Chapter 10

Asymmetric Encryption in Wireless Sensor Networks

Gustavo S. Quirino, Admilson R. L. Ribeiro and Edward David Moreno

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/48464

1. Introduction

A Wireless Sensor Network (WSN) is composed of autonomous devices called sensor nodes that generally have low computational power, limited data transmission and power constraints. A WSN consists of sensor nodes that capturing information from an environment, processing data and transmitting them via radio signals. WSNs are increasingly present in our days and can be found in environmental area (climatic measurements, presence of smoke), in health area (measurement of vital signs, temperature), home automation (motion sensor and image sensor) and other areas. Generally, WSNs have no fixed structure, and in many cases there is no monitoring station of sensor nodes during the operational life of the network, so a WSN must have mechanisms for self-configuration and adaptation in case of failure, inclusion or exclusion of a sensor node.

Security requirements of WSNs are similar to conventional computer networks, therefore parameters such as confidentiality, integrity, availability and authenticity must be taken into account in creation of a network environment. Due to limitations of WSNs, not all security solutions designed for conventional computer networks can be implemented directly in WSN. For a long time, it was believed that the public key cryptography was not suitable for WSNs because it was required high processing power, but through studies of encryption algorithms based on curves was verified the feasibility of that technique in WSN.

The cryptographic algorithm RSA is currently the most used among the asymmetric algorithms, working from the difficulty of factoring large prime numbers. Standardized by NIST[1], this algorithm is widely used in transactions on the Internet. The algorithms Elliptic / Hyperelliptic Curve Cryptography (ECC / HECC) were created in 80s, and are based on the difficulty of solving the discrete logarithm problem on elliptic curves and hyperelliptic respectively. Despite its complexity the algorithm based on elliptic and hyperelliptic curves have been extensively studied in academia. Recently, the public key algorithm called

[1] U.S. Agency for technology that has a partnership with industry to develop and apply technology, measurements and standards. Further information: www.nist.gov

Multivariate Quadratic Almost Group (MQQ) was proposed in academia. Experiments performed in the FPGA and PC platforms showed that MQQ is faster than algorithms such as RSA and ECC [1, 2]. Algorithms involved in this study are asymmetric, but each one works with a specific encryption mode.

Many studies have evaluated performance of cryptographic algorithms in WSNs, but there is no standardization in the performance analysis. As stated by Margi [3] studies on performance evaluation of cryptographic algorithms for WSNs are often quite different in terms of methodology, platform, metrics and focus of analysis, what difficult a direct comparison among the obtained results. Thus, this chapter describes a theoretical study of cryptographic such as RSA, ECC, HECC and MQQ as well as the performance analysis of these algorithms in WSN.

2. Wireless sensor network

Sensor nodes are electronic devices that have as main components units of storage, processing, sensing and transmission. Usually, these devices have low computational power, nevertheless play an important role in ubiquitous computing, because they have function of collecting data in a given environment, passing them through a wireless network. According to [4] WSNs can be seen as a special type of MANET (Mobile Ad hoc NETwork) that tend to run a collaborative basis where the elements (sensor nodes) provide data that are processed (or consumed) by special nodes called sink nodes.

The operation area of a WSN is very large and can be used in environmental monitoring, control temperature and humidity, vehicle traffic control, monitoring of human body organs, among others. Figure 1 illustrates a scenario of WSNs in the medical area where patients that are being monitored can be in a hospital, at home, or anywhere else performing an activity routine. Sensing data are sent to health professionals through the Internet.

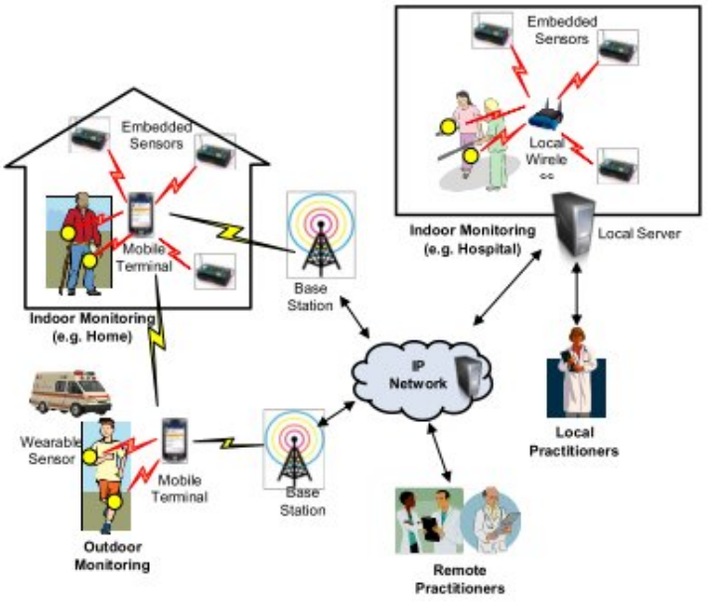

Figure 1. Scenario of Wireless Sensor Network [5]

Some application areas of WSNs require security in the information transport, such as the scenario illustrated by figure 1, where sensor nodes implanted in the human body reporting

to a hospital. According to [5] in the health field, authentication and access control are the main challenges of a mobile dynamic network topology with limited resources. Besides medical area, several other areas also need security in their transmissions as: industry, asset security and military applications. A U.S. security agency called DARPA[2] has been developed numerous studies involving security in WSNs for military purposes.

