Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **6**, November **2017**

Low-Space Complexity Digit-Serial Multiplier Based on Modified Polynomial Basis Over $GF(2^m)$

Jeng-Shyang, Pan[1,2]

[1]Fujian Provincial Key Lab of Big Data Mining and Applications, China
[2]Harbin Institute of Technology Shenzhen Graduate School, China
jspan@cc.kuas.edu.tw

Shu-Xia, Dong

Harbin Institute of Technology Shenzhen Graduate School, China
dsx_1994@126.com

Chun-Sheng, Yang

Harbin Institute of Technology Shenzhen Graduate School, China
starissim@126.com

Received March, 2017; revised July, 2017

Abstract. *The multiplication is one of the most time-consuming and hardware-consuming operations in finite field for the applications of elliptic curve cryptography. In this paper, in order to reduce the complexities of multiplication, a new polynomial basis is proposed, which is generated by the irreducible trinomial and called modified polynomial basis (MPB). The modified polynomial basis multiplication can be transformed into the matrix-vector form. The obtained matrix satisfies the properties of Toeplitz matrix. According to the properties of Toeplitz matrix, a digit-serial multiplier over $GF(2^m)$ by irreducible trinomials is presented. From theoretical analysis, the proposed multiplier involves lower area complexity, less energy consumption than the other existing digit-serial multipliers.*
Keywords: Elliptic curve cryptography; Irreducible trinomial; Toeplitz matrix

1. **Introduction.** The elliptic curve cryptography (ECC) algorithm has become a popular research field. Compared with other encryption algorithms, ECC has the advantages of short key at the same security conditions. The ECC algorithm contains a large number of arithmetic operations, such as point multiplication, point addition and multiples point. These operations are repeatedly achieved by the basic operations of addition, multiplication, inversion in large prime field $GF(p)$ or binary extension field $GF(2^m)$. Addition can be easily performed by 2-input XOR gate; inversion can be performed by repeating multiplication. Multiplication is a high frequency and high resource-cost operation. Therefore high-performance and low-latency multiplication design and implementation is essential, especially, in resource-constrained environments.

There is no carry-propagation in $GF(2^m)$ compared with $GF(p)$, so it is more conducive to the realization of modern digital. There are 2^m elements in the finite field $GF(2^m)$, and each element can be represented by a bit string of length m. The representation of elements is generally based on three basis, named as polynomial basis (PB), normal basis (NB) and Dual Basis (DB). According to polynomial basis representation, multiplication involves two steps : school multiplication and reduction by $F(x)$, where $F(x)$ is an irreducible polynomial. In order to reduce the computational complexity, $F(x)$ with a low number of nonzero term is a best choice. $F(x)$ is generally trinomial or pentanomial [1], where trinomial, do not exist for all degree m, it is conjectured that irreducible pentanomials exist for any degree $m \geq 4$ [2]. In addition to the trinomial and pentanomial, there are many variants. Many multipliers based on polynomial basis or it's variants have been studied in [3, 4, 5]. Itoh and Tsujii [3] based on all one polynomial (AOP) and equally spaced polynomial (ESP) designed two low-complexity multipliers in $GF(2^m)$. In [5, 6], time/area-efficient implementation based on shifted polynomial basis (SPB) have been introduced and in [7] previous multiplication results in [5] was optimizied. Parallel Polynomial Multiplication in $GF(2^m)$ for all degree m based on Generalized polynomial basis (GPB) was proposed in [8]. Recently, in [9] demonstrated that the SPB is a special class of GPB, hence SPB and GPB multiplication can be classified as one class. The design and implementing approaches of multiplication algorithm in $GF(2^m)$ are broadly divided into two categories: Karatsuba algorithm (KA) and Toeplitz matrix-vector product (TMVP). The algorithms extended by the KA algorithm have (b,2)-way KA, (a,b)-way KA etc. The algorithms derivative by TMVP have two-way TMVP, TMVP block recombination (TMVPBR). The proposed multipliers architecture can be divided into three structures [10, 11, 12, 13]: (1) bit-serial,with $O(m)$ area complexity but has large computation time [14]. (2) bit-parallel, with $O(m^2)$ area complexity but has less computation time [12]. (3) digit-serial, used to balance time and area complexities [15, 16].

In this paper, a new polynomial basis, which is called MPB, is proposed by transforming the polynomial basis in $GF(2^m)$. Based on MPB representation, MPB multiplication can be transform into Toeplitz matrix-vector product. According to the properties of Toeplitz, we proposed a digit-serial architecture to achieve low-space complexity multiplier.

The organized of this paper is as follows. Section 2 simply introduces polynomial basis multiplication and two-way TMVP. In section 3, we define a new polynomial basis MPB, then deduced the general formula of two element of MPB multiplication. Section 4 transforms MPB multiplication into Toeplitz matrix vector product. According to the properties of Toeplitz, we propose a digit-serial architecture and analyze its complexity, In section 5, we compare complexity of our proposed multiplier with the existing other digit-serial multipliers, the summary of this paper is given in Section 6.

2. **Mathematical Background.**

In this section, we briefly review the polynomial basis multiplication over $GF(2^m)$ and the two-way TMVP algorithm.

