Polynomial basis 방식의 3배속 직렬 유한체 곱셈기

문상국
목원대학교 정보전자공학부

3X Serial GF(2m) Multiplier on Polynomial Basis Finite Field

Sangook Moon
Mokwon University, School of Electronics & Information Engineering
E-mail : smoon@mokwon.ac.kr

요 약

정보 보호 응용에 새로운 이슈가 되고 있는 ECC 공개키 암호 알고리즘은 유한체 차원에서의 효율적인 연산처리가 중요하다. 직렬 유한체 곱셈기의 근간은 Mastrovito의 직렬 곱셈기에서 유래한다. 본 논문에서는 polynomial basis 방식을 적용하고 식을 유도하여 Mastovito의 직렬 유한체 곱셈방식의 3배 성능을 보이는 유한체 곱셈기를 제안하고, HDL로 기술하여 기능을 검증하고 성능을 평가한다. 설계된 3배속 직렬 유한체 곱셈기는 부분합을 생성하는 회로의 추가만으로 기존 직렬 곱셈기의 3배의 성능을 보여주었다.

ABSTRACT

Efficient finite field operation in the elliptic curve (EC) public key cryptography algorithm, which attracts much of latest issues in the applications in information security, is very important. Traditional serial finite multipliers root from Mastrovito's serial multiplication architecture. In this paper, we adopt the polynomial basis and propose a new finite field multiplier, inducing numerical expressions which can be applied to exhibit 3 times as much performance as the Mastrovito's. We described the proposed multiplier with HDL to verify and evaluate as a proper hardware IP. HDL-implemented serial GF (Galois field) multiplier showed 3 times as fast speed as the traditional serial multiplier's adding only partial-sum block in the hardware.

키워드
elliptic curve (EC), public key cryptography, finite field, finite field multiplier, HDL

Ⅰ. 서 론

타원 곡선 연산에 필수적으로 사용되는 유한체 곱셈 연산기에는 크게 직렬 유한체 곱셈기, 배열 유한체 곱셈기, 하이브리드 유한체 곱셈기가 존재한다. 직렬 유한체 곱셈기는 Mastrovito에 의하여 제안되어 유한체 곱셈기의 가장 기본적인 구조로 자리잡아 왔고 [1], 이를 병렬로 처리하기 위해 m배의 자원을 투자하여 m배의 속도를 얻어낸 결과가 2차원 배열 유한체 곱셈기이다 [2]. 배열 유한체 곱셈기는 가격대 성능비에 있어서 효율이 떨어지는 반면, 이런 기존 방식의 장점만을 취하여 제안된 방식이 가격대 성능비가 가장 우수한 하이브리드 곱셈기이다 [3]. 하지만 이 하이브리드 곱셈기는 사용 가능한 유한체로서 차수가 합성수인 합성수 유한체를 사용하기 때문에 암호학적인 안전도가 떨어져 많은 응용 분야에의 적용에 힘들다. 이에 Mastrovito의 곱셈기를 새롭게 응용하여 제안된 곱셈기가 공통인수 후처리방식에 기반한 유한체 곱셈기이다 [4]. 본 논문에서는 공통인수 후처리방식에 기반한 유한체 곱셈기를 응용하여 기존 직렬 유한체 곱셈기의 성능을 3배로 확장한 곱셈기를 구현하여 HDL로 구현하여 기능을 검증하고 성능을 평가한다.

II. 공통인수 후처리방식 유한체 곱셈기

이 방식은 Mastrovito의 직렬 곱셈기의 구조를 바탕으로 한다. 유한체 원소 표현 방식으로는 표준 기저 방식을 사용하며, 주어진 유한체로서 GF(2m)을 사용한다. 주어진 유한체의 임의의 두 원소를 각각 $A(x)=\sum_{i=0}^{m-1}a_ix^i$, $B(x)=\sum_{i=0}^{m-1}b_ix^i$라고 하면 곱셈결과 값인 $Z(x)$는 다음 식 (1)과 같이 나타내어지고, 이를 전통적인 직렬 곱셈기의 형태로 구현하면 그림 (1)과 같다.

$$Z(x)=A(x)\sum_{i=0}^{m-1}b_ix^i=\sum_{i=0}^{m-1}b_i(x^iA(x)) \bmod P(x)$$
$$=[b_0A(x)+\ldots+b_{m-1}x^{m-1}A(x)] \bmod P(x) \quad (1)$$

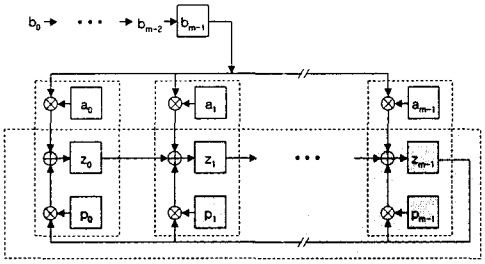

그림 1. Mastrovito의 직렬 곱셈기

공통인수 후처리방식의 유한체 곱셈기의 핵심적인 부분은 기준이 되는 식을 분리하는 것이다. 식 (1)을 t등분으로 분리하여 각각 하드웨어 자원을 할당함으로써 t배의 성능 향상을 유도하는 것이다. t배의 성능을 발휘하는 확장 구조식은 다음식 (2)와 같이 정리된다.