2.1. Sensor devices

Sensor devices are basically formed by a computational part responsible for storing and transmitting data, and a sensing portion which can be formed by one or more sensors, such as acoustic, seismic, infrared video camera, heat temperature and pressure [4]. In general two modulation formats are available: Frequency-Shift-Keyed (FSK) operating at 433 and 868-915 MHz and direct sequence spread spectrum (DSSS) operating at 2.4 GHz band that transmit 802.15.4 and ZigBee standards. The reach of the radios varies from 10 to 100 meters. The antenna configuration can cause transmission rates from 19.2kbps to 240kbps [6]. Currently, the sensor nodes can vary between mode activity, inactivity (idle) and low consumption (sleep) in order to save energy. The energy issue is important because most of sensor nodes are powered by batteries. Nowadays, the main sensor nodes available are LOTUS, IRIS, MICAz, Mica2, and TELOSB CRICKET.

Figure 2 illustrates the actual format of sensor nodes such as IRIS TelosB and MicaZ. During the development of this work were not found Brazilian companies that commercialize sensor nodes. A budget held in the Chinese company Mensic[3] in jan/2012 showed that a Micaz cost U$ 114.00 and a TelosB U$ 160.00, excluding import duties also should be minimum purchases of U$ 1000.00.

Figure 2. Sensors [6]

2.2. Application environments

WSNs can be applied in various areas. According to Loureiro [4], WSNs can be used in following situations:

- **Environment -** Monitoring of environmental variables such as buildings, residences and external locations such as oceans, volcanoes, deserts, etc..
- **Traffic -** Monitoring of vehicle traffic on highways, railroads, rivers, oceans, etc..

[2] Advanced Research Projects Agency. More information at: http://www.darpa.mil
[3] Chinese company specializing in electronic devices. Further information: http://www.memsic.com/

- **Security -** To provide security in homes, shopping centers, farms, among others.
- **Military -** To detect the presence of enemies, explosions, presence of hazardous materials as poison gas and radiation.

2.3. Security vulnerabilities

In most of applications, sensor devices are spread over large areas, what difficult a individual control of network components. Moreover, wireless communication allows an attacker can trigger attacks without having physical access to the device, so according to Shi and Perrig [7] attacks on WSNs can be divided into three main types: (1) Attack of authentication and confidentiality: Consists of attacks change, repetition or modification packages. (2) Availability network Attack: Generally known as DoS attacks or negation of service, this attack involves the application of techniques that make the network unavailable. (3)Attack on integrity: this type of attack the attacker's goal is to inject false data on the network, keeping the network available, but traveling fictitious data. Table 1 described by Wang [8] illustrates the most common types of attacks in WSN considering the network layer in which they operate.

Layer	**Types of Attacks**
Physical Layer	*jamming* ou ataque de interferï£¡ncia
Link Layer	*collision, exhaustion, unfairness*
Network Layer	*spoofed routing information and selective forwarding, sinkhole, sybil, wormhole, Hello flood, Ack Flooding,*
Transport Layer	*Flooding De-synchronization*
Application Layer	*Malicious Node*

Table 1. Possible attacks on a Wireless Sensor Network [8]

At the physical layer can occur the following attacks: jamming and tampering. The attack jamming consists in the interference of radio frequency signal that sensor nodes use to communicate. The tampering attack occurs due to physical vulnerability of sensor nodes spread over large areas, therefore susceptible to capture, breaking the circuit, setting modification or even replacement of a network node by a malicious sensor node [9]. At link layer attacks can be of the collision, when two sensor nodes attempt to transmit while at the same frequency, in this case the packet is discarded and must be retransmitted [10]. The attacker may cause intentional collisions by a malicious sensor node. Repeated collisions can lead to exhaustion of resources, making it unavailable sensor nodes. Also in the link layer unfairness attack is a type of DoS when the adversary causes degradation of real-time applications run on other sensor nodes by intermittent interruption of the transmission of their frames.

Denial of Service (DoS) attacks consist of flooding the receiver with no other requests for communication can be performed during the attack, leaving the involved nodes unavailable for new connections.

In the network layer attacks can occur of type Spoofed Routing Information, where the attacker modifies routing table information. The routes make false packets do not reach the correct destination, or even make the referral to consume more resources than normal [11]. The Selective Forwarding attack is the involvement of a sensor node by an attacker who causes

some messages to be routed and other discarded [11]. In the Sinkhole attack the attacker causes a compromised sensor node is seen as most efficient route to the sink of the network, thus the neighboring nodes will always use the attacker to send their data [12][11][10].

The Sybil attack happens when a malicious node takes over a network identity. According to Douceur [13] this attack was originally intended for distributed systems of redundant data storage, but it is also effective against routing algorithms, data aggregation, and resource allocation, among others. The Wormhole attack consists in a low latency link between two sensor nodes of a network through which an attacker generates messages with court order to exhaust the resources of the devices [11]. In the Hello Flood attack the attacker can use a high power transmitter to fool a large number of sensor nodes, making them believe they are close [11].

Subsequently the attacker sends a fake shortest path to base station, and all nodes receiving Hello packets, try to convey through the attacking node. However, these nodes are out of radio range of the malicious node. Some routing information algorithms use state of sensor nodes. The Acknowledgment Spoofing attack consists in spreading false information about the states of neighboring sensor nodes performed by a malicious sensor node in order to prevent packets from reaching their destinations [11].

