Light Weight McEliece Cryptographic Algorithm Implementation on FPGA for Wireless Sensor Networks

Vijaylaxmi, Prof. Nafeesath

***Abstract*— This paper gives a technique for secure communication in the presence of third parties. The McEliece cryptosystem is an asymmetric encryption algorithm. It was the first such scheme to use randomization in the encryption process. The algorithm has never gained much acceptance in the cryptographic community, but is a candidate for "post-quantum cryptography", as it is immune to attacks using Shor's algorithm and more generally measuring coset states using Fourier sampling. More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation.**

***Index Terms*— McEliece, FPGA- Field programmable gate arrays**

I. Introduction

In cryptography, the McEliece cryptosystem is an asymmetric encryption algorithm developed by Robert McEliece. It was the first such scheme to use randomization in the encryption process. The algorithm has never gained much acceptance in the cryptographic community, but is a candidate for "post-quantum cryptography", as it is immune to attacks using Shor's algorithm and more generally measuring coset states using Fourier sampling.

The algorithm is based on the hardness of decoding a general linear code. For a description of the private key, an error-correcting code is selected for which an efficient decoding algorithm is known, and which is able to correct t errors. The original algorithm uses binary Goppa codes (subfield codes of geometric Goppa codes of a genus-0 curve over finite fields of characteristic 2); these codes are easy to decode, thanks to an efficient algorithm due to Patterson. The public key is derived from the private key by disguising the selected code as a general linear code. For this, the code's generator matrix G is perturbated by two randomly selected invertible matrices S and P. Variants of this cryptosystem exist, using different types of codes. Most of them were proven less secure; they were broken by structural decoding. McEliece with Goppa codes has resisted cryptanalysis so far. The most effective attacks known use information-set decoding algorithms. A 2008 paper describes both an attack and a fix. Another paper shows that for quantum computing, key sizes must be increased by a factor of four due to improvements in information set decoding.

The McEliece cryptosystem has some advantages over, for example, RSA. The encryption and decryption are faster (for comparative benchmarks see the eBATS benchmarking project at bench.cr.yp.to), and with the growth of the key size, the security grows much faster. For a long time, it was thought that McEliece could not be used to produce signatures. One exceptional case that used McEliece for encryption is the Freenet-like application Entropy.

A. Scheme definition

McEliece consists of three algorithms: a probabilistic key generation algorithm which produces a public and a private key, a probabilistic encryption algorithm, and a deterministic decryption algorithm. All users in a McEliece deployment share a set of common security parameters.

B. Key generation

Key generation is a complicated process involving algebraic methods and Goppa codes. As this is computationally intensive, it requires more hardware resources and consumes more power. Hence this is not suitable for implementing on a micro-controller which constitutes the wireless sensor networks. Moreover, this is a one-time process of generating a key. This key is generally an nxn matrix, theoretically. We choose to use a 3x3 matrix as working with this matrix is not too simple or too complex. Once a matrix is generated this way, another validation should be done as well, that is, checking it the matrix has a determinant. If it does not have the determinant, than generating the private key will not be possible.

C. Message encryption

- Form a 3x3 message matrix: M. If there are not enough characters spaces will be added.
- The matrix M is multiplied with the key matrix K: C = M.K
- A error matrix Z is added to C: D = C + Z
- This method is repeated for several times until the message text is completed.

Manuscript received April 07, 2015.

Vijaylaxmi, VLSI Design and Embedded Systems, Dept. of Electronics and Communication Engineering, P A College of Engineering, Mangalore

Prof. Nafeesath, Guide/Asst. Proffesor, Dept. of Electronics and Communication Engineering, P A College of Engineering, Mangalore

D. *Message decryption*

- Remove Z from the received message D
- Calculate the inverse of the public key (K) using the matrix reduction method
  - To find the inverse of matrix K, using Gauss-Jordan elimination, we must find a sequence of elementary row operations that reduces K to the identity and then perform the same operations on I_n to obtain K^{-1}.
- Multiply K^{-1} with D, to get the original message M

