Abstract—This paper presents a novel algorithm for type-t Gaussian normal basis (GNB) binary finite field multiplication using Toeplitz matrix-vector representation. It is shown that the GNB multiplication can be realized through block Toeplitz matrix-vector-products. A digit-serial systolic GNB multiplier is proposed where each processing element is comprised of a Toeplitz multiplier and three registers. Analytical results indicate that our proposed architecture has significantly lower area complexity than existing digit-serial multipliers.

Index: Toeplitz Matrix-Vector, Gaussian Normal Basis, digit-serial multiplier

I. INTRODUCTION

Efficient design and implementation of finite field multipliers have received considerable attentions in recent years due to their applications in elliptic curve cryptography (ECC) and error control coding [1-3]. Although both channel codes and cryptographic algorithms adopt the finite field GF($2^m$), they require different field orders. Channel codes are typically restricted to field elements represented by up to eight bits, whereas ECC relies on field sizes of several hundred bits. Therefore, to have high performance security functions, high-speed algorithms and hardware architectures are necessary for the finite field arithmetic operations.

Amongst the basic finite field arithmetic operations over GF($2^m$), multiplication is one of the most important and time consuming operations. Several architectures for finite field multiplications are therefore suggested in the literature. Finite field multipliers are generally classified as either bit-parallel or serial architecture. A bit-serial multiplier [4,5] requires a smaller area, but it is slow, for example, it takes $m$ clock cycles to carry out a multiplication for a field of order $m$. Conversely, parallel multipliers [6-9] are faster, but involve higher hardware cost.

The digit-serial structures process $d$-bits in one cycle, where $d$ varies from one bit to the word length of $N$ bits, such that it becomes a bit-parallel structure when $d$ equals $N$ and will reduce to a bit-serial one when $d$ equals 1. Such kind of exibility of digit-serial designs enables the designers to have the necessary trade-off between the hardware cost and the throughput rate. A variety of digit-serial architectures have been proposed for computations in the binary extended fields [10-14]. In general, the polynomial basis (PB) multipliers are suitable for implementing digit-serial architectures through systematic unfolding. Besides, these architectures could be scaled by any digit width $d$, whereas the normal basis multiplier [14] can only be realized by the digit-serial architecture with digit width $d$ by which the degree $m$ is divisible. As $m$ should be a prime for cryptographic applications [15,18], the conventional normal basis multiplier cannot be scaled by any digit width for implementing the digit-serial architectures. In this paper we present a novel digit-serial GNB multiplier which could be scaled by any digit width $d$ independent of the degree $m$.

We utilize the Toeplitz matrix-vector representation to derive the GNB multiplication algorithm. It is shown that the proposed GNB multiplication can be realized through block Toeplitz matrix-vector products, which is suitable for implementing digit-serial multipliers. We have proposed a digit-serial systolic multiplier using Toeplitz multiplier as processing elements. The analytical results reveal that, if the digit size $d$ is not smaller than 8 bits, then the proposed architecture has lower area-time complexity than traditional digit-serial systolic PB multipliers. And the area complexity of the proposed architectures are lower than existing digit-serial multipliers.

II. GAUSSIAN NORMAL BASIS MULTIPLICATION

The finite field GF($2^m$) is commonly represented as a vector space of dimension $m$ over GF(2). A set $N = \{\alpha, \alpha^2, \ldots, \alpha^{2^m-1}\}$ is the normal basis of GF($2^m$), and $\alpha$ is called the normal element of GF($2^m$). Let an element $A \in$GF($2^m$) be given by $A = \sum_{i=0}^{m-1} a_i \alpha^i$, where $a_i \in$GF(2), $0 \leq i \leq m-1$, denotes the $i$th coordinate of $A$. In the hardware implementation, squaring is performed by a cyclic shift of the binary representation. The multiplication of elements in GF($2^m$) is uniquely determined by the $m$ cross products $\alpha \alpha^2 = \sum_{i=0}^{m-1} \mu_{ij} \alpha^{2i}, \mu_{ij} \in$GF(2), where $M = \{\mu_{ij}\}$ is called a multiplication matrix. Let $A = (a_0, a_1, \ldots, a_{m-1})$ and $B = (b_0, b_1, \ldots, b_{m-1})$ indicate two normal basis elements in GF($2^m$), and $C = (c_0, c_1, \ldots, c_{m-1}) \in$GF($2^m$) represents their product, i.e., $C = AB$. Coordinate $c_i$ of $C$ can then be represented by $c_i = A^{i-1} \cdot M \cdot (B^{i-1})^T$, $0 \leq i \leq m-1$, where $A^{i-1}$ denotes a left cyclic shift of the element $A$ by $i$ positions.