In the transport layer, Flooding attack consists in the flood of requests to new connections in order to exhaust the resources of memory and prevent the closure of legitimate requirements of provisions. The De-synchronization attack refers to the interruption of an existing connection [10]. In this attack the attacker captures messages forcing the sender to resend them expending energy unnecessarily.

There are also attacks that exploit vulnerabilities in authentication and data confidentiality. The attack consists of setting replication of a malicious node assumes the identity of a network node. This false node can forward packets in corrupt or false routes. If the attacker has physical access to network, it can copy cryptographic keys and use them in false messages. Also the attacker can deploy the malicious node at strategic locations in order to divide the WSN.

Preserving privacy in data transmission in WSN is challenging, since this type of network allows remote access. Moreover, a single adversary can monitor multiple networks simultaneously [14]. Eavesdropping and passive monitoring are the most common and easiest attack to data privacy. In this type of attack the spy monitors the data transfer and can access its contents if no encryption mechanism implemented in the network being monitored. The traffic analysis is usually applied in conjunction with the attack of listening and passive monitoring. It consists of the preliminary analysis of network traffic to identify nodes that are generating data exchange that interest to the attacker. Finally, the camouflage attack, wherein the malicious attacker deploys a node in the network forwards packets to sensor nodes being monitored.

Through this analysis one can see that there is a range of attacks for WSNs in all layers of the TCP / IP protocol stack. Furthermore, it is apparent that a common point in most attacks is the exploitation of low computing capacity of sensor nodes, as are injected false data and routes are always altered in order to occupy the lower transmission capacity of the sensor nodes, or eliminate its reserve energy. Others attacks yet unidentified may occur in WSN, and protect the network from these threats can be a difficult task.

2.4. Defense mechanisms

Different types of WSN applications require different security requirements. In an environment of temperature monitoring, where researchers collect data for research, it may be that safety requirements are not very important, but the monitoring of radiation near a nuclear power plant requires authenticity assurance, confidentiality, availability and integrity. Various architectures have been developed to provide security in WSNs, among them are: SPINS, TinySec MiniSec and besides these the IEEE 802.15.4 include a security framework to meet the services of data integrity, confidentiality and authenticity [3].

SPINS (Security Protocols for Sensor Networks) developed by Perrig [15] consists of a set of security protocols that acts through encryption and message authentication codes. The TinySec was designed and implemented in the TinyOS operating system to be a mechanism for providing confidentiality, integrity and authenticity of the data link layer. It uses the CBC mode of operation that may be combined with various block ciphers as RC5 and skipjack [16]. The MiniSec is a protocol layer of security to WSN using OCB (Offset Codebook) mode for operating the block cipher, which eliminates the need of adding filler to the clear text blocks [17]. The standard IEEE 802.15.4 provides integrity, access control, confidentiality and replay protection in the link layer. The cryptographic algorithm used in this standard is AES [18].

According to Loureiro [4], a WSN tends to be autonomous and requires a high degree of cooperation to perform the tasks defined for the network. This means that traditional distributing algorithms, such as communication protocols and election of leader, should be reviewed for this type of environment before being used directly. Taking account also the limited computational power and especially of limited energy of devices is possible to deduce that not everything that works efficiently in traditional computer networks can be used in WSNs The computational limitations of a device restricting the choice of cryptographic algorithms and protocols safety. Furthermore, the lifetime of the batteries using techniques preclude the complex of security because it drastically decreases the life span of the network. [18].

Encryption is the security solution most applicable in computing. In recent years asymmetric algorithms have been extensively studied in embedded systems with low computational power. The next section discusses concepts of cryptography, and the description of the algorithms RSA, ECC, HECC and MQQ.

3. Concepts of cryptography

Data encryption emerged before the invention of computer. Diplomats, enthusiasts and mainly militaries contributed to the evolution of this art that consists in distort the information that is being transported, so that only the authorized recipient can decipher it. In this regard, a cryptographic algorithm can be set as a function that converts encrypted message in clear messages and vice versa, making use of a cryptographic key.

Most cryptographic algorithms are public, according to Tanembaum [19] keeping the algorithm public gets rid of the creator from eager cryptologist to decode the system in order to publish articles, and that after five years of their exposure and no decoding was successful, the algorithm is assumed to be solid. Secrecy is the key that has the function to parameterize the cryptographic function, ie only with the key can encrypt or decrypt a message. Another important factor is that the key have the ability to change the output of the algorithm, so every

change of key cryptographic algorithm generates a new encrypted message. The key size is critical in a project, because the longer the key, more work will be crypto analyst to try to decipher the message. In general, keys have sizes of 64, 128 or 256 bits and may be higher or lower, according to security needs.

Currently, in addition to confidentiality, encryption also operates in the fields of integrity of authentication and is described below:

- **Confidentiality:** ensuring that only the sender and receiver have the ability to understand the message being exchanged.
- **Integrity:** Ability to check if a message was altered during transmission.
- **Authentication:** Medium to prove the identity of an individual communication.