2.1. **PB multiplication.**

The binary extension field $GF(2^m)$ can be view as the m dimension vector over $GF(2)$. All field element can be represented by the m dimension vector. The ordered set $N = \{1, x, x^2, \cdots, x^{m-1}\}$ is called the polynomial basis in $GF(2^m)$, then the filed element A can be represented as $A = a_0 + a_1 x + \cdots + a_{m-1}x^{m-1}$, where $a_i \epsilon \{0, 1\}, 0 \leq i \leq m-1$.

In $GF(2^m)$, field elements are generated by an irreducible polynomial $F(x) = x^m + \sum_{i=0}^{m-1} f_i x^i$, where $f_i \epsilon \{0, 1\}$. In order to reduce the complexity of PB multiplication , $F(x)$ with a low number of nonzero term is a best choice. $F(x)$ is generally trinomial or pentanomial, where trinomial, do not exist for all degree m, it is conjectured that irreducible pentanomials exist for any degree $m \geq 4$. Let $A(x) = \sum_{i=0}^{m-1} a_i x^i$, $B(x) = \sum_{i=0}^{m-1} b_i x^i$ are two elements in $GF(2^m)$ and the polynomial basis multiplication represent as $C(x) = AB \mod F(x)$, which can be carried out by two steps:

(1) School multiplication

$$T = AB = (\sum_{j=0}^{m-1} a_j x^j)(\sum_{k=0}^{m-1} b_k x^k) = \sum_{i=0}^{2m-2} t_i x^i, \tag{1}$$

where $t_i = \sum_{j+k=i} a_j b_k$, for $0 \leq j, k < m$ and $0 \leq i \leq 2m-2$.

(2) Reduction

$$C = T \mod F(x) = \sum_{i=0}^{m-1} c_i x^i. \tag{2}$$

Recently, a kind of low-latency digit serial multiplication methods proposed, such as, Lee et al. [9], have presented a digit-serial and scalable SPB/GPB multiplier with low-space complexity using (b,2) - way KA decomposition. In 2015, Lee et al. [17] have used Toeplitz Matrix-Vector Product Decomposition achieved efficient subquadratic space Complexity multiplier for All Trinomials. Liu et al. [18] based on Karatsuba algorithm proposed a efficient digit-serial multiplier in $GF(2^m)$.

2.2. **Two-way TMVP.** Let T be a $n \times n$ Toepltiz matrix, V be a $n \times 1$ column vector, where $n = 2^k$. TV can be called as a TMVP. T can be split into (T_0, T_1, T_2), where T_0,T_1 and T_2 are $(\frac{n}{2}) \times (\frac{n}{2})$ Toeplitz matrices and V can be split into (V_0, V_1) , where V_0 and V_1 are $(\frac{n}{2}) \times 1$ column vector. A $n \times n$ Toeplitz matrix is determined by $(2n-1)$ elements of the first row and the first column. The product of T and V can be written as:

$$\begin{aligned} TV &= \begin{bmatrix} T_1 & T_0 \\ T_2 & T_1 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \end{bmatrix}, \\ &= \begin{bmatrix} (T_0+T_1)V_1 + T_1(V_0+V_1) \\ (T_1+T_2)V_0 + T_1(V_0+V_1) \end{bmatrix}, \\ &= \begin{bmatrix} P_0 + P_2 \\ P_1 + P_2 \end{bmatrix}. \end{aligned}$$

where $P_0 = (T_0 + T_1)V_1$, $P_1 = (T_1 + T_2)V_0$ and $P_2 = T_1(V_0 + V_1)$. The Toeplitz matrix-vector product TV is decomposed in to three partial products P_0, P_1 and P_2, where P_0, P_1 and P_2 are $(\frac{n}{2}) \times 1$ sizes TMVP.

We assume #XOR, #AND respectively represent the number of XOR gates, AND gates. T_A, T_X respectively instead of the delay of AND gate, XOR gate. According to [19], the complexities of two-way TMVP can represented as:

$$\begin{cases} \#AND = n^{\log_2 3}, \\ \#XOR = 5.5n^{\log_2 3} - 6n + 0.5, \\ D = T_A + (2\log_2 n)T_X. \end{cases} \tag{3}$$

3. **Modified Polynomial Basis Multiplication.** Let $N = \{1, x, \cdots, x^m\}$ be a polynomial basis of $GF(2^m)$, $F(x) = x^m + x^n + 1$ be a irreducible trinomial to generate all the elements, where $n < m$. From (1) and (2), $F(x)$ is used to reduce the degree of $t_i x^i$ for $m \leq i \leq 2m-2$. In the reduction of $x^m, x^{m+1}, \cdots, x^{2m-2}$, we can find some of elements can be reused, based on the observing, we defined a new polynomial basis.

