INTEGRATION, the VLSI journal 45 (2012) 205–210

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

Explicit formulae of polynomial basis squarer for pentanomials using weakly dual basis

Sun-Mi Park *

KAIST, Department of Mathematics, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea

ARTICLE INFO

Article history:
Received 24 March 2011
Received in revised form
19 July 2011
Accepted 29 July 2011
Available online 7 August 2011

Keywords:
Finite field arithmetic
Pentanomials
Bit-parallel squarer
Polynomial basis
Weakly dual basis

ABSTRACT

I present a new method to compute a bit-parallel polynomial basis squarer for $GF(2^m)$ generated by an arbitrary irreducible polynomial using weakly dual basis. I apply the proposed method to irreducible pentanomial and derive the explicit formulae for squarer. It is the first time that gives the explicit formulae and an upper complexity bound of squarer for irreducible pentanomials. Moreover, such formulae permit one to choose pentanomial for any odd $m \in [19,2000]$ whose multiplier, as well as squarer, can be performed more efficiently.

1. Introduction

Arithmetic operations over finite field $GF(2^m)$ are essential in a great variety of cryptosystems and error correcting code applications. Among them, multiplication and squaring are the most important ones since other arithmetic operations, such as exponentiation and inversion, can be performed using multiplication and squaring.

In this paper, I study squarer for $GF(2^m)$. Explicit formulae for squarer have been presented when $GF(2^m)$ is defined by irreducible trinomial [1–3]. However, in the case of pentanomial, there exist squaring formulae for only $GF(2^m)$ defined by some special pentanomials. Ref. [4] only shows that squaring using irreducible pentanomials requires at most $4(m-1)$ XOR gates. Ref. [5] presented the complexity of squaring for some pentanomials. Ref. [6] gives closed formulae of Montgomery squaring for type II pentanomial $x^m+x^{n+2}+x^{n+1}+x^n+1$ with an odd m.

In general, the previously proposed squaring needs the reduction steps. Such reduction steps are too complicated to derive squaring formula for general pentanomial. I present a new method to implement a bit-parallel polynomial basis squarer for $GF(2^m)$ generated by an arbitrary irreducible polynomial using weakly dual basis. Recently, [7] proposed a bit-parallel shifted polynomial basis multiplier for $GF(2^m)$ using weakly dual basis. I modify the proposed multiplier in [7] to implement squaring operation. In particular, I derive the explicit formulae for squarer when $GF(2^m)$ is defined by irreducible pentanomial $x^m+x^{k_3}+x^{k_2}+x^{k_1}+1 (1 \le k_1 < k_2 < k_3 \le m/2)$ and show that its complexities are less than or equal to $(3m+7k_3-k_2-3k_1+25)/2$ XOR gates and $3T_X$, where T_X is the delay of one XOR gate.

2. Bit-parallel polynomial basis squarer

2.1. Weakly dual basis

In this section, I give some definitions and properties of finite fields in [7]. For an element $\beta \in GF(2^m)$, the trace $\mathrm{tr}(\beta)$ of β over $GF(2)$ is defined by $\mathrm{tr}(\beta) = \sum_{j=0}^{m-1} \beta^{2^j}$, which is a linear function from $GF(2^m)$ to $GF(2)$.

Lemma 1 (*Fenn et al. [8, Lemma 1] or Lidl and Niederreiter [9, Theorem 2.24]*). *For any linear function φ from $GF(2^m)$ to $GF(2)$, there exists $\gamma \in GF(2^m)$ corresponding to the linear function φ which satisfies that $\varphi(z) = \mathrm{tr}(\gamma z)$ for all $z \in GF(2^m)$.*

Definition 1. Let $\{\alpha_i\} := \{\alpha_0, \ldots, \alpha_{m-1}\}$ and $\{\beta_i\} := \{\beta_0, \ldots, \beta_{m-1}\}$ be bases for $GF(2^m)$ and $\gamma \neq 0 \in GF(2^m)$. Then, the bases $\{\alpha_i\}$ and $\{\beta_i\}$ are said to be weakly dual to each other with respect to γ if $\mathrm{tr}(\gamma\alpha_i\beta_j) = \delta_{ij}$ for all $0 \le i,j < m$, where δ_{ij} is the Kronecker delta function, which is equal to 1 if $i=j$ and 0 otherwise.

Lemma 2 (*Fenn et al. [8, Theorem 2]*). *Every basis has a weakly dual basis (WDB) with respect to any nonzero $\gamma \in GF(2^m)$.*

Throughout this paper, I assume that $GF(2^m)$ is defined by an irreducible polynomial $f(x)$ with a root α. Let $\gamma \neq 0 \in GF(2^m)$ and $\{\beta_i\} := \{\beta_0, \beta_1, \ldots, \beta_{m-1}\}$ be the WDB of the polynomial basis (PB)

* Tel.: +82 42 350 8111; fax: +82 42 350 8114.
E-mail address: smpark@korea.ac.kr

$\{\alpha^i\} := \{1,\alpha,\ldots,\alpha^{m-1}\}$ with respect to γ (see Lemma 2). It holds that $\mathrm{tr}(\gamma\alpha^i\beta_j)=\delta_{ij}$ for $0\le i,j<m$. I use the notations b_is and b_i^* s to denote the coordinates of an element $B=\sum_{i=0}^{m-1} b_i\alpha^i=\sum_{i=0}^{m-1} b_i^*\beta_i$ in $GF(2^m)$ with respect to the bases $\{\alpha^i\}$ and $\{\beta_i\}$, respectively. For an element $B=\sum_{i=0}^{m-1} b_i^*\beta_i$, I have

$$\mathrm{tr}(\gamma\alpha^j B)=\mathrm{tr}\left(\gamma\alpha^j\sum_{i=0}^{m-1} b_i^*\beta_i\right)=\sum_{i=0}^{m-1} b_i^*\mathrm{tr}(\gamma\alpha^j\beta_i)=b_j^* \tag{1}$$

for $0\le j<m$. Consider the basis transformation matrix $(d_{k,j})_{0\le k,j<m}$ from the PB $\{\alpha^i\}$ to the WDB $\{\beta_i\}$ such that $\beta_j=[1,\alpha,\ldots,\alpha^{m-1}][d_{0,j},d_{1,j},\ldots,d_{m-1,j}]^T=\sum_{k=0}^{m-1} d_{k,j}\alpha^k$, where $d_{k,j}\in GF(2)$ for $0\le k,j<m$. I have