Definition 1: [15]. Let $p = mt + 1$ represent a prime number, and $gcd(mt/k, m) = 1$, where $k$ denotes the multiplicative order of 2 module $p$. Let $\gamma$ be a primitive $p$ root of unity.
The type-$t$ GNB is employed by $\alpha = \gamma + \gamma^{2^m} + \cdots + \gamma^{2^{m(t-1)}}$ to generate a normal basis $N$ for $\text{GF}(2^m)$ over $\text{GF}(2)$.

Note that GNBs exist for $\text{GF}(2^m)$ whenever $m$ is not divisible by 8. According to Definition 1, an arbitrary element $A$ of $\text{GF}(2^m)$ can be given by

$$A = \sum_{i=0}^{m-1} a_i \gamma^i = \sum_{i=0}^{m-1} a_i (\gamma + \gamma^{2^m} + \cdots + \gamma^{2^{m(t-1)}})^i \tag{1}$$

Since $\gamma$ is a primitive $p$ root of unity, we have

$$\gamma^i = \begin{cases} 
\gamma^{i+1} & \text{if } i \neq p - 1 \\
1 & \text{if } i = p - 1 
\end{cases} \tag{2}$$

Thus, the field element $A$ in (1) can be alternated to define the following formula:

$$A = \sum_{i=0}^{p-1} a_{F(i)} \gamma^i \tag{3}$$

where

$$F(2^i 2^j \mod p) = i \text{ for } 0 \leq i \leq m - 1 \text{ and } 0 \leq j \leq t - 1. \tag{4}$$

Using this representation, the coefficients of $A$ are replicated by $t$-term coefficients of the original normal basis element $A = (a_0, a_1, \ldots, a_{m-1})$ if the field element $A$ are represented by a type-$t$ normal basis of $\text{GF}(2^m)$. Thus, by using the function $F(2^i 2^j \mod p) = i$, the field element $A = (a_{F(1)}, a_{F(2)}, \ldots, a_{F(p-1)})$ with the redundant representation can be transformed into the following representation, $A = (a_0, a_0, a_1, \ldots, a_1, \ldots, a_{m-1}, \ldots, a_{m-1})$, so that the redundant basis element is easily converted into a normal basis element, without any extra hardware.

Let $A$ and $B$ indicate two normal basis elements in $\text{GF}(2^m)$, and $C$ represents their product, i.e., $C = AB$. Assuming the elements $A$ and $B$ in GNB representation if the sequence $F(1), F(2), \ldots, F(p - 1)$ is defined by (4), and $p = m + 1$ is prime, the first coordinate of the product word $C$ can be calculated using the following formula [15]:

$$c_0 = G(A, B) = \sum_{j=0}^{m-1} a_{F(j+1)} b_{F(p-j)} \tag{5}$$

As given by (5), $c_0$ is the first coordinate of the product $C = AB$. $c_1$ is the first coordinate of the product $C^2 = A^2 B^2$ which can be obtained from the formula for $c_0$ by cycling the subscripts modulo $m$. Therefore, the GNB multiplication algorithm for even values of $t$ is modified as follows.

Algorithm 1: (GNB multiplication for even $t$)[15]

Input: $A = (a_0, a_1, \ldots, a_{m-1})$ and $B = (b_0, b_1, \ldots, b_{m-1}) \in \text{GF}(2^m)$

Output: $C = (c_0, c_1, \ldots, c_{m-1}) = AB$

1. initial : $c_0 = 0$
2. for $k = 0$ to $m - 1$
3. $c_k = G(A, B)$
4. $A = A^{(-1)}$ and $B = B^{(-1)}$
5. }

III. PROPOSED DIGIT-SERIAL GNB MULTIPLIER USING TOEPLITZ MATRIX-VECTOR REPRESENTATION

In this section, the GNB multiplication algorithm uses the Toeplitz matrix-vector representation to construct a novel multiplier.

Let $A = \sum_{i=0}^{p-1} a_{F(i)} \gamma^i$ and $B = \sum_{i=0}^{p-1} b_{F(i)} \gamma^i$ with $a_{F(0)} = b_{F(0)} = 0$ and $a_{F(i)}, b_{F(i)} \in \text{GF}(2^m)$ for $1 \leq i \leq p - 1$ denote two type-$t$ GNB elements in $\text{GF}(2^m)$, where $\gamma$ represents the root of $x^p + 1$. Selecting a digital size of $d$-bit, and $n = \lfloor p/d \rfloor$, the element $B$ can thus be expressed as $B = \sum_{i=0}^{d-1} b_{F(id+j)} \gamma^j$. The multiplication of $A$ and $B$ can then be obtained by

$$C = AB \mod (x^p + 1) = \sum_{i=0}^{n-1} A^{(id)} B_i \tag{6}$$

where

$$A^{(i)} = A \gamma^i \mod (x^p + 1) = \sum_{i=0}^{n-1} a_{F(j-i)} \gamma^j. \tag{7}$$