Definition 3.1. *Let $N = \{1, x, \cdots, x^m\}$ be a polynomial basis in $GF(2^m)$, $F(x) = x^m + x^n + 1$, where $n < m$, be the irreducible trinomial. A new basis is given to instead of the PB N, which is called as modified polynomial basis (MPB), denoted as N'. The MPB N' can be expressed as:*

$$N' = \begin{cases} \{\beta_0, \beta_1, \cdots, \beta_{k-1}, \beta_k, \cdots, \beta_{m-1}\}, & n > \frac{m}{2} \\ \{\gamma_0, \gamma_1, \cdots, \gamma_{n-1}, \gamma_n, \cdots, \gamma_{m-1}\}, & n < \frac{m}{2} \end{cases} \tag{4}$$

where $k = m - n$ and

$$\beta_i = \begin{cases} x^i + x^{i+n}, & 0 \le i \le k-1 \\ x^i, & i \ge k \end{cases}, \tag{5}$$

$$\gamma_i = \begin{cases} x^{m-i} + x^{n-i}, & 0 \le i \le n-1 \\ x^{m-n}, & n \le i \le m-1 \end{cases} \tag{6}$$

Lemma 3.1. *According to definition 3.1, we can obtain following lemma. Let A be a element of PB, denoted as $A = \sum_{i=0}^{m-1} a_i x^i$. Using MPB, A can be expressed as:*

$$A = \begin{cases} \sum_{i=0}^{n-1} a_i \beta_i + \sum_{i=n}^{m-1} a_{i(i-n)} \beta_i, & n > \frac{m}{2} \\ a_0 \gamma_0 + \sum_{i=1}^{m-n} a_{m-i} \gamma_i + \sum_{i=m-n+1}^{m-1} a_{(m-i)(2m-i-n)} \gamma_i, & n < \frac{m}{2} \end{cases} \tag{7}$$

where $a_{i(i-n)} = a_i + a_{i-n}$, $a_{(m-i)(2m-i-n)} = a_{(m-i)} + a_{(2m-i-n)}$.

Let A and B be two elements of PB in $GF(2^m)$. Based on the new MPB representation, we consider the multiplication $C = AB$.

(a) $n \ge m/2$

The product of A and B can be written as:

$$AB = b_0 A + \cdots + b_{k-1} A x^{k-1} + b_k A x^k + \cdots + b_{m-1} A x^{m-1} \tag{8}$$

According to (5) we can obtain:

$$x\beta_i = \begin{cases} \beta_{i+1} & 0 \le i \le k-2 \\ \beta_k + \beta_0 & i = k-1 \\ \beta_{i+1} & k \le i \le m-2 \\ \beta_0 & i = m-1 \end{cases} \tag{9}$$

we assume that $A^{(j)} = Ax^j, A^{(j)} = \sum_{i=0}^{m-1} a_i^{(j)} \beta_i$, $0 \le j \le m-1$. According to (7) and (9), $A^{(0)}$ can be written as:

$$A^{(0)} = A = \sum_{i=0}^{n-1} a_i \beta_i + \sum_{i=n}^{m-1} a_{(i-n)i} \beta_i, \tag{10}$$

According to (9) and (10), $A^{(j+1)}$ for $0 \le j \le m-2$ can be written as:

$$\begin{aligned} A^{(j+1)} &= xA^{(j)} \\ &= a^{(j)}_{(k-1)(m-1)} \beta_0 + \sum_{i=1}^{n} a^{(j)}_{i-1} \beta_i + \sum_{i=n+1}^{m-1} a^{(j)}_{(i-n-1)(i-1)} \beta_i. \end{aligned} \tag{11}$$

where $a_{(i-n)i} = a_{(i-n)} + a_i$, $a^{(j)}_{(k-1)(m-1)} = a^{(j)}_{(k-1)} + a^{(j)}_{(m-1)}$, $a^{(j)}_{(i-n-1)(i-1)} = a^{(j)}_{(i-n-1)} + a^{(j)}_{(i-1)}$.

According to the above formula (11) we can conclude that the coefficients of $A^{(j+1)}$ is obtained by cycle right shift k-bit and one XOR gates from $A^{(j)}$, where $A^{(0)} = A$, $0 \le j \le m-2$, when $n \ge m/2$ and $m - n = k$.

(b) $n < m/2$

The product of A and B can be written as:

$$\begin{aligned} AB = {} & b_0 A + b_1 Ax + \cdots + b_{k-1} A x^{k-1} + b_k A x^k \\ & + \cdots + b_{m-1} A x^{m-1} \end{aligned} \tag{12}$$

According to (6) we can obtain:

$$x\gamma_i = \begin{cases} \gamma_{m-1} & i = 0 \\ \gamma_{i-1} & 1 \le i \le n \\ \gamma_{n-1} + \gamma_{m-1} & i = n \\ \gamma_{i-1} & n < i < m \end{cases} \tag{13}$$

we assume that$A^{(j)} = Ax^j$, $A^{(j)} = \sum_{i=0}^{m-1} a_i^{(j)}\gamma_i$, $0 \le j \le m-1$. According to (9) and (13), $A^{(0)}$ can be written as:

$$A^{(0)} = A = a_0\gamma_0 + \sum_{i=1}^{m-n} a_{m-i}\gamma_i + \sum_{i=m-n+1}^{m-1} a_{(m-i)(2m-i-n)}\gamma_i, \tag{14}$$

According to (14) and (13), $A^{(j+1)}$ for $0 \le j \le m-2$ can be written as:

$$\begin{aligned} A^{(j+1)} &= xA^{(j)} \\ &= \sum_{i=0}^{m-n-1} a_{m-n-i}^{(j)}\gamma_i + \sum_{i=m-n}^{m-2} a_{(m-i-1)(2m-i-n)}^{(j)}\gamma_i \\ &\quad + a_{0(m-n)(2m-2n)}^{(j)}\gamma_{m-1}. \end{aligned} \tag{15}$$

Based on the above formula (15) we can conclude that the coefficients of $A^{(j+1)}$ is obtained by cycle left shift k-bit and one XOR gates from $A^{(j)}$. $A^{(0)} = A$, $0 \le j \le m-2$, when $n < m/2$ and $m - n = k$

According to the above analysis, we can obtain following summary: the modified polynomial basis multiplication can be transformed into the matrix-vector form. The obtained matrix satisfies the properties of Toeplitz matrix. Therefore we can use two-way TMVP perform MPB multiplication.

