Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.1, Feb. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.1.41

가우시안 정규기저를 갖는 $GF(2^n)$의 곱셈에 대한 오류 탐지*

김 창 한,[1†] 장 남 수,[2] 박 영 호[2‡]
[1]세명대학교, [2]세종사이버대학교

Fault Detection Architecture of the Field Multiplication Using Gaussian Normal Bases in $GF(2^n)$*

Chang Han Kim,[1†] Nam Su Chang,[2] Young Ho Park[2‡]
[1]Semyung University, [2]Sejong Cyber University

요 약

본 논문에서는 가우시안 정규기저를 갖는 유한체 $GF(2^n)$의 곱셈기 오류 탐지 방법을 제시한다. 제안하는 오류 탐지 방법은 하드웨어로 단순하게 구성된다. 즉 n-bit 출력 직렬 곱셈기에서는 1 개의 AND gate, n+1 개의 XOR gate, 그리고 1 개의 1-bit register로 구성되며, 병렬 곱셈기의 경우 n 개의 AND gate와 2n-1 개의 XOR gate로 구성된다. 제안하는 방법은 C=AB 연산에 홀수개의 오류가 발생하는 경우 탐지가 된다.

ABSTRACT

In this paper, we proposed an error detection in Gaussian normal basis multiplier over $GF(2^n)$. It is shown that by using parity prediction, error detection can be very simply constructed in hardware. The hardware overheads are only one AND gate, n+1 XOR gates, and one 1-bit register in serial multipliers, and so n AND gates, 2n-1 XOR gates in parallel multipliers. This method are detect in odd number of bit fault in C = AB.

Keywords: Finite Fields, Normal Basis, Multiplication, Error Detection

I. 서 론

유한체 $GF(2^n)$은 암호분야[1,2,3,4]와 Reed-Solomon Codes 분야[5]에 많이 응용된다. 최근 암호분야 오류주입 공격에 대한 연구가[6] 활발해 지면서 이를 방지하기 위한 방법으로 유한체 연산의 오류 탐지 방법에 관심이 고조되고 있다[7,8,9, 10,11]. 유한체 $GF(2^n)$의 연산은 덧셈, 곱셈으로 구성되나 경우에 따라 제곱 연산을 분리하기도 한다. 덧셈연산은 n번의 XOR 연산으로 쉽게 구성되지만 곱셈 연산은 복잡하게 구성된다. 또한, 유한체의 표현 방법에 따라 다양하게 구성되며, 대표적인 방법은 다항식기저와 정규기저를 이용하여 표현하는 것이다. 특히, 정규기저로 표현할 경우 제곱연산이 Cyclic Shift로 표현되어 하드웨어로 구현할 경우 연산이 없는 장점이 있다.

$GF(2^n)$의 곱셈연산 오류 탐지를 위한 연구는 1998년 FEN 등[7]이 AOP(All One Polynomial) 곱셈기, Massey-Omura 곱셈기, Berlekamp 곱셈기에 대한 오류 탐지 방법을 제안하였다. AOP와 Berlekamp 곱셈기는 다항식기저를 Mas-

접수일(2013년 9월 26일), 게재확정일(2013년 10월 10일)
* 이 논문은 2013학년도 세명대학교 교내학술연구비 지원에 의해 수행된 연구임.
† 주저자, chkim@semyung.ac.kr
‡ 교신저자, youngho@sjcu.ac.kr (Corresponding author)

sey-Omura 곱셈기는 정규기저를 이용한 곱셈기이다. 이 곱셈기는 1-bit 단위로 출력하는 직렬(Serial) 곱셈기이다. 2006년에는 A. Reyhani-Masoleh와 M.A. Hasan[8]이 다항식기저를 사용한 병렬(parallel) 곱셈기와 n-bit를 한 번에 출력하는 직렬곱셈기에 대한 오류 탐지 방법을 발표하였다. 또한 2009년에는 C.W. Chiou 등[11]이 가우시안 정규기저(Gassian Normal Basis)를 이용한 세미 시스톨릭(Semi-Systolic) 구조의 곱셈기의 오류 탐지 방법을 제시하였다.

본 논문에서는 가우시안 정규기저로 표현되는 유한체 $GF(2^n)$의 병렬곱셈기와 n-bit를 한 번에 출력하는 직렬곱셈기에 대한 오류 탐지 방법을 제안한다. Fen 등[7]이 제안한 Massey-Omura 곱셈기의 경우 Cyclic Function H가 존재하는 것을 보였으나 본 논문에서는 하나의 계수에 의해서 표현된다는 것을 보였다. 또한, $GF(2^n)$이 타입 k(even)의 가우시안 정규기저를 갖는 경우 n 개의 AND 와 $n-1$개의 XOR로, 그리고 $k=1$인 경우는 $n+1$ 개의 AND 연산과 n 번의 XOR 연산으로 곱셈 오류 탐지를 위한 Parity Bit를 계산할 수 있음을 보였다.

본 논문은 2장에서는 유한체와 가우시안 정규기저에 관한 기본적인 이론 설명과 Parity Prediction에 관해 정의하고, 3장에서는 곱셈의 Parity 확인을 위한 관련 이론을 제안한다. 4장에서는 직렬 및 병렬 곱셈기에 오류 탐지를 적용하였을 경우의 곱셈기와 복잡도를 제시하며 5장에서는 결론을 제시한다.

