

Subquadratic Space Complexity Multiplier for $GF(2^n)$ Using Type 4 Gaussian Normal Bases

Sun-Mi Park, Dowon Hong, and Changho Seo

Subquadratic space complexity multipliers for optimal normal bases (ONBs) have been proposed for practical applications. However, for the Gaussian normal basis (GNB) of type $t > 2$ as well as the normal basis (NB), there is no known subquadratic space complexity multiplier. In this paper, we propose the first subquadratic space complexity multipliers for the type 4 GNB. The idea is based on the fact that the finite field $GF(2^n)$ with the type 4 GNB can be embedded into fields with an ONB.

Keywords: Finite field arithmetic, subquadratic space complexity multiplier, normal basis, Gaussian normal basis.

I. Introduction

Finite field $GF(2^n)$ arithmetic is important in coding theory, computer algebra, and public key cryptography, such as elliptic curve cryptography, pairing-based cryptography, and ElGamal cryptography [1], [2]. The choice of basis for $GF(2^n)$ over $GF(2)$ has a great influence on the efficiency of field arithmetic. The polynomial basis and the normal basis (NB) are frequently used to represent a field element in $GF(2^n)$. The merit of the NB is that the squaring of a field element can be performed by a cyclic shift of the coordinates of the element and so it is free in hardware. However, a multiplier for the NB is generally less efficient than the other bases. Therefore, a special class of NBs, such as the optimal normal basis (ONB) or the Gaussian normal basis (GNB), has been considered (it is known that type 1 and type 2 GNBS are the same as type I and type II ONBs, respectively).

Until now, most of the known bit-parallel multipliers using the NB have quadratic space complexity (that is, the number of 2-input AND and XOR gates is greater than or equal to $O(n^2)$). However, for such practical applications as cryptographic applications, subquadratic space complexity multipliers have been required. In recent years, subquadratic space complexity multipliers for ONBs (that is, GNBS of types 1 and 2) were proposed. The researchers in [3] presented the first subquadratic space complexity multiplier for the type I ONB using the Karatsuba algorithm. The researchers in [4] proposed subquadratic space complexity multipliers for types I and II ONBs using the Toeplitz matrix-vector product (TMVP) scheme. However, for the GNB of type $t > 2$ as well as the NB, there is no known subquadratic space complexity design of multiplier. In this paper, we present subquadratic space complexity multipliers for the type 4 GNB. We first show that

Manuscript received Sept. 3, 2012; revised Nov. 8, 2012; accepted Dec. 5, 2012.

This work was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology, Rep. of Korea (2011-0029927) and Basic Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0021531).

Sun-Mi Park (phone: +82 41 850 8560, smpark@kongju.ac.kr), Dowon Hong (corresponding author, dwhong@kongju.ac.kr), and Changho Seo (chseo@kongju.ac.kr) are with the Department of Applied Mathematics, Kongju National University, Gongju, Rep. of Korea.

<http://dx.doi.org/10.4218/etrij.13.0112.0596>

the finite field $GF(2^n)$ with a type 4 GNB can be embedded into fields with an ONB. Then, to derive subquadratic space complexity multipliers for the type 4 GNB, we use the subquadratic space complexity multipliers for ONBs. As far as we know, it is the first time that a subquadratic space complexity multiplier for the type 4 GNB has been proposed. Furthermore, we present asymptotic complexities of proposed multipliers for the type 4 GNB and compare those with known multipliers presented in the literature.

II. Preliminaries

1. Gaussian Normal Basis

In this section, we give the definition of GNB. For a given NB $\{\beta, \beta^2, \dots, \beta^{2^{n-1}}\}$ of $GF(2^n)$ over $GF(2)$, the element β is called a “normal element” and any element A in $GF(2^n)$ can be represented as

$$A = \sum_{i=0}^{n-1} a_i \beta^{2^i} = a_0 \beta + a_1 \beta^2 + \dots + a_{n-1} \beta^{2^{n-1}},$$

where $a_i \in GF(2)$ for $0 \leq i < n$. The squaring of A is simply implemented by the right cyclic shift:

$$A^2 = \sum_{i=0}^{n-1} a_i \beta^{2^{i+1}} = a_{n-1} \beta + a_0 \beta^2 + a_1 \beta^{2^2} + \dots + a_{n-2} \beta^{2^{n-1}}.$$

Definition 1 [5]. Let $p = n \cdot t + 1$ be a prime. Let $\tau \in Z_p^*$ be an element of order t and k the multiplicative order of 2 in Z_p^* . If $\gcd(n \cdot t / k, n) = 1$, then

$$\beta := \gamma + \gamma^\tau + \gamma^{\tau^2} + \dots + \gamma^{\tau^{t-1}}$$

is a normal element for $GF(2^n)$ over $GF(2)$, where $\gamma \in GF(2^{p-1})$ is a primitive p -th root of unity. The NB $\{\beta, \beta^2, \dots, \beta^{2^{n-1}}\}$ is called a GNB of type t .