4. **Proposed Multiplier Based On MPB.**

4.1. **Digit Serial Architecture.**

According to MPB, the product of A and B can be permed by a Toeplitz matrix-vector product as:

$$C = TV = \begin{bmatrix} t_{m-1} & \cdots & t_{2m-3} & t_{2m-2} \\ \vdots & \vdots & \vdots & \vdots \\ t_1 & \cdots & t_{m-1} & t_m \\ t_0 & \cdots & t_{m-2} & t_{m-1} \end{bmatrix} \begin{bmatrix} v_0 \\ \vdots \\ v_{m-2} \\ v_{m-1} \end{bmatrix}$$

where T is an $m \times m$ Toeplitz matrix and V is an $m \times 1$ column vector.

Let $m = kd$, d is the digit size, T and V are divide into k parts:

$$T = \begin{bmatrix} T_{k-1} & T_k & \cdots & T_{2k-3} & T_{2k-2} \\ T_{k-2} & T_{k-1} & \cdots & T_{2k-4} & T_{2k-3} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ T_1 & T_2 & \vdots & T_{k-1} & T_k \\ T_0 & T_1 & \cdots & T_{k-2} & T_{k-1} \end{bmatrix} \text{ and}$$

$$V = \begin{bmatrix} V_0 & V_1 & \cdots & V_{k-1} \end{bmatrix}^T,$$

where T_i is $d \times d$ size and V_i is $d \times 1$ size. then $C = TV$ can be written as:

$$TV = \begin{bmatrix} T_{k-1}V_0 + T_kV_1 + \cdots + T_{2k-2}V_{k-1} \\ \vdots \\ T_1V_0 + T_2V_1 + \cdots + T_kV_{k-1} \\ T_0V_0 + T_1V_1 + \cdots + T_{k-1}V_{k-1} \end{bmatrix}$$

$$= \begin{bmatrix} T_{k-1}V_{0(k-1)} + T_kV_{1(k-1)} + \cdots + T_{k-1\sim 2k-2}V_{k-1} \\ \vdots \\ T_1V_{01} + T_{1\sim k}V_1 + \cdots + T_{k-1}V_{1(k-2)} + T_kV_{1(k-1)} \\ T_{0\sim k-1}V_0 + T_1V_{01} + \cdots + +T_{k-1}V_{0(k-1)} \end{bmatrix}$$

$$= \begin{bmatrix} p_{0(k-1)} + p_{1(k-1)} + \cdots + p_{(k-2)(k-1)} + p_{k-1} \\ \vdots \\ p_{01} + p_1 + \cdots + p_{1(k-2)} + p_{1(k-1)} \\ p_0 + p_{01} + \cdots + p_{0(k-2)} + p_{0(k-1)} \end{bmatrix} \quad (16)$$

where $T_{i\sim j} = \sum_{j=i}^{i+k-1} T_j$ and $i < k-1$, $V_{ij} = V_i + V_j$ and $i < j \le k-1$, $p_i = T_{i\sim k-1+i}V_i$ and $i \le k-1$, $p_{ij} = T_{i+j}V_{ij}$ and $i < j \le k-1$.

Due to the symmetry of (16), we just need to compute $\frac{k^2+k}{2}$ multiplication, The Figure 4.1 shows the proposed MPB multiplier architecture. Figure 4.1 contains four parts, respectively T Generator, V Generator, TMVP Multiplier and Reconsyrction. Multiplier architecture involves nine components (S_0, S_1, S_2, S_3, S_4, P, ACC1, ADD, ACC2). Next, we introduce the function of each component and estimate the complexities of these component. The space complexity of multiplier is expressed by the number of 2-input XOR gate, 2-input AND gate and 2-input MUX gate. The number of 2-input XOR gates, 2-input AND gates and 2-input MUX respectively expressed as #XOR, #AND and #MUX. The time complexity of multiplier is expressed by the delay of 2-input XOR gate, 2-input AND gate and 2-input MUX gate. T_A, T_X, T_{MUX} respectively instead of the delay of 2-input AND gate, 2-input XOR gate and 2-input MUX gate.

