ISBN 978-981-11-7861-0
Proceedings of 2018 the 8th International Workshop on Computer Science and Engineering (WCSE 2018)
Bangkok, 28-30 June, 2018, pp. 526-530
doi: 10.18178/wcse.2018.06.090

Concurrent Error Detection Scheme for Montgomery Multiplication Over $GF(2^m)$

Kee-Won Kim [1 +], Hyun-Ho Lee [2], and Seung-Hoon Kim [1]

[1] Department of Applied Computer Engineering, Dankook University, Yongin, Republic of Korea
[2] Department of Computer Science, Dankook University, Yongin, Republic of Korea

Abstract. Finite fields over GF(2m) have great interest for their applications like cryptography, where it is important to detect an error. Multiplication is one of the most crucial operations and the concurrent error detection scheme for multiplication over finite fields is very useful to increase the reliability in such application. In this paper, we propose a concurrent error detection scheme to be efficiently employed for the Montgomery multiplication over GF(2m). Our scheme uses two Montgomery factors for deriving an efficient concurrent error detection. We expect that the multiplier with concurrent error detection using our scheme can save about 50% time complexity as compared to the existing structures. In future research, we will implement the detailed architecture to compute Montgomery multiplications with concurrent error detection.

Keywords: cryptography, Montgomery multiplication, finite field arithmetic, fault-tolerant computing, concurrent error detection

1. Introduction

Finite fields $GF(2^m)$ have several applications in such areas of communications as error-correcting codes [1] and cryptography [2]. Among the basic arithmetic operations over $GF(2^m)$, multiplication is the most important, complex, and time consuming.

Since the reliability of computation over $GF(2^m)$ against fault-based cryptanalysis [3,4] is an important issue, an efficient multiplier with a concurrent error detection capability is required. Various the finite field multipliers with or without concurrent error detection capability have received the most attention in the literature [5-9].

Chiou et al. [6] proposed the semi-systolic array implementation of the Montgomery multiplication used a time-redundancy-based error detection approach. Their approach has used recomputing with shifted operands (RESO) method and alternate data retry. Also, Hariri and Reyhani-Masoleh [7] proposed the improved time-redundancy-based approach for the semi-systolic array implementation, as well as a single-bit parity-based technique for concurrent error detection in the bit-serial Montgomery multiplication. Hariri and Reyhani-Masoleh [8] proposed the concurrent error detection scheme for three different Montgomery multipliers, namely the bit-serial, digit-serial, and bit-parallel multipliers and implemented for each of them. Recently, Kim and Jeon [9] proposed the efficient Montgomery multiplier over $GF(2^m)$ with about half latency as compared to related multipliers.

In this paper, we propose a concurrent error detection scheme using the time redundancy. Our concurrent error detection scheme uses two Montgomery factors for deriving an efficient architecture and can be efficiently adopted to Kim-Jeon's multiplier [9]. Since Kim-Jeon's multiplier has half latency as compared to related multipliers, we expect that the multiplier with concurrent error detection using our scheme can save about 50% time complexity as compared to the existing structures.

+ Corresponding author. Tel.: +82-31-8005-3689; fax: +82-31-8021-7422.
E-mail address: nirkim@dankook.ac.kr.

The rest of the paper is organized as follows. In Section 2, we provide a brief background which includes the Montgomery multiplication over GF(2^m) and RESO. In Section 3, we propose a concurrent error detection scheme for the Montgomery multiplier over $GF(2^m)$. Finally, we conclude the paper in Section 4.

2. Preliminaries

This section briefly reviews the Montgomery multiplication over $GF(2^m)$ and RESO.