It is known that the GNBs for $GF(2^n)$ always exist when n is not divisible by 8 [5].

2. Toeplitz Matrix-Vector Product

In this subsection, we introduce TMVP schemes and present their asymptotic complexities, shown in [6]. Elements of the matrix are in $GF(2)$.

Definition 2. An $n \times n$ Toeplitz matrix is a matrix $(m_{i,j})_{1 \leq i, j \leq n}$ with the property that $m_{i,j} = m_{i-1,j-1}$ for all $1 < i, j \leq n$.

Let T be an $n \times n$ Toeplitz matrix and V an $n \times 1$ vector. If $2|n$, Fan and Hasan [6] use the following two-way split of matrix T and vector V to compute the TMVP TV :

$$TV = \begin{pmatrix} T_1 & T_0 \\ T_2 & T_1 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \end{pmatrix} = \begin{pmatrix} P_0 + P_2 \\ P_1 + P_2 \end{pmatrix}, \quad (1)$$

where T_0 , T_1 , and T_2 are $(n/2) \times (n/2)$ Toeplitz matrices, V_0

and V_1 are $(n/2) \times 1$ vectors, $P_0 = (T_0 + T_1)V_1$, $P_1 = (T_1 + T_2)V_0$, and $P_2 = T_1(V_0 + V_1)$. Then, the TMVP TV of size n is reduced to three TMVPs, that is, P_0 , P_1 , and P_2 of sizes $n/2$.

We use the notations $T^\oplus(n)$, $T^\otimes(n)$, $D_T^\oplus(n)$, and $D_T^\otimes(n)$ to respectively denote the number of XOR gates and AND gates and the time delays by XOR gates and AND gates to compute the TMVP TV of size n . According to Section 2.1 of [6], we have the following recurrence relations

$$\begin{cases} T^\oplus(n) = 3T^\oplus(n/2) + 3n - 1, \\ T^\otimes(n) = 3T^\otimes(n/2), \\ D_T^\oplus(n) = D_T^\oplus(n/2) + 2T_X, \\ D_T^\otimes(n) = D_T^\otimes(n/2), \end{cases} \quad (2)$$

where T_X is the delay due to one XOR gate. In the case of $n = 2^i$ ($i > 0$), the two-way split (1) may be used recursively to compute the TMVP TV . Using the recurrence relations (2), the complexities for the TMVP TV are computed in Section 2.1 of [6]. We summarize its complexities in Table 1, where T_A is the delay due to one AND gate.

If $3|n$, a three-way split approach is used:

$$TV = \begin{pmatrix} T_2 & T_1 & T_0 \\ T_3 & T_2 & T_1 \\ T_4 & T_3 & T_2 \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} P_0 + P_3 + P_4 \\ P_1 + P_3 + P_5 \\ P_2 + P_4 + P_5 \end{pmatrix}, \quad (3)$$

where T_i ($0 \leq i \leq 4$) are $(n/3) \times (n/3)$ Toeplitz matrices and V_0 , V_1 , V_2 are $(n/3) \times 1$ vectors,

$$\begin{aligned} P_0 &= (T_0 + T_1 + T_2)V_2, & P_3 &= T_1(V_1 + V_2), \\ P_1 &= (T_1 + T_2 + T_3)V_1, & P_4 &= T_2(V_0 + V_2), \\ P_2 &= (T_2 + T_3 + T_4)V_0, & P_5 &= T_3(V_0 + V_1). \end{aligned}$$

Then, the TMVP TV of size n is reduced to six TMVPs, that is, P_0, \dots, P_5 of sizes $n/3$.

In the case of $n = 3^i$ ($i > 0$), we compute TV using the three-way split (3) recursively. Similarly to the case $n=2^i$, the complexities for computing the TMVP TV are derived in Section 2.2 of [6] and given in Table 1.

3. Subquadratic Space Complexity Multiplier for ONBs

In this subsection, we recall the subquadratic space complexity multipliers for type I and type II ONBs, proposed in [4].

We use the notations $S^\oplus(n)$ and $S^\otimes(n)$ to denote the space complexity of a $GF(2^n)$ multiplier, that is, the number of XOR gates and the number of AND gates, respectively. $D_S^\oplus(n)$ and $D_S^\otimes(n)$ denote its time delay due to XOR gates and AND gates, respectively.

Let $X := \{x_1, x_2, \dots, x_n\}$ be a basis of $GF(2^n)$ over $GF(2)$.

