A Comprehensive and Comparative Study of Elliptic Curve Cryptography Hardware Implementations for WSN

Nabil Ghanmy, Lamia Chaari Fourati, Lotfi Kamoun
Electronic and Information Technology Lab (LETI), ENIS
SFAX University, Tunisia

Abstract

Key management and authentication are essential modules for network security provisioning especially for Wireless Sensor Networks (WSN). Make available these two services by using symmetric cryptosystems in software implementation are challenging that's not providing a perfect trade-off between resilience and performance. Asymmetric approaches with public key cryptosystems, specifically Elliptic Curve Cryptography (ECC) with implementation in hardware target can meet this challenge. The main focus of this paper is to provide a comprehensive use and comparative hardware implementation study of ECC for WSN security. In our study, we addressed the state of the art in research works related to ECC hardware implementations for WSN and discussed some enhancements and future works.

1. Introduction

Advanced technology in wireless communication and microelectronics allow today to build wireless sensor nodes. It's small, inexpensive, and battery powered sensing devices. The set build Wireless sensor networks (WSN) is exploited for divers services as habitat monitoring [1], structural health monitoring [2], emergency medical care [3], health care [4], vehicular tracking and even military operations [5]. WSNs are, in general, more vulnerable to attack and unauthorized access that can be easily compromised than wired networks. Indeed, security is fundamental for this type of applications mainly if WSN is deployed in security-critical applications as hostile environments. A more specific, authentication and key management to derive shared secret keys are the two most essential security services to maintain the proper operation for these applications. To ensure these services WSN use two families of cryptographic symmetric (shared-key cryptography) and asymmetric (public-key cryptography) algorithms. The former, despite these advantages that its low-resource requirements, it has some drawback: inflexible from key distribution as they require pre-distribution of keys, not robust by using one global key that lost its would compromise security on the entire network and problems of storage of keys in large networks with n nodes since for pair-wise key distribution scheme each node requires 2n keys. The last can be used to solve these problems. It allows for flexible key management and authentication by session key distribution and digital signature. Using Elliptic Curve Cryptography (ECC) leads to more performance to WSN devices that are often limited in terms of energy [6], clock frequency, and memory size [7]. ECC Public-key cryptosystem compared with RSA offers equivalent security at much smaller key sizes [8]. It reduces computation time and also the amount of data transmitted and stored. As illustration for this idea the original Diffie-Hellman algorithm for key exchange [9] requires 1024 bits to achieve sufficient security but Diffie-Hellman based on elliptic curve can achieve the same security level with 160 bit [10]. Therefore, ECC is an appropriate, and viable, choice for WSN. It performs three basic cryptographic schemes such as Elliptic Curve Diffie-Hellman (ECDH) key generation [11], Elliptic Curve Integrated Encryption Scheme (ECIES) for encryption and digital signature with Elliptic Curve Digital Signature (ECDSA) [12], [13].

The software implementation solutions of ECC [14]–[17] are still too high for applications that have several constraints as power, performance and area for WSN. To overcome these constraints, all efforts have been aimed on providing hardware support to ECC in order to create simple and energy-efficient cryptographic hardware designs [18]. This paper is structured as follows: section two provides a description of WSN. In the third section, we briefly describe ECC. Section four presents ECC security services for WSN such as ECDH and ECDSA schemes. A comparative study related to ECC hardware implementations for WSN is carried out in section five. Finally, section six concludes this paper.

2. Wireless sensor network presentation

Wireless Sensor Network (WSN) is a wireless ad hoc network consisting of numerous sensor nodes or motes and one or more base stations. Sensor node is small, low cost multi-functional devices, equipped with battery, radio communications, micro-controller, and sensors. It has limited processing capability, battery power and storage [19]. WSN is used in a wide range of application domains as to monitor physical or environmental conditions [20]

and industrial maintenance [21]. Security and privacy issues of WSN pose a big challenge especially when it deployed in hostile environments and security-critical applications as health care [3] and even military operations [19]. In order to provide a high level of security to these applications, ECC can be used in first step to establishment of shared secret keys between nodes and authentication in order to improve security services in WSN.

