with key size of 1 - 2K bits. To date, no significant breakthroughs have been made in determining weaknesses in the ECC algorithm, which is based on the discrete logarithm problem over points on an elliptic curve. The fact that the problem appears so difficult to crack means that key sizes can be reduced in size considerably, even exponentially [27]. This made ECC become a serious challenge to RSA. The advantage of ECC is being recognized recently where it is being incorporated in many standards. ECCs have gained popularity for cryptographic applications because of the short key compared with earlier public key cryptosystems such as RSA [25] and ElGamal [26]. They are considered particularly suitable for implementations on smart cards or mobile devices.

Extensive research has been done on the underlying math, security strength and efficient implementations of elliptic curve cryptosystems. Among the different fields that can underlie elliptic curves, prime fields $GF(p)$ and binary fields $GF(2^m)$ have shown to be best suited for cryptographic applications. An elliptic curve E over the finite field $GF(p)$ defined by the parameters $a, b \in GF(p)$ with $p > 3$, consists of the set of points $P = (x, y)$, where $x, y \in GF(p)$, that satisfy the equation:

$$y^2 = x^3 + ax + b \quad (1)$$

where $a, b \in GF(p)$ and $4a^3 + 27b^2 \neq 0 \bmod p$, together with the additive identity of the group point O known as the "point at infinity" [1]. The number of points $\#E$ on an elliptic curve over a finite field $GF(q)$ is defined by Hasse's theorem [16]. The set of discrete points on an elliptic curve form an abelian group, whose group operation is known as point addition. Elliptic curve point addition is defined according to the "chord-tangent process". Point addition over $GF(p)$ is described as follows:

Let P and Q be two distinct points on E defined over $GF(p)$ with $Q \neq P$ (Q is not the additive inverse of P). The addition of the two points P and Q is the point R ($R = P + Q$), where R is the additive inverse of S, and S is a third point on E intercepted by the straight line through points P and Q. The additive inverse of a point $P = (x, y) \in E$, over $GF(p)$, is the point $-P = (x, -y)$ which is the reflection of the point P with respect to the *x-axis* on E. When $P = Q$ and $P \neq -P$ the addition of P and Q is the point R ($R = 2P$), where R is the additive inverse of S and S is the third point on E intercepted by the straight line tangent to the curve at point P. This operation is referred to as point doubling.

Equation (2) defines the non-supersingular elliptic curve equation for $GF(2^m)$ fields. Only non-supersingular curves over $GF(2^m)$ are considered since supersingular curves are not secure. Supersingular elliptic curves define a special class of curves with some special properties that make them unstable for cryptography [28].

$$y^2 + xy = x^3 + ax^2 + b \quad (2)$$

where $a, b \in GF(2^m)$ and $b \neq 0$ together with the point at infinity denoted by O. It is well known that E forms a commutative finite group, with O as the group identity, under the addition operation known as the tangent and chord method. Explicit rational formulas for the addition rule involve several arithmetic operations (adding, squaring, multiplication and inversion) in the underlying finite field. In affine coordinate system, the elliptic group operation is given by the following.

Let $P = (x_1, y_1) \in E$; then $-P = (x_1, x_1 + y_1)$. For all $P \in E$, $O + P = P + O = P$. If $Q = (x_2, y_2) \in E$ and $Q \neq -P$, then $P + Q = (x_3, y_3)$,

where

$$x_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a$$

$$y_3 = (\frac{y_1 + y_2}{x_1 + x_2}) \cdot (x_1 + x_3) + x_3 + y_1$$

if $P \neq Q$ and,

$$x_3 = x^2{}_1 + \frac{b}{x^2{}_1}$$

$$y_3 = x^2{}_1 + (x_1 + \frac{y_1}{x_1})x_3 + x_3$$

if $P = Q$.

Computing $P + Q$ is called elliptic curve point addition if $P \neq Q$ and is called elliptic curve point doubling if $P = Q$.