$$\begin{aligned}\delta_{ij}&=\mathrm{tr}(\gamma\alpha^i\beta_j)=\textstyle\sum_{k=0}^{m-1} d_{k,j}\mathrm{tr}(\gamma\alpha^{i+k})\\ &=[\mathrm{tr}(\gamma\alpha^i),\mathrm{tr}(\gamma\alpha^{i+1}),\ldots,\mathrm{tr}(\gamma\alpha^{i+m-1})]\times[d_{0,j},d_{1,j},\ldots,d_{m-1,j}]^T\end{aligned}$$

for $0\le i,j<m$. These equations imply that

$$[\beta_0,\beta_1,\ldots,\beta_{m-1}]=[1,\alpha,\ldots,\alpha^{m-1}]M^{-1},$$

where M is the $m\times m$ matrix whose entry $M_{i,j}$ in the i-th row and j-th column is defined by

$$M_{i,j}=t_{i+j}\quad \text{for } 0\le i,\ j<m \tag{2}$$

and $t_l := \mathrm{tr}(\gamma\alpha^l)$ for $l\in\mathbb{Z}$. Moreover, for an element $B=\sum_{i=0}^{m-1} b_i\alpha^i=\sum_{i=0}^{m-1} b_i^*\beta_i$ in $GF(2^m)$:

$$[b_0,b_1,\ldots,b_{m-1}]^T=M^{-1}[b_0^*,b_1^*,\ldots,b_{m-1}^*]^T. \tag{3}$$

2.2. New squarer method

Let $A=\sum_{i=0}^{m-1} a_i\alpha^i$ be an element in $GF(2^m)$. For the squaring $C:=\sum_{i=0}^{m-1} c_i\alpha^i=\sum_{i=0}^{m-1} c_i^*\beta_i(=A^2)$ of A, I have

$$c_i^*=\mathrm{tr}(\gamma\alpha^i C)=\mathrm{tr}(\gamma\alpha^i A^2)=\sum_{j=0}^{m-1}\mathrm{tr}(\gamma\alpha^{i+2j})a_j=\sum_{j=0}^{m-1} t_{i+2j}a_j$$

for $0\le i<m$ from (1). Therefore

$$[c_0^*,c_1^*,\ldots,c_{m-1}^*]^T=K[a_0,\ldots,a_{m-1}]^T,$$

where K is the $m\times m$ matrix defined by

$$K_{i,j}=t_{i+2j}\quad \text{for } 0\le i,\quad j<m.$$

From (3), I conclude that

$$[c_0,c_1,\ldots,c_{m-1}]^T=M^{-1}[c_0^*,c_1^*,\ldots,c_{m-1}^*]^T=M^{-1}K[a_0,\ldots,a_{m-1}]^T. \tag{4}$$

Thus, in order to compute C, it suffices to compute the matrix $M^{-1}K$ whose entries are either 0 or 1.

3. Squarer for irreducible pentanomial

In this section, I apply the proposed method to irreducible pentanomial $f(x)=x^m+x^{k_3}+x^{k_2}+x^{k_1}+1$ with $1\le k_1<k_2<k_3\le m/2$ and derive the explicit formulae of PB squarer. I divide into two cases according to the cases $k_2\le 2k_1$ or $2k_1<k_2$ and take the different values for γ according to each cases to easily compute the inverse matrix M^{-1} of M.

3.1. The case $k_2\le 2k_1$

Assume $k_2\le 2k_1$. I choose $\gamma\in GF(2^m)$ which corresponds to the following linear function:

$$t_l := \mathrm{tr}(\gamma\alpha^l)=\begin{cases}1 & \text{if } l\in\{k_2-k_1-1,k_2-1\},\\ 0 & \text{if } l\in[0,m-1]-\{k_2-k_1-1,k_2-1\}.\end{cases} \tag{5}$$

(Of course, it may be possible to choose different value for γ. From experience, I just choose such value γ that the inverse matrix M^{-1} is of simple form.) Let $\{\beta_i\}$ be the WDB of the PB $\{\alpha^i\}$ with respect to γ.

3.1.1. Squaring formulae

First, I compute trace value t_l for $0\le l\le 2m-2$ to find the inverse matrix M^{-1}. Since $\alpha^l+\alpha^{l-m+k_3}+\alpha^{l-m+k_2}+\alpha^{l-m+k_1}+\alpha^{l-m}=0$ for $l\ge m$, I have

$$t_l=t_{l-m}+t_{l-m+k_1}+t_{l-m+k_2}+t_{l-m+k_3}\quad \text{for } l\ge m. \tag{6}$$

Using (5) and (6), I can compute t_l for $m\le l<2m-k_3$ (i.e., $0\le l-m<m-k_3$ and $k_i\le l-m+k_i<m-k_3+k_i$ for $1\le i\le 3$). Such trace values t_l can be computed by considering all values l such that at least one of trace values in right side of (6) is 1: $t_{l-m}=1\Leftrightarrow l-m\in\{k_2-k_1-1,\ k_2-1\}$ and $t_{l-m+k_1}=1\Leftrightarrow l-m+k_1=k_2-1$. I compute t_l for $l\in\{m+k_2-k_1-1,m+k_2-1\}$. For example, $t_{m+k_2-k_1-1}=t_{k_2-k_1-1}+t_{k_2-1}+t_{2k_2-k_1-1}+t_{k_3+k_2-k_1-1}=1+1+0+0=0$. So I get that

$$t_l=\begin{cases}1 & \text{if } l=m+k_2-1,\\ 0 & \text{if } l\in[m,2m-k_3-1]-\{m+k_2-1\}.\end{cases} \tag{7}$$

Next, using (5)–(7), I compute t_l for $2m-k_3\le l\le 2m-2$ (i.e., $m-k_3\le l-m<m-2$ and $m-k_3+k_i\le l-m+k_i<m+k_i-2$ for $1\le i\le 3$). I consider l such that at least one of trace values in right side of (6) is 1: $t_{l-m+k_3}=1\Leftrightarrow l-m+k_3=m+k_2-1\Leftrightarrow l=2m-k_3+k_2-1$. I have $t_{2m-k_3+k_2-1}=t_{m-k_3+k_2-1}+t_{m-k_3+k_2+k_1-1}+t_{m-k_3+2k_2-1}+t_{m+k_2-1}=0+0+0+1=1$. Therefore, I get that

$$t_l=\begin{cases}1 & \text{if } l=2m-k_3+k_2-1,\\ 0 & \text{if } l\in[2m-k_3,2m-2]-\{2m-k_3+k_2-1\}.\end{cases} \tag{8}$$

