Relationship between $GF(2^m)$ Montgomery and Shifted Polynomial Basis Multiplication Algorithms

Haining Fan and M. Anwar Hasan
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1
Emails: hfan@vlsi.uwaterloo.ca and ahasan@ece.uwaterloo.ca

Abstract - Applying the matrix-vector product idea of the Mastrovito multiplier to the $GF(2^m)$ Montgomery multiplication algorithm, we present a new multiplier for irreducible trinomials. This multiplier and the corresponding shifted polynomial basis (SPB) multiplier have the same circuit structure for the same set of parameters. Furthermore, by establishing isomorphisms between the Montgomery and the SPB constructions of $GF(2^m)$, we show that the Montgomery algorithm can be used to perform the SPB multiplication without any changes, and vice versa.

Index Terms - Finite field, multiplication, Montgomery multiplication algorithm, polynomial basis, shifted polynomial basis, irreducible trinomial.

1 INTRODUCTION

Efficient VLSI implementation of multipliers for the finite field $GF(2^m)$ is important for many cryptosystems. To this end, several algorithms and hardware architectures have been proposed in the literature (see, for example, [1–13]). Among the existing algorithms and architectures, the use of a polynomial basis (PB) to represent elements of $GF(2^m)$ appears to be more common than other bases, such as normal and dual bases.

For hardware implementation of a $GF(2^m)$ multiplier using PB, there are two types of approaches:

1) Type-I: multiplication of two binary polynomials, each of degree m-1 or less, followed by a modulo reduction operation using an irreducible polynomial $f(x)$ of degree m.

2) Type-II: formation of a binary $m \times m$ matrix, which depends on one input and the reduction irreducible polynomial, followed by a multiplication of the matrix and the other input vector.

Type-II approach is suitable when the reduction polynomial is fixed or is chosen from a small set of polynomials.

For the purpose of representing elements of $GF(2^m)$, a generalization of PB is the so-called shifted polynomial basis (SPB) proposed in [11]. As the name implies, for any integer v, the set $\{x^{i-v} | 0 \le i \le m-1\}$ is a SPB. For $GF(2^m)$ multiplication, the use of SPB also results in two approaches, which are similar to those of PB, namely, type-I and type-II.

For multiplication $C=AB$, where $A, B \in GF(2^m)$, whether PB or SPB is used, both type-I and type-II approaches generate C as the output, which is represented in the same basis as the inputs. A slightly different kind of multiplication scheme exists, known as the Montgomery multiplication for finite fields, where for inputs A and B, the output is $AB\tilde{R} \bmod f(x)$ with $\tilde{R}$ being the inverse of a carefully chosen field element $R \in GF(2^m)$. This type of multiplication scheme is hereafter referred to as the Montgomery form (MF). The existing MF algorithms for $GF(2^m)$ in [6], [7] fall

under the type-I approach mentioned earlier. Although the type-II approach is possible, there appears to be no mention of this in the literature. One of the contributions of the report is a multiplication scheme based on this approach. The following table summarizes the three representations and the available approaches for the multiplication in $GF(2^m)$.

TABLE 1: Approaches for PB, SPB and MF multiplications

	PB	SPB	MF
Type-I	[4, 5, 9, 10]	[11]	[6, 7, 13, 14]
Type-II	[1, 2, 3, 8]	[11]	This work

Each of SPB and MF can be described in terms of a number of parameters. Using irreducible trinomials, we first show that if SPB and MF have the same parameters then the type-II approach results in the same multiplier structure for these two representations. We then use the field isomorphism to investigate this phenomenon further. The homomorphism is not only a powerful method to study algebraic relationship of different algebraic structures but also an important tool to design efficient algorithms, e.g., the Discrete Fourier Transform based polynomial multiplication algorithm. We determine all isomorphisms among three representations of $GF(2^m)$: the PB representation, the Montgomery representation, and the SPB representation. The main result of this report is that the Montgomery multiplication algorithm can be used to perform the SPB multiplication without any change for the same parameters, and vice versa.

This report is organized as follows: In Section 2, we summarize the previous work on PB and SPB multiplication algorithms. Details of the type-II approach based multiplier using MF is proposed in Section 3. The relationship between the Montgomery multiplication and the SPB multiplication is presented in Section 4. Finally, concluding remarks are made in Section 5.