Table 1. Complexities for TMVP TV in case that $n = b^i$ ($i > 0$).

b	$T^\oplus(n)$	$T^\oplus(n)$	$D_T^\oplus(n) + D_T^\otimes(n)$
2	$n^{\log_2 3}$	$5.5n^{\log_2 3} - 6n + 0.5$	$(2 \log_2 n)T_X + T_A$
3	$n^{\log_3 6}$	$4.8n^{\log_3 6} - 5n + 0.2$	$(3 \log_3 n)T_X + T_A$

For elements A and B in $GF(2^n)$, $A = \sum_{i=1}^n a_i x_i$ and $B = \sum_{i=1}^n b_i x_i$, where $a_i, b_i \in GF(2)$ for $1 \leq i \leq n$. Then,

$$\begin{aligned} AB &= (\sum_{i=1}^n a_i x_i)B \\ &= [x_1 B, x_2 B, \dots, x_n B][a_1, a_2, \dots, a_n]^t \\ &= [x_1, x_2, \dots, x_n]Z[a_1, a_2, \dots, a_n]^t, \end{aligned} \quad (4)$$

where Z is an $n \times n$ matrix. To compute the product AB of A and B , it suffices to compute a matrix-vector product $Z[a_1, a_2, \dots, a_n]^t$.

A. Type I ONB Multiplier

Assume that $GF(2^n)$ has a type I ONB (that is, type 1 GNB) $M := \{\gamma, \gamma^2, \dots, \gamma^{2^{n-1}}\}$, where γ is an $(n+1)$ th primitive root of unity. According to subsection III.A of [4], the set $X := \{\gamma, \gamma^2, \dots, \gamma^n\}$ is also a basis for $GF(2^n)$ obtained by permuting the NB M . Moreover, the matrix Z in (4) corresponding to this basis X has the form $Z = Z_1 + Z_2$, where

$$Z_1 = \begin{pmatrix} 0 & b_n & b_{n-1} & \cdots & b_3 & b_2 \\ b_1 & 0 & b_n & \cdots & b_4 & b_3 \\ b_2 & b_1 & 0 & \cdots & b_5 & b_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{n-2} & b_{n-3} & b_{n-4} & \cdots & 0 & b_n \\ b_{n-1} & b_{n-2} & b_{n-3} & \cdots & b_1 & 0 \end{pmatrix},$$

$$Z_2 = \begin{pmatrix} b_n & b_{n-1} & b_{n-2} & \cdots & b_2 & b_1 \\ b_n & b_{n-1} & b_{n-2} & \cdots & b_2 & b_1 \\ b_n & b_{n-1} & b_{n-2} & \cdots & b_2 & b_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ b_n & b_{n-1} & b_{n-2} & \cdots & b_2 & b_1 \\ b_n & b_{n-1} & b_{n-2} & \cdots & b_2 & b_1 \end{pmatrix}.$$

Hence, the matrix-vector product $Z[a_1, a_2, \dots, a_n]^t$ may be computed via

$$Z[a_1, a_2, \dots, a_n]^t = Z_1[a_1, a_2, \dots, a_n]^t + Z_2[a_1, a_2, \dots, a_n]^t.$$

The computation of $Z_2[a_1, a_2, \dots, a_n]^t$ requires n AND gates and $n-1$ XOR gates. The complexities of the TMVP $Z_1[a_1, a_2, \dots, a_n]^t$ are $T^\oplus(n)$ AND gates, $T^\oplus(n)$ XOR gates, and $D_T^\oplus(n) + D_T^\otimes(n)$ delay, which are given in Table 1. Finally, we add two vectors, $Z_1[a_1, a_2, \dots, a_n]^t$ and $Z_2[a_1, a_2, \dots, a_n]^t$. As a result, the complexities to

Table 2. Complexities of ONB multipliers for $n = b^i$ ($i > 1$).

ONB	b	# AND	# XOR	Time delay
Type I	2	$n^{\log_2 3} + n$	$5.5n^{\log_2 3} - 4n - 0.5$	$(2 \log_2 n + 1)T_X + T_A$
	3	$n^{\log_3 6} + n$	$4.8n^{\log_3 6} - 3n - 0.8$	$(3 \log_3 n + 1)T_X + T_A$
Type II	2	$2n^{\log_2 3}$	$11n^{\log_2 3} - 12n + 1$	$(2 \log_2 n + 1)T_X + T_A$
	3	$2n^{\log_3 6}$	$9.6n^{\log_3 6} - 10n + 0.4$	$(3 \log_3 n + 1)T_X + T_A$

compute the matrix-vector product $Z[a_1, a_2, \dots, a_n]^t$ are as follows.

$$\begin{cases} S^\oplus(n) = T^\oplus(n) + (n-1) + n, \\ S^\otimes(n) = T^\otimes(n) + n, \\ D_S^\oplus(n) = D_T^\oplus(n) + T_X, \\ D_S^\otimes(n) = D_T^\otimes(n). \end{cases} \quad (5)$$

If $n = 2^i$, then we have

$$\begin{cases} S^\oplus(n) = T^\oplus(n) + 2n - 1 = 5.5n^{\log_2 3} - 4n - 0.5, \\ S^\otimes(n) = T^\otimes(n) + n = n^{\log_2 3} + n, \\ D_S^\oplus(n) = D_T^\oplus(n) + T_X = (2 \log_2 n + 1)T_X, \\ D_S^\otimes(n) = D_T^\otimes(n) = T_A \end{cases}$$

using Table 1. The case $n=3^i$ is dealt with similarly. The complexities for the type I ONB multiplier are given in Table 2.