By (5), (7), and (8), the matrix M defined in (2) is given by

$$M=\begin{bmatrix} R_1 & O_{k_2\times(m-k_2)}\\ O_{(m-k_2)\times k_2} & R_2\end{bmatrix}, \tag{9}$$

where $O_{i\times j}$ is the $i\times j$ zero matrix, J_l is the $l\times l$ exchange matrix defined by $(J_l)_{i,l-1-i}=1$ for $0\le i<l$ and 0 otherwise,

$$R_1=J_{k_2}+\begin{bmatrix} J_{k_2-k_1} & O_{(k_2-k_1)\times k_1}\\ O_{k_1\times(k_2-k_1)} & O_{k_1\times k_1}\end{bmatrix}\quad\text{and}$$

$$R_2=J_{m-k_2}+\begin{bmatrix} O_{(m-k_3)\times(m-k_3)} & O_{(m-k_3)\times(k_3-k_2)}\\ O_{(k_3-k_2)\times(m-k_3)} & J_{k_3-k_2}\end{bmatrix}.$$

Since $k_2-k_1\le k_2/2$ and $k_3-k_2<(m-k_2)/2$, I can verify

$$R_1^{-1}=J_{k_2}+\begin{bmatrix} O_{k_1\times k_1} & O_{k_1\times(k_2-k_1)}\\ O_{(k_2-k_1)\times k_1} & J_{k_2-k_1}\end{bmatrix}\quad\text{and}$$

$$R_2^{-1}=J_{m-k_2}+\begin{bmatrix} J_{k_3-k_2} & O_{(k_3-k_2)\times(m-k_3)}\\ O_{(m-k_3)\times(k_3-k_2)} & O_{(m-k_3)\times(m-k_3)}\end{bmatrix}.$$

So I obtain that

$$M^{-1}=\begin{bmatrix} R_1^{-1} & O_{k_2\times(m-k_2)}\\ O_{(m-k_2)\times k_2} & R_2^{-1}\end{bmatrix}\quad\text{and}\quad M^{-1}K=\begin{bmatrix}K_1\\K_2\\K_3\\K_4\end{bmatrix}, \tag{10}$$

where

- K_1 is the $k_1\times m$ matrix with

  $$(K_1)_{i,j}=t_{2j-i+k_2-1}\quad \text{for } 0\le i<k_1,$$

- K_2 is the $(k_2-k_1)\times m$ matrix with

  $$(K_2)_{i,j}=t_{2j-i+k_2-k_1-1}+t_{2j-i+k_2-1}\quad \text{for } 0\le i<k_2-k_1,$$

- K_3 is the $(k_3-k_2)\times m$ matrix with
  $(K_3)_{i,j}=t_{2j-i+k_3-1}+t_{2j-i+m-1}$ for $0\le i<k_3-k_2$,
- K_4 is the $(m-k_3)\times m$ matrix with
  $(K_4)_{i,j}=t_{2j-i+m-k_3+k_2-1}$ for $0\le i<m-k_3$

for $0\le j<m$. In order to find the matrix $M^{-1}K$ in (10), I have to compute the following trace values:

(a) t_l for $k_2-k_1\le l\le 3m-k_3+k_2-3$,
(b) $t_l+t_{l+k_1}$ for $0\le l\le 2m+k_2-k_1-3$,
(c) $t_l+t_{l+m-k_3}$ for $k_2\le l\le 2m+k_3-3$.

Such computations are implemented similar to (7) or (8). I leave the computation of trace values (a), (b), (c) for Appendix A.

Using (A.1)–(A.3) in Appendix A, I can locate the positions of one of the matrices K_i for $1\le i\le 4$, and so compute $[c_0,\dots,c_{m-1}]^T=M^{-1}K[a_0,\dots,a_{m-1}]^T$. For the matrix

$$K_1=\begin{bmatrix} t_{k_2-1} & t_{k_2+1} & \cdots & t_{2m+k_2-3}\\ t_{k_2-2} & t_{k_2} & \cdots & t_{2m+k_2-4}\\ t_{k_2-3} & t_{k_2-1} & \cdots & t_{2m+k_2-5}\\ t_{k_2-4} & t_{k_2-2} & \cdots & t_{2m+k_2-6}\\ \vdots & \vdots & \cdots & \vdots\\ t_{k_2-k_1} & t_{k_2-k_1+2} & \cdots & t_{2m+k_2-k_1-2}\end{bmatrix},$$

I have $t_l=1$ for $l\in\{k_2-1,m+k_2-1,2m-k_3+k_2-1,2m-1\}\cup\{2m+k_2-k_1-1\,\text{if}\,k_1\ge 2\}$ by (A.1). Let $\mathbf{S_1}$ be the set that consists of the difference $l-(k_2-1)$ of indices between $t_{k_2-1}(=(K_1)_{0,0})$ and t_l such that $(K_1)_{s,t}=t_l=1$. Then $\mathbf{S_1}=\{0,m,2m-k_3,\ 2m-k_2\}\cup\{2m-k_1\,\text{if}\,k_1\ge 2\}$. I use the notation $S^e:=\{j\in S|j\,\text{is even}\}$ and $S^o:=\{j\in S|j\,\text{is odd}\}$ for any set S. Since $(K_1)_{0j}=t_{(k_2-1)+2j}$ and $(K_1)_{1j}=t_{(k_2-1)+2j-1}$ for $0\le j<m$, I have $c_0=\sum_{j\in S_1^e}a_{\frac{j}{2}}$ and $c_1=\sum_{j\in S_1^o}a_{(j+1)/2}$. Moreover, since the entries of the first column of K_1 except for $(K_1)_{0,0}$ are zero and $(K_1)_{s,t}=(K_1)_{s+2,t+1}$ for $0\le s\le k_1-3$ and $0\le t\le m-2$, I have

$$\begin{cases} c_{0+2i}=\sum\limits_{j\in S_1^e}a_{i+j/2} & \text{for } 0\le i<\left\lceil\dfrac{k_1}{2}\right\rceil\\ c_{1+2i}=\sum\limits_{j\in S_1^o}a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor\dfrac{k_1}{2}\right\rfloor.\end{cases}$$