2 PREVIOUS WORK

2.1 PB Multiplication Algorithms

Let $f(u)$ be an irreducible polynomial over $GF(2)$. All elements of $GF(2^m)=GF(2)[u]/(f(u))$ can be represented using a PB $\{x^i | 0 \le i \le m-1\}$, where x is a root of f. Given two field elements $A=(a_0,a_1,...,a_{m-1})^T=\sum_{i=0}^{m-1} a_i x^i$ and $B=(b_0,b_1,...,b_{m-1})^T=\sum_{i=0}^{m-1} b_i x^i$, where $a_i, b_i \in GF(2)$, the PB type-I multiplication algorithm computes the product $C=(c_0,c_1,...,c_{m-1})^T=\sum_{i=0}^{m-1} c_i x^i$ of A and B using the following two steps:

(i) Polynomial multiplication:

$$T = AB = \sum_{i=0}^{2m-2} t_i x^i \text{, where}$$

$$t_i = \sum_{\substack{h+j=i \\ 0\le j,h\le m-1}} a_h b_j = \begin{cases} \sum_{j=0}^{i} a_j b_{i-j} & 0 \le i \le m-1, \\ \sum_{j=i+1-m}^{m-1} a_j b_{i-j} & m \le i \le 2m-2; \end{cases} \quad (1)$$

(ii) Reduction modulo the irreducible polynomial f:

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

Mastrovito proposed a structure for the VLSI implementation using the type-II approach based on the matrix-vector product, i.e., $C=(c_0,c_1,...,c_{m-1})^T=Z(a_0,a_1,...,a_{m-1})^T$ [8]. The structure is somewhat modular. The $m \times m$ matrix $Z = (z_{i,j})_{0 \le i,j \le m-1}$, which is referred to as the Mastrovito matrix [1], depends on both B and f. In order to compute $C=Z(a_0,a_1,...,a_{m-1})^T$, Z is computed first, then $c_t\,(0 \le t \le m-1)$ is computed in a vector inner-product module whose output is $c_t = \sum_{i=0}^{m-1} a_i z_{t,i}$.

2.2 SPB Multiplication Algorithm

The SPB of $GF(2^m)$ over $GF(2)$ is a modification of the PB, and it is defined as follows [11]:

Definition 1. Let v be an integer and the ordered set $M=\{x^i \mid 0 \le i \le m-1\}$ be a polynomial basis of $GF(2^m)$ over $GF(2)$. The ordered set $x^vM := \{x^{i+v} \mid 0 \le i \le m-1\}$ is called the shifted polynomial basis (SPB) with respect to M.

Using SPB, an element $A \in GF(2^m)$ can be represented as $A = x^{-v} \sum_{i=0}^{m-1} a_i x^i$. Corresponding to the two PB parallel multipliers as given in subsection 2.1, two SPB analogues for the irreducible trinomial are proposed in [11], namely, the SPB type-I and type-II multipliers.

2.3 Montgomery Multiplication Algorithm for Finite Fields

The $GF(2^m)$ Montgomery multiplication algorithm is presented in [7], and a generalized version is proposed in [6]. They follow the type-I approach. Let $f(u)$ be an irreducible polynomial over $GF(2)$ and $\{x^i \mid 0 \le i \le m-1\}$ be a PB of $GF(2^m)=GF(2)[u]/(f(u))$ over $GF(2)$, where x is a root of f. Let $\tilde{R}$ be the multiplicative inversion of $R = x^w \in GF(2^m)$ $(0 \le w \le m-1)$. We known that there exists $\tilde{f}(x) \in GF(2^m)$ such that $\tilde{R}R + \tilde{f}(x) f(x) = 1$. The type-I Montgomery multiplication algorithm is as follows [6]:

Algorithm A1: Type-I Montgomery multiplication algorithm for $GF(2^m)$
Input: A, B, R, $\tilde{f}(x)$ represented in PB and $f(x)$.
Output: $D = AB\tilde{R} \mod f(x)$ represented in PB.

S1: $T = AB = \sum_{h=0}^{2m-2} t_h x^h = t_L(x) + x^w t_M(x) + x^{m+w} t_H(x)$, where $t_L(x) = \sum_{h=0}^{w-1} t_h x^h$,

$$t_M(x) = \sum_{h=w}^{m+w-1} t_h x^{h-w} \text{ and } t_H(x) = \begin{cases} \sum_{h=m+w}^{2m-2} t_h x^{h-m-w} & 0 \le w < m-1, \\ 0 & w = m-1. \end{cases}$$

S2: $U = t_L(x)\, \tilde{f}(x) \mod R$;
S3: $D' = (T+Uf(x))/R$;
S4: If $\deg(D') > m-1$ then $D = D' \mod f(x)$ else $D = D'$.