3.Elliptic curve cryptography description

ECC was originally proposed by Koblitz and Miller independently in the 1980s [22], [23]. Since then, there has been a multitude of research on the security of ECC. In the 1990's ECC began to get accepted by several accredited organizations, and several security protocols based on ECC [24] were standardized. ECC over conventional asymmetric crypto systems [8] uses smaller key sizes without degradation of security level (as shown in table 1). For example, a 256 bit key in ECC provides the same security level as a 3072 bit RSA key. The smaller key sizes leads to less space for key storage, time saving when keys are transmitted and less costly arithmetic computations. These characteristics make ECC the best choice for low-bandwidth communication, low-storage and low-computation environments as well as wireless networks [25].

ECC cryptosystem [22], [23] is defined as the tuple $T = (F_q, a, b, P, n, h)$, where F_q is a finite field, a and b define the elliptic curve on F_q, P is a generator point of the elliptic curve, n is the order of P, that is, the smaller integer such that $n.P = O$ (identity point in the additive group), h is called the co-factor and it is equal to the total number of points in the curve divided by n. Mostly two types of underlying finite fields F_q are technically used: prime fields F_p with large primes p and corresponding characteristic and binary fields F_2^m of characteristic two. Prime finite fields F_p elliptic curve (equation 1) may provide more security than binary finite fields F_2^m [25]. Binary finite fields F_2^m with polynomial arithmetic are better preferred for fast hardware implementations because some arithmetic operations are easier to compute, like addition which is an XOR operation and the elements are binary strings that are well represented in m-bit registers[26]. Two kinds of curves over F_2^m are given: pseudo-random (equation 2) and Koblitz curves (equation 3). The last type is selected to optimize elliptic curve operations.

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

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

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

ECC's security is based on the discrete logarithm problem, called the Elliptic Curve Discrete Logarithm Problem (ECLDP). Thus, a cryptosystem could be built on this problem. The ECLDP consists on given two points P, Q Є F_q, to find the positive integer K such as Q = K.P. This problem is of exponential complexity. On the contrary, knowing the scalar k and the point P, the operation K.P is relatively easy to compute. K.P is the point multiplication. Security services are provided by ECC cryptographic schemes for key agreement, digital signatures.

Elliptic curve crypto systems have a layered hierarchy as a pyramid as shown in Fig.1. At the top layer there are elliptic curve primitives as key agreement with ECDH and digital signatures with ECDSA. These primitives, will be detailed in the next section, are based mainly on point multiplication which is the result of adding the point P to itself *n*-1 times. That is K.P=P+P+P+…+P, K times. It is achieved by repeated elliptic curve operations: point addition, which consists of the sum of two different points (P + Q) and point doubling, which consist of the sum of the same point (P + P). The bottom layer constituting the arithmetic on the underlying finite field: The operators are multiplication, inversion, squaring, adding and reduction. Performance of Elliptic curve crypto systems is related to these operators. It can be enhanced by selection the multiplier operators and by using the projective coordinate to avoid costly inversion operators [23].

TABLE I. Equivalent key size (bits) for symmetric, ECC and RSA

Symetric key	Asymetric key	
	ECC key	RSA key
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