A major operation required by ECC is the point scalar multiplication [27]. The scalar multiplication operation, denoted as kP, where k is an integer and P is a point on the elliptic curve represents the addition of k copies of point P as given by Equation 3.

$$kP = P + P + \ldots + P \ (k \text{ times}) \quad (3)$$

The finite $GF(2^m)$ field, with characteristic 2, has particular importance in cryptography since it leads to efficient hardware. Elements of the $GF(2^m)$ field are represented in terms of a basis. Most implementations use either a *Polynomial Basis* or a *Normal Basis*. Normal basis is more suitable for hardware implementations than polynomial basis since operations are mainly comprised of rotation, shifting and exclusive-ORing which can be efficiently implemented in hardware [27, 34].

In Elliptic Curve Diffie-Hellman Protocol, the base point P and the elliptic curve equation are public. User's A private and public keys are k_A and P_A respectively. User's A public key is equal to k_AP. User's B, on the other hand, private and public keys are k_B and P_B respectively. Similarly, User's A public key is equal to k_BP. The message to be encrypted is embedded into the x-coordinate of a point on the elliptic curve ($P_m = (x_m, y_m)$) [35]. The shared secret key S between two parties A and B is easily calculated by

$$S = k_A(k_BP) = k_B(k_AP) \quad (4)$$

Whenever one of the users need to send a message to the other party, he/she needs to add the shared secret key to the message to produce the ciphertext point P_c which is

$$P_c = P_m + S \quad (5)$$

To decrypt the ciphertext point, the secret key is subtracted from the ciphertext point to give the plaintext point P_m as follows

$$P_m = P_c + S \quad (6)$$

In elliptic curve ElGamal protocol, on the other hand, for some user to encrypt and send the message point P_m to user A, he/she chooses a random integer "l" and generates the ciphertext C_m which consists of the following pair of points:

$$C_m = (lP, P_m + lP_A) \quad (7)$$

The ciphertext pair of points uses A's public key, where only user A can decrypt the plaintext using his/her private key. To decrypt the ciphertext C_m, the first point in the pair of C_m, lP is multiplied by A's private key to get the point $k_a(lP)$. This point is subtracted from the second point of C_m to produce the plaintext point P_m.

The complete decryption operations can be summarized in the following equation

$$P_m = (P_m + lP_A) - k_A(lP) = P_m + l(k_AP) - k_A(lP) \quad (8)$$

4. Hardware Implementations of ECC in WSN

Several hardware implementations of ECC in WSN were reported [13 - 19]. The first hardware implementation of ECC was reported in 2005 by Gaubatz *et. al.* [13, 14] over GF(p). A custom-designed low power co-processor was presented in [15, 16]. The architecture of the presented co-processor occupies a chip area equivalent to 18,720 gates, using TSMC 0.13μ CMOS standard cell technology, and consumes less than 400 μW of power at a clock frequency of 500 *kHz*. Field operations are implemented in a bit-serial fashion to reduce the area. Figure 1 shows the block diagram of the arithmetic unit used in [13, 14].

Wolkerstorfer [15] in 2005 implemented an ECC processor over dual-field performing both prime and binary field operations using polynomial basis. The presented processor has an area complexity of around 23,000 gates implemented in 0.35 μm CMOS technology, operates at 68.5 *MHz*, consumes 500 μW of power and features a latency of 6.67 *ms* for one point multiplication. Figure 2 presents the architecture of the proposed processor in [15].

Batina *et al.* [16] in 2006 reported a low-power ECC processor over the binary field $GF(2^{131})$ using polynomial basis. The consumed power in the presented processor in [16] was less than 30 μW when the operating frequency is 500 *kHz*. The chip area of the presented work in [16] requires 6,718 gates using 0.13 μm CMOS technology.

Bertoni *et al.* [17] in 2006 proposed an efficient ECC coprocessor over GF(2^{163}) using polynomial basis. It computes the scalar multiplication in 17 *ms* @ 8 *MHz*. The reported chip area was 11,957 gates using the 0.18 μm CMOS technology library by ST Microelectronics. The consumed power, on the other hand, was 305 μW. Figure 3 depicts the structure of the proposed coprocessor in [17].

Kumar and Paar [18] in 2006 reported an ECC processor over $GF(2^m)$ using polynomial basis. The bit size range of implemented processor was between 113-193 bits. The presented architecture in [20]

consists of three units: $GF(2^m)$ addition (ADD), $GF(2^m)$ multiplication (MUL), and $GF(2^m)$ squaring (SQR) (see Figure 4). The area of the presented designs in [18] is between $10k$ and $18k$ gates on a $0.35\mu m$ CMOS technology.