Since the output of algorithm **A1** is not $AB \mod f(x)$ but $AB\tilde{R} \mod f(x)$, which is represented in PB, some pre- and post-processings are likely to be required in the

case of exponentiation operation. This is similar to the integer case [16, 17], i.e., we may first change inputs of algorithm **A1** from A and B to AR mod $f(x)$ and BR mod $f(x)$, respectively, and then perform the Montgomery multiplication. The output, which is $ARBR\widetilde{R}$ mod $f(x) = ABR$ mod $f(x)$, may now be used as an input to a subsequent multiplication.

We note that no conversion is required for exponentiation operations using PB or SPB multiplication algorithms, since the output of each of these algorithms is the exact product represented in the same basis as the inputs.

3 TYPE-II MONTGOMERY MULTIPLIER FOR IRREDUCIBLE TRINOMIALS

In this section we assume that $f(u)=u^m+u^k+1$ is an irreducible trinomial. In [11], it is shown that the best values of the SPB parameter v are k and k-1 for the implementation of the irreducible trinomial-based bit parallel multiplier. We now determine the best values of w in Montgomery multiplication algorithm **A1**.

If $f(u)=u^m+u^k+1$ then $\widetilde{f}(x)=1$, and steps S2, S3 and S4 of Algorithm **A1** may be simplified as follows:

S2′: $D' = (x^{m-w}+x^{k-w})t_L(x)+t_M(x)+(x^k+1)t_H(x)$;
S3′: If $\deg(D')>m-1$ then $D = D'$ mod $f(x)$ else $D= D'$.

If degrees of all terms of D' are in the range of 0 and m-1, then the mod operation in S3′ is not performed. Hence if this condition is always satisfied, for hardware implementation no gate is required for S3′. It is easy to see that this condition is equivalent to the following inequalities:

$$\begin{cases} 0 \le m-w+w-1 \le m-1, \\ 0 \le k-w \le m-1, \\ 0 \le k+m-w-2 \le m-1. \end{cases}$$

After solving these inequalities, we find that the values of w are k and k-1. Hence, the best values of w for the Montgomery multiplication and those of v for the SPB multiplication are the same. In this section, we assume that w is either k or k-1.

The computational procedure of the Montgomery multiplication Algorithm **A1** is similar to that of the type-I PB multiplier, i.e., the product of the two polynomials A and B is calculated first, and then the reduction operation is performed. We now apply the type-II approach to the computation of $D = AB\widetilde{R}$ mod $f(x)$, and obtain a formula of the Montgomery multiplication algorithm for the irreducible trinomial, which is called the type-II Montgomery multiplier.

The type-II Montgomery algorithm computes $D = AB\widetilde{R}$ mod $f(x)=(d_0,d_1,...,d_{m-1})^T$ by a single matrix-vector product, i.e., $D=(d_0,d_1,...,d_{m-1})^T=Z(a_0,a_1,...,a_{m-1})^T$. The $m \times m$ matrix $Z=(z_{i,j})_{0\le i,j\le m-1}$, which is also called the Mastrovito matrix, depends on R, B and f. In order to compute D, Z is computed first, then $d_t = \sum_{i=0}^{m-1} a_i z_{t,i}$ $(0 \le t \le m-1)$ is computed in a vector inner-product module. So we need to determine explicit expressions of entries of Z. From S1 and S4, we have

$$D = \sum_{h=0}^{w-1} t_h x^{h+m-w} + \sum_{h=0}^{w-1} t_h x^{h+k-w} + \sum_{h=w}^{m+w-1} t_h x^{h-w} + \sum_{h=m+w}^{2m-2} t_h x^{h+k-m-w} + \sum_{h=m+w}^{2m-2} t_h x^{h-m-w}$$

$$= \sum_{i=m-w}^{m-1} t_{i+w-m}x^i + \sum_{i=k-w}^{k-1} t_{i+w-k}x^i + \sum_{i=0}^{m-1} t_{i+w}x^i + \sum_{i=k}^{k+m-2-w} t_{i+w+m-k}x^i + \sum_{i=0}^{m-2-w} t_{i+m+w}x^i . \quad (2)$$

To obtain explicit expressions of entries of Z, we may first apply (1) to (2) and then compare the coefficients of x^i in the new expression. This method has been used in [11]. After obtaining the entries of matrix Z, we may find that type-II multipliers based on MF and SPB share the same Mastrovito matrix Z if $v=w$. In the following, we do not adopt this method since it is quite complicated. We give a simpler explanation. The SPB multiplication formula is obtained from equations (3), (6), (7) of [11], i.e.,

$$C = \sum_{t=m-2v}^{m-1-v} t_{t+2v-m}x^t + \sum_{t=k-2v}^{k-1-v} t_{t+2v-k}x^t + \sum_{t=-v}^{m-1-v} t_{t+2v}x^t + \sum_{t=k-v}^{k+m-2-2v} t_{t+2v+m-k}x^t + \sum_{t=-v}^{m-2-2v} t_{t+m+2v}x^t . \quad (3)$$