(a) T Generator: Includes S_0, ACC1, S_1 components. These three components are used to generate the entire T sequence $\{T_{0\sim k-1}, \cdots, T_{k-1\sim 2k-2}\}$and $\{T_1, \cdots, T_{2k-3}\}$. T consists two parts,S_0 and ACC1 are selected from the input set $\{T_0, T_1, \cdots T_{2k-2}\}$ to accumulate produce the first part $\{T_{0\sim k-1}, \cdots, T_{k-1\sim 2k-2}\}$, then $S_1$from the inputs set $\{T_{i\sim j}, T_1, \cdots T_{2k-3}\}$ generate the whole T sequence, where $T_{i\sim j} = \sum_{j=i}^{i+k-1} T_j$, $i < k-1$. The input of S_0 is $\{T_0, T_1, \cdots T_{2k-2}\}$, where T_i for $i \le 2k-2$ is $d \times d$ size. The output of S_0 is T_i, for $, i \le 2k-2$. So the complexities of S_0 are $(2k-2)(2d-1)$ MUX and delay is $\lceil log_2(2k-2)\rceil T_{MUX}$. ACC1 accumulation unit consists of $(2d-1)$ XOR gates to compute addition of $d \times d$ size T_i for $i \le 2k-2$, and delay is T_x. S_1 requires $2k-3$ MUX when output $T_i, i \le 2k-2$ from $2k-2$ inputs. The same like S_0, the complexities are $(2k-3)(2d-1)$ MUX and the delay is $\lceil log_2(2k-3)\rceil T_{MUX}$.

(b) V Generator: Consist of S_2,S_3, ADD components. The three components are used to generate the V sequence $\{V_0, \cdots, V_{k-1}\}$ and $\{V_{01}, \cdots, V_{(k-2)(k-1)}\}$corresponding to the T sequence. S_2 and S_3 respectively generate sequences $\{V_0, \cdots, V_{k-2}\}$ and $\{V_1, \cdots, V_{k-1}\}$ from input set $\{V_0, \cdots, V_{k-1}\}$, ADD component add the output sequence of $S_0$and S_1, produce whole V sequence. S_2,S_3 are output a $d \times 1$ size $V_i, i \le k-1$ from the$k-1$ inputs. Hence the complexities are $(k-2)d$ MUX , delay is $\lceil log_2(k-2)\rceil T_{MUX}$. ADD achieve the addition of $d \times 1$size V_i, therefore the complexities of ADD are d XOR gates and T_x delay.

(c) TMVP Multiplier: P component performs the product of the outputs of (a) and (b), denoted as p_i for $i = 0, \cdots, k-1$ or p_{ij} for $0 \leq i < j \leq k-1$. According to (3) so the complexities are $d^{log_2 3}$AND gates, $\frac{11}{2}d^{log_2 3} - 6d + \frac{1}{2}$ XOR gates and delay is $T_A + 2\lceil log_2 d\rceil T_X$.

(d) Reconstruction: S_4 and ACC2 reconstruct the result of C by using the output of P components. S_4 extend p_i or p_{ij} from d size to m size; ACC2 is an accumulator. The complexities are m AND gates, m XOR gates and delay is $T_A + T_x$.

Here, we analysis the complexity of the each component of figure 4.1, Table 1 lists the number of logical gates required and required delay for each component.

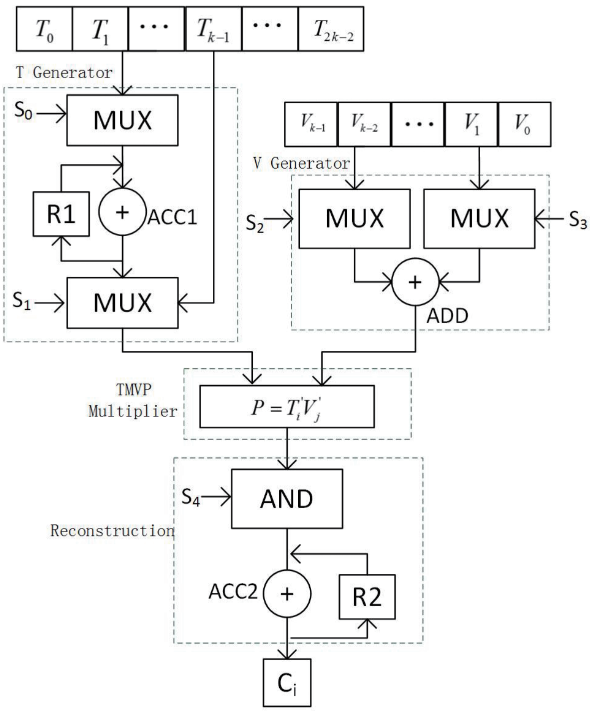

FIGURE 1. Architecture of proposed MPB multiplier

TABLE 1. Complexities of each component of proposed MPB

components	#AND	#XOR	#MUX	Delay
S_0	-	-	$(2k-2)(2d-1)$	$\lceil log_2(2k-2)\rceil T_{MUX}$
ACC1	-	$(2d-1)$	-	T_X
S_1	-	-	$(2k-3)(2d-1)$	$\lceil log_2(2k-3)\rceil T_{MUX}$
S_2	-	-	$(k-2)d$	$\lceil log_2(k-2)\rceil T_{MUX}$
S_3	-	-	$(k-2)d$	$\lceil log_2(k-2)\rceil T_{MUX}$
ADD		d	-	T_X
P	$d^{log_2 3}$	$\frac{11}{2}d^{log_2 3} - 6d + \frac{1}{2}$	-	$T_A + 2\lceil log_2 d\rceil T_X$
S_4	m	-	-	T_A
ACC2		m	-	T_X

