IOP Conf. Series: Materials Science and Engineering **199** (2017) 012049 doi:10.1088/1757-899X/199/1/012049

DNA-Cryptography-Based Obfuscated Systolic Finite Field Multiplier for Secure Cryptosystem in Smart Grid

Shaobo Chen, Pingxiuqi Chen, Qiliang Shao, Shaik Nazeem Basha and Jiafeng Xie

Department of Electrical Engineering, Wright State University, Dayton, OH 45435

Corresponding author: Jiafeng Xie, jiafeng.xie@wright.edu

Abstract. The elliptic curve cryptography (ECC) provides much stronger security per bits compared to the traditional cryptosystem, and hence it is an ideal role in secure communication in smart grid. On the other side, secure implementation of finite field multiplication over $GF(2^m)$ is considered as the bottle neck of ECC. In this paper, we present a novel obfuscation strategy for secure implementation of systolic field multiplier for ECC in smart grid. First, for the first time, we propose a novel obfuscation technique to derive a novel obfuscated systolic finite field multiplier for ECC implementation. Then, we employ the DNA cryptography coding strategy to obfuscate the field multiplier further. Finally, we obtain the area-time-power complexity of the proposed field multiplier to confirm the efficiency of the proposed design. The proposed design is highly obfuscated with low overhead, suitable for secure cryptosystem in smart grid.

1. Introduction

Secure communication in smart grid is gaining substantial attention recently [1-2], as shown by one example of Figure 1. The elliptic curve cryptography (ECC) gives much stronger security per bits compared to the RSA, and therefore it can be a suitable candidate to secure communication in smart grid [3-4]. For ECC, one has to perform point addition operations, which consist of addition, squaring, and multiplications in binary field. The field addition and squaring are fast operations, while field multiplication involves much more complexity, i.e., larger area occupation, longer computation time, and higher power consumption. As National Institute of Standards and Technology (NIST) has suggested 5 irreducible polynomials for ECC implementation [4-5], a lot of efforts have been made on efficient realization of the field multiplication over $GF(2^m)$ based on the recommended polynomials.

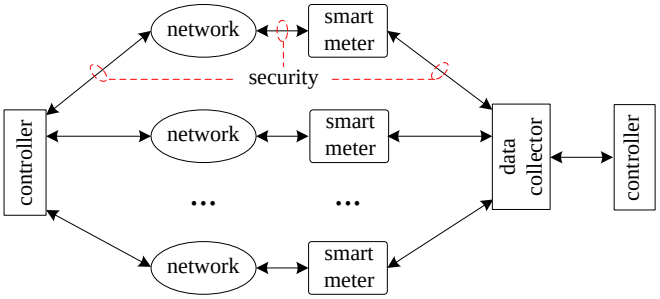

Figure 1. Security issues related to a typical data communication system within smart grid.

In terms of the design style, there are basically two types of multipliers over $GF(2^m)$: non-systolic and systolic [5]. The non-systolic designs usually have low area-complexity at the cost of long computation time. The systolic designs usually provide high-throughput computation since each processing element (PE) in the structures performs only a small part of the whole computation and passes the results to its neighboring PE for pipelined processing [5].

Efficient implementation of systolization of finite field multiplication over $GF(2^m)$ has been explored a lot in the past a few years, as well as efficient realization of ECC [5]. On the other side, challenges toward security related to the ECC have been risen up as technology advances, e.g., advanced tools are developed to attack the ECC through side-channel attacks or brute-force attacks [5-7]. There are a number of reports have been released on protecting the operation of ECC [5-7].

In this paper, we employ a novel obfuscation method to enhance the security level of ECC for smart grid. For the first time, we have proposed a novel "all-one-polynomial (AOP)-based obfuscation" method to obfuscate the functionality of the systolic finite field multiplier. Then, a DNA cryptography strategy is applied to encode the activation key. We have also tested and confirmed the efficiency of the proposed design, especially on the complexity and obfuscation performance.

The rest of the paper is organized as follows: Section 2 gives some introduction of the proposed techniques, while Section 3 focuses on the designing of obfuscated systolic multiplier. Comparison of area-time-power complexities is shown in Section 4. And conclusion is given in Section 5.

2. Preliminaries

In this section, we give here a brief introduction of hardware obfuscation technique, DNA cryptography, and the systolic finite field multiplier based on NIST recommended trinomials.