B. Type II ONB Multiplier

Let $GF(2^n)$ have a type II ONB (that is, type 2 GNB) $N := \{\beta, \beta^2, \dots, \beta^{2^{n-1}}\}$, where $\beta := \gamma + \gamma^{-1}$ and $\gamma \in GF(2^{2n})$ is a primitive $(2n+1)$ th root of unity. Let $\beta_i := \gamma^i + \gamma^{-i}$ for $1 \leq i < n$ and $X := \{\beta_1, \beta_2, \dots, \beta_n\}$. According to subsection III.B of [4], the set X is equal to the set N , and so X is a basis for $GF(2^n)$. Also, the matrix Z in (4) corresponding to the basis X can be decomposed as the summation of two matrices, that is, $Z = Z_1 + Z_2$, where

$$Z_1 = \begin{pmatrix} b_2 & b_3 & \cdots & b_{n-1} & b_n & b_n \\ b_3 & b_4 & \cdots & b_n & b_n & b_{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ b_{n-1} & b_n & \cdots & b_5 & b_4 & b_3 \\ b_n & b_n & \cdots & b_4 & b_3 & b_2 \\ b_n & b_{n-1} & \cdots & b_3 & b_2 & b_1 \end{pmatrix},$$

$$Z_2 = \begin{pmatrix} 0 & b_1 & \cdots & b_{n-3} & b_{n-2} & b_{n-1} \\ b_1 & 0 & \cdots & b_{n-4} & b_{n-3} & b_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ b_{n-3} & b_{n-4} & \cdots & 0 & b_1 & b_2 \\ b_{n-2} & b_{n-3} & \cdots & b_1 & 0 & b_1 \\ b_{n-1} & b_{n-2} & \cdots & b_2 & b_1 & 0 \end{pmatrix}.$$

The matrix Z_1 is a Hankel matrix, that is, entries at (i, j) and $(i-1, j+1)$ are equal. The Hankel matrix-vector product can be changed into the TMVP without using any logic gates (one first exchanges the i -th column and $(n+1-i)$ th column of Z_1 for $1 \leq i \leq n/2$ and reverses the column vector $[a_1, a_2, \dots, a_n]^t$). Therefore, its computation requires the same complexity with the TMVP. The matrix Z_2 is a Toeplitz matrix. Therefore, two TMVPs and the next addition of two vectors are used to compute the matrix vector product $Z[a_1, a_2, \dots, a_n]^t$. Thus, we have the following:

$$\begin{cases} S^\oplus(n) = 2T^\oplus(n) + n, \\ S^\otimes(n) = 2T^\otimes(n), \\ D_S^\oplus(n) = D_T^\oplus(n) + T_X, \\ D_S^\otimes(n) = D_T^\otimes(n). \end{cases} \quad (6)$$

Table 2 gives the complexities for the type II ONB multiplier in the case of $n=2^i$ or $n=3^i$ using Table 1.

III. Multiplier for GNB of Type 4

In this section, we propose subquadratic space complexity multipliers for the type 4 GNB. For the rest of the paper, we assume that $GF(2^n)$ has a type 4 GNB and so $p = 4 \cdot t + 1$ is a prime. The following lemma was proven in [7].

Lemma 1. (Lemma 1, [7]) Suppose that n is odd and $GF(2^n)$ has a type 4 GNB. Then, the multiplicative order k of 2 in Z_{4n+1}^* is $4n$, and so $GF(2^{4n})$ has a type I ONB.

Here, we remove the constraint of Lemma 1 that n is odd. Moreover, we prove that if $GF(2^n)$ has a type 4 GNB, then $GF(2^{2n})$ has a type II ONB.

Lemma 2. Assume that $GF(2^n)$ has a GNB of type 4. Then, n is odd.

Proof. Assume that $GF(2^n)$ has a GNB of type 4. From Definition 1, $p := 4 \cdot n + 1$ is a prime and $\gcd(4n/k, n) = 1$, where k is the multiplicative order of 2 in Z_p^* . The condition $\gcd(4n/k, n) = 1$ implies that $k \in \{n, 2n, 4n\}$. If n is even, $p \equiv 1 \pmod{8}$ and 2 is quadratic residue mod p . Hence, $k \neq 4n$, and so 2 divides $\gcd(4n/k, n)$, which leads to the contradiction of the condition $\gcd(4n/k, n) = 1$. So, n must be odd. \square

Lemma 3. If $GF(2^n)$ has a type 4 GNB, then the finite field $GF(2^{2n})$ has a type II ONB.

Proof. By Lemmas 1 and 2, if $GF(2^n)$ has a GNB of type 4, then the multiplicative order k of 2 in Z_{4n+1}^* is $4n$. So, $p := 4 \cdot n + 1 = 2 \cdot 2n + 1$ is prime and $\gcd(4n/k, 2n) = 1$. Therefore, by Definition 1, the finite field $GF(2^{2n})$ has a type II ONB. \square

By Lemmas 1, 2, and 3, the field $GF(2^n)$ with a type 4 GNB can be embedded into finite fields $GF(2^{4n})$ and $GF(2^{2n})$ with

ONBs.