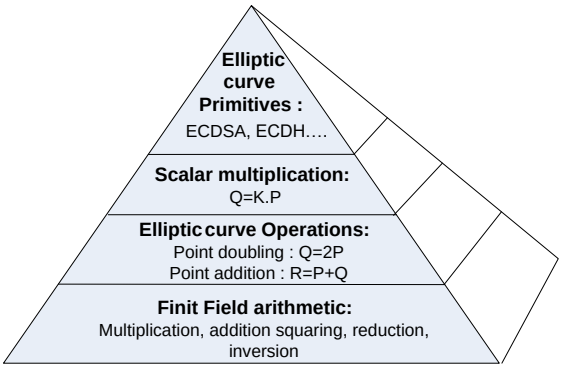

Figure 1. Hierarchy of ECC

4. ECC services security for WSN

As mentioned below WSN security systems based on symmetric-key algorithms is not perfect for all security issues, mainly in key management procedure and authentication services. These functionalities require asymmetric key algorithms [27]. Key management for pair-wise key is used to secure

unicast communication between two nodes in the network. Authentication service including the following three phases: identifing user and verifing that's allowed to access some restricted services, identifying node or verifing its identity and proving that data is fresh and unchanged from the source. ECC public key resolves key management and authentication problems with respectively ECDH key exchange and ECDSA.

4.1 Elliptic Curve Diffie-Hellman (ECDH) key exchange

Elliptic Curve Diffie-Hellman is a widely standardized variant of the classical Diffie-Hellman by the combination of ECC with this apprauch [28]. It allows providing key exchange scheme for two communication nodes in WSNs. It's most effective for pair-wise keying only with every direct neighbor. These main advantages are perfect resilience to node capture, excellent scalability, and low storage memory as well as communication overhead that does not need any trusted third party (trusted authority). It does not require a pre-distribution of keying material, and hence it also works for entities which have never met in advance or do not possess a secret key with a trusted third party. ECDH scheme between two nodes works as shown in the fig.2. Node A and node B generate their public keys by multiplying agree point P with their private keys K_A and K_B. After exchange public keys between them, they generate a shared secret key by multiplying public keys by their private keys. The secret key is R = K_B *K_A*P. Finding the private keys K_B and K_A by an eavesdropper with public values of Q_A, Q_B and P is computationally intractable. The problem of ECDH key exchange does not authenticate the entities A and B, and is therefore vulnerable to man-in-the-middle attacks [29]. It needs ECDSA to reform this weakness.

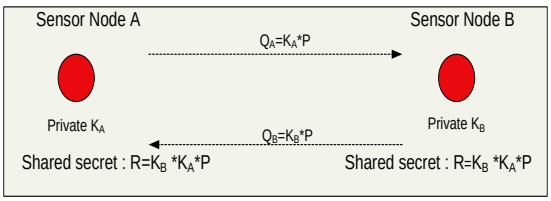

Figure 2. Elliptic Curve Diffie-Hellman scheme between two nodes

4.2 Elliptic Curve Digital Signature

ECDSA is a variant of DSA, which uses ECC [13]. As application ECDSA allows WSN authentication entities to resolve the weakness of ECDH [29] and authenticating broadcasted packets especially as in WSN broadcast is the fundamental communication primitive [30]. Broadcast authentication implicitly provides two additional cryptographic services that are data integrity and non-repudiation of the signed message. Data integrity service allows the prevention of data alteration by using secure hash algorithm (SHA) [31]. Non-repudiation of sensor data is achieved by generating ECDSA on the sensor devices to ensure that a transaction cannot be denied. In comparison with asymmetric-key algorithms, µTesla [32], symmetric-key algorithms, is a lightweight cryptographic primitive designed for efficient authentication of broadcast messages in WSNs. It does not support non-repudiation of sensor data [33]. It need for time synchronization and the delayed authentication, it has made them vulnerable to a variety of attacks [34]–[36]. Moreover, scalability is another concern for symmetric key based solutions [34].

ECDSA is used in WSN for several scenarios: from base station to all sensor nodes or to new node, from the user to several nodes and between nodes [28]. Its principal version is generated in two steps there are detailled below; the first is generated by the sender and called signature generation and the second is generated by the receiver and called signature verification. This version of signature is carried out only after system initialization wherein each entity has its trusted private and public key respectively d and Q = d *P.