II. 유한체와 Parity Prediction

2.1 유한체 $GF(2^n)$

유한체 $GF(2^n)$의 연산은 덧셈과 곱셈, 제곱으로 구성되어 있다. 덧셈과 제곱은 간단하게 구성되는 반면 곱셈은 유한체의 표현 방법에 따라 다양하게 구성된다. 유한체의 표현은 어떤 기저를 사용하여 표현하느냐에 따라 구분하는데, 대표적으로 다항식기저와 정규기저를 사용하고 있다. 즉, $GF(2^n)$을 구성하는 $GF(2)$ 위의 기약다항식을

$$f(x)= x^n+a_{n-1}x^{n-1}+\cdots+a_1x+1$$

이라 할 때 $f(x)$의 근을 α라 하면 $GF(2^n)$의 원소

$$A= a_0+a_1\alpha+\cdots+a_{n-1}\alpha^{n-1},\quad a_i\in GF(2)$$

와 같이 표현하는 것이 다항식기저를 이용한 표현이다. 이때 A를 벡터로 표시하면

$$A = (a_0,a_1,\cdots,a_{n-1}).$$

이다. 반면에 정규기저를 이용한 표현은 $\beta\in GF(2^n)$의 conjugate들이 $GF(2)$ 위에서 $GF(2^n)$ 의 기저가 될 때, 이 기저를 사용하는 것이다. 즉, $\{\beta,\beta^2,\cdots,\beta^{2^{n-1}}\}$이 기저가 되어 $GF(2^n)$의 원소

$$A=a_0\beta+a_1\beta^2+\cdots+a_{n-1}\beta^{2^{n-1}},\ a_i\in GF(2)$$

와 같이 표현하는 것이다. 이때 β를 정규기저 생성자라 한다. 또한 A를 벡터로 표시하면

$$A=(a_0,a_1,\cdots,a_{n-1})$$

이다.

정규기저를 이용할 경우 제곱 연산은 계수들의 위치이동으로 표시되므로 하드웨어적으로는 연산이 없는 장점이 있다. 정규기저를 구성하는 방법은 여러 방법이 있으나 다음 정리에 제시된 Gaussian 정규기저 방법이 가장 널리 쓰인다.

정리 1. n,k는 $nk+1$이 2 보다 큰 소수인 조건을 만족하는 양의 정수이고, α는 $GF(2^{nk})$에서 $nk+1$의 원시근이라 하자. Z^*_{nk+1} 에서 2의 위수(order)를 e라 할 때 $\gcd(nk/e,n)=1$라 하자. 그러면 Z_{nk+1} 에서 k의 원시근 τ에 대하여

$$\beta= \sum_{i=0}^{k-1}\alpha^{\tau^i}$$

는 $GF(2)$ 위에서 $GF(2^n)$의 정규기저 생성자이다[4].

참고. 양의 정수 n에 대하여 정리 1을 만족하는 k가 존재할 때 $GF(2^n)$은 $GF(2)$ 위에서 타입 k인 가우시안 정규기저를 갖는다고 한다.

정규기저를 이용한 곱셈의 경우

$$C=A\cdot B,\ A=\sum_{i=0}^{n-1}a_i\beta^{2^i},\ B=\sum_{i=0}^{n-1}b_i\beta^{2^i},\ C=\sum_{i=0}^{n-1}c_i\beta^{2^i}$$

라 하면

$$c_0=(a_0,a_1,\cdots,a_{n-1})\cdot M\cdot(b_0,b_1,\cdots,b_{n-1})^T,$$
$$M=(m_{ij}):n\times n\text{matrix},\ \mathrm{m_{ij}}\in\mathrm{GF(2)},$$

로 표시되고, 이때 행렬 M의 1의 개수(Weight)를 기저 $N=\{\beta,\beta^2,\cdots,\beta^{2^{n-1}}\}$의 곱셈에 대한 복잡도(Complexity) C_N이라 한다.

참고. 모든 정규기저 N의 복잡도는 $C_N \geq 2n-1$이다. 특히 $C_N = 2n-1$인 경우 N을 최적정규기저라 한다. 위 정리에서 $k=1,2$인 경우 최적정규기저가 되며 각각을 타입 I, II의 최적정규기저라 한다. 위 정리에서 k가 짝수인 경우 $C_N \leq kn-1$이다[4].

2.2 Parity Prediction

정규기저로 표현된 $A,B \in GF(2^n)$의 곱의 결과를 $C = A \cdot B$라 하고, 각각의 parity bit를

$$A_P = \sum_{i=0}^{n-1} a_i, B_P = \sum_{i=0}^{n-1} b_i, C_P = \sum_{i=0}^{n-1} c_i,$$
$$A_P, B_P, C_P \in GF(2)$$

라 하자. A와 B의 곱셈 연산에서 얻은 C의 parity bit를 $C_P(Act)$, 예상되는 parity bit를 $C_P(pred)$라 하자. 오류 없이 정상적인 곱셈이 이루어지면 $C_P(Act) = C_P(pred)$이다. 곱셈 과정을 CUT(circuit under test), $C_P(pred)$를 구하는 과정을 PP(parity prediction)라 하자. 비트 병렬 곱셈기와 n-bit 동시 출력 직렬 곱셈기의 경우 곱셈의 결과부터 $C_P(Act)$를 구하는 과정을 PG(parity generation) 이라 하자. 그러면 정상적으로 곱셈이 이루어졌는지 확인하기 위한 연산 과정은 Fig.1.과 같다. PG 연산 전체와 그 결과와 $C_P(pred)$의 연산 모두 XOR 로 구성된다.