The product of A and B can be performed by a Toeplitz matrix-vector product $C = TV$, where T is an $m \times m$ Toeplitz matrix and V is an $m \times 1$ column vector. To illustrate the multiplexer control table , let $m = kd$, $k = 4$ as a example, T and V are divide into k segmentation and $C = TV$ can be written as:

$$TV = \begin{bmatrix} T_3 & T_4 & T_5 & T_6 \\ T_2 & T_3 & T_4 & T_5 \\ T_1 & T_2 & T_3 & T_4 \\ T_0 & T_1 & T_2 & T_3 \end{bmatrix} \begin{bmatrix} V_3 \\ V_2 \\ V_1 \\ V_0 \end{bmatrix} = \begin{bmatrix} T_3V_{03} + T_4V_{13} + T_5V_{23} + T_{3\sim6}V_3 \\ T_2V_{02} + T_3V_{12} + T_{2\sim5}V_2 + T_5V_{23} \\ T_1V_{01} + T_{1\sim4}V_1 + T_3V_{12} + T_4V_{13} \\ T_{0\sim3}V_0 + T_1V_{01} + T_2V_{02} + T_3V_{03} \end{bmatrix} = \begin{bmatrix} p_{03} + p_{13} + p_{23} + p_3 \\ p_{02} + p_{12} + p_2 + p_{23} \\ p_{01} + p_1 + p_{12} + p_{13} \\ p_0 + p_{01} + p_{02} + p_{03} \end{bmatrix} \quad (17)$$

where the size of T_i is $d \times d$, the size of V_i is $d \times 1$.

From (17), we can find that the product C includes ten partial products: $p_0 = T_{0\sim3}V_0$, $p_1 = T_{1\sim4}V_1$, $p_2 = T_{2\sim5}V_2$, $p_3 = T_{3\sim6}V_3$, $p_{01} = T_1V_{01}$, $p_{02} = T_2V_{02}$, $p_{03} = T_3V_{03}$, $p_{12} = T_3V_{12}$, $p_{13} = T_4V_{13}$, $p_{23} = T_5V_{23}$. Next lists generated sequences and complexity analysis for MUX component in the figure 4.1.

(a) T Generator: $S_0$and ACC1 generate the sequence $\{T_{0\sim3}, T_{1\sim4}, T_{2\sim5}, T_{3\sim6}\}$ from the input set $T_0, \cdots, T_6$, required 6 MUX and $2d-1$ XOR gates. S_1 generate entire T sequence $\{T_{0\sim3}, T_{1\sim4}, T_{2\sim5}, T_{3\sim6}\}$ and $\{T_1, T_2, T_3, T_4, T_5\}$ needs 5 MUX.

(b) V Generator: S_2 and S_3 respectively produce the sequence $\{V_0, V_1, V_2\}$ and $\{V_1, V_2, V_3\}$ and ADD add the two sequences generate $\{V_0, V_1, V_2, V_3\}$ and $\{V_{01}, V_{02}, V_{03}, V_{12}, V_{13}, V_{23}\}$. Therefore S_2 and S_3 required 2 MUX and ADD needs d XOR gates.

(c) TMVP Multiplication: product the ten partial products $p_0, p_1, p_2, p_3, p_{01}, p_{02}, p_{03}, p_{12}, p_{13}, p_{23}$.

(d) Reconstruction: reconstruct the m size C.

Table 2 lists four control vector for $S_0, S_1, S_2, S_3$to be used to determine the partial products during each cycle. for example, S_0 generate the T_0 and S_1generate the $T_{0\sim3}$ when $i = 4$.

5. **Comparison.** Recently, various digit-serial multipliers have been proposed in [9], [20],and [21]. In [9], Lee et al. have presented a (b,2) - way KA decomposition to achieve digit-serial multiplier with low-space complexity multiplier. Talapatra et al. [21] have used the TMVP scheme to develop an efficient digit-serial systolic Montgomery multiplier for trinomials polynomials. Pan et al. [20], have used double basis multiplication which combines the polynomial basis and the modified polynomial basis to develop a new efficient digit-serial systolic multiplier. In this paper, we use modified polynomial basis develop a low-space complextiy digit-serial multiplier for trinomial. Table 3 lists the comparison results of our proposed multiplier and the existing digit-serial multipliers proposed in [20], [21]. It can be seen from the table 4 that the architecture we design is larger than the other two existing multiplier in the latency cycle, but in the number of logical gates be less than [21, 20].

In order to make the results more close to the actual implementation, we using the standard Nan-gate Open Cell Library_typical obtained the ASIC synthesis results to compares performance and complexities of the proposed multipier with the other two multipliers, presented in Talapatra et al.[21] and Jeng-Shyang-Pan et al. [20]. We choose the $F(x) = x^{409} + x^{87} + 1$ as the irreducible trinomial. The comparision of latency, area, power and total-time of synthesis tabulated in Tables 4.