According to Boyle and Newe [20] encryption is the standard method for defending a WSN of most possible attacks, and the various levels of encryption implicate variations in overhead in the form of growth in the size of the package data, code size, processor usage, memory, etc.. The choice of a cryptographic algorithm to get efficient for a WSN is a large debate among researchers. According to Chen [21] the cryptographic methods used in WSN should meet the constraints of computational devices, and go through evaluation before being implanted.

3.1. Classes of cryptographic algorithms

Traditionally users of encryption algorithms used simple, but currently the goal is to make the algorithm so complex that without the key is practically impossible to extract some information through a cryptanalysis. The classes of cryptographic algorithms say about it as an encryption key is changed and also the quantity of keys involved in the application of the method. Most existing cryptographic algorithms can be classified as symmetric or asymmetric.

3.1.1. Symmetric encryption

Symmetric encryption or secret key cryptography is the use of only a key, both in the encryption and decryption of data. By the year 1976 this was the only known method for the use of encryption, but to be effective you need a secure channel for communication in which a cryptographic key can be changed.

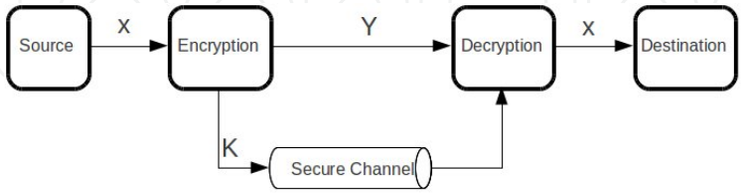

Figure 3. Symmetric Cryptography [22]

Figure 3 illustrates a communication through symmetric encryption. The text is encrypted X and Y become the message through the encryption algorithm and key k. The message Y is sent to the receiver, which uses the key k to decrypt it, turning it on again in the text X. Also according to figure 3 you can see that the key k is transported by a secure channel, for the possession of it, a potential attacker could easily make the reading the original text. AES and DES are two examples of algorithms that are part of the class symmetrical.

3.1.2. Asymmetric encryption

The public key cryptography or asymmetric cryptography came up with a radical change of paradigms. According to Stallings [22] public key algorithms are based on mathematical functions, instead of permutation and substitution. Besides the single most important thing is that the public key cryptography is asymmetric, involving the use of two different keys, in contrast to the conventional symmetric encryption, which uses only one key. The use of two keys has profound consequences in the areas of confidentiality, key distribution and authentication. The main distinguishing feature of asymmetric encryption is that it allows the establishment of a secure communication between individuals, without the requirement of the previous share a single cryptographic key.

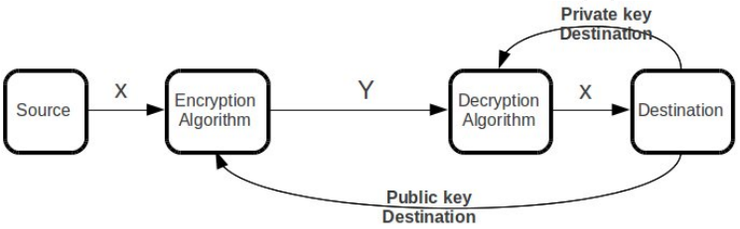

Figure 4. Asymmetric Cryptography [22]

In this class of cryptographic algorithms are used two different keys for encryption and decryption: a public key and its corresponding private key. In this model, in accordance with figure 4, the receiver releases its public key to the sender can encrypt the message, but only the private key of the receiver, which is kept secret is able to decrypt it.

3.1.3. Symmetric x asymmetric cryptography

The IEEE 802.15.4 standard of 2011 defines parameters for low-range personal area networks (LR-WPANs). The first version of this standard was launched in 2003, and the second one [20] was appointed to be the standard communication protocol for WSNs. The encryption mechanism specified in IEEE 802.15.4 standard is based on encryption symmetric key. But according to Sen [23] recent studies have shown that it is possible to implement public key encryption using the right selection of algorithms and associated parameters, and optimization techniques for low power. In some cases the public-key cryptography efficiently obtained similar or even greater than symmetric key encryption using keys smaller. According to Struik [24] is already proven that public-key algorithms developed are suitable for hardware in WSNs.

3.2. RSA algorithm

In the introductory paper about RSA, the authors [25] proposed a method to implement a public key cryptosystem whose security is based on the difficulty to be factoring large prime numbers. Through this technique it is possible to encrypt data and to create digital signatures. It was so successful that today is the RSA public key algorithm used most in the world. The encryption scheme uses RSA and signature of the fact that:

$$m^{ed} \equiv m(modn) \tag{1}$$

for m integer. The encryption and decryption schemes are presented in Algorithms 1 and 2. The decryption works because $c^d \equiv (m^e)^d \equiv m(modn)$. The safety lies in the difficulty of computing a clear text m from a ciphertext c$m^e modn$ and the public parameters n (e).

Algorithm 1: RSA Encryption

Input: RSA public key (n,e), Plain text m $\in$ [0, n-1]
Output: Cipher text c
begin
| **1.** Compute c = m^e mod n **2.** Return c.
end

Algorithm 2: Decryption RSA

Input: Public key (n,e),Private key d, Cipher text c
Output: Plain text m
begin
| **1.** Compute m = c^d mod n
| **2.** Return m.
end

3.3. Algorithms based on curves

The main idea of the algorithms based on curves is to build a set of points of an elliptic curve for which the discrete logarithm problem is intractable. According to Blake [26] cryptosystems based on elliptic curves is an interesting technology because they reach the same level of security systems such as RSA, using minor keys, and thus consuming less memory and processor resources. This characteristic makes them ideal for use in smart cards and other environments where features such as storage, time and energy are limited.