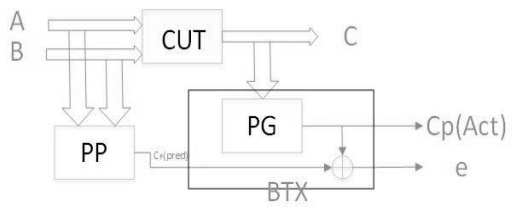

Fig.1. Error Detection Multiplier

III. 소가우시안 정규기저 곱셈기의 오류 탐지

3.1 기본 개념

$n+1$을 소수라 하자. α를 $x^n + x^{n-1} + \cdots + 1$의 완전 확창체(Splitting field) F의 $n+1$의 원시근($n+1$-th primitive root)라 하자. 그러면 F는 $GF(2^n)$의 부분체이다.

$$\overline{A} = \sum_{i=0}^{n} A_i \alpha^i, \overline{B} = \sum_{j=0}^{n} B_j \alpha^j, \ A_i, B_j \in GF(2)$$

라 하면 $\overline{A}, \overline{B} \in GF(2^n)$이다.

$\overline{C} = \overline{A} \cdot \overline{B} = \sum_{k=0}^{n} C_k \alpha^k$라 하면

$$(C_0\, C_1 \cdots C_n)^T = \begin{pmatrix} A_0 & A_n & \cdots A_2\, A_1 \\ A_1 & A_0 & \cdots A_3\, A_2 \\ & & \\ A_n\, A_{n-1} & & \cdots A_1\, A_0 \end{pmatrix} \begin{pmatrix} B_0 \\ B_1 \\ \\ B_n \end{pmatrix}$$

이다.

유한체 원소 A를 정규기저로 표현한 것과 같이 parity bit를 $\overline{A}$에 대하여 다음과 같이 정의하자.

정의 1. $\overline{A} \in GF(2^n)$인 경우

$\overline{A_P} = \sum_{i=0}^{n} A_i, \overline{A_P} \in GF(2)$

라 하자.

정리 2. $\overline{C_P} = \overline{A_P} \cdot \overline{B_P}$.

보조정리 1. 정리 1의 가정을 만족하는 n,k,τ의 경우, Z^*_{nk+1}의 원소는

$$r = \tau^j 2^i, \ 0 \leq j \leq k-1, \ 0 \leq i \leq n-1,$$

로 유일하게 표현된다[4].

3.2 타입 I

타입 I인 경우는 $n+1$이 소수이고 2가 Z^*_{n+1}의 생성자이므로

$$Z^*_{n+1} = \{1,2,\cdots,\mathrm{n}\} = \{2^{\mathrm{i}} \mid 0 \leq \mathrm{i} \leq \mathrm{n}-1\}$$

이다. 따라서 $GF(2)$ 위에서 $GF(2^n)$의 정규기저 생성자 $\beta = \alpha$이므로 정규기저 N은 다음과 같다.

$N = \{\alpha, \alpha^2, \alpha^{2^2}, \cdots, \alpha^{2^{n-1}}\}$,
$\alpha \in GF(2^n) : n+1th \ primitive$ root of unity.

또한, 집합적으로는 $Z^*_{n+1} = <2>$ 이므로

$$\{\alpha,\ \alpha^2,\ \alpha^{2^2},\ \cdots,\ \alpha^{2^{n-1}}\} = \{\alpha, \alpha^2, \cdots, \alpha^n\}$$

이다. 그러므로 $GF(2^n)$의 원소 A의 계수 위치를 정리하면 $A = \sum_{i=1}^{n} a_i \alpha^i$이고 A의 parity bit는 동일하게

표시된다. 또한, $\overline{A}=\sum_{i=0}^{n}A_i\alpha^i$로 표시하면 $A_0=0$이고 $A_i=a_i, 1\le i\le n$이다.

A, B를 $\overline{A},\overline{B}$로 표시하고 A와 B의 곱을 표시한 결과를 $\overline{C}=\sum_{i=0}^{n}C_i\alpha^i$라 하면,

$$(C_0\,C_1\cdots C_n)^T=\begin{pmatrix}0 & A_n & \cdots A_2\,A_1\\ A_1 & 0 & \cdots A_3\,A_2\\ & & \\ A_n\,A_{n-1}\cdots A_1 & & 0\end{pmatrix}\begin{pmatrix}0\\ B_1\\ \\ B_n\end{pmatrix}$$

이다. 그리고 $1=\alpha+\alpha^2+\cdots+\alpha^n$이므로 실제로

$$C=A\cdot B=\sum_{i=1}^{n}(C_0+C_i)\alpha^i$$

이다. 이 경우 $\overline{A_P}=A_P$, $\overline{B_P}=B_P$, n이 짝수,

$$C_0=\sum_{i=1}^{n}A_{n+1-i}B_i=\sum_{i=1}^{n}a_{n+1-i}b_i$$

이므로

$$C_P=\sum_{i=1}^{n}(C_0+C_i)=\sum_{i=0}^{n}C_i+\sum_{i=2}^{n}C_0=\overline{C_P}+C_0$$
$$=\overline{A_P}\overline{B_P}+C_0=A_PB_P+\sum_{i=1}^{n}a_{n+1-i}b_i.$$

정리 3. $A,B\in GF(2^n)$에 대해 $C=A\cdot B$라 하면

$$C_P(pred)=A_P\cdot B_P+\sum_{i=1}^{n}a_{n+1-i}b_i$$

이다.