Table 2. Four Control Tables. (a) S_0 control table; (b) S_1 control table; (c) S_2 and S_3 control table; (d) S_4 control table

(a)S_0

i	s_{00}	s_{01}	s_{02}	s_{03}	s_{04}	s_{05}	s_{06}
0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
2	0	0	1	0	0	0	0
3	0	0	0	1	0	0	0
4	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	0	0	1	0
8	0	0	1	0	0	0	0
9	0	0	0	0	0	0	1

(b)S_1

i	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}
0	0	1	0	0	0	0
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	0	0	0	0	1	0
4	1	0	0	0	0	0
5	0	0	0	1	0	0
6	1	0	0	0	0	0
7	0	0	0	0	1	0
8	0	0	0	0	0	1
9	1	0	0	0	0	0

(c).S_2

i	s_{20}	s_{21}	s_{22}	s_{23}
0	1	0	0	0
1	1	0	0	0
2	1	0	0	0
3	0	0	0	0
4	1	0	0	0
5	0	1	0	0
6	0	1	0	0
7	0	1	0	0
8	0	0	1	0
9	0	0	0	0

(c).S_3

i	s_{30}	s_{31}	s_{32}	s_{33}
0	0	1	0	0
1	0	0	1	0
2	0	0	0	1
3	0	0	0	0
4	0	0	0	0
5	0	1	0	0
6	0	0	0	0
7	0	0	0	1
8	0	0	0	1
9	0	0	0	1

(d)S_4

i	s_{40}	s_{41}	s_{42}	s_{43}
0	0	0	1	1
1	0	1	0	1
2	1	0	0	1
3	0	0	0	1
4	0	1	1	0
5	1	0	1	0
6	0	0	1	0
7	1	1	0	0
8	0	1	0	0
9	1	0	0	0

Table 3. Comparisons of Various Digit-Serial Multipliers over $GF(2^m)$

Multipliers	Talapatra *et al.*[21]	Pan *et al.*[20]	Proposed Figure 4.1
Architecture	Digit-Serial	Digit-Serial	Digit-Serial
Basis	Montgomery	DB	PB
Polynomial type	Trinomial	Trinomial	Trinomial
#AND	kd^2	$k^{\frac{3}{2}}P_2$	$m+P_2$
#XOR	kd^2+2d	$m+k^{\frac{1}{2}}P_9+d+P_4$	$m-4d+\frac{11}{2}P_2-\frac{1}{2}$
#MUX	$2kd$	$-$	$10m-4k-14d+5$
Latency	$2k$	$2k^{\frac{1}{2}}$	$\frac{k^2+k}{2}$
Critical path delay	$T_A+P_6T_x+T_{MUX}$	$(2+P_6)T_x$	$2T_A+(2+2P_6)T_x+P_{10}T_{MUX}$

Note: $P_2=d^{log_2 3}$, $P_3=k(2.5d^{log_2 3}-3d+0.5)+d^{log_2 3}-d$, $P_4=k(2d^{log_2 3}-2d)$, $P_5=kd^{log_2 3}$, $P_6=log_2 d$, $P_7=\lceil log_2(2k-2)\rceil$, $P_8=\lceil log_2(2k-3)\rceil$,$P_9=2d+P_3+P_5$,$P_{10}=P_7+P_8$

Proposed multiplier is based on the 2-way TMVP structure and we have considered five different segmentation number k, i.e., 2, 4, 6, 8, and 10, for synthesizing and both multipliers [21, 20], are also synthesized for the same segmentation number k. The corresponding digit-sizes $d=\frac{m}{k}$ of the three multipliers are same, i.e., 205,103,69,52, and 41. We note that choosing the same k and d will have a consistent comparison for all multipliers. In table 4, the area of our proposed MPB multiplication architecture is lower than other multipliers under the same digit-size over $GF(2^{409})$. comparably, as seen in Table. The proposed architecture area - saving about 72.9% − 81.7% compared to the multiplier [21] and area - saving about 61.1% − 86.6% compared to the multiplier [20] when $k = 2, 4, 6, 8, 10$.

Table 4. The Comparison of Latency *cycles*, Area [μm^2] , Power [$\mu W/GHz$] and Total-time [$ns \times cycles$] for the Previously-Presented Multiplier Architectures over $GF(2^{409})$ for Different Digit-Sizes d

k		2	4	6	8	10
d-size		205	103	69	52	41
Talapatra *et al.*[21]	Latency	4	8	12	16	20
	Area	225,750	114,740	77,748	59,256	46,372
	Power	631,270	320,100	216,390	164,550	128,460
	Total-time	3.84	7.68	11.52	15.36	19.20
Pan *et al.*[20]	Latency	$2\sqrt{2}$	4	$2\sqrt{6}$	$4\sqrt{2}$	$2\sqrt{10}$
	Area	124,010	103,080	94,086	88,734	82,428
	Power	302,120	253,250	232,180	219,630	204,510
	Total-time	0.91	1.28	1.57	1.81	2.02
Proposed Figure 4.1	Latency	3	10	21	36	55
	Area	47,473	20,657	14,618	12,277	10,871
	Power	114,580	47,611	32,331	26,334	22,799
	Total-time	2.88	9.6	20.16	34.56	52.80

6. **Conclusions.** In this paper, we have proposed a novel low-space complexity digit-serial multiplier architecture for modified polynomial basis multiplication over $GF(2^m)$. The proposed new basis MPB is generated by irreducible trinomial $F(x) = x^m + x^n + 1$ when m and n satisfies $n \geq \frac{m}{2}$ or $n < \frac{m}{2}$. The MPB multiplication can transform into Toeplitz matrix-vector product. According to the property of Toeplitz we proposed a digit-serial architecture. In section 4 and Table 1, we have provided a theoritical analysis of the complexities of architecture component, including S_0, S_1, S_2, S_3, S_4, P, ACC1, ADD, ACC2. The proposed multiplier architecture involves significantly lower area complexity, less energy consumption than the other existing digit-serial multipliers.