For the matrix K_2, I have $t_l+t_{l+k_1}=1$ for $l\in I_2$ in (A.2). Let $\mathbf{S_2}:=\{l-(k_2-k_1-1):l\in I_2\}$ be the set that consists of the difference between k_2-k_1-1 and $l\in I_2$ (note that $(K_2)_{0,0}=t_{k_2-k_1-1}+t_{k_2-1}$ and $(K_2)_{s,t}=t_l+t_{l+k_1}=1$ for $l\in I_2$). Then $\mathbf{S_2}:=\{k_1,m,m+k_1,2m-k_3\}\cup\{2m-k_2\,\text{if}\,k_3\ne\ k_2+k_1,2m-k_1\,\text{if}\,k_1\ge 2,k_2\ne 2k_1,\text{and}\,k_3\ \ne 2k_1,2m-k_3+k_1\,\text{if}\,k_3\ne 2k_1\,\text{and}\ \ k_3\ne k_2+k_1,2m-k_2+\ \ k_1\,\text{if}\,k_1+2\le k_2<2k_1\}$. Similar to the matrix K_1, I have

$$\begin{cases} c_{k_1+2i}=\sum\limits_{j\in S_2^e}a_{i+j/2} & \text{for } 0\le i<\left\lceil\dfrac{k_2-k_1}{2}\right\rceil\\ c_{k_1+1+2i}=\sum\limits_{j\in S_2^o}a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor\dfrac{k_2-k_1}{2}\right\rfloor.\end{cases}\tag{11}$$

The conditional sentences in $\mathbf{S_1}$ (resp. $\mathbf{S_2}$) may be omitted if I eliminate the elements in $\mathbf{S_1}$ (resp. $\mathbf{S_2}$) such that are greater than $2m-2$ or appear twice. For example, if $k_3=k_2+k_1$, then two elements $2m-k_2$ and $2m-k_3+k_1$ in $\mathbf{S_2}$ are equal. In that case, I eliminate such two elements in $\mathbf{S_2}$. (In fact, two terms in the summation (11) induced by same elements in $\mathbf{S_2}$ are eliminated.) Then, I may simply rewrite

$\mathbf{S_1}=\langle 0,m,2m-k_3,2m-k_2,2m-k_1\rangle$,

$\mathbf{S_2}=\langle k_1,m,m+k_1,2m-k_3,2m-k_2,2m-k_1,2m-k_3+k_1,2m-k_2+k_1\rangle$,

where the set $<\ >$ means the set that the elements in $<\ >$ which appear twice or are greater than $2m-2$ are eliminated. (See Example.) Similar to above, I can write the formulae for the coefficients c_i for $k_2\le i<m$ as follows:

$$\begin{cases} c_{k_2+2i}=\sum_{j\in S_3^e}a_{i+j/2} & \text{for } 0\le i<\left\lceil\dfrac{k_3-k_2}{2}\right\rceil\\ c_{k_2+1+2i}=\sum_{j\in S_3^o}a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor\dfrac{k_3-k_2}{2}\right\rfloor\end{cases}$$

$$\begin{cases} c_{k_3+2i}=\sum_{j\in S_4^e}a_{i+j/2} & \text{for } 0\le i<\left\lceil\dfrac{m-k_3}{2}\right\rceil\\ c_{k_3+1+2i}=\sum_{j\in S_4^o}a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor\dfrac{m-k_3}{2}\right\rfloor,\end{cases}\tag{12}$$

where $a_l:=0$ for $l\ge m$ and

$\mathbf{S_3}:=\langle k_2,m,m+k_2-k_1,m+k_2,2m-k_3,2m-k_3+k_2-k_1,$
$\quad 2m-k_3+k_2,2m-k_2,2m+k_2-2k_1\rangle$,

$\mathbf{S_4}:=\langle k_3,m,m+k_3-k_2,m+k_3-k_1,m+k_3,2m-k_3,$
$\quad 2m+k_3-2k_2,2m+k_3-2k_1\rangle$.

Set $\mathbf{S_5}:=\langle k_3,m,m+k_3-k_2,m+k_3-k_1,m+k_3\rangle(\subseteq\mathbf{S_4})$. Since $a_l=0$ for $l\ge m$, (12) can be rewritten as follows:

$$\begin{cases} c_{k_3+2i}=\begin{cases}\sum_{j\in S_4^e}a_{i+j/2} & \text{for } 0\le i\langle\left\lceil\dfrac{k_3}{2}\right\rceil\\ \sum_{j\in S_5^e}a_{i+j/2} & \text{for } \left\lceil\dfrac{k_3}{2}\right\rceil\le i\langle\left\lceil\dfrac{m-k_3}{2}\right\rceil,\end{cases}\\ c_{k_3+1+2i}=\begin{cases}\sum_{j\in S_4^o}a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor\dfrac{k_3}{2}\right\rfloor\\ \sum_{j\in S_5^o}a_{i+(j+1)/2} & \text{for } \left\lfloor\dfrac{k_3}{2}\right\rfloor\le i<\left\lfloor\dfrac{m-k_3}{2}\right\rfloor.\end{cases}\end{cases}$$

3.1.2. *Upper bound of complexity*

I consider an upper complexity bound of the proposed squarer. It is easily verified that $1\le|\mathbf{S}_i^e|\le 8$ and $1\le|\mathbf{S}_i^o|\le 8$ for any m,k_3,k_2,k_1, and $1\le i\le 5$. So the time complexity of squarer is $(\max_{1\le i\le 5}\{\lceil\log_2(|\mathbf{S}_i^o|)\rceil,\lceil\log_2(|\mathbf{S}_i^e|)\rceil\})T_X\le 3T_X$ by binary XOR tree. And the space complexity is $\le$