3.3 타입 II

타입 II인 경우 Z_{2n+1}에서 $\tau=-1$이므로 $\beta=\alpha+\alpha^{-1}$이다. 그리고 보조정리 1에 의하여 $Z_{2n+1}^{*}=\{\pm 2^i\,|\,0\le i\le n-1\}$이므로 타입 I의 경우와 같이 $GF(2^n)$의 정규기저는 집합적으로 다음과 같다.

$$N=\{\beta,\beta^2,\beta^{2^2},\cdots,\beta^{2^{n-1}}\}$$
$$=\{\alpha+\alpha^{-1},\alpha^2+\alpha^{-2},\cdots,\alpha^n+\alpha^{-n}\}.$$

따라서 $GF(2^n)$의 원소 A의 계수 위치를 다시 정리하면

$$A=\sum_{i=1}^{n}a_i(\alpha^i+\alpha^{-i})$$

이다. 또한, A를 $GF(2^{2n})$의 원소 $\overline{A}$로 표시하면 다음과 같다.

$$\overline{A}=\sum_{k=1}^{2n}A_k\alpha^k\ .$$

이 경우, $1\le i\le n$에 대해 $A_i=A_{2n+1-i}=a_i$이다. 그러므로 정리 3의 타입 I 경우의 곱의 parity check bit를 적용하면 다음과 같다.

$$\overline{C_P}=\overline{A_P}\,\overline{B_P}+\sum_{k=1}^{2n}A_{2n+1-k}B_k \qquad (1)$$

정리 4. n이 타입 II 최적 정규기저의 경우 $C=A\cdot B$의 parity bit는 다음과 같다.

$$C_P(pred)=\sum_{i=1}^{n}a_ib_i.$$

증명. $GF(2^n)$의 원소 $A=\sum_{i=1}^{n}a_i(\alpha^i+\alpha^{-i})$를 $GF(2^{2n})$의 원소로 표시하면

$\overline{A}=\sum_{k=1}^{2n}A_k\alpha^k$, $A_i=A_{2n+1-i}=a_i$, $1\le i\le n$

이므로 정수로 $\overline{A_P}=2A_P$이고, 식 (1)과 $1\le i\le n$에 대하여 $A_i=A_{2n+1-i}=a_i$, $B_i=B_{2n+1-i}=a_i$ 이므로 정수로

$$2C_P=2A_P\cdot 2B_P+2\sum_{i=1}^{n}a_ib_i$$

이다. 그러므로 $C_P(pred)=\sum_{i=1}^{n}a_ib_i$이다. □

예제 1. $n=3$은 타입 II의 경우다. $f(x)=x^3+x^2+1$의 근 β는 $GF(2^3)$의 타입 II의 최적 정규기저 생성자이다.

〔I〕 $A=a_1\beta+a_2\beta^2+a_3\beta^{2^2}$, $B=b_1\beta+b_2\beta^2+b_3\beta^{2^2}$라 하고, $\beta^3=1+\beta^2=\beta+\beta^{2^2}$, $\beta^{2^3}=\beta$를 이용하면

$$C=AB$$
$$=a_1b_1\beta^2+(a_1b_2+a_2b_1)\beta^3+a_2b_2\beta^4+(a_1b_3+a_3b_1)\beta^5$$
$$+(a_2b_3+a_3b_2)\beta^6+a_3b_3\beta^8$$
$$=a_1b_1\beta^2+(a_1b_2+a_2b_1)(\beta+\beta^{2^2})+a_2b_2\beta^{2^2}$$
$$+(a_1b_3+a_3b_1)(\beta^2+\beta^{2^2})+(a_2b_3+a_3b_2)(\beta+\beta^2)$$
$$+a_3b_3\beta$$

이므로 $C_P(pred)=a_1b_1+a_2b_2+a_3b_3$이다.

〔II〕 I에서 $\beta=\alpha+\alpha^{-1}$, $a:(2\cdot 3+1)th$ primitive root of unity라 하면

$$\beta = \alpha + \alpha^{-1} = \alpha + \alpha^6,$$
$$\beta^2 = \alpha^2 + \alpha^{-2} = \alpha^2 + \alpha^5,$$
$$\beta^{2^2} = \alpha^{2^2} + \alpha^{-2^2} = \alpha^4 + \alpha^3$$

이므로

$$\overline{A} = a_1\alpha + a_2\alpha^2 + a_3\alpha^3 + a_3\alpha^4 + a_2\alpha^2 + a_1\alpha^6,$$
$$\overline{B} = b_1\alpha + b_2\alpha^2 + b_3\alpha^3 + b_3\alpha^4 + b_2\alpha^2 + b_1\alpha^6$$

이다. $\alpha^7 = 1$과 $1 = \alpha + \alpha^2 + \cdots + \alpha^6$을 이용하여 다음을 계산할 수 있다.

$$\begin{aligned}\overline{A} \cdot \overline{B} = {} & a_1(b_1, 0, b_1 + b_2, b_1 + b_3, b_1 + b_3, b_1 + b_2) \\ & + a_2(b_1 + b_2, b_2, b_1 + b_2, 0, b_2 + b_3, b_2 + b_3) \\ & + a_3(b_2 + b_3, b_1 + b_3, b_3, b_1 + b_3, b_2 + b_3, 0) \\ & + a_3(0, b_2 + b_3, b_1 + b_3, b_3, b_1 + b_3, b_2 + b_3) \\ & + a_2(b_2 + b_3, b_2 + b_3, 0, b_1 + b_2, b_2, b_1 + b_2) \\ & + a_1(b_1 + b_2, b_1 + b_3, b_1 + b_3, b_1 + b_2, 0, b_1)\end{aligned}$$