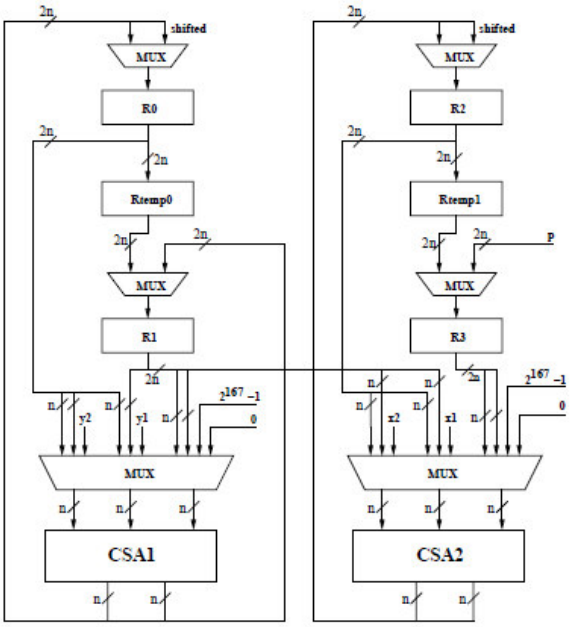

Figure 1: Block diagram of the arithmetic unit presented in [13, 14]

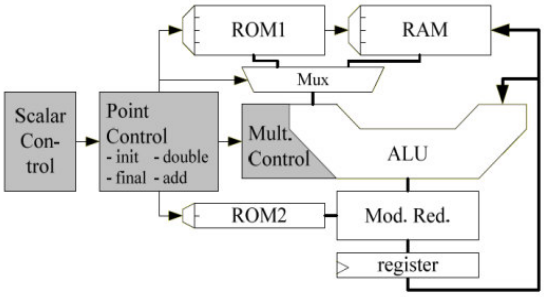

Figure 2: Architecture for ECC processor in [15]

Recently, Portilla *et al.* [19] in 2010 reported an implementation of ECC over $GF(2^m)$ using polynomial basis on an FPGA, which incorporates a mixed solution based on an 8052 compliant microcontroller and a Xilinx XC3S200 Spartan 3 FPGA. An additional XC2V2000 Virtex 2 FPGA is attached to the custom platform due to size limitations. The implemented field multiplier is generic and supports curve sizes from 163 up to 571 bits. The reported chip area is 98275 and 180317 for the bit sizes 283 and 571 bits respectively, using the Xilinx XC2V2000 Virtex 2 FPGA. The reported power consumption, on the other hand, is 253 and 484 *mA* @ 25 *MHz* for the bit sizes 283 and 571 bits respectively.

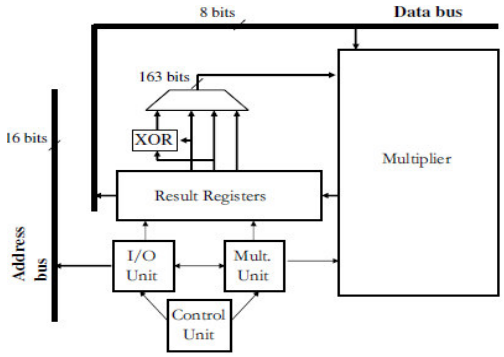

Figure 3: Structure of the 3-register coprocessor presented in [17].

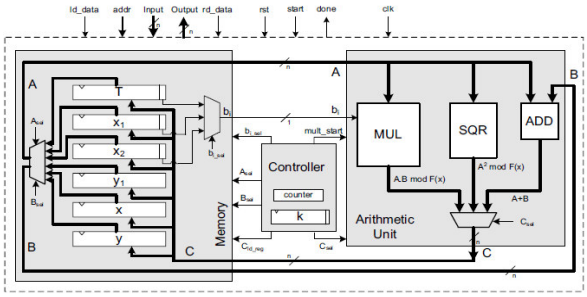

Figure 4: The ECC processor presented in [18]

5. Discussion

The main focus of this paper is in conducting a comparative study of the hardware implementations of ECC in WSN, and emphasizing on key parameters like the underlying finite field, representation basis, occupied chip area, consumed power, and time performances of these implementations (See table1). As shown in table 1, the majority of the reported implementations used the $GF(2^m)$ binary fields [15 - 19], and only two of these implementations used prime fields $GF(p)$ [13 - 15]. This is due to the reason that $GF(2^m)$ has shown to be best suited for cryptographic applications [27, 34]. Although it is known that normal basis representation provides more efficient hardware, Table 1 shows that only polynomial basis was used for all hardware implementations that used binary fields $GF(2^m)$ [15 - 19]. This opens an opportunity to explore and inspect the performance of normal basis based ECC implementations in WSN.

