Architectural Evaluation of Asymmetric Algorithms in ARM Processors

Gustavo S. Quirino and Edward David Moreno
UFS/PROCC, Aracaju, Brasil
Email: {gucefet, edwdavid}@gmail.com

***Abstract*—This paper presents the performance evaluation of asymmetric cryptographic algorithms oriented to embedded platforms used in Wireless Sensor Networks (WSN). The algorithms RSA, ECC and MQQ were evaluated on ARM platform. We have used three criteria in our comparison: the processing time, memory and processor usage. We used the SimpleScalar tool for our simulations analysis. The MQQ algorithm achieved the best results in most of the evaluated criteria. Considering the same key sizes, the processing time for MQQ is at least 16 times smaller than the ECC and 230 times smaller than RSA. Regarding memory consumption, the MQQ had an occupation 61% lower than the RSA and 24% less than in the ECC. Besides these, other criteria such as misses on cache level 1, branches, replacements and write-backs were recorded in order to improve our assessment. Finally, we show the MQQ is a good algorithm for embedded systems since it is better than ECC e RSA.**

***Index Terms*—Embedded systems, Public-key cryptosystems, Performance Analysis and Design Aids.**

I. 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 National Institute of Standard and Technology (NIST), this algorithm is widely used in transactions on the Internet. The Elliptic Curve Cryptography (ECC) was created in 80s, and are based on the difficulty of solving the discrete logarithm problem on elliptic curves. Despite its complexity the algorithm based on elliptic have been extensively studied in academia. Recently, the public-key algorithm called 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 paper describes a theoretical study of cryptographic such as RSA, ECC and MQQ as well as the performance analysis of these algorithms in ARM embedded platforms used in wireless sensor networks. This paper is organized as follows. Section II gives some background about asymmetric algorithms RSA, ECC and MQQ. Section III discusses on the implementations and observation in the performance evaluation. Finally, some concluding remarks and planning for future works are outlined in Section IV.

II. ASSYMETRIC ALGORITHMS

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 [4] 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

Manuscript received January 23, 2013; revised March 13, 2013.

doi: 10.12720/ijeee.1.1.39-43

according to Sen [5] 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 [6] is already proven that public-key algorithms developed are suitable for hardware in WSNs.

A. *RSA Algorithm*

In the introductory paper about RSA, the authors Rivest and Shamir [7] 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:

$$m^{ed} \equiv m(mod\ n) \qquad (1)$$

The decryption works because $c^d \equiv (m^e)^d \equiv m(mod\ n)$. The safety lies in the difficulty of computing a clear text m from a cipher text $c = m^e$ mod n and the public parameters n (e).

B. *Elliptic Curve Cryptography (ECC)*

In the mid-80, Koblitz [8] and Miller [9] proposed a method of cryptography based on elliptic curves ECC. According to creators of the ECC, an elliptic curve is a plane curve defined in (2):

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

The efficiency of this algorithm is based on finding a discrete logarithm of a random element that is part of an elliptic curve. According to Blake [10] 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.

C. *Multivariate Quadratic Quasigroup (MQQ)*

In 2008, it was proposed a new scheme called multivariate quadratic public-key near group (MQQ) [11]. This algorithm is based on multivariate polynomial transformations of nearly quadratic and groups. A generic description for the scheme is a typical system MQQ multivariate quadratic, as in (3):

$$T \circ P' \circ S : \{0,1\}^n \rightarrow \{0,1\}^n \qquad (3)$$

where T and S are two nonsingular linear transformations and P' is a multivariate mapping bijective quadratic over $\{0, 1\}^n$.

According to Maia [11] and Ahlawat [12], 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 [2] and Maia [11], 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.

III. Architectural Evaluation

The embedded platform consists of StrongARM which is a 32-bit RISC processor using 206MHz, 16KB and 16KB of instruction cache using writeback strategy. The gcc compiler and the Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) were used in coding. The code was turned into ARM via compiler arm-linux-gcc. The encrypted files by algorithms were identical, respecting the equivalence between key sizes, which can be viewed in Table I.