Multiplying x^v to (3) and changing the range of the exponent of x from [$-v$, m-v-1] to [0, m-1], we have

$$Cx^v = \sum_{i=m-v}^{m-1} t_{i+v-m}x^i + \sum_{i=k-v}^{k-1} t_{i+v-k}x^i + \sum_{i=0}^{m-1} t_{i+v}x^i + \sum_{i=k}^{k+m-2-v} t_{i+v+m-k}x^i + \sum_{i=0}^{m-2-v} t_{i+m+v}x^i . \quad (4)$$

Careful comparisons of formulae (2) and (4) reveal that $D=Cx^v$ if $v=w$. Since the type-II SPB matrix is derived from (3), and multiplying x^v to (3) does not change this matrix, we obtain that type-II multipliers based on MF and SPB share the same Mastrovito matrix if $v=w$. Therefore we may conclude that both multipliers have the same architecture, and summarize complexities of the type-II Montgomery multiplier for the irreducible trinomial $f(u)=u^m+u^k+1$ as follows [11]:

Time delay = $T_A + (1+\lceil \log_2 m \rceil)T_X$;

AND gates = m^2;

XOR gates = $\begin{cases} m^2-1 & 2k \neq m, \\ m^2-m/2 & 2k = m. \end{cases}$

4 RELATIONSHIP BETWEEN THE MONTGOMERY MULTIPLICATION AND THE SPB MULTIPLICATION

In this section, $f(u)$ denotes a general irreducible polynomial over $GF(2)$ and x is a root of f. Let $M=\{x^i \mid 0 \le i \le m-1\}$ be the PB of $GF(2^m)=GF(2)[u]/(f(u))$ over $GF(2)$ and $\{x^{i-w} \mid 0 \le i \le m-1\}$ be the SPB with respect to M.

First, we introduce some notations. Let set $S = \left\{\sum_{i=0}^{m-1} a_i x^i \mid a_i \in \{0,1\}\right\}$, $A = \sum_{i=0}^{m-1} a_i x^i$ and $B = \sum_{i=0}^{m-1} b_i x^i$ be two elements of S. Let set $L = \left\{\sum_{i=-w}^{m-w-1} p_{w+i} x^i \mid p_i \in \{0,1\}\right\}$, $P = \sum_{i=-w}^{m-w-1} p_{w+i} x^i$ and $Q = \sum_{i=-w}^{m-w-1} q_{w+i} x^i$ be two elements of L. Let $\vec{X} = (1,x,x^2,...,x^{m-1})^T$ and $\vec{Y} = (x^{-w},x^{-w+1},...,x^{m-w-1})^T$ denote the basis column vectors of the PB and SPB. To facilitate description, we use symbols $\vec{A}$ and $\vec{P}$ to denote the coordinate row vectors of $A \in S$ and $P \in L$, i.e., $\vec{A} = (a_0,a_1,...,a_{m-1})$ and $\vec{P} = (p_0,p_1,...,p_{m-1})$, respectively. Let the symbol $\circ$ denote the vector inner product, then $A \in S$ and $P \in L$ may also be written as $A = \vec{A} \circ \vec{X}$ and $P = \vec{P} \circ \vec{Y}$.

We now present definitions of the three constructions of $GF(2^m)$. Defining two operations + and $\cdot$ in S as follows: $A+B:=\sum_{i=0}^{m-1}(a_i \oplus b_i)x^i$ and $A \cdot B:=AB \bmod f(x)$, where $\oplus$ denotes the binary XOR operation and AB denotes the conventional polynomial multiplication, we know that the algebraic structure F_{PB}:=<S, +, $\cdot$> is the well-known PB representation of $GF(2^m)$.

Let $R=x^w \in S$ and $\tilde{R}$ is the multiplicative inverse of R in F_{PB}=<S, +, $\cdot$>. Let the symbol * denote the Montgomery multiplication operation. We use the multiplication operation $\cdot$ of F_{PB}=<S, +, $\cdot$> to describe the operation *, and obtain $A*B:=A \cdot B \cdot \tilde{R}$. It is easy to verify that the algebraic structure F_M:=<S, +, *> is a field, and the identity element of the * operation is R.