1. Construction

We would like to consider the embedding of $GF(2^n)$ into $GF(2^{4n})$ or $GF(2^{2n})$ in more detail. Let γ be a p -th root of unity in $GF(2^{p-1}) = GF(2^{4n})$. Then, γ is a normal element of a type I ONB for $GF(2^{4n})$ over $GF(2)$ by Definition 1. Note that the multiplicative order of 2 in $Z_p^* = Z_{4n+1}^*$ is $4n$ by Lemmas 1 and 2, and so the multiplicative order of 2^n in Z_p^* is 4. Therefore, by Definition 1,

$$\begin{aligned} \beta &:= \gamma + \gamma^{-1} = \gamma + \gamma^{2^{2n}} \quad \text{and} \\ \beta &:= \gamma + \gamma^{2^n} + \gamma^{2^{2n}} + \gamma^{2^{3n}} = \beta_0 + \beta_0^{2^n} \end{aligned} \quad (7)$$

are normal elements of a type II ONB for $GF(2^{2n})$ and a type 4 GNB for $GF(2^n)$ over $GF(2)$, respectively. From (7), an element $D = \sum_{i=0}^{n-1} d_i \beta^{2^i} \in GF(2^n)$ may be considered as an element

$$D = \sum_{i=0}^{n-1} d_i \gamma^{2^i} + \sum_{i=n}^{2n-1} d_{i-n} \gamma^{2^i} + \sum_{i=2n}^{3n-1} d_{i-2n} \gamma^{2^i} + \sum_{i=3n}^{4n-1} d_{i-3n} \gamma^{2^i} \in GF(2^{4n})$$

or

$$D = \sum_{i=0}^{n-1} d_i \beta_0^{2^i} + \sum_{i=n}^{2n-1} d_{i-n} \beta_0^{2^i} \in GF(2^{2n}).$$

Moreover, we have the following.

Lemma 4:

- (1) (Theorem 1 [7]) For an element $A = \sum_{i=0}^{4n-1} a_i \gamma^{2^i} \in GF(2^{4n})$, $A \in GF(2^n)$ if and only if $a_i = a_{i+n} = a_{i+2n} = a_{i+3n}$ for $0 \leq i < n$.
- (2) For an element $B = \sum_{i=0}^{2n-1} b_i \beta_0^{2^i} \in GF(2^{2n})$, $B \in GF(2^n)$ if and only if $b_i = b_{i+n}$ for $0 \leq i < n$.

Proof. (1) is proved in Theorem 1 of [7]. The proof of (2) is similar to that of (1). \square

To derive a subquadratic space complexity multiplier for $GF(2^n)$, we perform the field multiplication in $GF(2^n)$ as follows. For elements $A, B \in GF(2^n)$, we regard A and B as elements in field $GF(2^{4n})$ or field $GF(2^{2n})$ and compute the product C of A and B in $GF(2^{4n})$ or $GF(2^{2n})$ using known subquadratic space complexity multipliers. Finally, we derive the product C as an element in $GF(2^n)$ using one of the following equations (Lemma 4):

$$C = \sum_{i=0}^{4n-1} c_i \gamma^{2^i} = \sum_{i=0}^{n-1} c_i \beta^{2^i},$$

or

$$C = \sum_{i=0}^{2n-1} c_i \beta_0^{2^i} = \sum_{i=0}^{n-1} c_i \beta^{2^i}.$$

Note that the complexities of the multiplier for $GF(2^n)$ equal

Table 3. Complexities of multipliers for $GF(2^n)$ with type 4 GNB using embedding into fields with an ONB.

	n	# AND	# XOR	Time delay
Embedding into $GF(2^{4n})$	2^i	$9n^{\log_2 3} + 4n$	$49.5n^{\log_2 3} - 16n - 0.5$	$(2\log_2 n + 5)T_X + T_A$
	3^i	$9n^{\log_3 6} + 4n$	$43.2n^{\log_3 6} - 7n - 3.2$	$(3\log_3 n + 5)T_X + T_A$
Embedding into $GF(2^{2n})$	2^i	$6n^{\log_2 3}$	$33n^{\log_2 3} - 22n + 1$	$(2\log_2 n + 3)T_X + T_A$
	3^i	$6n^{\log_3 6}$	$28.8n^{\log_3 6} - 16n - 0.8$	$(3\log_3 n + 3)T_X + T_A$

those of the used multiplier for $GF(2^{4n})$ or $GF(2^{2n})$.

2. Complexities for Subquadratic Space Multiplier for Type 4 GNB

In this section, we give asymptotic complexities of the $GF(2^n)$ multiplier for type 4 GNB. That is, we compute the complexities of the multiplier for field $GF(2^{4n})$ or field $GF(2^{2n})$ in which $GF(2^n)$ is embedded. For subquadratic space complexity multipliers for $GF(2^{4n})$ and $GF(2^{2n})$, we use multipliers based on the TMVP scheme in [4].