$$Z_{tk}(x)=[b_0A(x)+b_tx^tA(x)+\ldots]\bmod P(x)$$
$$Z_{tk+1}(x)=x[b_1A(x)+b_{t+1}x^tA(x)+\ldots]\bmod P(x)$$
$$Z_{tk+2}(x)=x^2[b_2A(x)+b_{t+2}x^tA(x)+\ldots]\bmod P(x)$$
$$\vdots$$
$$Z_{tk+(t-1)}(x)=x^{t-1}[b_{t-1}A(x)+\ldots]\bmod P(x) \quad (2)$$

공통인수 후처리 방식의 곱셈기의 구현에는 연산 블록별로 해당되는 제어 mux와 xt 제곱 연산기가 필요하다. 이 구조의 가장 큰 장점은, 기존의 직렬 곱셈기와 거의 비슷한 최대 임계 경로 (critical path delay)를 가지므로 고속 연산에 적하다.

III. 3배속 직렬 유한체 곱셈기

3배속 직렬 유한체 곱셈기를 구현하기 위해, 식 (2)를 변형하면 다음식 (3)과 같이 나타낼 수 있다.

$$Z_{3k}(x)=[b_0A(x)+b_{43}x^3A(x)+\ldots]modP(x)$$
$$Z_{3k+1}(x)=x[b_1A(x)+b_4x^3A(x)+\ldots]modP(x) \quad (3)$$
$$Z_{3k+2}(x)=x^2[b_2A(x)+b_5x^3A(x)+\ldots]modP(x)$$

식 (3)의 첫 번째 식을 분석해보면, Mastrovito의 직렬 곱셈방식의 기본적인 형태로 이루어져 있지만, 곱셈의 차수가 3씩 증가한다는 점을 알 수가 있고, 두 번째, 세 번째 식은 첫 번째 기본적인 수식에 b의 차수만 다르고 같은 형태를 취하며, 각각 최종 합에 x와 x2을 곱해서 처리해야 하는 구조로 이루어져 있다는 점을 알 수가 있다.

3배속 직렬 유한체 곱셈 구현의 핵심적인 부분이 되는 x3A(x) 회로의 구조를 얻기 위해서 다음식 (4)와 같이 정리할 수 있다.

$$x^3A(x)=a_0x^3+a_1x^4+\ldots+a_{m-4}x^{m-1}$$
$$+a_{m-3}x^m+a_{m-2}x^{m+1}+a_{m-1}x^{m+2} \quad (4)$$

이 식에서 xm, xm+1, xm+2 항을 변형하여 최고차 항이 xm-1 이 되도록 만들기 위해, 오른쪽 끝에서 3 항을 다음 식 (5)와 같이 정리하였다.

$$x^m=p_0+p_1x+\ldots+p_{m-2}x^{m-2}+p_{m-1}x^{m-1}$$
$$x^{m+1}=p_0x+p_1x^2+\ldots+p_{m-2}x^{m-1}+p_{m-1}x^m \quad (5)$$
$$x^{m+2}=p_0x^2+p_1x^3+\ldots+p_{m-2}x^m+p_{m-1}x^{m+1}$$

여기서 계산을 간편하게 하기 위하여 소수 다항식 (primitive polynomial)의 특성을 이용한다. 높은 암호학적인 복잡도의 특성을 나타내는 소수 다항식은 주로 3항 다항식 (trinomial)이나 5항 다항식 (pentanomial)이 사용된다 [5]. 이 3항 다항식이나 5항 다항식 공통으로 해당되는 특성은 계수 pm-1과 pm-2가 각각 0의 값을 가진다는 것이다. 따라서 이 성질을 이용하면 식 (5)는 다음과 같이 간이화되어 차수를 m-1차로 줄일 수 있다.

$$x^{m+1}=p_0x+p_1x^2+\ldots+p_{m-2}x^{m-1}$$
$$x^{m+2}=p_0x^2+p_1x^3+\ldots+p_{m-3}x^{m-1} \quad (6)$$

식 (5)와 식 (6)을 식 (4)에 대입하여 정리하면 다음과 같은 식 (7)을 얻는다.

$$
\begin{aligned}
x^3A(x) &= a_{m-3}p_0 \\
&+ (a_{m-3}p_1 + a_{m-2}p_0)x \\
&+ (a_{m-3}p_2 + a_{m-2}p_1 + a_{m-1}p_0)x^2 \\
&+ (a_0 + a_{m-3}p_3 + a_{m-2}p_2 + a_{m-1}p_1)x^3 \\
&+ (a_1 + a_{m-3}p_4 + a_{m-2}p_3 + a_{m-1}p_2)x^4 \\
&+ \cdots \qquad (7) \\
&+ (a_{m-4} + a_{m-3}p_{m-1} + a_{m-2}p_{m-2} + a_{m-1}p_{m-3})x^{m-1}
\end{aligned}
$$

식 (7)을 회로도로 표현하면 3배속 유한체 곱셈기의 핵심이 되는 x3 곱셈 회로인 다음 그림 2로 나타낼 수 있다.