Let the symbol $\times$ denote the SPB multiplication operation. We also use the multiplication operation $\cdot$ of F_{PB}=<S, +, $\cdot$> to describe the operation $\times$. The operation $\times$ is defined as follows:

$$P \times Q = (\vec{P} \circ \vec{Y}) \times (\vec{Q} \circ \vec{Y}) := \overrightarrow{((\vec{P} \circ \vec{X}) \cdot (\vec{Q} \circ \vec{X}) \cdot \tilde{R})} \circ \vec{Y}. \quad (5)$$

Here, we also use the symbol + to denote the addition operation in L, which is defined as $P+Q:=\sum_{i=-w}^{m-w-1}(a_{w+i} \oplus b_{w+i})x^i$. It is easy to verify that the algebraic structure F_{SPB}:=<L, +, $\times$> is also a field, and the identity element of the $\times$ operation is 1.

The above discussions may be summarized as follows:

Proposition 1. F_{PB}=<S, +, $\cdot$>, F_M=<S, +, *> and F_{SPB}=<L, +, $\times$> are three isomorphic representations of the finite field $GF(2^m)$.

Before analyzing their isomorphic mappings, we give the following basis conversion formulae between the PB and SPB representations. They address the import and export problems of [15].

Export from PB to SPB:

$$A = \vec{A} \circ \vec{X} = A \cdot R \cdot \tilde{R} = \overrightarrow{(A \cdot R)} \circ \vec{Y}, \quad (6)$$

Import from PB to SPB:

$$A = \overrightarrow{(A \cdot R)} \circ \vec{Y} = \overrightarrow{((\vec{A} \circ \vec{X}) \cdot (\overrightarrow{R^2} \circ \vec{X}) \cdot \tilde{R})} \circ \vec{Y} = (\vec{A} \circ \vec{Y}) \times (\overrightarrow{R^2} \circ \vec{Y}), \quad (7)$$

Export from SPB to PB:

$$P = \vec{P} \circ \vec{Y} = P \times x^{-w} \times R = \overrightarrow{(P \times x^{-w})} \circ \vec{X}, \quad (8)$$

Import from SPB to PB:

$$P = \vec{P} \circ \vec{Y} = (\vec{P} \circ \vec{X}) \cdot \tilde{R} = \overrightarrow{(\vec{P} \circ \vec{X}) \cdot \tilde{R}} \circ \vec{X}. \quad (9)$$

In (7), $R^2=R \cdot R$ denotes the square of R in F_{PB}=<S, +, $\cdot$>. In (8), x^{-w} is an element of the field F_{SPB}=<L, +, $\times$>. Please note that $\tilde{R}$ is the multiplicative inverse of R in F_{PB}=<S, +, $\cdot$>.

In the following, we will find all isomorphisms among the above three representations of $GF(2^m)$. From Theorem 2.21 of [14], we know that the distinct

automorphisms of $GF(2^m)$ are exactly the mappings $\sigma_0, \sigma_1, ..., \sigma_{m-1}$, defined by $\sigma_i(\alpha) = \alpha^{2^i}$ for $\alpha \in GF(2^m)$ and $0 \le i \le m-1$. Similar to the proof of Theorem 2.21, we now prove the following proposition.

Proposition 2. The distinct isomorphisms of F_{PB}=<*S*, +, $\cdot$> onto F_M=<*S*, +, *> are exactly the mappings g_0, g_1,..., g_{m-1} defined by $g_i(A) = R \cdot A^{2^i}$ for $A \in F_{PB}$ and $0 \le i \le m-1$.

Proof. It is easy to see that each g_i is one-to-one, and for all $A, B \in F_{PB}$, we have

$$g_i(A+B) = R \cdot (A+B)^{2^i} = R \cdot A^{2^i} + R \cdot B^{2^i} = g_i(A) + g_i(B);$$
$$g_i(A \cdot B) = R \cdot (A \cdot B)^{2^i} = (R \cdot A^{2^i}) \cdot (R \cdot B^{2^i}) \cdot \tilde{R} = g_i(A) * g_i(B).$$

So each g_i is an isomorphism of F_{PB} onto F_M.

Let β be a primitive element of F_{PB}. The mappings g_0, g_1,...,g_{m-1} are distinct since $g_i(\beta) \ne g_j(\beta)$ for $0 \le i < j \le m-1$.