2.1. Montgomery multiplier over finite fields

$GF(2^m)$ is a kind of finite field that contains 2^m different elements. This finite field is an extension of $GF(2)$ and any $\alpha \in GF(2^m)$ can be represented as a polynomial of degree $m-1$, such as $\alpha = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_{m-2}x^{m-2} + \alpha_{m-1}x^{m-1}$, where $\alpha_j \in \{0,1\}$, for $0 \le j \le m-1$. Let x be a root of the polynomial, then the irreducible polynomial P is represented as $P = p_0 + p_1 x + p_2 x^2 + \cdots + p_{m-2}x^{m-2} + p_{m-1}x^{m-1} + x^m$, where $p_j \in \{0,1\}$, for $0 \le j \le m-1$.

The Montgomery multiplication algorithm is an efficient method for computing modular multiplications and squarings required for exponentiation [10]. A binary Montgomery multiplication algorithm over the bit-level is introduced by Koc et al. [5].

Let α and β be two elements of $GF(2^m)$, then we define $\delta = \alpha \cdot \beta \bmod P$. Also, let A and B be two Montgomery residues, then they are defined as

$$A = \alpha \cdot r \bmod P = a_0 + a_1 x + a_2 x^2 + \cdots + a_{m-2}x^{m-2} + a_{m-1}x^{m-1}, \quad (1)$$

$$B = \beta \cdot r \bmod P = b_0 + b_1 x + b_2 x^2 + \cdots + b_{m-2}x^{m-2} + b_{m-1}x^{m-1}, \quad (2)$$

where a Montgomery factor r and an irreducible polynomial P are relatively prime, and $\gcd(r, P) = 1$. Then, the Montgomery multiplication algorithm [9] over $GF(2^m)$ can be formulated as

$$C = A \cdot B \cdot r^{-1} \bmod P. \quad (3)$$

Then, (3) can be expressed as the following by the definition of the Montgomery residue as shown in (1) and (2)

$$C = (\alpha \cdot r) \cdot (\beta \cdot r) \cdot r^{-1} \bmod P = \delta \cdot r \bmod P. \quad (4)$$

For deriving an efficient parallel architecture, Kim and Jeon [9] choose $r = x^{\lfloor m/2 \rfloor}$ as the Montgomery factor. Then, the Montgomery multiplication over $GF(2^m)$ can be formulated as

$$C = A \cdot B \cdot r^{-1} \bmod P = A \cdot B \cdot x^{-\lfloor m/2 \rfloor} \bmod P. \quad (5)$$

C is represented by substituting (2) in (5) as follows:

$$\begin{aligned} C &= \left(\sum_{j=0}^{m-1} b_j A x^j \right) \cdot x^{-\lfloor m/2 \rfloor} \bmod P \\ &= \sum_{j=0}^{\lfloor m/2 \rfloor - 1} b_j A x^{j-\lfloor m/2 \rfloor} \bmod P + \sum_{j=\lfloor m/2 \rfloor}^{m-1} b_j A x^{j-\lfloor m/2 \rfloor} \bmod P \\ &\equiv S + T, \end{aligned} \quad (6)$$

where $S = \sum_{j=0}^{\lfloor m/2 \rfloor - 1} b_j A x^{j-\lfloor m/2 \rfloor} \bmod P$ and $T = \sum_{j=\lfloor m/2 \rfloor}^{m-1} b_j A x^{j-\lfloor m/2 \rfloor} \bmod P$.

S and T can be simultaneously executed because there are no data dependency between them. Using this property, they proposed the polynomial basis multiplication architecture using Montgomery multiplication algorithm over $\mathrm{GF}(2^{\mathrm{m}})$ with about half time complexity compared to the typical related architectures.

2.2. Recomputing with shifted operand (RESO)

Patel and Fung [11, 12] used the RESO method for providing concurrent error detection capability on a traditional arithmetic logic unit with multiply/divide array. Their method is based on time redundancy, using the existing hardware to shift and replicate operations for developing the concurrent error detection capability of the circuits. In the RESO method, every operation is executed twice, once for the basic operation and once for the shifted input operation. The results from both operations are compared to detect errors. A mismatch indicates the presence of errors.