$$\begin{aligned}&(|\mathbf{S}_1^e|-1)\left\lceil\frac{k_1}{2}\right\rceil+(|\mathbf{S}_1^o|-1)\left\lfloor\frac{k_1}{2}\right\rfloor+(|\mathbf{S}_2^e|-1)\left\lceil\frac{k_2-k_1}{2}\right\rceil+(|\mathbf{S}_2^o|-1)\left\lfloor\frac{k_2-k_1}{2}\right\rfloor\\&\quad+(|\mathbf{S}_3^e|-1)\left\lceil\frac{k_3-k_2}{2}\right\rceil+(|\mathbf{S}_3^o|-1)\left\lfloor\frac{k_3-k_2}{2}\right\rfloor+(|\mathbf{S}_4^e|-1)\left\lceil\frac{k_3}{2}\right\rceil\\&\quad+(|\mathbf{S}_4^o|-1)\left\lfloor\frac{k_3}{2}\right\rfloor+(|\mathbf{S}_5^e|-1)\left\lceil\frac{m-2k_3}{2}\right\rceil+(|\mathbf{S}_5^o|-1)\left\lceil\frac{m-2k_3}{2}\right\rceil\\&\le(|\mathbf{S}_1|-2)\left\lceil\frac{k_1}{2}\right\rceil+(|\mathbf{S}_2|-2)\left\lceil\frac{k_2-k_1}{2}\right\rceil+(|\mathbf{S}_3|-2)\left\lceil\frac{k_3-k_2}{2}\right\rceil\\&\quad+(|\mathbf{S}_4|-2)\left\lceil\frac{k_3}{2}\right\rceil+(|\mathbf{S}_5|-2)\left\lceil\frac{m-2k_3}{2}\right\rceil\\&\le\frac{3m+7k_3-k_2-3k_1+25}{2}.\end{aligned}$$

3.1.3. *Example*

I illustrate the formulae of the squarer for $GF(2^{163})$ defined by $f(x)=x^{163}+x^8+x^6+x^4+1$ with $k_1=4,k_2=6,\ k_3=8,$ and

$2m-2=324$. I have

$\mathbf{S_1}=\langle 0,163,318,320,322\rangle=\{0,163,318,320,322\}$,

$\mathbf{S_2}=\langle 4,163,167,318,320,322,322,324\rangle$
$=\{4,163,167,318,320,324\}$,

$\mathbf{S_3}=\langle 6,163,165,169,318,320,324,320,324\rangle$
$=\{6,163,165,169,318\}$,

$\mathbf{S_4}=\langle 8,163,165,167,171,318,322,326\rangle$
$=\{8,163,165,167,171,318,322\}$,

$\mathbf{S_5}=\{8,163,165,167,171\}$

Then the coefficients c_i of the squaring $C=\sum_{i=0}^{162} c_i\alpha^i$ $(=A^2=(\sum_{i=0}^{162} a_i\alpha^i)^2)$ can be written as follows:

- $c_{0+2i}=a_i+a_{i+159}+a_{i+160}+a_{i+161}$ for $0\le i<2$,
- $c_{1+2i}=a_{i+82}$ for $0\le i<2$,
- $c_4=a_2+a_{159}+a_{160}+a_{162}$,
- $c_5=a_{82}+a_{84}$,
- $c_6=a_3+a_{159}$,
- $c_7=a_{82}+a_{83}+a_{85}$,
- $c_{8+2i}=\begin{cases} a_{i+4}+a_{i+159}+a_{i+161} & \text{for } 0\le i<4,\\ a_{i+4} & \text{for } 4\le i<78,\end{cases}$
- $c_{9+2i}=a_{i+82}+a_{i+83}+a_{i+84}+a_{i+86}$ for $0\le i<77$.

I note that c_{8+2i} can be rewritten by $c_{8+2i}=a_{i+4}+a_{i+159}$ for $2\le i<4$ since $a_l=0$ for $l\ge m$. The terms $a_{i+159}+a_{i+161}$ in c_{0+2i} may be reused in c_{8+2i} for$0\le i<2$. Also, c_9 is computed reusing the term $a_{82}+a_{83}$ in c_7. Therefore, the space complexity of the squarer is $(6+3+1+1+2+8+77\cdot 3)-5=247$ and the time delay is $2T_X$ by binary XOR tree (see Table 1).

3.2. The case $k_2>2k_1$

I now consider the case $k_2>2k_1$. For this case, I take γ which satisfies that

$$\mathrm{tr}(\gamma\alpha^l)=\begin{cases}1 & \text{if } l\in\{k_1-1,2k_1-1\},\\ 0 & \text{if } l\in[0,m-1]-\{k_1-1,2k_1-1\}.\end{cases}$$

By the similar process to the case $k_2\le 2k_1$, I have

$$M=\begin{bmatrix} R'_1 & O_{2k_1\times(m-2k_1)}\\ O_{(m-2k_1)\times 2k_1} & R'_2\end{bmatrix} \quad\text{and}$$

$$M^{-1}=\begin{bmatrix} R'^{-1}_1 & O_{2k_1\times(m-2k_1)}\\ O_{(m-2k_1)\times 2k_1} & R'^{-1}_2\end{bmatrix},$$

where

$$R'_1=J_{2k_1}+\begin{bmatrix} J_{k_1} & O_{k_1\times k_1}\\ O_{k_1\times k_1} & O_{k_1\times k_1}\end{bmatrix},$$

$$R'^{-1}_1=J_{2k_1}+\begin{bmatrix} O_{k_1\times k_1} & O_{k_1\times k_1}\\ O_{k_1\times k_1} & J_{k_1}\end{bmatrix},$$

$$R'_2=J_{m-2k_1}+\begin{bmatrix} O_{(m-k_3)\times(m-k_3)} & O_{(m-k_3)\times(k_3-2k_1)}\\ O_{(k_3-2k_1)\times(m-k_3)} & J_{k_3-2k_1}\end{bmatrix} +\begin{bmatrix} O_{(m-k_2)\times(m-k_2)} & O_{(m-k_2)\times(k_2-2k_1)}\\ O_{(k_2-2k_1)\times(m-k_2)} & J_{k_2-2k_1}\end{bmatrix},$$

$$R'^{-1}_2=J_{m-2k_1}+\begin{bmatrix} J_{k_3-2k_1} & O_{(k_3-2k_1)\times(m-k_3)}\\ O_{(m-k_3)\times(k_3-2k_1)} & O_{(m-k_3)\times(m-k_3)}\end{bmatrix} +\begin{bmatrix} J_{k_2-2k_1} & O_{(k_2-2k_1)\times(m-k_2)}\\ O_{(m-k_2)\times(k_2-2k_1)} & O_{(m-k_2)\times(m-k_2)}\end{bmatrix}.$$