2.1. Hardware obfuscation

Hardware obfuscation [6-7] is a technique to conceal a certain hardware circuit's functionality and structurality to protect the attacks from reverse engineer. Some hardware obfuscation methods are based on encrypting the source hardware description language (HDL) codes, while some are focusing on designing obfuscated finite state machine (FSM) to obfuscate the circuits. Recently, a high-level transformation-based obfuscation technique is reported to protect the digital signal processing circuits.

On the other side, however, there is still no report available in literature on obfuscating the systolic-array-based designs, especially on the systolic finite field multiplier. In this paper, for the first time, we propose an efficient obfuscation technique to equip the systolic finite field multiplier with obfuscation ability to increase the difficult for reverse engineer.

2.2. DNA cryptography

DNA cryptography is a new cryptographic system has emerged recently from the research of DNA computing [8-9]. It is initially based on biological problems: a DNA computer have a function which current computers does not. First, DNA chains have a high level of parallelism, and its calculating speed can upto 1 billion times per second. Second, the DNA molecule has a huge capacity to contain data: one trillion bits of binary data only occupy one cubic decimeter of a DNA solution. Last, a DNA-based computer is efficient in power consumption, as low as one-billionth of a current computer [8-9].

Comparing with current data encoding system where everything is encoded by 0 and 1, DNA encoding scheme has four basic units: 1. Adenine (A); 2. Thymine (T); 3. Cytosine (C); 4. Guanine (G). Consequently, we can use another way to represent these four states as: 1. A(0)-00; 2. T(1)-01; C(2)-10; G(3)-11. Thus, there will in total 4!=24 possible encoding ways. A number of efficient encoding strategies have been reported to improve the encoding safety and speed, yet the whole area is still in an early stage [8-9].

2.3. Systolic finite field multiplier

A lot of efforts have been made on efficient implementation of finite field multipliers over $GF(2^m)$ based on NIST polynomials, especially based on trinomials [5]. In this paper, we propose an efficient

technique to obfuscate a newly reported trinomial-based low register-complexity systolic multiplier based on trinomials. The existing algorithms are given below [5]:

Let f(x) be an irreducible polynomial over $GF(2)$ as

$$f(x) = x^m + f_{m-1}x^{m-1} + \cdots + f_1 x + 1, \tag{1}$$

where $f_j \in GF(2) = \{0, 1\}$. $\{1, x, \ldots, x^{m-1}\}$ is the polynomial basis in $GF(2^m)$, such that we can have the Montgomery multiplication algorithm as [5]

$$C = A \cdot B \cdot r^{-1} \bmod f(x), \tag{2a}$$

where

$$\begin{aligned} A &= a_{m-1}x^{m-1} + \cdots + a_1 x + a_0 \\ B &= b_{m-1}x^{m-1} + \cdots + b_1 x + b_0 \\ C &= c_{m-1}x^{m-1} + \cdots + c_1 x + c_0, \end{aligned} \tag{2b}$$

and a_j b_j, $c_j \in GF(2)$, for $j = 0, 1, \ldots, m\text{-}1$. r is Montogomery factor that satisfies $\gcd(r, f(x)) = 1$, where gcd means the greatest common divisor. In [2], $r = x^t = x^{(m-1)/2}$ as the Montgomery factor, and then (2) can be expressed as

$$C = \sum_{i=0}^{m-1} b_i (A \cdot x^i \cdot x^{-t} \bmod f(x)) = C_1 + C_2, \tag{3a}$$

where

$$\sum_{i=0}^{t-1} b_i \cdot A \cdot x^{i-t} \bmod f(x) = C_1 \tag{3b}$$

$$\sum_{i=t}^{m-1} b_i \cdot A \cdot x^{i-t} \bmod f(x) = C_2. \tag{3c}$$

The detailed algorithm can then be seen in Algorithm 1 of [5].

3. Proposed Obfuscated Systolic Finite Field Multiplier

In this section, we first present the proposed obfuscated systolic field multiplier, and then based on the DNA-cryptography-based coding method, we obfuscate the proposed design further.