Each partial product $C^{(i)} = A^{(id)} B_i$ can also be calculated by

$$C^{(i)} = \sum_{k=0}^{d-1} A^{(id)} b_{F(id+k)} \gamma^k = \sum_{k=0}^{d-1} A^{(id+k)} b_{F(id+k)} \tag{8}$$

Next, let $A^{(id+k)}$ be split into $n$ subwords, we have

$$A^{(id+k)} = \sum_{j=0}^{n-1} A_j^{(id+k)} \gamma^{dj} \tag{9}$$

where

$$A_j^{(id+k)} = \sum_{w=0}^{d-1} a_{F(id+w-(id+k))} \gamma^w \tag{10}$$

Thus, (8) can be re-expressed by

$$C^{(i)} = \sum_{k=0}^{d-1} \sum_{j=0}^{d-1-n} A_j^{(id+k)} \gamma^{dj} b_{F(id+k)} \tag{11}$$

where

$$C_j^{(i)} = \sum_{k=0}^{d-1} \sum_{w=0}^{d-1-n} A_{F(id+w-(id+k))} b_{F(id+k)} \tag{12}$$
Using the matrix-vector representation, $C_j^{(i)}$ in (12) can be obtained as

$$
\begin{bmatrix}
C_{jd}^{(i)} \\
C_{jd+1}^{(i)} \\
\vdots \\
C_{jd+d-1}^{(i)}
\end{bmatrix} = 
\begin{bmatrix}
aF(jd-id) & aF(jd-id-1) \\
aF(jd-id+1) & aF(jd-id) \\
\vdots & \vdots \\
aF((j+1)d-id-1) & aF((j+1)d-id-2)
\end{bmatrix}
\begin{bmatrix}
bF(id) \\
bF(id+1) \\
\vdots \\
bF(id+d-1)
\end{bmatrix}
$$

Observing the structure of (13), the result $C_j^{(i)}$ is formed by the Toeplitz matrix-vector multiplication. For clarity, the Toeplitz matrix-vector multiplication in (13) is represented by

$$
C_j^{(i)} = t_{k-1-i+j} \otimes B_i
$$

Adopting the GNB representation to convert the normal basis, Wu et al. in [16] showed that the minimum representation in (3) has a Hamming weight equal to or less than $mt/2$ if $m$ is even, and equal to $(mt - t + 2)/2$ if $m$ is odd. For example, let $A = (a_0, a_1, a_2, a_3, a_4)$ be a type-2 GNB element of $\text{GF}(2^5)$, and let $a = a_0 + a_1\gamma + a_2\gamma^2 + a_3\gamma^3 + a_4\gamma^4$ generate the GNB. Applying (4), the field element $A$ can be represented by $A = a_0\gamma + a_1\gamma^2 + a_2\gamma^3 + a_3\gamma^4 + a_4\gamma^5 + a_5\gamma^6$. We find that $A = a_0\gamma + a_1\gamma^2 + a_2\gamma^3 + a_3\gamma^4 + a_4\gamma^5$ is enough for the conversion from the GNB to the NB. Therefore, we can assume that the coordinate numbers of the partial product $C_j^{(i)}$ in (11) are selected by $q = dk$ consecutive coordinates to satisfy the corresponding normal basis representation, where $q \geq mt/2$ if $m$ is even, and $q \geq (mt - t + 2)/2$ if $m$ is odd. Employing the Toeplitz matrix-vector representation, $C_j^{(i)}$ can then be re-expressed as follows:

$$
C_j^{(i)} = \sum_{i=0}^{k-1} t_{k-1-i+j} \otimes B_i^j
$$

Applying (14) to (6), the product $C$ can be obtained by

$$
C = \sum_{i=0}^{n-1} C_j^{(i)} = \sum_{i=0}^{n-1} \sum_{j=0}^{k-1} t_{k-1-j+i} \otimes B_i^j \gamma^{jd} = \sum_{i=0}^{n-1} C_j \gamma^{jd}
$$

Using the matrix-vector representation, the product $C$ can also be written as

$$
\begin{bmatrix}
C_0 \\
C_1 \\
\vdots \\
C_{k-1}
\end{bmatrix} = 
\begin{bmatrix}
t_{k-1} & t_k & \cdots & t_{k+n-2} \\
t_{k-2} & t_k & \cdots & t_{k+n-3} \\
\vdots & \vdots & \ddots & \vdots \\
t_0 & t_1 & \cdots & t_{n-1}
\end{bmatrix}
\begin{bmatrix}
B_0 \\
B_1 \\
\vdots \\
B_{n-1}
\end{bmatrix}
$$