Now suppose that φ is an arbitrary isomorphism of F_{PB} onto F_M. For the primitive element $\beta \in F_{PB}$, let $\rho(x) = \sum_{i=1}^{t} x^{e_i}$ be its minimal polynomial over the prime field <{0,1}, +, $\cdot$ >, where $0 = e_1 < e_2 < ... < e_{t-1} < e_t = m$. Using the identity $\varphi(A^i) = \varphi(A)^i \cdot \tilde{R}^{i-1}$, where $A \in F_{PB}$ and *i* is a non-negative integer, we have

$$0 = \varphi(\rho(\beta)) = \sum_{i=1}^{t} \varphi(\beta^{e_i}) = \sum_{i=1}^{t} (\varphi(\beta))^{e_i} \cdot \tilde{R}^{e_i - 1} = R \cdot \sum_{i=1}^{t} (\varphi(\beta) \cdot \tilde{R})^{e_i}.$$

So $\varphi(\beta) \cdot \tilde{R}$ is a root of $\rho(x)$ in F_{PB}. Hence from Theorem 2.14 of [14] we know that $\varphi(\beta) \cdot \tilde{R} = \beta^{2^j}$ for some *j*, $0 \le j \le m-1$. Since φ is an isomorphism, then we have $\varphi(A) = R \cdot A^{2^j}$ for all $A \in F_{PB}$. □

Because g_is are bijections, we have the following corollary from the above proposition.

Corollary 1. The distinct isomorphisms of F_M=<*S*, +, *> onto F_{PB}=<*S*, +, $\cdot$> are exactly the mappings g_0^{-1}, g_1^{-1},..., g_{m-1}^{-1} defined by $g_i^{-1}(A) = (A \cdot \tilde{R})^{2^i}$ for $A \in F_M$ and $0 \le i \le m-1$. Especially, $g_0^{-1}(A) = A \cdot \tilde{R}$.

Please note that the exponential operation in the field F_M is not defined in this report, so we use the multiplication operation $\cdot$ of the field F_{PB}=<*S*, +, $\cdot$> to represent the mapping g_i^{-1}. This is also one of the reasons that we do not prove Corollary 1 directly. The other reason is that each of the non-zero coefficients of the minimal polynomial of the element in F_M is *R*, which is the identity element of the * operation in F_M=<*S*, +, *>.

The distinct isomorphisms of F_{PB} onto F_{SPB} are determined by the following proposition.

Proposition 3. The distinct isomorphisms of F_{PB}=<*S*, +, $\cdot$> onto F_{SPB}=<*L*, +, $\times$> are exactly the mappings h_0, h_1,..., h_{m-1} defined by $h_i(A) = \overrightarrow{(A^{2^i} \cdot R)} \circ \vec{Y}$ for $A \in F_{PB}$ and $0 \le i \le m-1$. Especially, $h_0(A) = h_0(\vec{A} \circ \vec{X}) = \overrightarrow{(A \cdot R)} \circ \vec{Y}$, and this map is just the basis

conversion formula (6).

Proof. It is easy to see that each h_i is one-to-one, and for all $A, B \in F_{PB}$, we have

$$h_i(A+B) = \overrightarrow{((A+B)^{2^i} \cdot R)} \circ \vec{Y} = \overrightarrow{(A^{2^i} \cdot R)} \circ \vec{Y} + \overrightarrow{(B^{2^i} \cdot R)} \circ \vec{Y} = h_i(A) + h_i(B);$$

$$h_i(A) \times h_i(B) = \left(\overrightarrow{(A^{2^i} \cdot R)} \circ \vec{Y} \right) \times \left(\overrightarrow{(B^{2^i} \cdot R)} \circ \vec{Y} \right)$$

$$= \overrightarrow{(\overrightarrow{(A^{2^i} \cdot R)} \circ \vec{X}) \cdot (\overrightarrow{(B^{2^i} \cdot R)} \circ \vec{X}) \cdot \tilde{R})} \circ \vec{Y} = \overrightarrow{(A^{2^i} \cdot R \cdot B^{2^i} \cdot R \cdot \tilde{R})} \circ \vec{Y}$$

$$= \overrightarrow{((A \cdot B)^{2^i} \cdot R)} \circ \vec{Y} = h_i(A \cdot B).$$

So each h_i is an isomorphism of F_{PB} onto F_{SPB}.

Let β be a primitive element of F_{PB}. The mappings $h_0, h_1, ..., h_{m-1}$ are distinct since $h_i(\beta) \neq h_j(\beta)$ for $0 \le i < j \le m-1$.