TABLE I. Equivalence of Key size to RSA, ECC and MQQ [11] [13]

RSA	1024	2048	3072	7680	15360
ECC (Prime field)	192	224	256	384	521
MQQ 160	160	—	—	—	—

The key generation was not part of this study because it was considered expensive for embedded platforms. The codes were simulated by SimpleScalar tool, which returned information about simulation time in cycles, number of reads and writes, memory pages, use of cache memory, among others. The performance evaluation was separated into sections like processing time, processor usage and memory consumption.

A. *Processing Time*

Considering the frequency of StrongArm processor in 206MHz and having the number of cycles of each algorithm, it was possible to calculate the time in milliseconds (ms). Fig. 1 shows that in the ARM platform, the processing time of RSA was slower than ECC and MQQ for all variations of key.

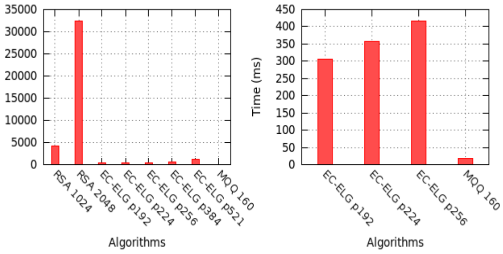

Figure 1. Processing time

The MQQ 160 had the best performance among the algorithms analyzed, since it had a time of 18.7 ms, against 305.8 ms of the ec-elg_p192 and 4302.8 ms of RSA 1024. These data show that MQQ in the ARM platform is 16 times faster than ec-elg_p192 and 230 times faster than RSA 1024.

Regarding the number of cycles per instruction (CPI), Fig. 2 shows that the algorithm based on curves showed the highest mean related to CPI values, followed by the RSA algorithm and finally, the algorithm MQQ. The

MQQ algorithm presents a good value of CPI, emphasizing that the algorithm is fast and has less processing time.

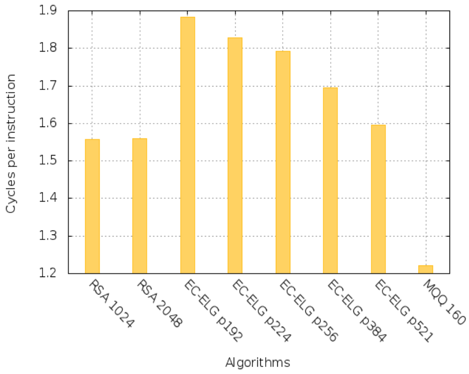

Figure 2. Cycles per instruction

B. *Evaluation of Processor*

The performance evaluation took into account criteria such as number of instructions, number of reads and writes, as well as the amount of branches, which are a code sequences that are conditionally executed according to a flow control. When the number of branches is high, the program presents a lower performance because tests require conditional processing time and increase the program code.

The Table II shows that the number of instructions executed by the MQQ-160 algorithm was 277 times smaller than the number of instructions executed by the RSA 1024 algorithm. Regarding ec-elg_p192, the number of instructions executed by MQQ 160 was 13 times smaller.

TABLE II. NUMBER OF INSTRUCTIONS [11] [13]

Algorithm	Instructions (bilion)
RSA 1024	1,110
RSA 2048	8,426
ec-elg p192	0,055
ec-elg p224	0,067
ec-elg p256	0,080
ec-elg p384	0,140
ec-elg p521	0,248
MQQ 160	0,004

The Fig. 3 shows that the algorithm RSA 2048 showed 0.22 billion of branches, against 0.03 billion from RSA 1024. The algorithms based on curves and the MQQ showed smaller number of branches. The ec-elg_p192 presented 0,004 billions of branches compared to 0,0002 billions presented by MQQ160.

The Fig. 3 shows that the MQQ 160 presented a number of branches 21 times lower than presented by ec-elg_p192 and 150 times smaller than presented by RSA 1024. This is another reason that explains why the MQQ presents better results when compared to RSA and ECC.