4.2.2 Signature generation. To sign the message, the signer:

1. Choose an integer k from [1, n-1]
2. Compute K.P = (x1, y1)
3. Compute r = x1 mod n
4. Compute e = h (m) with m the message, e the message digest and h the hash function
5. Compute $s = k^{-1} (e + d.r) \bmod n$.

The signature is the set (r, s). This step needs a pseudo random number generator to choose k, one point multiplication, one modular reduction, one hash function, addition and modular division.

4.2.3. Signature verification. To verify the signature, the receiver:

1. Compute e = h (m)
2. Compute $u_1 = e.s^{-1} \bmod n$
3. Compute $u_2 = r.s^{-1} \bmod n$
4. Compute $u_1.P + u_2.Q = (x_2, y_2)$
5. Compute $v = x_2 \bmod n$

The receiver must compare v and r. If v = r then the signature is valid else it is invalid.
The demonstration is as the following:

$$u_1.P + u_2.Q = e.s^{-1}.P + r.s^{-1}.Q$$
$$= e.s^{-1}.P + r.s^{-1}.d.P$$
$$= (e + d.r).s^{-1}.P$$

Since $s = k^{-1}.(e + d.r)$ *so* $k = s^{-1}.(e + d.r)$

As a result $u_1 .P + u_2 .Q = k.P$
Consequently $(x_2, y_2) = (x_1, y_1)$
Therefore, $x_2 \bmod n = x_1 \bmod n$
Which gives that the signature is well verified v = r.

This step needs one hash function, two modular divisions, two point multiplications, one point addition and one modular reduction.

5. Hardware implementations of ECC for WSN

Running strong cryptographic algorithms on wireless sensor nodes is extremely difficult due to their limited resources as WSN. In fact, software implementations of PKC in sensor nodes environment became possible in 2004 [28]. Because of several constraints of WSN, this type of implementation is still too high for applications that have several constraints as power, performance and area. Hardware implementations are a suitable alternative in order to speed up the computation and reduce power to prolong life time of sensor nodes.

5.1 Hardware Implementations

There are several published FPGA-based hardware implementations that concentrate on implementing only the point multiplication operation or fully functional ECC algorithm for ECDSA and ECDH under the form of accelerator (coprocessor) or processor. ECC hardware implementation is beginning by Gaubatz et.al [8] in 2005 whose aim is to study the power of ECC Public Key with other primitive schemes for Wireless Sensor Networks. They found that the use of PKC can reduce the amount of traffic overhead due to key management in WSN. Their implementation is more oriented to highly constrained sensor nodes, since its operational frequency is only 500 KHz. In their ECC point multiplication architecture, operations are performed over F_{100}^{p}, occupying a chip area equivalent to 18720 gates in 0,13µm CMOS technology, and consuming just under 400µW in signing and encrypting messages. ECDSA signatures decryptions are performed at around 410 ms, and ECDSA verifications and ECMV encryptions are performed at around 820 ms. As for the optimizations, they efficiently implement modular reduction, achieve a fast inversion algorithm, and implement all arithmetic primitives in a bit serial fashion.
In 2005 and 2006 respectively Wolkerstorfer et al., Kumar et al. developed integrated chips that can provide an excellent performance for signing a message using ECDSA, but at the cost of a high operating frequency [36], [37]. In the case of Wolkerstorfer, the chip has an area of 23000 gates implemented in 0.35 µm CMOS technology, operates at 68.5, and is able to execute a one point multiplication over F_2^{191} in only 6.67 ms. In the other side the chip created by Kumar and Paar is slower, providing a point multiplication over F_2^{131} in 18 ms, but has less gates with almost 12k , and works at a slower operating frequency (around 13 Mhz).