REFERENCES

[1] H. Fan, J. Sun, M. Gu, and K. Y. Lam, Overlap-Free Karatsuba-Ofman Polynomial Multiplication Algorithms, *IET Information Security*, vol. 4, pp. 8–14, March 2010.

[2] G. Seroussi, Table of low-weight binary irreducible polynomials, *Hp Labs Technical Reports*, pp. 98–135, 1998.

[3] T. Itoh and S. Tsujii, Structure of Parallel Multipliers for a Class of Fields $GF(2^m)$, *Information and Computation*, vol. 83, no. 1, pp. 21–40, 1989.

[4] F. Rodriguez-Henriguez and C. K. Koc, Parallel multipliers based on special irreducible pentanomials, *IEEE Transactions on Computers*, vol. 52, pp. 1535–1542, Dec 2003.

[5] H. Fan and Y. Dai, Fast bit-parallel $GF(2^m)$ multiplier for all trinomials, *IEEE Transactions on Computers*, vol. 54, pp. 485–490, April 2005.

[6] A. Cilardo, Efficient Bit-Parallel $GF(2^m)$ Multiplier for a Large Class of Irreducible Pentanomials, *IEEE Transactions on Computers*, vol. 58, pp. 1001–1008, July 2009.

[7] H. Fan and M. A. Hasan, Fast Bit Parallel-Shifted Polynomial Basis Multipliers in $GF(2^m)$, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, pp. 2606–2615, Dec 2006.

[8] A. Cilardo, Fast Parallel $GF(2^m)$ Polynomial Multiplication for All Degrees, *IEEE Transactions on Computers*, vol. 62, pp. 929–943, May 2013.

[9] C. Y. Lee and C. S. Yang and B. K. Meher and P. K. Meher and J. S. Pan, Low-Complexity Digit-Serial and Scalable SPB/GPB Multipliers Over Large Binary Extension Fields Using $(b, 2)$-Way Karatsuba Decomposition, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, pp. 3115–3124, Nov 2014.

[10] C. Y. Lee, C. C. Chen, Y. H. Chen, and E. H. Lu, Low-Complexity Bit-Parallel Systolic Multipliers over $GF(2^m)$, in *2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1160–1165, Oct 2006.

[11] J. S. Pan, C. Y. Lee, and P. K. Meher, Low-Latency Digit-Serial and Digit-Parallel Systolic Multipliers for Large Binary Extension Fields, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, pp. 3195–3204, Dec 2013.

[12] A. Reyhani-Masoleh and M. A. Hasan, Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over $GF(2^m)$, *IEEE Transactions on Computers*, vol. 53, pp. 945–959, Aug 2004.

[13] C.-Y. Lee, J.-S. Horng, I.-C. Jou, and E.-H. Lu, Low-Complexity Bit-Parallel Systolic Montgomery Multipliers for Special Classes of $GF(2^m)$, *IEEE Transactions on Computers*, vol. 54, pp. 1061–1070, Sept 2005.

[14] C. Y. Lee, P. K. Meher, and J. C. Patra, Concurrent Error Detection in Bit-Serial Normal Basis Multiplication Over $GF(2^m)$ Using Multiple Parity Prediction Schemes, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.18, no.8*, pp. 1234–1238, Aug 2010.

[15] C. Y. Lee and P. L. Chang, Digit-Serial Gaussian Normal Basis Multiplier over $GF(2^m)$ Using Toeplitz Matrix-Approach, in *2009 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, Dec 2009.

[16] C. S. Yang, J. S. Pan, and C. Y. Lee, Digit-Serial GNB Multiplier Based on TMVP Approach over $GF(2^m)$, in *2013 Second International Conference on Robot, Vision and Signal Processing*, pp. 123–128, Dec 2013.

[17] C. Y. Lee and P. K. Meher, Efficient Subquadratic Space Complexity Architectures for Parallel MPB Single- and Double-Multiplications for All Trinomials Using Toeplitz Matrix-Vector Product Decomposition, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 854–862, March 2015.

[18] C. H. Liu, C. Y. Lee, and P. K. Meher, Efficient Digit-Serial KA-Based Multiplier Over Binary Extension Fields Using Block Recombination Approach, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, pp. 2044–2051, Aug 2015.

[19] H. Fan and M. A. Hasan, A New Approach to Subquadratic Space Complexity Parallel Multipliers for Extended Binary Field , *IEEE Transactions on Computers*, vol. 56, pp. 224–233, Feb 2007.

[20] J. S. Pan, R. Azarderakhsh, M. M. Kermani, C. Y. Lee, W. Y. Lee, C. W. Chiou, and J. M. Lin, Low-Latency Digit-Serial Systolic Double Basis Multiplier over $GF(2^m)$ Using Subquadratic Toeplitz Matrix-Vector Product Approach, *IEEE Transactions on Computers*, vol. 63, pp. 1169–1181, May 2014.

[21] S. Talapatra, H. Rahaman, and S. K. Saha, Unified Digit Serial Systolic Montgomery Multiplication Architecture for Special Classes of Polynomials over $GF(2^m)$, in *2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, pp. 427–432, Sept 2010.