이다. 그러므로 유한체의 원소 A, B의 곱은 다음과 같다.

$$\begin{aligned}A \cdot B = {} & a_1(b_2, b_1 + b_3, b_2 + b_3) \\ & + a_2(b_1 + b_3, b_3, b_1 + b_2) \\ & + a_3(b_2 + b_3, b_1 + b_2, b_1)\end{aligned}$$

이다. 따라서

$$\begin{aligned}C &= A \cdot B \\ &= (a_1b_2 + a_2(b_1 + b_3) + a_3(b_2 + b_3))\beta \\ &\quad + (a_2b_3 + (a_1(b_1 + b_3) + a_3(b_1 + b_2))\beta^2 \\ &\quad + (a_3b_1 + a_1(b_2 + b_3) + a_2(b_1 + b_2))\beta^{2^2}\end{aligned}$$

이므로 $C_P = a_1b_1 + a_2b_2 + a_3b_3$이다.

3.4 타입 k, $k \geq 4$, k 는 짝수 인 경우

타입 k인 경우 정리 1에 의하여 $\beta = \sum_{i=0}^{k-1} \alpha^{\tau^i}$가 정규기저 생성자이다.

정리 5. $A = \sum_{i=1}^{n} a_i \beta^{2^{i-1}} \in GF(2^n)$라 하자. A를 $\overline{A} = \sum_{j=1}^{nk} A_j \alpha^j \in GF(2^{nk})$의 원소로 표시하면 $1 \leq j \leq nk$인 j에 대하여 $j = 2^i\tau^l$을 만족하는 $i \in \{0, \cdots, n-1\}$, $l \in \{0, \cdots, k-1\}$이 존재하면 $A_j = a_{i+1}$이다.

증명.

$A = \sum_{i=1}^{n} a_i \beta^{2^{i-1}} = \sum_{i=1}^{n} a_i \sum_{l=0}^{k-1} \alpha^{\tau^{l+i-1}} = \sum_{i=0}^{n-1} \sum_{l=0}^{k-1} a_{i+1} \alpha^{2^i\tau^l}$이고 보조정리 1에 의하여

$$Z^*_{nk+1} = \{\tau 2^i \mid 0 \leq l \leq k-1,\ 0 \leq i \leq n-1\}$$

이므로 고정된 $0 \leq i \leq n-1$에 대하여 $j = 2^i\tau^l$, $0 \leq l \leq k-1$인 경우 $A_j = a_{i+1}$이다. □

k가 짝수이므로 $\tau^{k/2} = -1 \in Z_{nk+1}$이다. 위의 정리에 의하여 $\overline{A}$의 계수는 A의 각 계수가 k번 반복되는 것을 알 수 있다. 즉, 정수로 $\overline{A_P} = kA_P$이다.

정리 6. n이 타입 k인 경우 $C = AB \in GF(2^n)$인 경우 parity bit는 $C_P(pred) = \sum_{i=1}^{n} a_i b_i$이다.

증명. A, B, C를

$$\overline{A} = \sum_{i=1}^{nk} A_i \alpha^i, \overline{B} = \sum_{i=1}^{nk} B_i \alpha^i, \overline{C} = \sum_{i=1}^{nk} C_i \alpha^i$$

로 표시하면 정리 3의 타입 I 경우의 Parity bit를 적용하면 다음과 같다.

$$\overline{C_P} = \overline{A_P}\,\overline{B_P} + \sum_{j=1}^{nk} A_{nk+1-j} B_j$$

그리고 정수로 $\overline{A_P} = kA_P$ 이고 $j = 2^i\tau^l$, $1 \leq j \leq nk$, $0 \leq i \leq n-1, 0 \leq l \leq k-1$인 경우

$$nk + 1 - j = \tau^{k/2} 2^i \tau^l = 2^i \tau^{k/2+l} \bmod nk + 1$$

이고 $A_j = A_{nk+1-j}$이므로

$$\sum_{j=1}^{nk} A_{nk+1-j} B_j = \sum_{j=1}^{nk} A_j B_j$$

이다. 따라서 정수로 보면

$$\begin{aligned}kC_P = \overline{C_P} &= \overline{A_P}\,\overline{B_P} + \sum_{j=1}^{nk} A_{nk+1-j} B_j \\ &= kA_P kB_P + k\sum_{i=1}^{n} a_i b_i\end{aligned}$$

이다. 그러므로 $C_P(pred) = \sum_{i=1}^{n} a_i b_i$ 이다. □

IV. 복잡도 분석

4.1 직렬 곱셈기

정규기저에 관한 직렬 곱셈기는 2005년에 A. Reyhani-Masolleh와 M.H.Hasan[12]이 AND, XOR, Time Delay 부분에서 효율적인 것을 제안하

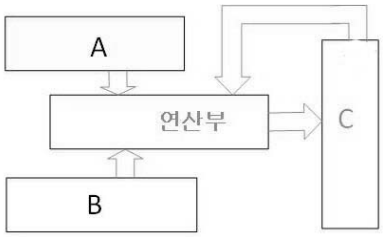

Fig.2. n-bit Output Serial-Multiplier

였다. C.H. Kim 등[13]은 타입 IV의 경우 A. Reyhani-Masolleh의 곱셈기와 같은 복잡도를 가지나 타입 X 의 경우 더 효율적인 Time Delay를 갖는 곱셈기를 제안하였다. 이 곱셈기들은 n-bit Input, n-bit Ouput으로 구성된 곱셈기이다. 이 곱셈기에 3장의 내용을 추가한 오류 탐지 곱셈기에 관하여 살펴보자. 먼저 직렬곱셈기는 Fig.2.와 같이 두 입력 A, B에 대하여 n 클락(clock)후 곱셈 결과를 n 비트 출력하도록 구성되어 있다.