The scenario of using public key cryptographic algorithms are changing, because according to Koc [27] in terms of public key encryption algorithm RSA continues to lead the number of implementations, but the number of applications that are using algorithms elliptic curves is increasing considerably thanks to the standardization performed by NIST. The algorithms based on curves are standardized according to the ANSI X9.62, FIPS 186-2, IEEE 1363-2000 and ISO / IEC 15946-2. According to Amin [28] public key encryption includes algorithms for key agreement, encryption and digital signatures. Among the algorithms that operate in key agreement, it can mention the Elliptic Curve Diffie-Hellman (ECDH), data encryption on the Elliptic Curve Integrated Encryption Standard (ECIES) and generating the digital signature Elliptic Curve Digital Signature Algorithm (ECDSA).

3.3.1. ECC algorithm

In the mid-80 [29] and [30] proposed a method of cryptography based on elliptic curves ECC . According to creators of the ECC [4], an elliptic curve is a plane curve defined by the following equation:

$$y^2 = x^3 + ax + b \tag{2}$$

[4] Elliptic curve cryptography. More information on the site of the workshop on Elliptic Curve Cryptography which is in issue. Site: http://ecc2011.loria.fr/index.html

The efficiency of this algorithm is based on finding a discrete logarithm of a random element that is part of an elliptic curve. To get an idea of the applicability of the algorithms based on elliptic curves on devices with computational constraints [31] argue that the efficiency of ECC cryptographic algorithm with key sizes of approximately 160 bits is the same obtained using the RSA algorithm with 1024 bit key. Algorithms several features are based on elliptic curves, including key management, encryption and digital signature. Key management algorithms are used to share secret keys, encryption algorithms enable a confidential communication and digital signature algorithms authenticate a participant communication as well as validate the integrity of the message.

The procedures of decryption and encryption through elliptic curve analogous to ElGamal encryption scheme are described in the algorithms 3 and 4. The pure text m is first represented as a point M, and then encrypted by the addition to kQ, where k is an integer chosen randomly, and Q is the public key.

Algorithm 3: *ElGamal elliptic curve encryption*

Input: Parameters field of elliptic curve (p, E, P, n), Public key Q, Plain text m
Output: Cipher text (C_1, C_2)
begin
1. Represent the message m as a point M in E (F_p)
2. Select $k \in R^{[1,n-1]}$.
3. Compute $C_1 = kP$
4. Compute $C_2 = M + kQ$.
5. Return (C_1, C_2)
end

Algorithm 4: *ElGamal elliptic curve decryption*

Entrada: Parameters field of elliptic curve (p, E, P, n), Private key d, Cipher text (C_1, C_2)
Saída: Plain text m
início
1. Compute $M = C_2 - dC_1$, and m from M.
2. Return (m).
fim

The transmitter transmits the points $C_1 = kPeC_2 = M + kQ$ to receiver who uses his private key d to compute:

$$dC_1 = d(kP) = k(dP) = kQ, \tag{3}$$

and then calculating $M = C_2 - kQ$. An attacker who wants to read of M need to calculate kQ. This model algorithm have been extensively studied since according to Amin [28] in recent years the ECC has attracted attention as a security solution for wireless networks, because the use of small keys and low computational overhead.

3.3.2. HECC algorithm

The HECC was created in 1988 by Koblitz [32] as a generalization of elliptic curves. According to Batina [33] the unique difference between ECC and HECC is at average level that in this

case consists of different sequences of operations. The HECC uses more complex operations, but works with smaller operands. According to Chatterjee [31] the hierarchy of operations in the HECC and ECC algorithms can be divided into three levels. The first level is the scalar multiplication on the second level are point operations group / splitter and the third level, finite field operations. The authors further inform that the main difference between the ECC and HECC is in the operations group, as different from the ECC, the points on the curve hiperelliptic not form a group. HECC is more complex than the ECC, but uses small numbers.

According to [27] a hyperelliptic curve is a special type of non-singular, projective curve. For our purposes, a hyperelliptic curve, of genus $g \geqslant 1$ over k is the set of points $(X, Y) \in k^2$ that satisfy

$$y^2 + h(X)Y = f(X) \tag{4}$$

where h and f are polynomials in k[X] with $deg(f) = 2g + 1, deg(h) \leq g$, together with a point "at infinity", P_∞. An elliptic curve is just a hyperelliptic curve of genus 1.

3.4. Multivariate Quadratic Quasigroup (MQQ)

The cryptographic algorithms presented above have their security based on computationally intractable mathematical problems: computational efficiency of calculating the discrete logarithm and integer factorization [1]. In 2008, it was proposed a new scheme called multivariate quadratic public key near group (MQQ) [34]. This algorithm is based on multivariate polynomial transformations of nearly quadratic and groups having the following properties [1, 34].

- Highly parallelizable unlike other algorithms that are essentially sequential.
- The encryption speed is comparable to other cryptosystems public key based on multivariate quadratic.
- The decryption speed is typical of a symmetric block cipher.
- Post-Quantum Algorithm

According to Ahlawat [34, 35] MQQ gives a new direction for the cryptography field and can be used to develop new cryptosystems the public key as well as improve existing cryptographic schemes. Furthermore according to El-Hadely and Maia [2, 34] experiments showed that the hardware MQQ can be as fast as a typical symmetric block cipher, being several orders of magnitude faster than algorithms such as RSA, DH and ECC.