II. LITERATURE SURVEY

We referred base paper was published on 1978 by R. J. McEliece for a code-based encryption scheme suggested the use of binary Goppa codes,While other types of codes proven less secure. in the current implementation we are implementing the security grows much faster. For a long time

III. OBJECTIVES

With the break of RSA and ECC cryptosystems in an era of quantum computing, asymmetric code based cryptography is an established alternative that can be a potential replacement. A major drawback are large keys in the range between 50 kByte to several MByte that prevented realworld applications of code based cryptosystems so far. A recent proposal by Misoczki et al. showed that quasi-cyclic moderate density parity check (QCMDPC) codes can be used in McEliece encryption – reducing the public key to just 0.6 KByte to achieve a 80bit security level. Despite of reasonably small key sizes that could also enable small designs, previous work only report high performance implementations with high resource consumptions of more than 13,000 slices on a large Xilinx Virtex6 FPGA for a combined en/decryption unit.

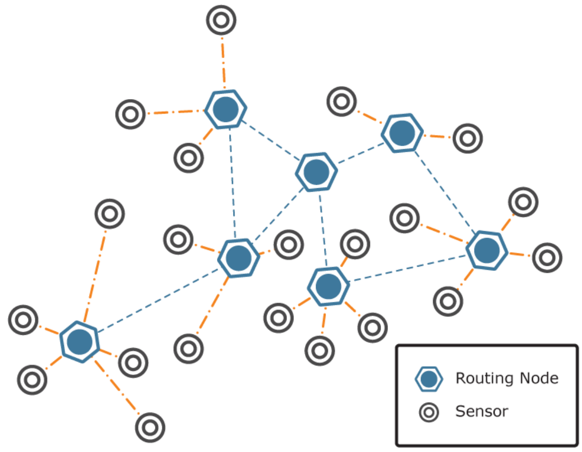
In this work we focus on lightweight implementations of code based cryptography and demonstrate that McEliece encryption using QCMDPC codes can be implemented with a Significantly smaller resource footprint still achieving reasonable performance sufficient for many applications, e.g., challenge response protocols or hybrid firmware encryption.

A comparison can be made after a literature survey of already implemented light weight cryptography systems such as HB-2.

IV. ORGANIZING AND OPTIMIZING THE SYSTEM

Practical implementation largely differs from theory of Mc-Eliese. As the key generation process is computationally intensive, in the context of WSN, it is moved to the server in the network. Key generation requires more resources and consumes more power. Generally, the nodes are power limited (they may be working on batteries or having limited power conditioning circuitry). Implementing individual key generation function in each node is thus not feasible. Developing key generation will not be the scope of this project a matrix with a determinant is assumed. In a typical sensor network, routing nodes generally possess enough computing power to implement key generation algorithm.

The encryption and decryption modules are implemented on the node and form a part of the application itself. Thus, it will be implemented using Embedded C APIs suitable for portability. The following figure shows the functionality.



Hardware Implementation

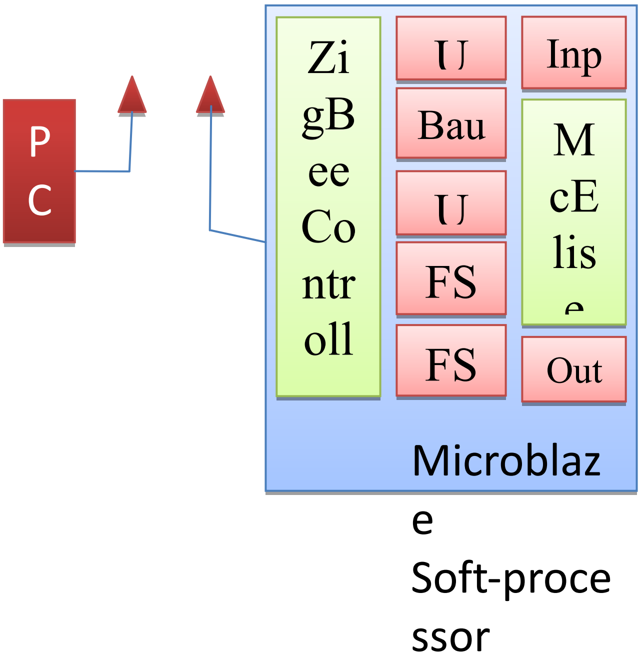


Fig 1: FPGA/Microblaze Implementation of McEliece Algorithm

a. *Methodology*

The project will be carried out in the following stages:

- Further literature survey has to carried out to extract more ideas
- A design plan should be created with an application
- RTL Design of the proposed design should made with the Microblaze soft-processor
- RTL verification should be done to make sure the design is working as decided
- First level FPGA implementation should be done. This will complete one full cycle.

- Next, speed, area, power and other parameters should be observed and tabulated.
- Comparison of the parameters should be done between regular and fault-tolerant models
- A demonstration should be and this can be done using the application decided
- Thesis/dissertation should be written.

V. APPLICATIONS

- Hybrid firmware encryption
- Wireless sensor networks
- Challenge response protocols
- Key-establishment protocols

A. Hardware& Software requirements

a. Software

- Xilinx Vivado Design Suite
- TMFT

b. Hardware

- Xilinx FPGA Kit
- PC
- USB-Serial cables
- ZigBee Modules
- Other application related hardware

VI. CONCLUSION

In this paper we presented lightweight implementations of the asymmetric cryptosystem McEliece with QC-MDPC codes. In addition to considerably reducing the resource requirements by using embedded block memories that are offered in Xilinx FPGAs, we achieved reasonableb performance for both encryption and decryption. Furthermore, the cryptosystem reduces the key sizes to a level that is much more appropriate for real-world usage.

REFERENCES

[1] P. W. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms On a Quantum Computer,” SIAM J. Comput, 1997.

[2] K. Chang, “I.B.M. Researchers Inch Toward Quantum Computer,” New York Times Article, February 28, 2012, http://www.nytimes.com/ .

[3] R. J. McEliece, “A Public-Key Cryptosystem Based On Algebraic Coding Theory,” Deep Space Network Progress Report, Jan. 1978.

[4] H. Niederreiter, “Knapsack-type cryptosystems and algebraic coding theory,” Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform., 1986.

[5] T. Eisenbarth, T. G¨uneysu, S. Heyse, and C. Paar, “MicroEliece: McEliece for Embedded Devices,” in CHES, ser. Lecture Notes in Computer Science, C. Clavier and K. Gaj, Eds. Springer, 2009.

[6] S. Ghosh, J. Delvaux, L. Uhsadel, and I. Verbauwhede, “A Speed Area Optimized Embedded Co-processor for McEliece Cryptosystem,” in Application-Specific Systems, Architectures and Processors (ASAP), 2012 IEEE 23rd International Conference on, july 2012.

[7] S. Heyse, “Implementation of McEliece Based on Quasi-dyadic Goppa Codes for Embedded Devices,” in Post-Quantum Cryptography, ser. Lecture Notes in Computer Science, B.-Y. Yang, Ed. Springer Berlin / Heidelberg, 2011.

[8] E. Persichetti, “Compact McEliece Keys based on Quasi-Dyadic Srivastava Codes,” J. Mathematical Cryptology, 2012.

[9] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. \ Barreto, “MDPCMcEliece: New McEliece Variants from Moderate Density Parity-Check Codes,” IEEE International Symposium on Information Theory, vol. 2013, 2013.

[10] S. Heyse, I. von Maurich, and T. G¨uneysu, “Smaller Keys for Code-Based Cryptography: QC-MDPC McEliece Implementations on Embedded Devices,” in CHES, ser. Lecture Notes in Computer Science, G. Bertoni and J.-S. Coron, Eds. Springer, 2013.

[11] K. Kobara and H. Imai, “Semantically Secure McEliece Public-Key Cryptosystems-Conversions for McEliece,” in Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography, ser. PKC ’01. London, UK: Springer- Verlag, 2001.

[12] R. Nojima, H. Imai, K. Kobara, and K. Morozov, “Semantic security for the McEliece cryptosystem without random oracles,” Des. Codes Cryptography, 2008.