타입 k (even) 가우시안 정규기저를 갖는 유한체 $GF(2^n)$의곱셈 오류 탐지를 위하여 $\sum_{i=1}^{n} a_i b_i$를 계산하는 것이 필요하다. 그리고 곱셈기 결과 값과 비교하기 위한 $C_P(Act)$ 계산이 필요하고 마지막으로 $C_P(Act)$와 $C_P(pred)$의 XOR 연산이 필요하다. 구체적으로 보면 $\sum_{i=1}^{n} a_i b_i$ 계산은 곱셈기가 연산하는 동안 1 번의 AND와 XOR로 구성된다. $C_P(Act)$의 계산은 곱셈이 끝난 후에 이루어진다. $GF(2^5)$의 구체적인 예는

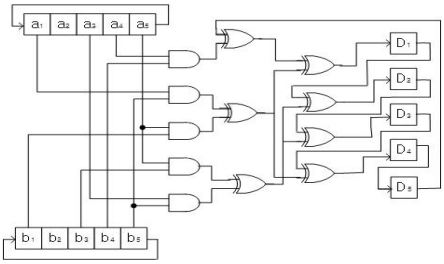

Fig.3(a). Serial-Multiplier of $GF(2^5)$

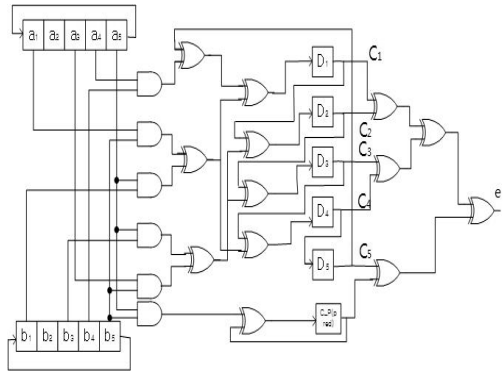

Fig.3(b). Error Detection Serial-Multiplier of $GF(2^5)$

Table 1. Complexities and overhead of Error Detection Serial-Multiplier

Serial multipliers		Complexities = original + [overhead]			Overhead*(%)
Reyhani-Masoleh [12]	Kim[13]	General	AND	n +[1]	
			XOR	#XOR+[n+1], #XOR $\le (C_n+n)/2$	
			#1-bit reg.	3n + [1]	
			Delay	D_G + [$\lceil \lg n+1 \rceil$] T_X , $D_G \le n(T_A+(1+\lceil \lg k \rceil)T_X)$	
	Yang[8], Kwon[14]	Type II	AND	n + [1]	0.4
			XOR	$(3n-1)/2$ +[n+1]	67
			#1-bit reg.	3n +[1]	0.14
			Delay	$n(T_A+2T_X)+(\lceil \lg(n+1) \rceil)T_X$	1.1
	Kim[13]	Type IV	AND	n + [1]	0.2(0.6)
			XOR	$(5n-7)/2+[n+1]$	40.2(40.59)
			#1-bit reg.	3n + [1]	0.08(0.2)
			Delay	$n(T_A+3T_X)+(\lceil \lg(n+1) \rceil)T_X$	0.7(1.23)
		Type X	AND	n + [1]	0.175
			XOR	$(11n-73)/2+[n+1]$	18.4
			#1-bit reg.	3n + [1]	0.058
			Delay	$n(T_A+5T_X)+(\lceil \lg(n+1) \rceil)T_X$	0.29

* $GF(2^n)$:NIST recommended field, II:233, IV:409,163, X:571, overhead/original×100%

Fig.3.과 같다. Fig.3(a).는 곱셈기이고 Fig.3(b).는 오류 탐지가 포함된 연산기이다[15]. $GF(2^5)$의 경우는 타입 II 최적 정규기저를 갖는 유한체로 전체 gate(AND+XOR)의 개수는 $m+(3m-1)/2$개 이고 $1T_A+2T_X$의 지연이 있는 곱셈기이다[14,15]. 2006년에 Reyhani-Masoleh [12]는 전체 gate(AND+XOR)의 개수가 타입IV인 경우 $(7n-3)/2$이고 일반적으로는 $(C_n+3m)/2$인 것을 제안하였다. 이러한 모든 경우에 오류 탐지 $C_P(pred)$에 필요한 gate 수는 AND gate 1개, XOR gate 1개, 1-bit register 1개가 필요하다. 구체적인 것은 표 1에 제시하였다. 시간 복잡도는 타입 k의 경우 1 clock 당 $T_A+(1+\lg\lceil k\rceil)T_X$ 이 필요하나 전체 계산 결과를 출력하는데 필요한 n clock을 고려하면 n $(T_A+(1+\lg\lceil k\rceil)T_X)$ 의 시간 복잡도가 필요하다. 오류 탐지 곱셈기는 A와 B의 곱 결과의 bit와 $C_P(pred)$의 XOR에 필요한 $\lg\lceil n+1\rceil T_X$가 추가로 더 필요하다. 복잡도는 Table 1.에 제시하였다.