A generic description for the scheme is a typical system MQQ multivariate quadratic $T \circ P' \circ S : \{0,1\}^n \rightarrow \{0,1\}^n$ where T and S are two nonsingular linear transformations and P'is a multivariate mapping bijetivo quadratic over $\{0,1\}^n$. The mapping $P' : \{0,1\}^n \rightarrow \{0,1\}^n$ is defined in the algorithm 5.

The algorithm for encryption with the public key is the direct application of the set of n multivariate polynomials $P = \{P_i(x_1, ..., x_n) | i = 1, ..., n\}$ on the vector $x = (x_1, ..., x_n)$, or is $y = P(x)$. Can be represented as $y = P(x) \equiv y \equiv A.X$. The algorithm 6 is described a decryption using the private key $(T, S, *_1, ..., *_8)$.

Algorithm 5: Non-linear mapping P'

Input: A vector $x = (f_1, ..., f_n)$ of n linear Boolean functions of n variable. We implicitly suppose that a multivariate quadratic quasigroup * is previously defined, and that n = 32k, k = 5,6,7,8 is also previously determined.

Output: : 8 linear expressions $P'_i(x_1, ..., x_n)$, i = 1,...,8 and n - 8 multivariate quadratic polynomials $P'_i(x_1, ..., x_n), i = 9, ..., n$.

begin

1. Represent a vector $x = (f_1, ..., f_n)$ of n linear Boolean functions of n variables $x_1, ..., x_n$ as a string $x = (X_1, ..., X_{n/s})$ where X_i are vector of dimension 8;

2. Compute $Y = Y_1, ..., Y_{n/8}$, where $Y_1 = X_1, Y_{j+1} = X_j * X_{j+1}$,for even j=2,4,...and $Y_{j+1} = X_{j+1} * X_j$, for odd j=3,5,...

3. Output (y)

end

Algorithm 6: Decryption Algorithm MQQ and sign with private key T, S, *1,...,*8

Input: Vector $y = y_1, ..., y_n$

Output: Vector $x = x_1, ..., x_n$ such that P(x) = y

begin

1. y' = $T_{-1}(y)$.

2. W = $y'_1, y'_2, y'_3, y'_4, y'_5, y'_6, y'_{11}, y'_{16}, y'_{21}, y'_{26}, y'_{31}, y'_{36}, y'_{41}$.

3. Z = $Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8, Z_9, Z_{10}, Z_{11}, Z_{12}, Z_{13} = Dob^{-1}(W)$.

4. $y'_1 \longleftarrow Z_1, y'_2 \longleftarrow Z_2, y'_3 \longleftarrow Z_3, y'_4 \longleftarrow Z_4, y'_5 \longleftarrow Z_5, y'_6 \longleftarrow Z_6, y'_{11} \longleftarrow Z_7, y'_{16} \longleftarrow Z_8, y'_{21} \longleftarrow Z_9, y'_{26} \longleftarrow Z_{10}, y'_{31} \longleftarrow Z_{12}, y'_{41} \longleftarrow Z_{13}$.

5. $y' = Y_1, ..., Y_k$ onde Y_i are vectors of dimension 5.

6. Considering $*_i$, i = 1,...,8, obter x' = $X_1, ..., X_k$, such that,
$X_1 = Y_1, X_2 = X_{1\backslash 1}Y_2, X_3 = X_{2\backslash 2}Y_3, X_1 = X_{i-1_{\backslash 3+((i+2) mod 6)}}Y_i$

7. x = $S^{-1}(x')$

end

4. Performance evaluation

Some authors [36] believed that HECC would be less efficient than ECC due to complex structure of the group's operations, but it was reported that there was a detailed analysis of the efficiency of these cryptosystems in embedded systems. The work done by [37] and [38] confirmed the superiority in efficiency of ECC compared to RSA.

[37] showed that the ECC 160 bits is two times better than RSA 1024 bits considering code size and power consumption. [37]performed the tests in 8051 and AVR platforms. [38] pointed out that ECC 160 bits uses four times less energy than RSA 1024 bits in Mica2dot platform.

Only [36] and [31] presented a general analysis, comparison of ECC and HECC, which showed a trend of superiority of HECC on embedded systems. [31] showed that in the encryption, the HECC reaches the same time of ECC using smaller keys. Regarding the time decrypting, the HECC always performed better. The scalar multiplication of HECC is two times faster than ECC. [31] used the platform jdk1.6 in his assessment.

[39] conducted a comparison of ECC and HECC over the computational time of point multiplication on a platform with limited resources. The results showed that the ECC consumed 210ms and the HECC consumed 546ms.

[40] have implemented ECC and HECC on diferent embedded platforms with high practical relevance, namely ARM, ColdFire, and PowerPC. Table 2 show that for the boards at hand they could achieve the best timings for the HECC implementation on the PowerPC. One scalar multiplication for HECC took 117 ms and 84.9 ms for genus-2 and genus-3 curves, respectively. The scalar multiplication for ECC can be performed fastest on the PowerPC at 50MHz resulting in 106.3 ms.