I can compute $M^{-1}K$ using (6) and the definition of γ as in Section 3.1.1. As a result, I obtain the coefficient c_i for $0\le i<m$ as follows:

$$\begin{cases}
c_{0+2i} = \sum_{j\in S_1^e} a_{i+j/2} & \text{for } 0\le i<\left\lceil \frac{k_1}{2}\right\rceil,\\
c_{1+2i} = \sum_{j\in S_1^o} a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor \frac{k_1}{2}\right\rfloor,\\
c_{k_1+2i} = \sum_{j\in S_2^e} a_{i+j/2} & \text{for } 0\le i<\left\lceil \frac{k_1}{2}\right\rceil,\\
c_{k_1+1+2i} = \sum_{j\in S_2^o} a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor \frac{k_1}{2}\right\rfloor,\\
c_{2k_1+2i} = \sum_{j\in S_3^e} a_{i+j/2} & \text{for } 0\le i<\left\lceil \frac{k_2-2k_1}{2}\right\rceil,\\
c_{2k_1+1+2i} = \sum_{j\in S_3^o} a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor \frac{k_2-2k_1}{2}\right\rfloor,\\
c_{k_2+2i} = \sum_{j\in S_4^e} a_{i+j/2} & \text{for } 0\le i<\left\lceil \frac{k_3-k_2}{2}\right\rceil,\\
c_{k_2+1+2i} = \sum_{j\in S_4^o} a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor \frac{k_3-k_2}{2}\right\rfloor,\\
c_{k_3+2i} = \begin{cases} \sum_{j\in S_5^e} a_{i+j/2} & \text{for } 0\le i<\left\lceil \frac{k_3}{2}\right\rceil\\ \sum_{j\in S_6^e} a_{i+j/2} & \text{for } \left\lceil \frac{k_3}{2}\right\rceil\le i<\left\lceil \frac{m-k_3}{2}\right\rceil\end{cases}\\
c_{k_3+1+2i} = \begin{cases} \sum_{j\in S_5^o} a_{i+(j+1)/2} & \text{for } 0\le i<\left\lfloor \frac{k_3}{2}\right\rfloor\\ \sum_{j\in S_6^o} a_{i+(j+1)/2} & \text{for } \left\lfloor \frac{k_3}{2}\right\rfloor\le i<\left\lfloor \frac{m-k_3}{2}\right\rfloor\end{cases}
\end{cases}$$

where $a_l := 0$ for $l\ge m$ and

$\mathbf{S_1} := \langle 0,m,2m-k_3,2m-k_2,2m-k_1\rangle$,

$\mathbf{S_2} := \langle k_1,m,m+k_1,2m-k_3,2m-k_2+k_1,2m-k_3+k_1,2m-k_2,$
$2m-k_1\rangle$,

$\mathbf{S_3} := \langle 2k_1,m+k_1,m+2k_1,2m-k_3+k_1,2m-k_3+2k_1,2m-k_2+k_1,$
$2m-k_2+2k_1\rangle$,

Table 1
Comparison of bit-parallel squaring for irreducible pentanomials.

Polynomial	Squarer	Basis	♯ XOR	Delay
$x^m+x^{k_3}+x^{k_2}+x^{k_1}+1$	[4]	PB	$\le 4(m-1)$	–
$(1\le k_1<k_2<k_3\le \frac{m}{2})$	This paper	PB	$\le \frac{3m+7k_3-k_2-3k_1+25}{2}$	$3T_X$
$x^{163}+x^7+x^6+x^3+1$	[5], this paper	PB	246	$3T_X$
$x^{163}+x^8+x^6+x^4+1$	This paper	PB	247	$2T_X$
$x^m+x^{n+2}+x^{n+1}+x^n+1$	[6]	Montgomery squaring	$\le \frac{m-3}{2}+m+4$	$2T_X$
(m is odd)	This paper	SPB	$\le \frac{m-3}{2}+m+3$	$2T_X$

$\mathbf{S_4} := \langle k_2, m, m+k_2-k_1, m+k_2, 2m-k_3, 2m-k_3+k_2-k_1, 2m-k_3+k_2,$
$2m-k_2 \rangle,$

$\mathbf{S_5} := \langle k_3, m, m+k_3-k_2, m+k_3-k_1, m+k_3, 2m-k_3, 2m+k_3-2k_2 \rangle,$

$\mathbf{S_6} := \langle k_3, m, m+k_3-k_2, m+k_3-k_1, m+k_3 \rangle.$

From above formulae, I induce that the time complexity of squarer is $\le 3T_X$ and the space complexity of it is $\le (3m+5k_3-k_2-k_1+28)/2$ as in Section 3.1.2.

4. Comparison and conclusion

I have proposed a new method to perform bit-parallel PB squarer for $GF(2^m)$. Using the proposed method, I have given the explicit formulae of PB squarer for pentanomials, which are the first reported. Table 1 shows that the proposed PB squarer has the similar efficiency with [5] which gives PB squarer in hardware for some polynomials. However, the explicit formulae permit one to choose pentanomial whose squaring can be performed with $2T_X$ time delay which is the lowest delay reported for squaring using pentanomials in the literature. For example, I consider the following two types of pentanomials $x^m+x^{k_3}+x^{k_2}+x^{k_1}+1$ $(1 \le k_1 < k_2 < k_3 \le m/2, m: \text{odd})$:

type(i) : k_3 : odd, k_2 : even, $k_1 = 1$, and $k_3 > 2k_2$,
type(ii) : k_2 or k_1 : even and $k_3 = 2k_1$.

Such types of pentanomials are very abundant. (For any odd $m \in [19, 2000]$, there exist such types of pentanomials.) Squarer using types (i) and (ii) of pentanomials need $2T_X$ time delay and at most $(m+k_3)+(m+2k_3-2k_2-1)/2$ XOR gates. Moreover, for type (i) pentanomials, [10] already gives an efficient multiplier which has the lowest time complexity $T_A+(3+\lceil\log_2(m-1)\rceil)T_X$ among known PB multipliers using pentanomials. For type (ii) pentanomials, I modify the multiplier proposed in [10]. For elements $A, B \in GF(2^m)$, I explicitly write the product AB by coordinates e_i's and d_j's defined in [10] and verify that the modified multiplier for type (ii) pentanomials has m^2 AND gates, $m^2+2m-k_2+2k_1-2$ XOR gates, and $T_A+(3+\lceil\log_2(m-1)\rceil)T_X$ delay. Therefore, operations over binary fields (for example, scalar multiplication) can be implemented more efficiently using (modified) multiplier and squarer for such special types of pentanomials. In Table 2, I particularly give type (i) or (ii) of pentanomials for five binary fields recommended by NIST which have low squaring complexities.