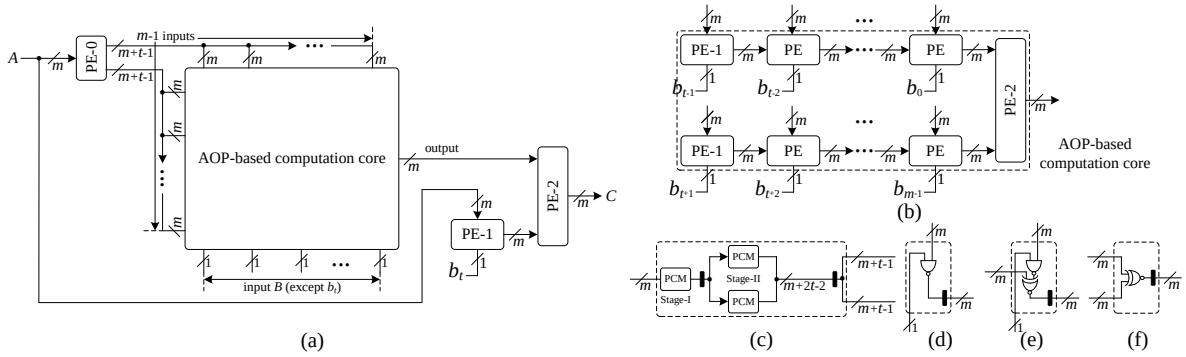

Figure 2. Low register-complexity systolic multiplier based on the AOP-based computation core, where the black box denotes the registers [5]. (a) Systolic structure. (b) Internal structure of the AOP-based computation core. (c) Detailed design of PE-0. (d) Detailed design of PE-1. (e) Detailed design of regular PE. (f) Detailed design of PE-2.

3.1. Proposed obfuscated systolic finite field multiplier: AOP-based obfuscation technique

The existing systolic finite field multiplier can be seen in Figure 2, where it consists of one AOP-based computation core and three extra PEs. PE-1 and PE-2 only involve simple calculation, while PE-0 is

complicated in calculation, as indicated in [5]. Here, we propose a novel obfuscation technique to conceal its functionality, as presented follows:

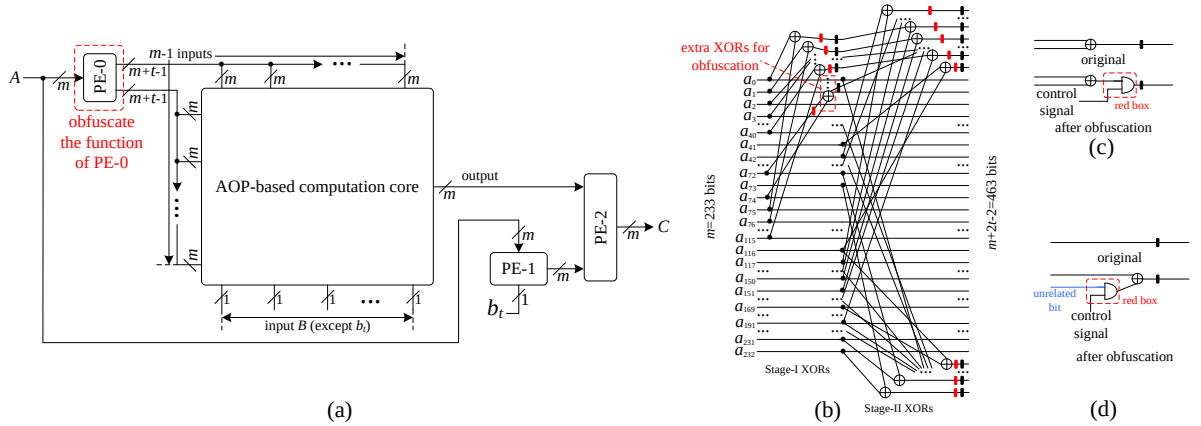

Figure 3. Proposed obfuscated systolic multiplier based on the AOP-based computation core, where the black box denotes the registers and the red box denotes the obfuscated element. (a) Obfuscated systolic structure. (b) Detailed obfuscation design of PE-0. (c) Detailed obfuscation design-I based on the red box. (d) Detailed obfuscation design-II based on the red box.