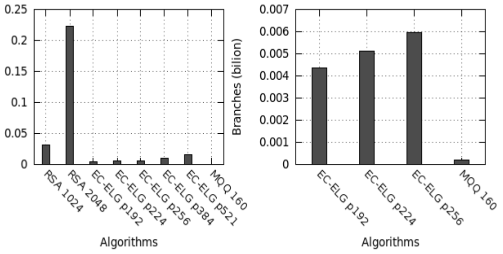

Figure 3. Number of branches

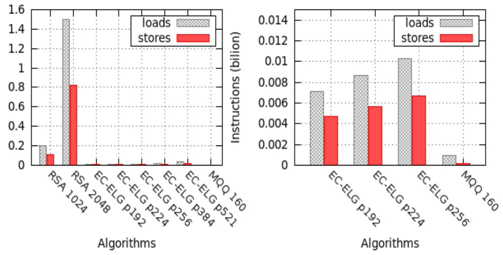

Figure 4. Number of loads and stores

The verification of the amount of reads and writes performed by each algorithm on ARM platform has not changed the scenario of the benchmark, since according to Fig. 4, the RSA algorithm performed more operations of this kind regardless of variations in key from our algorithms studied.

C. *Evaluation of Memory*

The evaluation of memory consumption aimed to study not only the amount of memory occupied by each algorithm, as well as specific issues of cache performance that can determine which algorithm is more suitable for WSN. Among these questions we have the replacements and writebacks. Both replacements as writebacks are detrimental to system performance. So, the RSA has a high quantity of these operations and the performance is lower when compared to ECC and MQQ algorithms.

In relation to the total size of memory pages allocated on the ARM platform, Fig. 5 illustrates that regardless of the key size, algorithms based on factorization of prime numbers (RSA) and algorithms based on elliptic curves (ECC) obtained similar results in their respective categories. The RSA-1024 algorithm allocated 708KB of memory pages, whereas RSA 2048 allocated 712KB of memory. Regarding the ECC, for all key sizes, it allocates 360KB of memory pages. The MQQ-160 algorithm occupied less memory then others, only 276KB of occupation. In this sense, the algorithm MQQ 160 occupied 76% of the memory occupied by the ec-elg_p192 and 39% of the memory occupied by the RSA 1024. Regarding the number of memory pages used by each algorithm, we note from Fig. 5 that RSA 1024 occupied 177 pages and RSA-2048 occupied 178 pages. All variations of ECC occupied 90 pages and MQQ-160 occupied 69 pages. These numbers shows that MQQ-160 is 61% more economical in memory consumption when compared to RSA 1024 and 23% more economical when compared to ec-elg_p192.

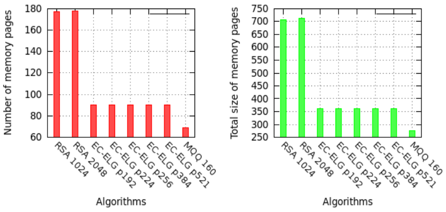

Figure 5. Memory pages

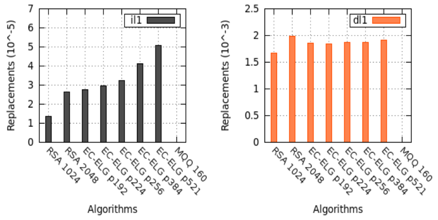

Figure 6. Number of replacements

The replacements, both in instruction and data level, were minimal. The Fig. 6 shows that only RSA and ECC presented computable data in this evaluation criterion. In instruction cache, ECC algorithm has an increased according the key size increases, moreover, the ECC always present levels higher than RSA. In the data cache, all algorithms analyzed showed an approximate number of substitutions, with the exception of MQQ, that in two caches analyzed obtained irrelevant values (near to zero replacements).

Again, this behavior, explains why MQQ has a good performance and usage of the processor. These characteristics make the MQQ is on the same level of ECC and present better performance.

Regarding to writebacks, it is possible to observe in Fig. 7, that RSA-1K and ECC algorithms maintained a range between 0.014 and 0.016. As expected, the RSA-2K algorithm obtained approximately 0.017 while MQQ algorithm does not show writebacks.

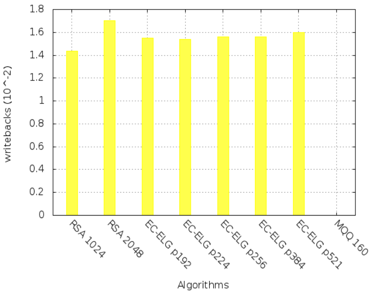

Figure 7. Number of writebacks

IV. Conclusion

This performance evaluation of asymmetric cryptographic algorithm for ARM architecture showed that the MQQ algorithm achieved better results when compared to RSA and ECC algorithms in all aspects analyzed.