In 2006 Batina et al. [38] further improved ECC processor over the binary field F_2^{131} using polynomial basis. Their design included a Modular Arithmetic Logic Unit (MALU) to compute modular additions and multiplications, using the same cells for both purposes without having a full-length array of multiplexors. Moreover, squaring is considered a special case of multiplication, and inversion is avoided mostly by use of projective coordinates. The results are promising: using 8104 gates (12 kgates including RAM) in 0,13 µm CMOS technology, one point multiplication is performed in only 115 ms, consuming less than 30µW in a operational frequency of 500 KHz. With the same basis that Batina et al. Bertoni et al. [39] proposed in 2006 an efficient implementation of ECC coprocessor over F_2^{163}. The duration of point multiplication is 17 ms with 8 MHz as frequency. The reported chip area was 11,957 gates using the 0.18µm CMOS technology library by ST Microelectronics and the consumed power was 305µW. In 2008 Murphy et al. [40] present an area-efficient processor for performing RSA and ECC operations for constrained platforms such as WSN. A 190-bit EC point multiplication requires 227.8s for 16 bits processor architecture with 65.3 MHz frequency. The chip area was 858 LUT with Xilinx Spartan-2E xc2s256e FPGA target. In 2011 Portilla et al. [41] presented an HW/SW implementation of an ECC based on an 8052 compliant microcontroller and a Xilinx XC3S200 Spartan 3 FPGA over F_2^{m} using polynomial basis and generic m sizes from 163 up to 571 bits. An additional XC2V2000 Virtex 2 FPGA is attached to the custom platform due to size limitations. The result of 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 for the bit sizes 283 and 571 bits is respectively 253mA and 484mA with 25 MHz frequency. Panic et.al [42] present a sensor node processor designed to support complex data encryption/decryption operations. Its hardware accelerators chip for crypto Cores which support AES, ECC and SHA-1. This chip consumes 0.055 mW in an operational frequency of 12.5 MHz with 233 bit key size. Point multiplication requires 84 µs with a complexity of 76 kgates using the 0.18 µm CMOS technology.

5.2 Discussion

From the below table that summarizes the set of hardware implementations of ECC in WSN already detailed in the previous sub-section, binary fields F_2^{m}

is used in the majority of implementations [36]–[39], [41], [42] and few over primary field F_p [28], [40]. The former is best suitable for hardware implementations and allows for better performance compared to primary field F_p. These implementations

Works	Finite field, key size(bits)	Point Multiplication (ms)	Frequency	Power	Technology	Chip area (gates)
[8]	F_p, 100	~ 400	500 kHz	> 400 µW	0,13µm CMOS	18720
[36]	F_2m, 191	6.67	68.5Mhz	N/A	0,35µm CMOS	23000
[37]	F_2m, 131	18	13 Mhz	N/A	0,35µm CMOS	12000
[38]	F_2m, 131	115	500kHz	30µW	0,13µm CMOS	8104
[39]	F_2m,163	17	8 MHz	305 µW	0.18 µm CMOS	11957
[40]	F_p,190	227.8	65.3	N/A	Xilinx spartan-2E xc2s256e	858 LUT
[41]	F_2m , [283;571 bits]	[0,750; 3,6]	25 MHz	[253;484 mA] for [283; 571 bits]	Xilinx Virtex 2-XC2V2000	[98275; 180317]for [283; 571 bits]
[42]	F_2^m, 233	0,084	12.5 Mhz	55 µW	0.25µm CMOS	76000

focus mainly in point multiplication, since it's the greediest operation in ECC. Evaluation of different works covers several parameters like occupied chip area, consumed power, and time performances. It's difficult since input parameters are not homogeneous such as underlying finite field, key size, technology and frequency. As illustrated in table II, implementation carried in [42] is the fastest with 84µs for point multiplication and relatively less consuming energy with 55µW comparing it with other primitives. Implementation in [38] performed ECC operation with area, time and consumption trade-off that are respectively 8104 gates, 115ms and 30µW. For WSN security, although most of the reviewed implementations were implemented on Application Specific Integrated Circuits (ASIC) [28], [36]–[39], [42], ECC can be implemented in FPGA [40], [41]. The lack of these works is the fact that most of them not implement the full scheme of key exchange and authentication with respectively ECDH and ECDSA which allows to strengthen the security of WSN and to resolve some constraints found in software implementation such as power, performance and area.