For systolic finite field multiplier, two main problems should be solved for hardware obfuscation: (a) The identical internal structure of nearly all PEs. (b) The real functionality of the multiplier. To solve these two problems successfully, we propose here a so-called "AOP-based obfuscation" technique. It is noted that AOP-based multiplier is unpreferable for ECC implementation due to security reasons. If we could hide the key operation of this multiplier while leaving the main part of AOP-based computation core there, the potential attacks will find that it is difficult to obtain the real functionality of this systolic multiplier. As PE-0 involves main pre-computation for trinomial-based multiplier, we here mainly focus on obfuscating the internal structure of PE-0, as indicated by the strategy in Figure 3(a), where the PE-0 is functionally obfuscated to achieve overall obfuscation of the systolic multiplier. The detailed obfuscation design of PE-0 is shown in Figure 3(b), where we use some extra XOR gates and red boxes to obfuscate the functionality. There are basically two types of obfuscation boxes, as shown in Figure 3(c) and (d), respectively. As shown in Figure 3(c), the original design is just an XOR operation of two bits followed by a bit-register, while in the obfuscated circuit we purposely add an extra AND gate such that if the control signal is "1", the circuit still works correctly and otherwise not. In Figure 3(d), the original circuit is just the direct connection between input and output (the bit-register can be removed to save the register-complexity). To add the obfuscation ability, we use an unrelated bit and one extra XOR as well as the red box to achieve circuit obfuscation. The circuit in Figure 3(d) works correctly if the control signal is "0" such that the output of the AND gate will be "0" and the original input does not have any change after the XOR operation with "0". The two types of obfuscation circuit can be employed randomly in any connection bit inside of the PE-0 to increase the overall functionality. Note that the employment of these two obfuscation circuits does not increase the critical-path of the systolic structure.

In practical implementation, one can always put as many obfuscation units as possible in PE-0 to increase the obfuscation level, e.g., if we add 120 extra obfuscation units into PE-0, we need a control signal of 120-bits to activate the real function of the systolic multiplier (if an attack tries to attack this multiplier, ideally there should be 2^{120} times of trial, which could be nearly impossible to complete that based on current technology). Even if the attackers are able to separate PE-0 from AOP-based computation core, it is still meaningless for the attackers since the AOP-based computation core is unsafe for ECC implementation, and thus the attackers' obtained information still does not match the real function. Note that we can define here the number of red box as **obfuscation level**.

3.2. Employing DNA cryptography method for further obfuscation

As mentioned in Subsection 2.2, we can use a way to represent the four states as: 1. A(0)-00; 2. T(1)-01; C(2)-10; G(3)-11 (there will in total 4!=24 possible encoding ways). These four states can be used to segment a bunch of binary signals into several groups, which facilitates the encoding process. For example, for a binary number "11100100011101", we can use "GCATTGT" to represent the original number. Of course, if we can design an obfuscated control unit where these four stages of letter can come out in an obfuscated way to activate the systolic multiplier, then the security level of the multiplier will be increased.

Here we have employed the strategy proposed in [8-9], where the DNA encoding process can be done through a number of XOR operations (see detailed algorithm in [8-9]). The whole structure is shown in Figure 4, where the activation key is encoded through DNA cryptography method, and then the encoded signal is used to activate the obfuscated systolic multiplier. Due to the protection from DNA cryptography method, it will be hard for the attacker to get the real key to activate the multiplier. As the DNA cryptography encoding is done mainly by XOR operations, we here just combine these XORs into PE-0, not only to increase the obfuscation ability of the multiplier, but also to save some resources as some XORs in PE-0 can be shared between each other.

4. Area-Time-Power Complexities

The area and time complexities (mainly number of logic gate count, register count, critical-path, and latency) of the proposed obfuscated systolic multiplier and the existing one of [5] are listed in Table 1.

Table 1. Area-time complexities of the existing systolic multiplier and the proposed obfuscated one.

design	AND	XOR	register	critical-path	latency
[5][1]	m^2	m^2-1	m^2+3m-1	T_A+T_X	$(m+7)/2$
proposed[2]	m^2+n	$\approx m^2+n+l$	$\approx m^2+3m-1+n$	T_A+T_X	$\approx(m+7)/2$

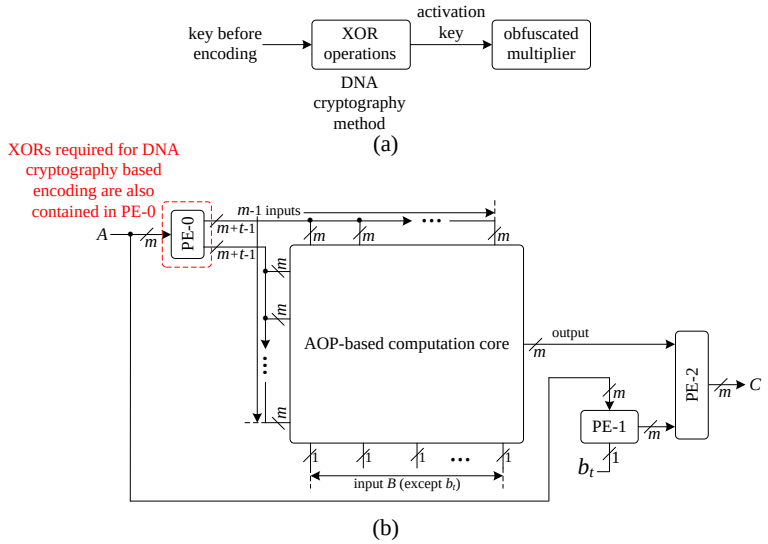

Figure 4. Proposed obfuscated systolic multiplier after employing DNA cryptography method. (a) Overall structural diagram. (b) Detailed obfuscation block diagram of the proposed obfuscated multiplier.