Data from evaluation showed that in ARM platform, the algorithm MQQ-160 is faster, consumes less amount of memory and uses less the architectural resources of the processor when compared to RSA 1024 and ec-elg_p192. Thus, the results confirm that among the algorithms studied, the MQQ-160 is more suitable for embedded platforms with limited resources since it uses lower quantity of resources and gives a good performance. As future work, we plan to investigate the key generation performance of algorithms and add the algorithm HECC in our analysis.

References

[1] D. Gligoroski, S. Markovski, and S. J. Knapskog. "Multivariate Quadratic Trapdoor Functions Based on Multivariate Quadratic Quasigroups," in *Proc. American Conference on Applied Mathematics (MATH '08),* Cambridge, Massachusetts, USA, March 24-26, 2008., pp. 44-49.

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

[3] C. Margi, M. Simplicio, T. C. M. B. Carvalho, and P. S. M. L. Barreto, "Segurança em redes de sensores sem fio," in *Proc. Simp ósio Brasileiro em Seguran ça da Informa ção*, Belo Horizonte, 2009, pp. 149–194.

[4] D. Boyle and T. Newe, "Securing wireless sensor networks: security architectures," *Journal of Networks*, vol. 3, no. 1, pp. 65–77, Jan. 2008.

[5] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security*, vol. 1, no. 2, pp. 55–78, Aug 2009.

[6] R. Struik, "Cryptography for highly constrained networks," presented at the NIST-CETA Workshop. Gaithersburg, USA, Nov. 7-8, 2011.

[7] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[8] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[9] V. Miller, "Use of elliptic curves in cryptography," in *Proc. Advances in Cryptology - CRYPTO Proceedings.* Springer, 1986, pp. 417–426.

[10] I. Blake, G. Seroussi, and N. smart, *Elliptic Curves in Cryptography*, vol. 265, Cambridge University Press, 1999.

[11] R. Maia, "Análise da viabilidade da implementa ção de algoritmos p ós-qu ânticos baseados em quase-grupos multivariados quadr áticos em plataformas de processamento limitadas," M. S. thesis, Dept. of Computer, USP Univ., S ão Paulo, 2010.

[12] R. Ahlawat, K. Gupta, and S. Pal, "From mq to mqq cryptography: Weaknesses new solutions," in *Proc. Western European Workshop on Research in Cryptology*, Austria, Jul. 7-9, 2009.

[13] I. Branovic, R. Giorgi, and E. Martinelli, "A workload characterization of elliptic curve cryptography methods in embedded environments," *ACM SIGARCH Computer Architecture News*, vol. 32, no. 3, pp. 27–34, 2003.

Gustavo da Silva Quirino holds a degree in Computer Science from Federal University of Tocantins (UFT - 2003), an expert in Linux Network Administration from the Federal University of Lavras (UFLA - 2010). He is currently a professor at the Federal Institute of Bahia (IFBA) and a Masters student in Computer Science at the Federal University of Sergipe (UFS). Conducts research in embedded systems, sensor networks and security systems.

Edward David Moreno holds a degree in Electrical Engineering - Universidad Del Valle (1991), and Ph.D. degrees in Electrical Engineering from the University of São Paulo (1994 and 1998). The doctorate was Sandwich with Univ. Toronto, Canada (1996) and Chalmers University of Technology, Goteborg, Sweden (1997). He did postdoctoral studies at UFSCAR Univ. Federal de São Carlos (2000).

He is currently a Professor in DCOMP (Department of Computing) - UFS (Univ. Federal de Sergipe), in the city of Aracaju, Brazil. It Editoral Board member of 4 international journals: IJCSNS (Intl. Journal of Computer Science and Network Security), TCS Transactions on Computational Science Springer Verlag, the JUCS (Journal on Universal Computer Science), JCP (Journal of Computers). It also assessor courses in computing and informatics and Institutional INEP / MEC. He has participated as program committee, approx., 100 international events and published 6 books in the area of systems design and digital reconfigurable hardware security, and the 3 of them by Springer: Security in Computing, published by Springer verlag, 2009, Security in Computing - Part I, 2010 and Part II, 2010. Has experience in the area of Computer Science and Computer Engineering, with emphasis in Computer Systems Architecture, acting on the following subjects: Computer Architecture, Embedded Systems, Hardware security, Power-Aware Computing, High Performance Computing and Performance Evaluation.