Literature Review: Elliptic Curve Cryptography on Field Programmable Gate Array

By Katie Groves, Electrical and Computer Engineering Department

Abstract

When it comes to encryption, low propagation error and high throughput with minimum resources are the goals to have for any cryptosystem. One type of encryption that fits these properties is the elliptic curve cryptography (ECC). While ECC uses little resources and is reliable for securely encrypt data, implementation onto a field programmable gate array (FPGA) chip may enhance the encryption throughput. The purpose of this literature review is to research different ways ECC is being used and accelerated via its basic point multiplication task. A more in-depth analysis on the advantages of having ECC on FPGA will be explored. Having a comparison of efficiency and resources will be taken into account during this review. Also, noting any limitations that may exist having ECC on a FPGA chip and how some researchers have developed answers to these problems.

Introduction

- Encryption is used for practically everything we do online, i.e. online banking, cloud storage, sending emails, etc.
- All encryption levels and types are determined by the National Institute of Science and Technology (NIST) as guidelines for companies and government branches.
- NIST has included a new type of encryption known as elliptic curve cryptography (ECC) invented by Miller and Kolblitz in 1985.
- The key size, or the designated size for a randomly generated number for encryption and decryption, for ECC designated by NIST can start at 161-bits which is the *fraction of the size* of another widely use encryption method RSA.
- ECC with a smaller key size has the *same level security* as RSA.
- To further increase the level of efficiency ECC, implementation onto a Field Programmable Gate Array (FPGA) chip *reduces the resources* needed to run ECC
- Focus of this review is the analysis on the advantages of ECC on an FPGA and different implementation methods.
- Reviewing advantages of improving ECC implementations

Coverage

The scope of this literature review is observing the cryptosystem ECC since this is a relatively new encryption method that is being researched. Implementation of ECC onto an FPGA chip is the focus of this literature review so this narrows down the research papers that analyze the method of ECC and how to accelerate the encryption process of ECC with the use of FPGA chips. Of course, the interest of this type of implementation needs to only observe current work within the past 3 to 4 years to stay up to date of the fastest, most efficient methods possible. These papers must include results of their method of implementation to compare the results.

Methodology

To compare each of these papers, this literature review will take use of a qualitative content analysis of three different papers. Each of the papers will have their own methods of encryption compared to the standard ECC encryption method, so defining *each paper's method* will be analyzed. Next, each paper used different FPGA chips so the *hardware* will be taken into account. Each chip runs at different *frequencies* so this must be taken into consideration. Following each chip, a type of measurement known as *slices* is used in the determination of how much resources the implementation will take. A slice in an FPGA chip is individual circuit and each implementation will use so many slices to perform its process. Finally, the comparison of the *time* it takes to complete the encryption implementation is compared to see which implementation has the quickest execution times and for what criteria makes the process fast.

Results

After reviewing three different articles, three different solutions were found to accelerate the encryption process of ECC:

The first article [1] improved their method by optimized multiplication by using Comba algorithm which is a bit-by-bit multiplication that is more efficient. Multiplication and add/subtraction run parallel to each other. All outputs are stored in RAM so multiplication and add/subtract don't wait on each other's output. RAM is large enough to store any size of ECC to make the system scalable.

The second article [2] also accelerated their method by improving multiplication by using Karatsuba multiplier and modified it to pipelined multiplier that requires lower clock frequencies. Values to be multiplied are segmented into *n* different chunks. After multiplication, the segments are concatenated back together. The segmentation speeds up the process and requires more resources.

Lastly, the third article [3] once again improves their process by accelerating the multiplication by using iterative interleaved addition chain technique which uses shifts and modulo functions. It also uses the architecture radix-4, a method of parallel multiplication. Pre-computations for the modulus operation to allow four multiplication to execute in parallel in the radix for architecture. Therefore, only one addition/subtraction needed for the process.

In **Table 1**, the results of each of these methods are tabulated.

Significance

ECC as it stands is efficient on its own. It offers the same level of security as another encryption scheme RSA but the key size needed and recourses needed to execute ECC are a faction less than RSA. To further increase the efficiency of ECC, it is suggested to implement the algorithm onto a FPGA chip as this literature review shows. The FPGA chip is portable, small, and cost efficient. With these properties, a FPGA could be considered "plug-n-play". There are many inputs and outputs on the board that the chip is integrated with, so this FPGA board can work with different systems.

Another property that makes integrating ECC on the FPGA board desirable is the algorithm on the board can run in parallel to itself, meaning, multiple operations for the process of ECC can run at the same time. Shown in some of the methods of other research papers, the use of parallelism in FPGA was taken advantage of to further accelerate the execution of ECC.

Work	Device	Bits	Slices (LUTs)	Time (ms)	Frequency (MHz)
[1]	Virtex-5	192	1,980	1.709	251.3
		224		2.652	
		256		3.951	
		384		11.81	
		521		28.04	
[1]	Virtex-4	192	7,020	2.361	182.0
		224		3.663	
		256		5.457	
		384		16.31	
		521		38.73	
[2]	Virtex-7	163(3x14)*	1476	.0105	397
		233(4x14+3)*	2647	.0160	370
		283(5x14+1)*	3728	.0210	345
		409(7x14+5)*	6888	.0327	316
		571(10x14+1)*	12965	.0576	350
[3]	Virtex-6	192	24.8k	.77	154
		224	28.9k	1.07	149
		256	32.4k	1.43	144
[3]	Virtex-4	192	28.1k	1.49	79
		224	31.6k	2.1	74
		256	35.7k	2.96	70

Table 1

*These values represent the segment size that were used in the segmentation multiplication

Conclusion

In conclusion, it is easy to determine that increasing the efficiency of the multiplication process will increase the speed of ECC. Each of these three different papers based their methods on making multiplication more efficient in different ways. The FPGA chip's ability to run processes in parallel was taken advantage of to be able to run different multiplication processes at the same time. As seen in Table 1, the segmentation method [2] worked the fastest out of all the other methods. Though it has a higher frequency than any of the other methods, which may be due to the type of board ECC is implemented on, the utilization it takes is either same or less than the other two. However, the fact that these implementation methods were all processed on a FPGA board and can run the process in milliseconds, shows that implementation of ECC on an FPGA board can accelerate ECC encryption.

References

[1] K. C. C. Loi and S. B. Ko, "Scalable Elliptic Curve Cryptosystem FPGA Processor for NIST Prime Curves," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 11, pp. 2753-2756, Nov. 2015.

[2] Z. U. A. Khan and M. Benaissa, "Throughput/Area-efficient ECC Processor Using Montgomery Point Multiplication on FPGA," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 62, no. 11, pp. 1078-1082, Nov. 2015.

[3] Javeed, K., and Wang, X. (2017) Low latency flexible FPGA implementation of point multiplication on elliptic curves over GF(p). Int. J. Circ. Theor. Appl., 45: 214–228.

Acknowledgements: Dr. Elkeelany and the Electrical and Computer Engineering Department
Contact: kmgroves42@students.tntech.edu