7. Conclusions

In this paper, a comparative study of ECC hardware implementations for WSN security is presented. ECDH and ECDSA are used in WSN to trade-off security and performance compared to other techniques specifically for key distribution and authentication phases. Hardware implementations as FPGA are the best target mainly with finite field F_2^m parameters. The set of studied related works can be improved to be ECDH and ECDSA in the same chip as accelerator or processor that will be integrated in motes of WSN.

8. References

[1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Application driver for wireless communications technology," ACM SIGCOMM Computer Communication Review, vol. 31(2), pp. 20–41, April 2001.

[2] V.A. Kottapalli, A.S. Kiremidjian, J.P. Lynch, E. Carryer, T.W. Kenny, K.H. Law, and Y. Lei, "Two-tiered wireless sensor network architecture for structural health monitoring," in SPIE's 10th Annual International Symposium on Smart Structures and Materials car, 2003.

[3] D. Malan, "Crypto for tiny objects," Harvard University, Cambridge, Massachusetts, USA, Tech. Rep., January 2004.

[4] P.Johnson and A. D.C., "Remote continuous monitoring in the home," Journal of Telemedicine and Telecare, vol. 2(2), pp. 107–113, 1996.

[5] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A wireless sensor network for targer detection, classification, and tracking," IEEE Computer Networks, vol. 46, pp. 605–634, 2005.

[6] K. Bicakci, H. Gultekin, and B. Tavli, "The impact of onetime energy costs on network lifetime in wireless sensor networks," IEEE Communications Letters, vol. 13(12), pp. 905–907, 2009.

[7] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," Communications of the ACM, vol. 47(6), pp. 53– 57, June 2004.

[8] N. Gura, A. Patel, A. Wander, H. Eberle, and S.-C. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in 6th International Workshop on Cryptographic Hardware and Embedded Systems, August 2004.

[9] I. P1363, "Standard specifications for public-key cryptography. available at http://grouper.ieee. org/groups/1363/," IEEE, Tech. Rep.

[10] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," Security and Networks, vol. 1, pp. pp. 127–137, 2006.