Group Order		**ECC**	HECC		
			g = 2	g = 3	g = 4
$\approx 2^{160}$	ARM @ 50Mhz	469.96	446.46	515.46	316.6
	ColdFire @ 90Mhz	152.1	155.6	219.4	123.6
	PowerPC @ 50Mhz	106.3	117	141.4	84.9

Table 2. Timings of the scalar multiplication of ECC and HECC on diferent embedded platforms (in ms). [40]

[41] conducted tests with authentication protocols based on RSA and HECC algorithms, comparing the computational time in the Palm III and Wireless Tolkit platforms. The results showed that the protocol based on HECC is 1.37 times faster in key generation and 1.38 times faster with respect to signal generation.

According to Gligoroski [42] in software, digital signature performed by MQQ is 300 to 7000 times faster than the signature of RSA and ECC algorithms. Already in hardware, the superiority of MQQ can reach 10,000 times. The speed of 59 bytes of authentication is compared by the authors [42] and the results are shown in Table 3. The results of the performance evaluation showed that MQQSIGN is at least 325 times faster than RSA and ECC.

[34] evaluated the time of encryption and decryption of algorithms RSA and MQQ in the MicaZ and TelosB platforms. The MQQ showed the time 825.1 ms to encrypt and 116.6 ms to decrypt in TelosB and 445ms to decrypt in MicaZ . Still according to [34] the MQQ 160bits is 909 times faster in the encryption and 5470 times faster in the decryption when compared to the RSA 8bits.

Algorithm	**Signing of 59 bytes**
RSA 1024	2,230,848
ECC 160	1,284,800
MQQSIGN 160	3,440
-	-
RSA 2048	14,815,324
ECC 224	2,108,556
MQQSIGN 224	4,160

Table 3. Performance comparison of RSA, ECC and MQQ in CPU cycles. [42]

According to [2] that implemented in FPGA a 160bit instance of the newly published public key scheme MQQ, the results of their implementation and the Table 4 show that in hardware, MQQ public key algorithm in encryption and decryption (that means also in verication and

signing) can be as fast as a typical block cipher and is several orders of magnitude faster than most popular public key algorithms like RSA, DH or ECC.

Algorithm	1024-bit RSA	160-bit MQQ	128-bit AES
Throughput	40Kbps	44.27Gbps / 399.04Mbps	7.78Gbps

Table 4. Synthesis Results for 160bit MQQ realized in Virtex-5 chip xc5vfx70t-2-ő1136 [2]

5. Conclusion and future work

It is natural that the spread of ubiquitous computing to increase the number of devices with low computing power scattered all over the planet. The security of data transmissions from these devices should be improved in a preventative manner to avoid possible attacks. Regarding WSNs, RSA public key algorithm is the most commonly used is standardized, and achieves efficiency relatively good. The algorithm based on elliptic curves have been extensively studied in academia as an alternative to RSA, and the results show that it is possible to achieve good results with smaller keys. The algorithm MQQ was discovered recently and showed significant results when compared to RSA and ECC, taking as parameters authenticity and digital signature. This algorithm is post-quantum, and may even be a good solution when the quantum computation is standardized. Despite the satisfactory results of MQQ front of RSA and ECC algorithms, there is not a work about performance evaluation specific to encryption and decryption of data.

Author details

Gustavo S. Quirino, Admilson R. L. Ribeiro and Edward David Moreno
Universidade Federal de Sergipe, Brasil

6. References

[1] D. Gligoroski, S. Markovski, and S.J. Knapskog. A public key block cipher based on multivariate quadratic quasigroups. *Arxiv preprint arXiv:0808.0247*, 2008.

[2] M. El-Hadedy, D. Gligoroski, and S.J. Knapskog. High performance implementation of a public key block cipher-mqq, for fpga platforms. In *Reconfigurable Computing and FPGAs, 2008. ReConFig'08. International Conference on*, pages 427–432. IEEE, 2008.

[3] Marcos Simplicio Margi, MS Jr, and Tereza C. M. B. Carvalho Barreto. Segurança em Redes de Sensores Sem Fio. In *Simpósio Brasileiro em Segurança da Informação*, pages 149–194, 2009.

[4] A.A.F. Loureiro, J.M.S. Nogueira, L.B. Ruiz, R.A. de Freitas Mini, E.F. Nakamura, and C.M.S. Figueiredo. Redes de sensores sem fio. In *Simpósio Brasileiro de Redes de Computadores (SBRC)*, pages 179–226, 2003.

[5] Xuan Hung Le, Ravi Sankar, Murad Khalid, and Sungyoung Lee. Public Key Cryptography - based Security Scheme for Wireless Sensor Networks in Healthcare. In *4th International Conference on Ubiquitous Information Management and Communication*, pages 1–7, 2010.

[6] MEMSIC. Mote Processor Radio Mote Interface Boards User Manual - Document Part Number: 7430-0021-09 Rev A, 2012

[7] E. Shi and A. Perrig. Designing secure sensor networks. *Wireless Communication Magazine*, 11(6):38–43, 2004.