3. The Concurrent Error Detection Scheme for Montgomery Multiplier

In this section, we propose a new concurrent error detection scheme based on the RESO method using the Montgomery multiplication algorithm in [9]. For convenience of deriving the relevant equations, we assume that m is even. Let $k = \lfloor m/2 \rfloor$. We use $r = x^k$ and $r' = x^{k+1}$ as two Montgomery factors for deriving an efficient concurrent error detection scheme. α and β are two elements in $GF(2^m)$ generated by P. A and B are given by multiplying α and β with $r = x^k$, respectively. They are represented as follows:

$$A = \alpha \cdot r \bmod P = \alpha \cdot x^k \bmod P,\ B = \beta \cdot r \bmod p = \beta \cdot x^k \bmod P. \tag{7}$$

Similarly, A' and B' are given by multiplying α and β with $r' = x^{k+1}$, respectively. They are represented as follows:

$$A' = \alpha \cdot r' \bmod P = \alpha \cdot x^{k+1} \bmod P,\ B' = \beta \cdot r' \bmod p = \beta \cdot x^{k+1} \bmod P. \tag{8}$$

We can perform two Montgomery multiplications as

$$C = A \cdot B \cdot r^{-1} \bmod P = A \cdot B \cdot x^{-k} \bmod P, \tag{9}$$

$$C' = A' \cdot B' \cdot r'^{-1} \bmod P = A' \cdot B' \cdot x^{-(k+1)} \bmod P. \tag{10}$$

We can rewrite (9) and (10) as

$$C = \left(\sum_{j=0}^{m-1} b_j A x^j \right) \cdot x^{-k} \bmod P$$

$$= \sum_{j=0}^{k-1} b_j A x^{j-k} \bmod P + \sum_{j=k}^{m-1} b_j A x^{j-k} \bmod P \equiv S + T \tag{11}$$

$$C' = \left(\sum_{j=0}^{m-1} b'_j A'^{x^j} \right) \cdot x^{-(k+1)} \bmod P$$

$$= \sum_{j=0}^{k} b'_j A' x^{j-(k+1)} \bmod P + \sum_{j=k+1}^{m-1} b'_j A' x^{j-(k+1)} \bmod P \equiv S' + T' \tag{12}$$

where $S = \sum_{j=0}^{k-1} b_j A x^{j-k} \bmod P$, $T = \sum_{j=k}^{m-1} b_j A x^{j-k} \bmod P$, $S' = \sum_{j=0}^{k} b'_j A' x^{j-(k+1)} \bmod P$, and $T' = \sum_{j=k+1}^{m-1} b'_j A' x^{j-(k+1)} \bmod P$.