[11] W. Diffie and H. M., "New directions in cryptography," IEEE Transactions on Information Theory, vol. 22(6), pp. 644–654, November 1976.
[12] SECG, "Sec 1: Elliptic curve cryptography," Certicom Research, Tech. Rep., 2000.
[13] ANSI, "Public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ecdsa)," American Bankers Association, Tech. Rep., 1999.
[14] P. Szczechowiak and M. C. M. D. R. Oliveira, L. B. and Scott, "Nanoecc: testing the limits of elliptic curve cryptography in sensor networks," in 5th European conference on Wireless sensor networks, 2008.
[15] A. Liu and P. Ning, "Tinyecc: a configurable library for elliptic curve cryptography in wireless sensor networks," in 5th European conference on Wireless sensor networks, 2008.
[16] A. S. M. C. M. Szczechowiak, P.and Kargl, "On the application of pairing based cryptography to wireless sensor networks," in 2nd ACM Conference on Wireless Network Security, 2009.
[17] H. Wang and Q. Li, "Efficient implementation of public key cryptosystems on mote sensors," in International Conference on Information and Communication Security, 2006.
[18] E. Ozturk, B. Sunar, and S. E., "Low-power elliptic curve cryptography using scaled modulararithmetic," in 6th International Workshop on Cryptographic Hardware and Embedded Systems, 2004.
[19] I.-F. Akyildiz, T. Melodia, and K.-R. Chowdhury, "A survey on wireless multimedia sensor networks," Computer Networks, Elsevier, vol. 51, pp. pp. 921–960, 2004.
[20] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in 1st Workshop on Wireless Sensor Networks and Applications, 2002.
[21] J. McCulloch, S. M. Guru, and D. Hugo, "Wireless sensor network deployment for water use efficiency in irrigation," in Workshop on Real-World Wireless Sensor Networks, 2008.
[22] N. Koblitz, "Elliptic curve cryptosystems," Mathematics of Computation, vol. 48, pp. pp. 203–209, 1987.
[23] V.-S. Miller, "Use of elliptic curves in cryptography," Advances in Cryptology, vol. 218, pp. pp. 417–426, 1986.
[24] NIST, "Recommended elliptic curves for federal government use." National Institute of Standards and Technology, Tech. Rep., 1999.
[25] M. A. V. S. Hankerson, D., Guide to Elliptic Curve Cryptography, Springer, Ed., 2004. [26] A. Menezes, Elliptic Curve Public Key Cryptosystems. Kluwer Academic Publishers (1993)., Springer, Ed. Kluwer Academic, 1993.
[27] NIST, "Key management guideline-workshop document http://csrc.nist.gov/encryption/kms /keymanagementguideline-(workshop).pdf." NIST, Tech. Rep.
[28] G. Gaubatz, J.-P. Kaps, E. ztrk, and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," in Third IEEE International Conference on Pervasive Computing and Communications Workshops, 2005.
[29] J. Lopez and J. Zhou, Wireless Sensor Network Security. IOS Press, 2008.
[30] U. D. of Commerce, "Secure hash standard (2009)." National Institute of Standards and Technology, Tech. Rep., 2009.
[31] N. P., A. Liu, and W. Du, "Mitigate dos attacks against broadcast authentication in wireless sensor networks," in ACM Transactions on Sensor Networks, 2008.
[32] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J.-D. Tygar, "Security protocols for sensor networks," in 7th Annual International Conference on Mobile Computing and Networking, 2001.
[33] K. Bicakci, "Pushing the limits of one-time signatures," in International Conference on Security of Information and Networks, 2009.
[34] K. Ren, W. Lou, and P.-J. Zeng, K.and Moran, "On broadcast authentication in wireless sensor networks," IEEE Transactions on Wireless Communication, vol. 6(11), pp. 4136–4144, 2007.
[35] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," IEEE Transactions on Vehicular Technology, vol. 58 (8), pp. 4554–4564, 2009.
[36] J. Wolkerstorfer, "Scaling ecc hardware to a minimum," in Cryptographic Advances in Secure Hardware, 2005.
[37] S. Kumar and C. Paar., "Are standards compliant elliptic curve cryptosystems feasible on rfid," in Workshop on RFID Security, July 2006.
[38] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, "Low-cost elliptic curve cryptography for wireless sensor networks," in 3rd European Workshop on Security and Privacy in Ad hoc and Sensor Networks, September 2006.
[39] G. Bertoni, L. Breveglieri, and M. Venturi, "Power aware design of an elliptic curve coprocessor for 8 bit platforms," in PERCOMW'06, 2006.
[40] G.-D. Murphy, E.-M. Popovici, and W.-P. Marnane, "Areaefficient processor for public-key cryptography in wireless sensor networks," in SENSORCOMM2008, 2008.
[41] A.-O. Portilla, J.and Marnotes, E. la Torre, T. Riesgo, O. Stecklina, S. Peter, and P. Langendrfer., "Adaptable security in wireless sensor networks by using reconfigurable ecc hardware coprocessors," International Journal of Distributed Sensor Networks, 2011.
[42] G. Panic, T. Basmer, O. Schrape, S. Peter, F. Vater, and T.-H. Klaus, "Sensor node processor for security applications," in 18th IEEE International Conference on Electronics, Circuits, and Systems, 2011.