[8] Y. Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 8(2):2–23, 2006.
[9] X. Wang, W. Gu, S. Chellappan, Dong Xuan, and Ten H. Laii. Search-based physical attacks in sensor networks: Modeling and defense. Technical report, Department of Computer Science and Engineering, Ohio State University, 2005.
[10] A.D. Wood and J.A. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, 2002.
[11] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *1st IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127. IEEE, 2003.
[12] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: Analysis and defenses. In *3rd International Symposium on Information Processing in Sensor Networks*, pages 259–268. ACM Press, 2004.
[13] J. Douceur. The sybil attack. In *1st International Workshop on Peer-to-Peer Systems - IPTPS*, 2002.
[14] H. Chan and A. Perrig. Security and privacy in sensor networks. *IEEE Computer Magazine*, pages 103–6105, 2003.
[15] A. Perrig, R. Szewczyk, JD Tygar, V. Wen, and D.E. Culler. Spins: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.
[16] C. Karlof, N. Sastry, and D. Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175. ACM, 2004.
[17] M. Luk, G. Mezzour, A. Perrig, and V. Gligor. Minisec: a secure sensor network communication architecture. In *Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on*, pages 479–488. IEEE, 2007.
[18] IEEE. Ieee 802.15.4 - wireless medium access control (mac)and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). Technical report, Park Avenue, New York, USA: IEEE, 2011.
[19] Andrew S. Tanembaum. *Redes de computadores*. Elsevier, Rio de Janeiro, 4ž edition edition, 2003.
[20] David Boyle and Thomas Newe. Securing Wireless Sensor Networks: Security Architectures. *Journal of Networks*, 3(1):65–77, January 2008.
[21] Xiangqian Chen, Kia Makki, Kang Yen, and Niki Pissinou. Sensor network security: a survey. *IEEE Communications Surveys & Tutorials*, 11(2):52–73, 2009.
[22] W. Stallings. *Network and internetwork security: principles and practice*. Prentice-Hall, Inc., 1995.
[23] Jaydip Sen. A Survey on Wireless Sensor Network Security. *International Journal of Communication Networks and Information Security (IJCNIS)*, 1(2):55–78, 2009.
[24] RenÃl' Struik. Cryptography for highly constrained networks. In *NIST - CETA Workshop 2011*, 2011.
[25] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
[26] I. Blake, G. Seroussi, and N. smart. *Elliptic Curves in Cryptography*. Cambridge, 1999.
[27] C.K. Koc, Nigel Boston, and Matthew Darnall. *About Cryptographic Engineering - Elliptic and Hyperelliptic Curve Cryptography*. Springer, 2009.
[28] F Amin, A H Jahangir, and H Rasifard. Analysis of Public-Key Cryptography for Wireless Sensor Networks Security. *World Academy of Science, Engineering and Technology*, 31(July):530–535, 2008.

[29] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[30] V. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology - CRYPTO Proceedings*, pages 417–426. Springer, 1986.

[31] Kakali Chatterjee, Asok De, and Daya Gupta. Software Implementation of Curve based Cryptography for Constrained Devices. *International Journal of Computer Applications*, 24(5):18–23, 2011.

[32] N. Koblitz. A family of jacobians suitable for discrete log cryptosystems. In *Advances in Cryptology - Crypto88*, pages 94–99. Springer, 1990.

[33] L. Batina, N. Mentens, B. Preneel, and I. Verbauwhede. Flexible hardware architectures for curve-based cryptography. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4–pp. IEEE, 2008.

[34] R.J.M. Maia. Análise da viabilidade da implementação de algoritmos pós-quânticos baseados em quase-grupos multivariados quadráticos em plataformas de processamento limitadas. Master's thesis, USP, 2010.

[35] R. Ahlawat, K. Gupta, and S.K. Pal. From mq to mqq cryptography: Weaknesses new solutions. In *Western European Workshop on Research in Cryptology*, 2009.

[36] Thomas Wollinger, Jan Pelzl, Volker Wittelsberger, Christof Paar, Gökay Saldamli, and Çetin K. Koç. Elliptic and hyperelliptic curves on embedded μP. *ACM Transactions on Embedded Computing Systems*, 3(3):509–533, August 2004.

[37] Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *In Proceedings of the 2004 Workshop on Cryptographic Hardware and Embedded Systems (CUES 2004), Boston Marriott Cambridge Cambridge (Boston)August*, 2004.

[38] Shish Ahmad, Mohd. Rizwan beg, and Qamar Abbas. Energy Saving Secure Framework for Sensor Network using Elliptic Curve Cryptography. *IJCA Special Issue on ?Mobile Ad-hoc Networks?*, pages 167–172, 2010.

[39] Lejla Batina, Nele Mentens, Kazuo Sakiyama, Bart Preneel, and Ingrid Verbauwhede. Public-Key Cryptography on the Top of a Needle. In *2007 IEEE International Symposium on Circuits and Systems*, pages 1831–1834. Ieee, May 2007.

[40] T. Wollinger. *Software and hardware implementation of hyperelliptic curve cryptosystems*. Europ. Univ.-Verl., 2004.

[41] S. Prasanna Ganesan. An Authentication Protocol For Mobile Devices Using Hyperelliptic Curve Cryptography. *International J. of Recent Trends in Engineering and Technology*, 3(2):2–4, 2010.

[42] D. Gligoroski, S.J. Knapskog, S. Markovski, R.S. Ødegård, R.E. Jensen, L. Perret, and J.C. Faugère. The digital signature scheme mqq-sig. *Arxiv preprint arXiv:1010.3163*, 2010.