A. Embedding into $GF(2^{4n})$ with Type I ONB

Let $GF(2^n)$ with a type 4 GNB be embedded into $GF(2^{4n})$ with a type I ONB. To compute the complexities of the $GF(2^n)$ multiplier, we must compute the complexities of the multiplier for $GF(2^{4n})$, that is, $S^\oplus(4n)$, $S^\otimes(4n)$, $D_S^\oplus(4n)$, and $D_S^\otimes(4n)$. Using a two-way split formula (2), we obtain the following equations:

$$\begin{cases} T^\oplus(4n) = 3T^\oplus(2n) + 12n - 1 = 9T^\oplus(n) + 30n - 4, \\ T^\otimes(4n) = 3T^\otimes(2n) = 9T^\otimes(n), \\ D_T^\oplus(4n) = D_T^\oplus(2n) + 2T_X = D_T^\oplus(n) + 4T_X, \\ D_T^\otimes(4n) = D_T^\otimes(2n) = D_T^\otimes(n). \end{cases}$$

Therefore, (5) implies that

$$\begin{cases} S^\oplus(4n) = T^\oplus(4n) + 8n - 1 = 9T^\oplus(n) + 38n - 5, \\ S^\otimes(4n) = T^\otimes(4n) + 4n = 9T^\otimes(n) + 4n, \\ D_S^\oplus(4n) = D_T^\oplus(4n) + T_X = D_T^\oplus(n) + 5T_X, \\ D_S^\otimes(4n) = D_T^\otimes(4n) = D_T^\otimes(n). \end{cases}$$

We obtain the formula for $S^\oplus(4n)$, $S^\otimes(4n)$, $D_S^\oplus(4n)$, and $D_S^\otimes(4n)$ using Table 1. For example, when $n=3^i$, we have

$$\begin{aligned} S^\oplus(4n) &= 9T^\oplus(n) + 38n - 5 \\ &= 9(4.8n^{\log_3 6} - 5n + 0.2) + 38n - 5 \\ &= 43.2n^{\log_3 6} - 7n - 3.2, \\ S^\otimes(4n) &= 9T^\otimes(n) + 4n = 9n^{\log_3 6} + 4n, \\ D_S^\oplus(4n) &= D_T^\oplus(n) + 5T_X = (3\log_3 n + 5)T_X, \\ D_S^\otimes(4n) &= D_T^\otimes(n) = T_A. \end{aligned}$$

The complexities for the $GF(2^{4n})$ multiplier are summarized in Table 3.

B. Embedding into $GF(2^{2n})$ with Type II ONB

Assume that $GF(2^n)$ with a type 4 GNB is embedded into $GF(2^{2n})$ with a type II ONB. We compute the complexities of the multiplier for $GF(2^{2n})$, that is, $S^\oplus(2n)$, $S^\otimes(2n)$, $D_S^\oplus(2n)$, and $D_S^\otimes(2n)$. A two-way split formula (2) implies that

$$\begin{cases} T^\oplus(2n) = 3T^\oplus(n) + 6n - 1, \\ T^\otimes(2n) = 3T^\otimes(n), \\ D_T^\oplus(2n) = D_T^\oplus(n) + 2T_X, \\ D_T^\otimes(2n) = D_T^\otimes(n). \end{cases}$$

From (6), we have

$$\begin{cases} S^\oplus(2n) = 2T^\oplus(2n) + 2n = 6T^\oplus(n) + 14n - 2, \\ S^\otimes(2n) = 2T^\otimes(2n) = 6T^\otimes(n), \\ D_S^\oplus(2n) = D_T^\oplus(2n) + T_X = D_T^\oplus(n) + 3T_X, \\ D_S^\otimes(2n) = D_T^\otimes(2n) = D_T^\otimes(n). \end{cases}$$

In Table 3, we present the complexities for the $GF(2^{2n})$ multiplier, that is, $S^\oplus(2n)$, $S^\otimes(2n)$, $D_S^\oplus(2n)$, and $D_S^\otimes(2n)$, for the case in which $n=2^i$ or $n=3^i$, using Table 1.

Remark 1. In Table 3, we consider the case in which $n=2^i$. However, n is odd according to Lemma 2. To illustrate the asymptotic complexities of the proposed type 4 GNB multiplier, we choose the smallest 2^i that is larger than n instead of choosing n . Then, we may expand the sizes of the matrices and vectors, as shown in Section 2.3 of [6]. Also, for other approaches, see Section 2.3 of [6].

Remark 2. According to Table 3, it is more efficient to use embedding in the field $GF(2^{2n})$ with the type II ONB than $GF(2^{4n})$ with the type I ONB.

IV. Comparison

In this section, we compare the proposed multiplier with other multipliers for $GF(2^n)$ with the type 4 GNB recently published in the literature. Here, we use an embedding of the

Table 4. Complexity comparison of bit-parallel multipliers for $GF(2^n)$ with type 4 GNB.