Concerning the other parameters, the implementations in [15] and [18] performed ECC operation (point multiplication) in short time (6.67 *ms* for [15] @ 68.5 *MHz*, and 18 *ms* for [18] @ 13.56 *MHz*), but at the cost of high operating frequency and power consumption of 500 μW and an area between $10k$ and $23k$ gates. On the other hand, implementation in [14] performed ECC operation in 410 *ms* @ 500 *kHz*, consuming just less than 400 μW and occupying

a chip area equivalent to 18,720 gates in 0.13 µm CMOS technology. The implementation in [16], however, is an enhancement of [14]. The presented design in [16] performed ECC operation in 115 ms @ 500 kHz, consuming less than 30 μW using 8,104 gates in 0.13 μm CMOS technology. The implementation in [17], on the other hand, performed ECC in 17 *ms* @ 8 *MHz*, consuming 305 μW and occupying a chip area of 11,957 using the 0.18 µm CMOS technology.

An important result of our study is found in the implementation of [19]. FPGAs were used in [19] showing that FPGAs can be used in WSN. It has been believed for a long time that FPGAs are not suitable for WSN applications because of their power consumption. However, the reported work in [19] opens the opportunity of exploring the performance of FPGAs in terms of area, time and power consumption.

6. Conclusion

In this paper, a comparative study of hardware implementations of Elliptic Curve Cryptography (ECC) in Wireless Sensor Networks (WSN) is presented. The study covers important parameters like the underlying finite field, representation basis, occupied chip area, consumed power, and time performances of these implementations. Additionally, most of the reviewed implementations were implemented on Application Specific Integrated Circuits (ASIC) and only one was implemented on Field Programmable Gate Array (FPGA). However, it has been believed for a long time that FPGAs are not suitable for WSN applications because of their power consumption.

Likewise, the study shows that most of these implementations were implemented over the binary fields GF(2^m). Despite that normal basis representation in $GF(2^m)$ are more efficient in hardware implementations, all of the reviewed implementations were implemented using polynomial basis representation. This opens an opportunity to explore the performance of ECC in WSN over $GF(2^m)$ using normal basis representation.

7. Acknowledgment

The authors would like to acknowledge the support of LIMOS, CNRS, University Blaise Pascal, Clermont-Ferrand II, France and the support of Umm Al-Qura University, Makkah, Saudi Arabia.

8. References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks", in Mobile Computing and Networking (MobiCom'99), Seattle, WA USA (1999) 263–270.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: a survey", Computer Networks, Volume 38, Issue 4, 15 March 2002, Pages 393-422.

[3] R.J. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology", in Proc. SASN, 2004, pp.59-64.

[4] L.B. Oliveira, D. Aranha, E. Morais, F. Daguano, J. L´opez, R. Dahab, "TinyTate: Computing the TinyTate in resource-constrained nodes", in 6th IEEE International Symposium on Network Computing and Applications, Cambridge,MA, 2007.

Table 1: A Summary of hardware implementations of ECC in WSN.