T_A: The delay time of an AND gate. T_X: The delay time of an XOR gate.

[1]: For simplicity of discussion, we here list only the AND and XOR gates, though we can use NAND and XNOR gates.

[2]: n refers to the red box added within PE-0. l refers to the number of XORs used for DNA cryptography-based encoding method.

As shown in Table 1, the proposed multiplier does not involve too much complexity overhead when compared with the one in [5], e.g., for a trinomial recommended by NIST $f(x)=x^{233}+x^{74}+1$, if we

select the obfuscation level n=120, the area-overhead will only roughly be ≈0.2% (including those for DNA cryptography encoding method), which is quite low compared with the key size of 2^{120} bits.

To confirm the efficiency of the proposed obfuscated systolic multiplier, we have also presented the FPGA synthesis results of the proposed design (obfuscation level = 120) and the existing one in Table 2 (based on the result of a Virtex 6 FPGA chip from Xilinx ISE 14.1).

Table 2. FPGA implementation results

design	area	delay time	power	ADP	PDP
[5]	54,032	128.2	2.336	6,926,902	299.48
proposed	54,326	129.3	2.378	7,024,352	307.48

Unit for Area: number of slice register; Unit for delay: ns; Unit for power: W (power is estimated at 100MHz).
Delay time = latency
ADP: Area-delay product = Area×Delay. PDP: Power-delay product = Power×Delay.

It can be seen that for an obfuscation level of 120 of the proposed systolic multiplier, it involves only 1.4% and 2.7% overhead of ADP and PDP, respectively, when compared with the existing one of [5]. The proposed design can be extended further for practical implementation of ECC in smart grid environment.

5. Conclusion

A novel obfuscated systolic multiplier for ECC in smart grid environment is presented here. We first employ a novel "AOP-based obfuscation" technique to a systolic multiplier that the functionality of the systolic multiplier is easily obfuscated. Then, we apply the DNA cryptography encoding method to obfuscate the activate key further. Through both theoretical and simulation comparison, the proposed obfuscated systolic multiplier is found to have low complexity overhead compared with existing one, yet with high level of obfuscation. To the authors' knowledge, this is the first design available in the literature on obfuscated field multiplier. Because of its efficiency in both complexity and obfuscation performance, the proposed multiplier can be extended for ECC in smart grid environment.

References

[1] Khurana H, Hadley M, Lu N, and Frincke D 2010 smart-grid security issues *IEEE Security & Privacy* 8(1) pp 81-85
[2] McDaniel P and McLaughlin S 2009 security and privacy chalenges in the smart grid *IEEE Security & Privacy* 7(3) pp 75-77
[3] Blake I, Seroussi G, and Smart N 1999 Elliptic Curves in Cryptography *London Mathematical Society Lecture Note Series* (Cambridge, U.K.: Cambridge Univ. Press) pp 5-10
[4] National Institute of Standards and Technology 2000 FIPS 186-2, Digital Signature Standard (DSS) *Federal Information Processing Standards Publication*
[5] Chen P, Basha S, Kermani M, Azarderakhsh R, and Xie J 2016 FPGA realization of low register systolic all-one-polynomial multipliers over $GF(2^m)$ and their applications in trinomial multipliers *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, pp 99-109
[6] Koushanfar F and Alkabani Y 2010 provably secure obfuscation of diverse watermarks for sequential circuits *Proc. Int. Symp. Hardw.-Oriented Security Trust* pp 42–47
[7] Lao Y and Parhi K K 2015 obfuscating DSP circuits via high-level tansformations *IEEE Transactions on VLSI Systems* **23**(5) pp 819-830
[8] Rahman N, Balamurugan C, and Mariappan R 2015 a novel DNA computing based encryption and decryption algorithm *International Conference on Information and Communication Technologies* pp 463-475
[9] Zhang Y, He L, and Fu B 2012 research on DNA cryptography *Applied Cryptography and Network Security* pp 357-376