Now suppose that φ is an arbitrary isomorphism of F_{PB} onto F_{SPB}. For the primitive element $\beta \in F_{PB}$, let $\rho(x) = \sum_{i=1}^{t} x^{e_i}$ be its minimal polynomial over the prime field $<\{0,1\}, +, \cdot>$, where $0 = e_1 < e_2 < ... < e_{t-1} < e_t = m$.

From (5) and (9), we have

$$P \times Q = \overrightarrow{(((\vec{P} \circ \vec{X}) \cdot (\vec{Q} \circ \vec{X}) \cdot \tilde{R}) \circ \vec{X}) \cdot \tilde{R}} = (\vec{P} \circ \vec{X}) \cdot (\vec{Q} \circ \vec{X}) \cdot \tilde{R}^2.$$

So we obtain the identity $\varphi(A^i) = (\overrightarrow{\varphi(A)} \circ \vec{X})^i \cdot \tilde{R}^i$, where i is a non-negative integer. Thus we have

$$0 = \varphi(\rho(\beta)) = \sum_{i=1}^{t} \varphi(\beta^{e_i}) = \sum_{i=1}^{t} \left((\overrightarrow{\varphi(\beta)} \circ \vec{X}) \cdot \tilde{R} \right)^{e_i}.$$

Therefore $(\overrightarrow{\varphi(\beta)} \circ \vec{X}) \cdot \tilde{R}$ is a root of $\rho(x)$ in F_{PB}, and from Theorem 2.14 of [14], we know that $(\overrightarrow{\varphi(\beta)} \circ \vec{X}) \cdot \tilde{R} = \beta^{2^j}$ for some j, $0 \le j \le m-1$. So we get $\overrightarrow{\varphi(\beta)} = \overrightarrow{R \cdot \beta^{2^j}}$. Since φ is an isomorphism, we get then $\varphi(A) = \overrightarrow{(A^{2^j} \cdot R)} \circ \vec{Y}$ for all $A \in F_{PB}$. □

Now we determine isomorphisms of F_M onto F_{SPB}. Since the composition of homomorphisms is a homomorphism, we may compose the isomorphisms $g_i^{-1} : F_M \to F_{PB}$ and h_j: $F_{PB} \to F_{SPB}$ to obtain all isomorphisms from F_M onto F_{SPB}:

$$h_j g_i^{-1}(A) = h_j\left(\left(A \cdot \tilde{R} \right)^{2^i} \right) = \overrightarrow{\left(A \cdot \tilde{R} \right)^{2^{i+j}} \cdot R} \circ \vec{Y}, \text{ where } 0 \le i, j \le m-1.$$

Using the identity $\alpha^{2^i} = \alpha^{2^{m+i}}$ ($\alpha \in GF(2^m)$), we have the following proposition.

Proposition 4. The distinct isomorphisms of F_M=<*S*, +, *> onto F_{SPB}=<*L*, +, ×> are exactly the mappings η_0, $\eta_1, ..., \eta_{m-1}$ defined by $\eta_i(A) = \overrightarrow{((A \cdot \tilde{R})^{2^i} \cdot R)} \circ \vec{Y}$ for $A \in F_M$ and $0 \le i \le m-1$. Especially, $\eta_0(A) = \eta_0(\vec{A} \circ \vec{X}) = \vec{A} \circ \vec{Y}$.

Now we have found all isomorphisms among the three representations of $GF(2^m)$, namely, F_{PB}=<*S*, +, ·>, F_M=<*S*, +, *> and F_{SPB}=<*L*, +, ×>. Given an isomorphism of F_M onto F_{SPB}, say σ, we have $A * B = \sigma^{-1}(\sigma(A) \times \sigma(B))$. Since we focus on the

efficient computation of $GF(2^m)$ multiplication, σ should be chosen such that the computation procedure of $\sigma^{-1}(\sigma(A)\times\sigma(B))$ is as simple as possible. The isomorphism η_0 is such a candidate. One method to compose η_0 is by choosing g_0^{-1} and h_0. The following commutative diagram illustrates the three isomorphisms.

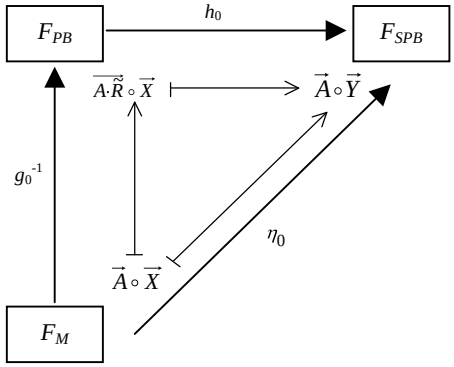

Fig. 1. Isomorphisms h_0, g_0^{-1} and η_0

In software or hardware implementations of the Montgomery and the SPB multiplication algorithms, only coordinates of the multiplier and multiplicand are involved. Therefore the isomorphism map η_0 implies that an implementation of the Montgomery multiplication algorithm in either hardware or software, for example, [5], [7] [12] and [13] etc., can be used to perform the SPB multiplication without any changes, and vice versa.