The proposed new method for squarer is also applicable for any shifted polynomial basis (SPB). The squaring formulae depend on chosen SPB as well as pentanomial. In particular, I consider SPB squarer for type (ii) irreducible pentanomial with the SPB $\{\alpha^{-n-1}, \alpha^{-n}, \ldots, \alpha^{m-n-2}\}$. It has the similar complexity with the squarer in [6] (see Table 1). I omit the proof since it is a similar work with Section 3.

Finally, I expect that the results in this paper would be usable by hardware implementations and possibly by software.

Table 2
Pentamomial $x^m+x^{k_3}+x^{k_2}+x^{k_1}+1$ with low squaring complexities.

$[m, k_3, k_2, k_1]$	♯ XOR	Delay
[163, 8, 6, 4]	247	$2T_X$
[233, 9, 4, 1]	355	
[283, 45, 14, 1]	437	
[409, 18, 16, 9]	630	
[571, 35, 6, 1]	861	

Acknowledgements

This work was supported by the SRC Program of Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (2011-0001184).

Appendix A. Computation of trace values

I compute trace values (a), (b), (c) given in Section 3.1.1. First of all, I consider (a) : t_l for $k_2-k_1 \le l \le 3m-k_3+k_2-3$. The values t_l for $k_2-k_1 \le m \le 2m-2$ are already given in (5), (7), and (8). The rest values t_l for $2m-1 \le l \le 3m-k_3+k_2-3$ are computed by the similar method to (7) or (8). First, in order to compute t_l for $2m-1 \le l < 3m-k_3-1$ (i.e., $m-1 \le l-m < 2m-k_3-1$ and $m+k_i-1 \le l-m+k_i < 2m-k_3+k_i-1$ for $1 \le i \le 3$), I consider l such that at least one of trace values in right side of (6) is 1: $l-m = m+k_2-1, l-m+k_1 = m+k_2-1$, $l-m+k_2 = m+k_2-1$, $l-m+k_3 = 2m-k_3+k_2-1$. For example, $t_{3m-k_3+k_2-1} = t_{2m-2k_3+k_2-1} + t_{2m-2k_3+k_2+k_1-1} + t_{2m-2k_3+2k_2-1} + t_{2m-k_3+k_2-1} = 0+0+0+1 = 1$ or $1+0+0+1 = 0$ according to $m \ne 2k_3$ or $m = 2k_3$. Therefore, I have

$$t_l = \begin{cases} 1 & \text{if } l \in T_1, \\ 0 & \text{if } l \in [2m-k_3, 3m-k_3-2]-T_1, \end{cases}$$

where $T_1 := \{2m-1, 2m+k_2-k_1-1\} \cup \{2m+k_2-1 \text{if } m \ne 2k_3, 3m-2k_3+k_2-1 \text{ if } m \ne 2k_3\}$. Next, I consider t_l for $3m-k_3-1 \le l \le 3m-k_3+k_2-3$. It suffices to compute t_l for l satisfying one of the followings:

- $l-m+k_1 \in \{2m-k_3+k_2-1 \quad \text{if } k_1 \ge 2,$
  $2m-1 \quad \text{if } k_3 \le k_2+k_1-2, 2m+k_2-k_1-1 \quad \text{if } k_3 \le 2k_1-2\},$
- $l-m+k_2 \in \{2m-k_3+k_2-1, 2m-1 \quad \text{if } k_3 \le 2k_2-2,$
  $2m+k_2-k_1-1 \text{ if } k_3 \le k_2+k_1-2\},$
- $l-m+k_3 \in \{2m-1, 2m+k_2-k_1-1 \text{ if } k_1 \ge 2\}.$

For example, if $k_3 \le 2k_2-2$, then $t_{3m-k_2-1} = t_{2m-k_2-1} + t_{2m-k_2+k_1-1} + t_{2m-1} + t_{2m+k_3-k_2-1} = 0+0+1+0 = 1$ or $0+1+1+1 = 1$ according to $k_3 \ne 2k_2-k_1$ or $k_3 = 2k_2-k_1$. For another example, if $k_1 \ge 2$, then $t_{3m-k_3+k_2-k_1-1} = t_{2m-k_3+k_2-k_1-1} + t_{2m-k_3+k_2-1} + t_{2m-k_3+2k_2-k_1-1} + t_{2m+k_2-k_1-1} = 0+1+0+1 = 0$ or $0+1+1+1 = 1$ according to $2k_2 \ne k_3+k_1$ or $2k_2 = k_3+k_1$. That is, if $k_1 \ge 2$, and $k_3 = 2k_2-k_1$, then $t_{3m-k_3+k_2-k_1-1} = t_{3m-k_2-1} = 1$. I note that this case is a special case of $l = 3m-k_2-1$ with $k_3 \le 2k_2-2$. As a result, I obtain that

$$t_l = \begin{cases} 1 & \text{if } l \in T_2, \\ 0 & \text{if } l \in [3m-k_3-1, 3m-k_3+k_2-3]-T_2, \end{cases}$$

where $T_2 := \{3m-k_2-1 \text{ if } k_3 \le 2k_2-2, 3m+k_2-2k_1-1 \text{ if } k_3 \le 2k_1-2\}$. Consequently, I have

$$(a) t_l = \begin{cases} 1 & \text{if } l \in I_1 \\ 0 & \text{if } l \in [k_2-k_1, 3m-k_3+k_2-3]-I_1, \end{cases} \tag{A.1}$$

where $I_1 :=$

$\{k_2-1, m+k_2-1, 2m-k_3+k_2-1, 2m-1, 2m+k_2-k_1-1\}$

$\cup\{2m+k_2-1 \text{ if } m \ne 2k_3,\ 3m-2k_3+k_2-1 \text{ if } m \ne 2k_3,$

$3m-k_2-1 \text{ if } k_3 \le 2k_2-2,\ 3m+k_2-2k_1-1 \text{ if } k_3 \le 2k_1-2\}.$