Ref.	Underlying finite field	$GF(2^m)$ Representation basis	Chip area (Gates)	Consumed power	Time performance
[13][14]	GF(p) Bit size: 100 bits		18,720 using TSMC 0.13μm CMOS technology	Under 400 μW @ 500 kHz	410.45 *ms* for one point multiplication @ 500 kHz
[15]	GF(p) and $GF(2^m)$ Bit size: 192 bits	Polynomial basis	23,000 using 0.35 μm CMOS technology	500 μW @ 68.5 MHz	6.67 *ms* for one point multiplication @ 68.5 MHz
[16]	$GF(2^m)$ Bit size: 131 bits	Polynomial basis	6,718 using 0.13 μm CMOS technology	less than 30 μW when the operating frequency is 500 kHz	115 *ms* for one point multiplication @ 500 kHz
[17]	$GF(2^m)$ Bit size: 163 bits	Polynomial basis	11,957 using the 0.18 μm CMOS technology library by ST Microelectronics	305 μW @ 8 MHz	17 *ms* for scalar multiplication @ 8 MHz
[18]	$GF(2^m)$ Bit size [113-193 bits]	Polynomial basis	between 10k and 18k using 0.35μm CMOS technology		[12.5, 16.8, 27.9, 38.8 ms] for scalar multiplication @ 13.56 MHz for bit size of [113, 131, 163, 193 bits] respectively.
[19]	$GF(2^m)$ Bit sizes [283, 571 bits]	Polynomial basis	[98275, 180317] for the bit sizes [283, 571 bits] using Xilinx XC2V2000 Virtex 2 FPGA	[253, 484 mA]@ 25 MHz for the bit sizes [283, 571 bits]	It computes the scalar multiplication in [750, 3600 μs] @ 25 MHz for the bit sizes [283, 571 bits]

[5] D.J. Malan, M. Welsh, and M.D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography", in Proc. of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04), pp. 71–80, Santa Clara, Calif, USA, 2004.

[6] S.C. Seo, D.-G. Han, H.C. Kim, S. Hong, "TinyECCK: Efficient Elliptic Curve Cryptography Implementation over GF(2^m) on 8-bit MICAz Mote", IEICE Transactions on Info and Systems E91-D(5), 1338-1347, 2008.

[7] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors", in Information and Communications Security — ICICS 2006, vol. 4307 of Lecture Notes in Computer Science, pp. 519–528. Springer Verlag, 2006.

[8] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks", in Wireless Sensor Networks — EWSN 2008, vol. 4913 of Lecture Notes in Computer Science, pp. 305–320. Springer Verlag, 2008.

[9] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks", in Proc. 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), pp. 245–256, 2008.

[10] N. Gura, A. Patel, A. S. Wander, H. Eberle, and S. Chang Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs", in Cryptographic Hardware and Embedded Systems — CHES 2004, vol. 3156 of Lecture Notes in Computer Science, pp. 119–132. Springer Verlag, 2004.

[11] C. Lederer, R. Mader, M. Koschuch, J. Großschädl, A. Szekely,and S. Tillich, "Energy-Efficient Implementation of ECDH Key Exchange for Wireless Sensor Networks", in WISTP 2009, Springer, pp. 112-127, 2009.

[12] S. Khajuria and H. Tange, "Implementation of diffie-Hellman key exchange on wireless sensor using elliptic curve cryptography", in Proc. 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology (Wireless VITAE '09), pp. 772–776, May 2009.

[13] E. Ozturk, B. Sunar, and E. Savas¸, "Low-power elliptic curve cryptography using scaled modular arithmetic", in CHES 2004, volume 3156 of LNCS, pages 92–106. Springer, 2004.

[14] G. Gaubatz, J.-P. Kaps, E. Öztürk, and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks", Third IEEE International Conference on Pervasive Computing and Communications Workshops, Workshop on Pervasive Computing and Communications Security–PerSec'05, IEEE Computer Society, pages 146–150, Mar,2005.

[15] J. Wolkerstorfer. "Scaling ECC Hardware to a Minimum", in ECRYPT workshop - Cryptographic Advances in Secure Hardware - CRASH 2005, September 6-7, 2005. invited talk.

[16] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, "Low-Cost Elliptic Curve Cryptography for Wireless Sensor Networks", in Proc. ESAS'06, pp.6-17, 2006.

[17] Bertoni, G., Breveglieri, L., and Venturi, M., "Power Aware Design of an Elliptic Curve Coprocessor for 8 bit Platforms", in Proc. of PERCOMW'06, p. 337, 2006.

[18] S. Kumar and C. Paar. "Are standards compliant elliptic curve cryptosystems feasible on RFID?", in Proc.of Workshop on RFID Security, Graz, Austria, July 2006.

[19] J. Portilla, A.O. Marnotes, E.de la Torre, T. Riesgo, O. Stecklina, St. Peter, and P. Langendörfer. "Adaptable Security in Wireless Sensor Networks by Using Reconfigurable ECC Hardware Coprocessors", in International Journal of Distributed Sensor Networks, 2010.