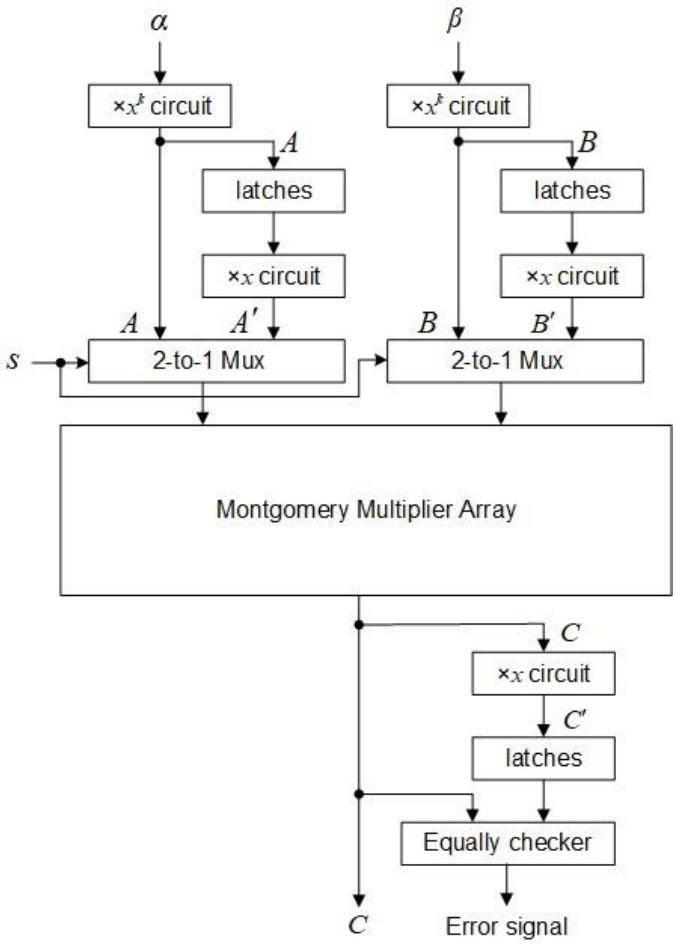

Fig. 1: Proposed concurrent error detection for the Montgomery multiplication.

We can modify the Montgomery multiplier in [9] for computing two consecutive Montgomery multiplications with two different Montgomery factors. The flowchart of the proposed error detection scheme is depicted in Fig. 1. The RESO method is employed for concurrent error detection of the fault existing in the multiplier array. The fundamental operations of the proposed error detection are shown as the following steps:

1) The first step is executed in normal multiplication mode using the inputs A and B based Montgomery factor $r = x^k$. Both inputs A and B are applied to the multiplier and the result C is converted by the function unit ×x circuit to C' and then such a C' is stored in latches.

2) The second step is performed in the error-checking operation mode applying with the alternate inputs A' and B' based Montgomery factor $r' = x^{k+1}$. Both A' and B' are input to the Montgomery multiplier array and the result C' is compared to the previously stored result C' in latches.

4. Conclusion

In this paper, we have presented the method of the concurrent error detection based on Kim-Jeon's Montgomery multiplier. Our scheme uses two Montgomery factors for deriving an efficient concurrent error detection. We expect that a modified Montgomery multiplier can perform Montgomery multiplications with two different factors. Also, we expect that the multiplier with concurrent error detection using our scheme can save about 50% time complexity as compared to the existing structures. We will implement the architecture to be able to compute Montgomery multiplications with two different factors in future research.

5. Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01059739).

6. References

[1] R.E. Blahut. Theory and Practice of Error Control Codes. Addison-Wesley, 1983.

[2] A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

[3] E. Biham, A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. In: *Proc. Crypto*, 1997, (LNCS, 1294), pp. 513-525.

[4] D. Boneh, R. Demillo, R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. *In: Proc. Advances in cryptology - EUROCRYPT'97*, 1997, pp. 37-51

[5] C. Koc, T. Acar. Montgomery Multiplication in $GF(2^k)$. *Des. Codes Cryptogr.* 1998, 14: 57-69.

[6] C.W. Chiou, C.Y. Lee, A.W. Deng, J.M. Lin. Concurrent Error Detection in Montgomery Multiplication over $GF(2^m)$. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*. 2006, E89-A(2): 566-574.

[7] A. Hariri, A. Reyhani-Masoleh. Fault Detection Structures for the Montgomery Multiplication over Binary Extension Fields. In: *Proc. Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC)*. 2007, pp. 37-46.

[8] A. Hariri, A. Reyhani-Masoleh. Concurrent Error Detection in Montgomery Multiplication over Binary Extension Fields. *IEEE Transactions on Computers*. 2011, 60 (9): 1341-1353.

[9] K.W. Kim, J.C. Jeon. Polynomial Basis Multiplier Using Cellular Systolic Architecture. *IETE Journal of Research*. 2014, 60(2): 194-199.

[10] P. Montgomery. Modular Multiplication without Trial Division. *Math. Comput.* 1985, 44: 519-521.

[11] J.H. Patel, L.Y. Fung. Concurrent Error Detection in ALU's by Recomputing with Shifted Operands. *IEEE Trans. Comput.* 1982, C-31: 589-595.

[12] J.H. Patel, L.Y. Fung. Concurrent Error Detection in Multiply and Divide Arrays. *IEEE Trans. Comput.* 1983, **C**-32: 417-422.