The trace values in (b) are computed from (5) and (A.1) by considering $l \in [0, 2m+k_2-k_1-3]$ such that at least one of t_l and t_{l+k_1} is 1. That is, I compute $t_l + t_{l+k_1}$ for $l \in \{k_2-k_1-1, k_2-1, m+k_2-1, 2m-k_3+k_2-1, 2m-1 \text{ if } k_2 \ge k_1+2\}$ or $l+k_1 \in \{k_2-1, m+k_2-1, 2m-k_3+k_2-1, 2m-1, 2m+k_2-k_1-1 \text{ if } k_1 \ge 2\}$. For example, if $k_2 \ge k_1+2$, $t_{2m-1} + t_{2m+k_1-1} = 1+0 = 1$ or $1+1 = 0$ according to $k_2 \ne 2k_1$ or $k_2 = 2k_1$. From hypothesis $k_2 \le 2k_1$, I have $t_{2m-1} + t_{2m+k_1-1} = 1$ if $k_1+2 \le k_2 < 2k_1$. I obtain the followings:

$$\text{(b)}\quad t_l + t_{l+k_1} = \begin{cases} 1 & \text{if } l \in I_2 \\ 0 & \text{if } l \in [0, 2m+k_2-k_1-3] - I_2, \end{cases} \tag{A.2}$$

where

$I_2 := \{k_2-1, m+k_2-k_1-1, m+k_2-1, 2m-k_3+k_2-k_1-1\}$

$\cup\{2m-k_1-1$ if $k_3 \neq k_2+k_1, 2m+k_2-2k_1-1$ if $k_1 \geq 2$,

$k_2 \neq 2k_1$, and $k_3 \neq 2k_1, 2m-k_3+k_2-1$ if $k_3 \neq 2k_1$ and

$k_3 \neq k_2+k_1, 2m-1$ if $k_1+2 \leq k_2 < 2k_1\}$.

I note that $t_l + t_{l+m-k_3} = 0$ for $k_2 \leq l < k_3$ by (5). Since $t_l + t_{l+m-k_3} = t_{l-k_3} + t_{l-k_3+k_1} + t_{l-k_3+k_2}$ for $k_3 \leq l$ (from (6)), the trace values $t_l + t_{l+m-k_3}$ for $k_3 \leq l \leq 2m+k_3-3$ are obtained from (5) and (A.1) by considering $l \in [k_3, 2m+k_3-3]$ such that at least one of $t_{l-k_3}, t_{l-k_3+k_1}$, and $t_{l-k_3+k_2}$ is 1, i.e., l satisfying one of the followings:

- $l-k_3 \in \{k_2-k_1-1, k_2-1, m+k_2-1,\ 2m-k_3+k_2-1$ if $k_3 \geq k_2+2\}$,
- $l-k_3+k_1 \in \{k_2-1, m+k_2-1, 2m-k_3+k_2-1,\ 2m-1$ if $k_1 \geq 2$, $2m+\ k_2-k_1-1$ if $k_2 \leq 2k_1-2\}$,
- $l-k_3+k_2 \in \{m+k_2-1, 2m-k_3+k_2-1, 2m-1,\ 2m+k_2-k_1-1$ if $k_1 \geq 2\}$.

Consequently, I obtain that

$$\text{(c)}\quad t_l + t_{l+m-k_3} = \begin{cases} 1 & \text{if } l \in I_3 \\ 0 & \text{if } l \in [k_2, 2m+k_3-3] - I_3, \end{cases} \tag{A.3}$$

where

$I_3 := \{k_3+k_2-1, m+k_3-1, m+k_3+k_2-k_1-1,$

$m+k_3+k_2-1, 2m-1\} \cup \{2m+k_2-k_1-1$ if $2k_2 \neq k_3+k_1$,

$2m+k_2-1$ if $k_3 \geq k_2+2, k_3 \neq 2k_1$, and $k_3 \neq 2k_2$,

$2m+k_3-k_2-1$ if $2k_2 \neq k_3+k_1$ and $k_3 \neq 2k_2$,

$2m+k_3+k_2-2k_1-1$ if $k_2 \leq 2k_1-2$ and $k_3 \neq 2k_1\}$.

References

[1] S.-M. Park, K.-Y. Chang, Low complexity bit-parallel squarer for $GF(2^n)$ defined by irreducible trinomials, IEICE Trans. Fundam. E89-A (9) (2006) 2451–2452.

[2] H. Wu, Montgomery multiplier and squarer for a class of finite fields, IEEE Trans. Comput. 51 (5) (2002) 521–529.

[3] H. Wu, Bit-parallel finite field multiplier and squarer using polynomial basis, IEEE Trans. Comput. 51 (7) (2002) 750–758.

[4] H. Wu, Low complexity bit-parallel finite field arithmetic using polynomial basis, in: Proceedings of the First International Workshop Cryptographic Hardware and Embedded Systems (CHES), 1999, pp. 280–291.

[5] J. Guajardo, T. Güneysu, S. Kumar, C. Paar, J. Pelzl, Efficient hardware implementation of finite fields with applications to cryptography, Acta Appl. Math. 93 (1) (2006) 75–118 (International Survey Journal on Applying Mathematics and Mathematical Applications).

[6] A. Hariri, A. Reyhani-Masoleh, Bit-serial and bit-parallel montgomery multiplication and squaring over $GF(2^m)$, IEEE Trans. Comput. 58 (10) (2009) 1332–1345.

[7] S.-M. Park, K.-Y. Chang, Fast bit-parallel shifted polynomial basis multiplier using weakly dual basis over $GF(2^m)$, IEEE Trans. Very Large Scale Integration (VLSI) Syst., in press, doi:10.1109/TVLSI.2010.2075946.

[8] S.T.J. Fenn, M. Benaissa, D. Talyor, $GF(2^m)$ multiplication and division over the dual basis, IEEE Trans. Comput. 45 (3) (1996) 319–327.

[9] R. Lidl, H. Niederreiter, An Introduction to Finite Fields and Their Applications, Cambridge University Press, Cambridge, 1986.

[10] A. Reyhani-Masoleh, M.A. Hasan, Low complexity bit parallel architectures for polynomial basis multiplication over $GF(2^m)$, IEEE Trans. Comput. 53 (8) (2004) 945–959.

Sun-Mi Park received her B.S. degree in mathematics education and M.S. and Ph.D. degrees in mathematics, all from Korea University, Seoul, Korea in 1997, 1999 and 2004, respectively. She is currently working as a researcher with Algebraic's Structure and the Applications Research Center (ASARC) at Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. Her research interests include number theory, algorithms and architectures for computations in Galois fields.