4.2 병렬 곱셈기

병렬 곱셈기는 Fig.1.의 CUT블록에서 $A\cdot B$ 를 병렬로 계산해서 결과를 출력한다. [16]의 유한체 $GF(2^3)$의 곱셈기는 Fig.4(a).와 같다. 또한 이 곱셈기에 오류 확인 연산을 추가하면 Fig.4(b).이다.

일반적으로 타입 k 가우시안 정규기저 병렬곱셈기는 $(nC_N+3n^2-2n)/2$ 이하의 공간복잡도를 가지며 타입 IV인 경우는 $(7n^2-9n)/2$이다[17,18]. 오류 탐지 곱셈기에는

$$\sum_{i=1}^{n}a_ib_i,\ C_P(Act),\ C_P(act)+C_P(pred)$$

계산에 $2n-1$개의 gate가 필요하다. 시간 복잡도는 $T_A+\lceil\lg C_N\rceil T_X$ 이하이고 오류 확인에는 $\sum_{i=1}^{n}a_ib_i$ 는 곱셈기와 같은 타임에 계산하므로 오류 확인에는 $C_P(Act)$, $C_P(act)+C_P(pred)$ 계산에 필요한 $\lceil\lg(n+1)\rceil T_X$가 더 필요하다. 유한체 $GF(2^3)$의 병렬 곱셈기와 병렬 오류 탐지 곱셈기의 예는 Fig.4(b).와 같다. 전제적인 복잡도는 Table 2.에 제시하였다.

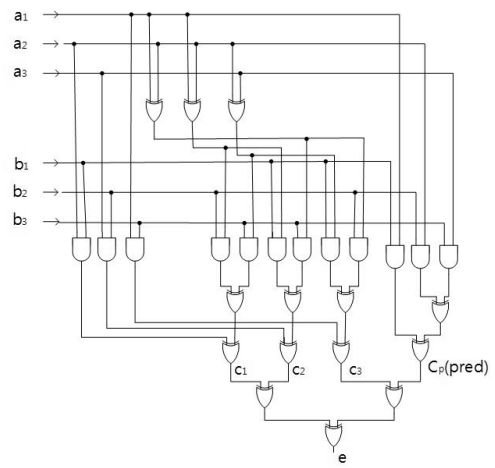

Fig.4(a). Parallel-Multiplier of $GF(2^3)$

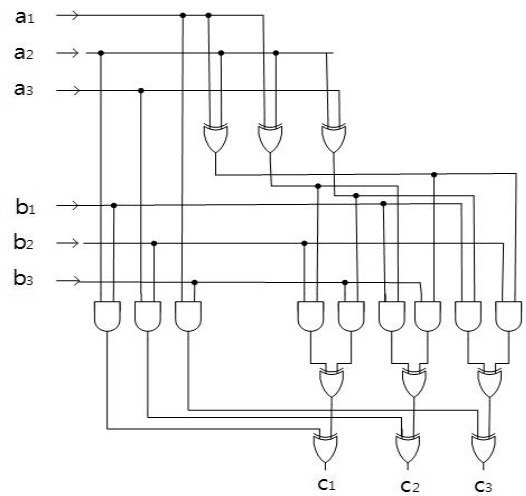

Fig.4(b). Error Detection Parallel-Multiplier of $GF(2^3)$

V. 결 론

정보보호 관련 오류 주입 공격에 대비하기 위한 유한체 연산의 오류 탐지 방법에 대한 관심이 높아지고 있다. 이 논문에서는 가우시안 정규기저를 사용한 곱셈 연산의 오류 탐지 방법을 제안하였다. 타입 k(even)의 유한체에 대하여 병렬곱셈기의 경우 n 개의 AND gate 와 $2n-1$ 개의 XOR gate, n-bit 출력의 직렬곱셈기의 경우 1 개의 AND gate, $n+1$ 개의 XOR gate, 1 개의 1-bit Register가 필요한 오류 탐지 방법을 제안하였다. 시간복잡도는 타입 IV 정규 기저를 갖는 $n=163$인 경우 직렬곱셈기는

Table 2. Complexities and overhead of Error Detection Parallel-Multiplier

Parallel multipliers		Complexities = original + (overhead)			Overhead*(%)
Reyhani-Masoleh [18]	Kim[17]	General	AND	n^2 +(n)	
			XOR	#XOR+[2n−1], #XOR ≤ n(C_n +n−2)/2	
			Delay	D_G + [⌈lg n+1⌉ T_X] , $D_G \le T_A+(1+\lceil \lg C_N \rceil)T_X$	
	Kim[16]	Type II	AND	n^2 + (n)	0.4
			XOR	$3n(n-1)/2$ +(2n-1)	0.57
			Delay	$T_A+(1+\lceil \lg n \rceil T_X+(\lceil \lg(n+1) \rceil)T_X$	80
	Kim[17]	Type IV	AND	n^2 + (n)	0.24(0.6)
			XOR	$n(5n-7)/2+[2n-1]$	0.196(0.49)
			Delay	$T_A+\lceil \lg(4n-7) \rceil T_X+(\lceil \lg(n+1) \rceil)T_X$	75(72.7)
		Type X	AND	n^2 +(n)	0.175
			XOR	$n(11n-75)/2+[2n-1]$	0.06
			Delay	$T_A+\lceil \lg(10n-73) \rceil T_X+[\lceil \lg(n+1) \rceil]T_X$	71.4

* $GF(2^n)$:NIST recommended field, II:233, IV:409, 163, X:571, overhead/original×100%

1.23%, 병렬곱셈기는 72.7% 증가한다. 모든 경우 $C=A\cdot B$ 연산에서 홀수 비트 개의 오류가 발생하면 탐지할 수 있다.