Note that each entity in (16) performs a $d \times d$ Toeplitz matrix-vector multiplication. We can find that the GNB multiplication in (16) is formed by the structure of a block Toeplitz matrix-vector. As given by (16), the block Toeplitz matrix-vector includes $(k + n - 1)$ entities of $t_i$ and we assume that the GNB multiplication is based on $t_i$, for $0 \leq i \leq k + n - 2$, to translate $(k + n - 1)$ terms $t_iB_i$, such as

$$
C = \sum_{i=0}^{n+k-2} t_iB_i
$$

where $B_i$ is defined as

$$
B_i = \begin{cases} 
(0, \ldots, 0, B_0, \ldots, B_1) & \text{if } 0 \leq i \leq k - 1 \\
(B_{i-k}, B_{i-k+1}, \ldots, B_i) & \text{if } k \leq i \leq n - 1 \\
(B_{i-k}, \ldots, B_{i-1}, 0, \ldots, 0) & \text{if } 0 \leq i \leq k - 1
\end{cases}
$$

Assume that $B_i = (b_{i,0}, b_{i,0}\gamma^d + \cdots + b_{i,k-1}\gamma^{d(k-1)})$ is represented by the polynomial representation. From (17), for the polynomials $B_i$ and $B_{i-1}$ we can obtain the relationship:

$$
B_i = \begin{cases} 
(b_{i,0}, B_{i-1})\gamma^d + B_{i-1} & \text{if } 0 \leq i \leq n - 1 \\
(b_{i,0}, B_{i-1}) & \text{if } n \leq i \leq n + k - 2
\end{cases}
$$

The proposed digit-serial systolic multiplier based on (14)-(18), is shown in Fig.1, which consists of $k$ processing elements (PE), where each PE performs one Toeplitz multiplication, one SW, three registers and one adder, as shown in Fig.2. During each cycle, the $j$th PE performs the computation:

$$
C_{i,j} = C_{i-1,j} + t_i b_{i,j}
$$

where $i$ denote clock cycle. In Fig.1, the signal $t_i$ is fed to all the PEs the proposed digit-serial multiplier simultaneously to get $t_i B_i$. The signal $ctr$ is used to control the output of the computational results in the PE. At the $i$th clock cycles, the $j$th PE in Fig.2 computes the product of $t_i$ and $b_{i,j}$ and adds that with the previous result. The SW in Fig. 2 is based on the signal $ctr$ to have two operations: (a) if $ctr = 0$, the result $C_{i,j}$ is stored in the register $L_2$; (2) if $ctr = 1$, the result $C_{i,j}$ is passed through SW to output the array multiplier. After the $(n - 1)$th cycle, the output of the PE$_{k-1}$ is obtained by the result $C_{i,k-1}$. After the $n$th cycle, the output of the PE$_{k-2}$ is obtained by the result $C_{i,k-2}$, and so on. According to (18), the latency of the proposed digit-serial multiplier demands $(n + k - 1)$ clock cycles.

Table 1 shows a comparison between the proposed digit-serial multiplier and the existing digit-serial multiplier in Reyhani-Masoleh and Hasan (RMH multiplier) [14]. The latency of RMH multiplier is $[m/d]$ cycles, while the proposed digit-serial multiplier has $n + k - 1$ cycles. For comparing the time-area complexity, the transistor count based on the standard CMOS VLSI realization is employed for comparison. Therefore, some basic logic gates: 2-input XOR, 2-input AND, 1×2 SW, MUX and 1-bit latch are assumed to be composed of.

![Fig. 1. The proposed digit-serial systolic multiplier over GF(2^m)](image-url)
6, 6, 6, and 8 transistors, respectively [17]. When \( m = 233 \) and \( d = 28 \), we have \( k = 9 \). The proposed multiplier is 88830 transistor count, while digit-serial multiplier [11] demands 152996 transistor count. In Table 2, the proposed scalable multiplier has lower time-area complexity than that of digit-serial multiplier [14] for digital size \( d \geq 8 \).

IV. CONCLUSIONS

We propose here a Toeplitz matrix-vector representation to derive novel multiplication algorithm for the GNB of GF(2^m). Using the proposed algorithm, the GNB multiplication is realized by block Toeplitz matrix-vector structure, and is suitable for implementing that in digit-serial architectures. Differing from other digit-serial multipliers, our proposed multipliers use systematic unfolding technique to achieve digit-serial multiplication architectures. For cryptographic applications, the field of degree \( m \) should be a prime number. The major feature of our multiplier is that it can realize digit-in digit-out digit-serial architectures. And, the selected digit-width \( d \) may or may not divide the field degree \( m \). Analytical results show that, in the finite field GF(2^{233}), the proposed multipliers have lower area-time complexity than existing digit-serial multipliers for polynomial basis and normal basis of GF(2^m). The proposed digit-serial multiplier is thus highly flexible, and are suitable for implementing all type-I GNB multiplications. Finally, the proposed architecture could have a better trade-off between area and speed for implementing cryptographic schemes in embedded systems.

REFERENCES