5 CONCLUSIONS

In this work, we have found all isomorphisms among three representations of $GF(2^m)$: PB representation, the Montgomery form representation and the SPB representation. We have shown that the Montgomery multiplication algorithm can be used to perform the SPB multiplication without any changes for the same parameters, and vice versa. Especially, we have presented a new design of the $GF(2^m)$ bit-parallel Montgomery multiplier, i.e., the matrix-vector product-based Montgomery multiplier, for irreducible trinomials.

REFERENCES

[1] B. Sunar and C. K. Koc, "Mastrovito Multiplier for All Trinomials," *IEEE Transactions on Computers*, vol. 48, no. 5, pp. 522-527, May 1999.

[2] T. Zhang and K. K. Parhi, "Systematic Design of Original and Modified Mastrovito Multipliers for General Irreducible Polynomials," *IEEE Transactions on Computers*, vol. 50, no. 7, pp. 734-749, July 2001.

[3] A. Halbutogullari and C. K. Koc, "Mastrovito Multiplier for General Irreducible Polynomials," *IEEE Transactions on Computers*, vol. 49, no. 5, pp. 503-518, May 2000.

[4] C. Paar, "A New Architecture for a Parallel Finite Field Multiplier with Low Complexity based on Composite Fields," *IEEE Transactions on Computers*, vol. 45, no. 7, pp. 856-861, July 1996.

[5] H. Wu, "Bit-parallel Finite Field Multiplier and Squarer Using Polynomial Basis," *IEEE Transactions on Computers*, vol. 51, no. 7, pp. 750-758, July 2002.

[6] H. Wu, "Montgomery Multiplier and Squarer for a Class of Finite Fields," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 521-529, May 2002.

[7] C. K. Koc and T. Acar, "Montgomery Multiplication in $GF(2^k)$," *Designs, Codes, and Cryptography*, vol. 14, pp. 57-69, 1998.

[8] E. D. Mastrovito, "VLSI Architectures for Multiplication over Finite Field $GF(2^m)$," *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, T. Mora, ed., pp. 297-309, Springer-Verlag, 1988.

[9] 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, no. 8, pp. 945-959, Aug. 2004.

[10] S. O. Lee, S. W. Jung, Ch. H. Kim, J. Yoon, J. Koh, and D. Kim, "Design of Bit Parallel Multiplier with Lower Time Complexity," *In Proc. ICICS'*2003, LNCS 2971, pp. 127-139, Springer-Verlag, 2004.

[11] H. Fan and Y. Dai, "Fast Bit-Parallel $GF(2^n)$ Multiplier for All Trinomials," *IEEE Transactions on Computers*, vol. 54, no. 4, pp. 485-490, 2005.

[12] A. Satoh and K. Takano, "A Scalable Dual-Field Elliptic Curve Cryptographic Processor," *IEEE Transactions on Computers*, vol. 52, no. 4, pp. 449-460, 2003.

[13] E. Savas, A.F. Tenca, and C.K. Koc, "A Scalable and Unified Multiplier Architecture for Finite Fields $GF(p)$ and $GF(2^m)$," *Cryptographic Hardware and Embedded Systems-CHES* 2000, C.K. Koc and C. Paar, eds., pp. 277-292, Springer-Verlag, Aug. 2000.

[14] R. Lidl and H. Niederreiter, *Finite Fileds*, Mass.: Addison-Wesley publishing company, 1983.

[15] B.S. Kaliski Jr. and Y.L. Yin, "Storage-Efficient Finite Field Basis Conversion," *Selected Areas in Cryptography*, S. Tavares and H. Meijer, eds., pp. 81-93, Springer-Verlag, 1998.

[16] S.E. Eldridge and C.D. Walter, "Hardware implementation of Montgomery's modular multiplication algorithm," *IEEE Transactions on Computers*, vol. 42, no. 6, pp. 693-699, 1993.

[17] S.R. Dusse and B.S. Kaliski Jr., "A cryptographic library for the Motorola DSP56000," In *Advances in Cryptology - EUROCRYPT* 90, I.B. Damgard, editor, LNCS-473, pp. 230–244, Springer-Verlag, 1990.