[20] N. Biggs, Discrete Mathematics. 2nd edition, Oxford University Press, New York, ISBN-10: 0198507178, ISBN-13: 978-0198507178, 2003.

[21] R. McEliec, Finite Fields for Computer Scientists and Engineers. Kluwer Academic Publishers, Boston, 1st edition, ISBN-13: 978-0898381917, 1987.

[22] R. Lidl, and H. Niederreiter, "Introduction to finite fields and their applications", Cambridge University Press, Cambridge, UK, 2nd edition, ISBN-13: 978-0521460941, 1994.

[23] N. Koblitz, Elliptic curve cryptosystems. Math. Comput., Vol. 48, No. 177, pp. 203-209, 1987.

[24] V.S. Miller, "Use of elliptic curves in cryptography", in Proc. of the Advances in Cryptology CRYPTO '85, Springer-Verlag, LNCS 218, pp: 417–426, DOI : 10.1007/3-540-39799-X_31, ISBN: 978-3-540-16463-0, 1986. Available from: http://www.springerlink.com/content/w475304616327668/fulltext.pdf

[25] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems". Commun. ACM, Vol. 21, No.2, pp. 120-126, DOI: 10.1145/357980.358017, 1978. http://portal.acm.org/citation.cfm?doid=357980.358017

[26] T. ElGamal,. "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Advances in Cryptology", in Proc. of CRYPTO 84, August 19-22, Springer Verlag, pp. 10-18, 1985. DOI: 10.1007/3-540-39568-7_2, 1985. Available from: http://www.springerlink.com/content/jl0mkpm32tn8ve3q/.

[27] D. Hankerson, A. Menezes, and S. Vanstone, "Guide to Elliptic Curve Cryptography". 1st edition, Springer, ISBN-13: 978-0387952734, 2004.

[28] A. Menezes, "Elliptic Curve Public Key Cryptosystems", 1st edition, Kluwer Academic Publishers, ISBN-13: 978-0792393689, 1993.

[29] K. Koyama and Y. Tsutuoka, "Speeding up elliptic cryptosystems by using signed binary window method" in Proc. of the Advances in Cryptology Crypto '92, Springer-Verlag, LNCS 740, pp: 345-357, ISBN: 978-3-540-57340-1, 1993. DOI: 10.1007/3-540-48071-4_25, Available from: http://www.springerlink.com/content/bdjju843tvx6ak9n/?p=b939852b15254c62909220736c018a75&pi=0.

[30] H. Cohen, A. Miyaji and T. Ono, "Efficient elliptic curve exponentiation", in Proc. of Advances in Cryptology ICICS '97, Dec. 12-15, 1997, Zhengzhou, China, Springer-Verlag. LNCS 1334, pp: 282-290, Available from: http://www.springerlink.com/content/w536l58355350v50/?p=9afa5450617f468f9bb11ffc69911364&pi=1, DOI: 10.1007/BFb0028484, ISBN 978-3-540-63696-0.

[31] J. Lopez, and R. Dahab, "Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation", in Proc. of the Cryptographic Hardware and Embedded Systems - CHES '99. LNCS 1717, pp: 316-327, 1999. DOI : 10.1007/3-540-48059-5_27, Available from: http://www.springerlink.com/content/0q3wn0y79w1lma0b/fulltext.pdf.

[32] H. Cohen, A. Miyaji and T. Ono, "Efficient Elliptic Curve Exponentiation Using Mixed Coordinates", in Advances in Cryptology -ASIACRYPT '98, 18-22 October 1998, Kazuo Ohta, Dingyi Pei (Eds.), LNCS 1514, Springer-Verlag, New York, pp. 51-65, ISBN: 3-540-65109-8.

[33] M. Rosing, "Implementing Elliptic Curve Cryptography". Manning Publications Company, 1999.ISBN-10: 1884777694, ISBN-13: 978-1884777691.

[34] H. Cohen, G. Frey, R.M. Avanzi, C. Doche, T. Lange, K. Nguyen and F. Vercauteren, "Handbook of Elliptic and Hyperelliptic Curve Cryptography. Discrete Mathematics and Its Applications", Vol. 34, Chapman and Hall, CRC, USA, 2005, ISBN: 9781584885184.