Multiplier	# AND	# XOR	Time delay
[9]	n^2	$4n^2 - 4n$	$T_A + (2 + \lceil \log_2 n \rceil)T_X$
[10]	n^2	$2.5n^2 - 4.5n$	$T_A + (1 + \lceil \log_2(2n-1) \rceil)T_X$
DLGMP [11], [12] ($d=n$)	n^2	$2.5n^2 - 1.5n$	$T_A + (2 + \lceil \log_2 n \rceil)T_X$
DLGMD [11] ($d=n$)	n^2	$2.5n^2 - 2.5n$	$T_A + (2 + \lceil \log_2 n \rceil)T_X$
[13]	n^2	$\leq 2n^2 - 2n$	$T_A + (2 + \lceil \log_2 n \rceil)T_X$
This paper with [4] ($n=2^i$)	$6n^{\log_2 3}$	$33n^{\log_2 3} - 22n + 1$	$T_A + (3 + 2\log_2 n)T_X$
This paper with [8] ($n=2^i$)	$8n^{\log_2 3} + n$	$20.5n^{\log_2 3} - 15n + 1.5$	$T_A + (3 + 2\log_2 n)T_X$

Table 5. Complexity comparison for type 4 GNB multipliers.

Multiplier	$n = 409$ ($\approx 2^4 \cdot 3^3$)			$n = 573$ ($\approx 2^6 \cdot 3^2$)			$n = 759$ ($\approx 2^8 \cdot 3^1$)		
	#AND	#XOR	Delay	#AND	#XOR	Delay	#AND	#XOR	Delay
[9]	167,281	667,488	$T_A + 11T_X$	328,329	1,311,024	$T_A + 12T_X$	576,081	2,301,288	$T_A + 12T_X$
[10]	167,281	416,362	$T_A + 11T_X$	328,329	818,244	$T_A + 12T_X$	576,081	1,436,787	$T_A + 12T_X$
DLGMP [11], [12]	167,281	417,589	$T_A + 11T_X$	328,329	819,963	$T_A + 12T_X$	576,081	1,439,064	$T_A + 12T_X$
DLGMD [11]	167,281	417,180	$T_A + 11T_X$	328,329	819,390	$T_A + 12T_X$	576,081	1,438,305	$T_A + 12T_X$
[13]	167,281	333,744	$T_A + 11T_X$	328,329	655,512	$T_A + 12T_X$	576,081	1,150,644	$T_A + 12T_X$
This paper with [4]	104,976	507,358	$T_A + 20T_X$	157,464	781,210	$T_A + 21T_X$	236,196	1,223,134	$T_A + 22T_X$
This paper with [8]	140,400	321,936	$T_A + 20T_X$	210,528	493,278	$T_A + 21T_X$	315,696	765,960	$T_A + 22T_X$

field $GF(2^n)$ into the field $GF(2^{2n})$ with the type II ONB (see Remark 2). Many authors have proposed quadratic space complexity multipliers for the type 4 GNB. Their space and time complexities are given in rows 1 through 5 of Table 4. Row 6 of Table 4 presents the space and time complexities of the proposed multiplier, referred to in part B of subsection III.2. It uses a multiplier presented in [4] for the type II ONB, described in part B of subsection II.3. Recently, [8] proposed a subquadratic space complexity multiplier for ONBs that is based on the block recombination of the TMVP given in subsection II.2. The type II ONB multiplier in [8] has a fewer number of XOR gates but larger number of AND gates compared with that presented in [4]. However, the total number of gates (that is, the sum of the number of XOR gates and AND gates) of [8] is fewer than that presented in [4]. The proposed multiplication scheme for $GF(2^n)$ with the type 4 GNB in this paper has no restriction on multipliers for $GF(2^{2n})$ with the type II ONB in which $GF(2^n)$ is embedded. A more efficient type II ONB multiplier induces a more efficient type 4 GNB multiplier. The last row of Table 4 presents the complexities of the type 4 GNB multiplier that uses the type II ONB $GF(2^{2n})$ multiplier presented in [8]. As predicted, it has a fewer number of XOR gates but a larger number of AND gates

compared with the multiplier shown in row 6 of Table 4.

Table 4 shows a comparison of multipliers in terms of the total number of gates (the sum of the number of XOR gates and the number of AND gates). The total number of gates of the multipliers using the proposed scheme, that is, the data shown in rows 6 and 7 of Table 4, is $39n^{\log_2 3} - 22n + 1$ and $28.5n^{\log_2 3} - 14n + 1.5$, respectively. The best known quadratic space complexity multiplier for the type 4 GNB has a $3n^2 - 2n$ ($n^2 + (2n^2 - 2n)$) total number of gates (row 5 of Table 4). It is larger than the total number of gates reflected in rows 6 and 7 of Table 4 if $n \geq 467$ and $n \geq 218$, respectively.