References

[1] R.Lidl and H. Niederreiter, " Introduction to finite fields and their applications," Cambridge University Press, Revised Edition, 1994.

[2] National Institute for of Standards and Technology, "Digital Signature Standard," FIPS Publication 186-2, Jan. 2000.

[3] IEEE Std. 1363-2000, "IEEE Standard Specifications for Public - key Cryptography," Jan. 2000.

[4] A.J. Menezes, I.F. Blake, X. Gao, R.C. Muliin, S.A. Vanstone, and T. Yaghoobian, "Applications of Finite Fields," Kluwer Academic Publishers, 1993.

[5] E.R. Berlekamp, "Bit-Serial Reed-Solomon Encoders," IEEETrans. on info. Theory, vol. 28, pp. 869-874, Nov. 1982.

[6] M. Joye and M. Tunstall, "Fault Analysis in Cryptography," Springer, 2012.

[7] S. Fenn, M. Gossel, M. Benassia, and D. Taylor. "On-Line Error Detection for Bit-Serial Multipliers in $GF(2^m)$," Journal of Electronic Testing : Theory and Applications, vol. 13, pp. 29-40, Aug. 1998.

[8] A. Reyhani-Masoleh and M.H. Hasan, "Fault Detection Architectures for Field Multiplication Using Polynomial Bases," IEEE Trans. computers Vol. 55, no. 9, pp. 1089-1103, Sep. 2006.

[9] A. Hariri and A. Reyhani-Masoleh, "Fault Detection structures for the Montgomery Multiplication over Binary Extension Fields," 2007 Workshop on Fault and Tolerance in Cryptography, pp. 37-42, Sep. 2007.

[10] C.Y. Lee, C.W. Chiou, and J.M. Lin, "Concurrent Error Detection in a Polynomial Basis Multiplier over $GF(2^n)$," Journal of Electronic Testing: Theory and Applications, vol. 22, no. 2, pp. 143-150, Apr. 2006.

〔11〕 C.W. Chiou, C.C. Chang, C.Y. Lee, and T.W. Hou, "Concurrent Error Detection and Correction in Gaussian Normal Basis Multiplier over GF(2^m)," IEEE Trans. computers Vol. 58, no. 6, pp. 851-857, June. 2009.

〔12〕 A. Reyhani-Masoleh, "Efficient Algorithms and Architecture for Finite Multiplication Using Gaussian Normal Bases," IEEE Trans. computers, vol. 55 no. 1, pp. 34-47, Jan. 2006.

〔13〕 C.H. Kim, N.S. Chang and Y.I. Cho, "Modified Sequential Multipliers for Type-k Gaussian Normal Bases," MUE 2011, pp. 220-225, June. 2011.

〔14〕 S. Kwon, K. Gaj, C.H. Kim, and C.P. hong, "Efficient Linear Array for Multiplication in $GF(2^m)$ Using a Normal Basis for Elliptic Curve Cryptography," CHES 2004, LNCS 3156, pp. 76-91, 2004.

〔15〕 D.J. Yang, C.H. Kim, Y. Park, and J. Lim, "Modified Sequential Normal Basis Multipliers for Type II Optimal Normal Basis," ICCA 2005, LNCS 3481, pp. 647-656, 2005.

〔16〕 C.H. Kim, Y. Kim, S.Y. Ji and I. Park, "A New Parallel Multiplier for Type II Optimal Normal Basis," CIS 2006, pp.460-469, 2006.

〔17〕 C.H. kim and Y. Kim, "A parallel Architecture for Type k Gaussian Normal Basis Multiplier over $GF(2^n)$," The Workshop of 2005 International Conference on Computational Intelligence and Security, pp. 109-114, Dec. 2005.

〔18〕 A. Reyhani-Masoleh and M.H. Hasan, "A New Construction of Massey-Omura Parallel Multiplier over $GF(2^m)$," IEEE Trans. computers, Vol. 51, no. 5, pp. 512-520, May. 2002.

〈저 자 소 개〉

김 창 한 (Chang Han Kim) 종신회원
1985년 2월: 고려대학교 수학과 이학사
1987년 2월: 고려대학교 수학과 이학석사
1992년 2월: 고려대학교 수학과 이학박사
1992년~현재: 세명대학교 정보통신학부 교수
〈관심분야〉 정수론, 공개키암호, 암호프로토콜

장 남 수 (Nam Su Chang) 종신회원
2002년 2월: 서울 시립대학교 수학과 이학사
2004년 8월: 고려대학교 정보보호대학원 공학석사
2010년 2월: 고려대학교 정보경영공학전문대학원 공학박사
2010년 7월~현재: 세종사이버대학교 정보보호학과 조교수
〈관심분야〉 암호칩 설계 기술, 부채널 공격, 공개키 암호 알고리즘, 공개키 암호 암호분석

박 영 호 (Young-Ho Park) 종신회원
1990년 2월: 고려대학교 수학과 이학사
1993년 2월: 고려대학교 수학과 이학석사
1997년 2월: 고려대학교 수학과 이학박사
2002년 3월~현재: 세종사이버대학교 정보보호학과 부교수
〈관심분야〉 정수론, 공개키암호, 암호프로토콜, 부채널분석