For an odd $n \leq 2,000$, there are 193 values of n for which there exists a type 4 GNB. For only 19 values (about 10%) of the 193 values of n , there exists an ONB. Table 5 shows a comparison of the multipliers in Table 4 for three specific fields, namely, $GF(2^{409})$, $GF(2^{573})$, and $GF(2^{759})$. Those fields have a type 4 GNB but no ONBs. Since neither 409, 573, nor 759 is a power of 2 or 3, we choose sizes $2^k 3^l$, which is close to n , to use the proposed multiplication scheme. Then, we expand the sizes of the matrices and vectors as needed and use a combination of two-way and three-way split approaches (see Remark 1).

V. Conclusion

In this paper, we proposed subquadratic space complexity multipliers for the type 4 GNB. Moreover, we derived the complexities of multipliers for the type 4 GNB and compared those with known multipliers. However, there is no known subquadratic space complexity multiplier for general GNBs except for types 1, 2, and 4.

References

- [1] R. Lidl and H. Niederreiter, *An Introduction to Finite Fields and Their Applications*, Cambridge: Cambridge University Press, 1986.
- [2] A.J. Menezes et al., *Applications of Finite Fields*, Kluwer Academic, 1993.
- [3] M. Leone, “A New Low Complexity Parallel Multiplier for a Class of Finite Fields,” *Proc. 3rd Int. Workshop Cryptographic Hardware and Embedded Systems, LNCS 2162*, 2001, pp. 160-170.
- [4] H. Fan and M.A. Hasan, “Subquadratic Computational Complexity Schemes for Extended Binary Field Multiplication Using Optimal Normal Bases,” *IEEE Trans. Computers*, vol. 56, no. 10, Oct. 2007, pp. 1435-1437.
- [5] D.W. Ash, I.F. Blake, and S.A. Vanstone, “Low Complexity Normal Bases,” *Discrete Appl. Mathematics*, vol. 25, no. 3, 1989, pp. 191-210.
- [6] H. Fan and M.A. Hasan, “A New Approach to Subquadratic Space Complexity Parallel Multipliers for Extended Binary Fields,” *IEEE Trans. Computers*, vol. 56, no. 2, Feb. 2007, pp. 224-233.
- [7] C.H. Kim et al., “Modified Serial Multipliers for Type-IV Gaussian Normal Bases,” *INDOCRYPT, LNCS 3797*, 2005, pp. 375-388.
- [8] J. Adikari et al., “Improved Area-Time Tradeoffs for Field Multiplication Using Optimal Normal Bases,” *IEEE Trans. Computers*, vol. 62, no. 1, Jan. 2013, pp. 193-199.
- [9] L. Gao and G.E. Sobelman, “Improved VLSI Designs for Multiplication and Inversion in $GF(2^M)$ over Normal Bases,” *Proc. 13th Annual IEEE Int. ASIC/SOC Conf.*, 2000, pp. 97-101.
- [10] A. Reyhani-Masoleh and M.A. Hasan, “A New Construction of Massey-Omura Parallel Multiplier over $GF(2^m)$,” *IEEE Trans. Computers*, vol. 51, no. 5, 2002, pp. 511-520.
- [11] A. Reyhani-Masoleh, “Efficient Algorithms and Architectures for Field Multiplication Using Gaussian Normal Bases,” *IEEE Trans. Computers*, vol. 55, no. 1, 2006, pp. 34-47.
- [12] C.H. Kim, S. Kwon, C.P. Hong, “FPGA Implementation of High Performance Elliptic Curve Cryptographic Processor over $GF(2^{163})$,” *J. Syst. Architecture*, vol. 54, no. 10, 2008, pp. 893-900.
- [13] R. Azarderakhsh and A. Reyhani-Masoleh, “A Modified Low Complexity Digit-Level Gaussian Normal Basis Multiplier,” *Proc. 3rd Int. Workshop Arithmetic Finite Fields, LNCS 6087*, 2010, pp. 25-40.



Sun-Mi Park received her BS in mathematics education and her MS and PhD in mathematics, all from Korea University, Seoul, Rep. of Korea, in 1997, 1999, and 2004, respectively. She is currently working as a researcher with the Department of Applied Mathematics at Kongju National University, Gongju, Rep. of Korea.

Her research interests include number theory, algorithms, and architectures for computations in Galois fields.



Dowon Hong received his BS, MS, and PhD in mathematics from Korea University, Seoul, Rep. of Korea, in 1994, 1996, and 2000, respectively. He was a principal member of the engineering staff of ETRI, Daejeon, Rep. of Korea, from 2000 to 2012. Since March 2012, he has been an associate professor in the Department of Applied Mathematics at Kongju National University, Gongju, Rep. of Korea. His research interests include cryptography and information security, data privacy, and digital forensics.



Changho Seo received his BS, MS, and PhD in 1990, 1992, and 1996, respectively, from the Department of Mathematics at Korea University, Seoul, Rep. of Korea. Currently, he is a full professor in the Department of Applied Mathematics at Kongju National University, Gongju, Rep. of Korea. His research interests include cryptography, information security, and system security.