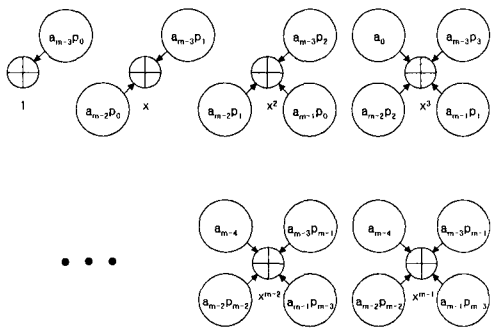

그림 2. x3 곱셈 회로의 블록도

IV. 구현 및 성능 평가

그림 2에서 나타낸 x3 곱셈 회로를 이용하여 식 (3)에 적용시켜 그림 3과 같은 3배속 유한체 곱셈기를 HDL로 구현하였다. 그림 3에서 각각의 Z 모듈 블록에는 그림 2에서 얻은 x3 곱셈 회로를 사용하였다.

3배속 유한체 곱셈기를 검증하기 위해서, 동작이 검증된 Mastrovito의 직렬 곱셈기를 C 언어로 구현하고 테스트 벡터로서 193비트 임의의 난수를 발생시켜 입력하였다. C 언어로 기술된 직렬 곱셈기는 알고리즘에 기초하여 기술되었기 때문에 오류를 발생시키지 않으며 결과는 알고리즘의 흐름에 따른 부분 결과를 살핌으로써 확인되었다. 다음으로 본 논문의 III장에서 제시한 방법에 의해 유한체 곱셈기를 C 언어로 구현한 다음 결과가 확인된 C 프로그램의 결과와 비교하여 결과가 맞음을 확인하여 수식적으로 검증된 알고리즘을 실제 구현적으로 검증하는 것을 확인하였다.

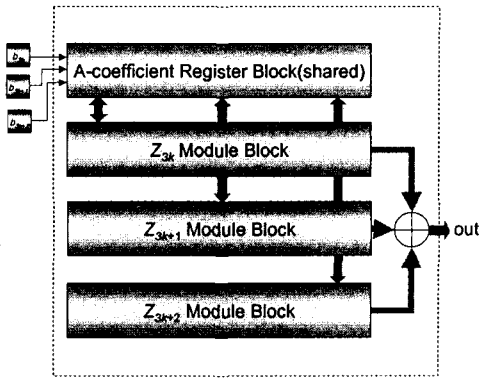

그림 3. 3배속 유한체 곱셈기

마지막으로 검증된 알고리즘을 HDL로 구현하여 그 결과를 C 프로그램을 통하여 얻은 결과와 비교하여 올바른 결과를 산출하는지 확인하였다. 합성 결과는 표 1과 같고 이 결과는 유한체 곱셈기가 적용될 수 있는 IC 카드에는 물론 고속 어플리케이션에 사용되기에 충분한 결과를 보인다.

표 1. 193비트 3배속 유한체 곱셈기 합성결과

	3배속 유한체 곱셈기
임계경로 지연	0.7ns
동작 주파수	1.43 GHz
게이트 수	6,563

V. 결 론

본 논문에서는 타원 곡선 암호 시스템에서 적은 면적을 소비하면서도 빠른 암호화, 복호화를 수행할 수 있는 polynomial 기저 직렬 3배속 유한체 곱셈기를 구현하여 HDL로 기술하여 그 성능을 검증하고 평가하였다. 구현의 의도는 허용하는 만큼의 자원을 투자하여 성능을 극대화 하는데 초점을 맞추었다. 설계 및 합성 결과는 IC 카드와 같이 수행 연산 규모나 자원 규모가 제한되는 어플리케이션에 적합한 것으로 나타났다. 특히, 차세대 비밀 키 암호 알고리즘인 AES에 채택된 Rijndael에는 내부적으로 8비트 유한체 곱셈 방식을 사용하기 때문에 차후 이 8비트 유한체 곱셈이 큰 비트 수로 확장이 될 경우 본 논문에서 제시한 곱셈 방식이 적용될 수도 있어 많은 응용을 기대할 수 있다.

참고문헌

[1] E. D. Mastrovito, "VLSI Architectures for

Computations in Galois Fields", Linkoping Studies in Science and Technology Dessertations, No. 242, 1991.

[2] C. Wang and J. Lin, "Systolic Array Implementation of Multipliers for Finite Fields GF(2m)", IEEE Transactions on Circuits and Systems, vol. 38, no. 7, pp. 796-800, Jul. 1991.

[3] C. Paar, "Efficient VLSI Architectures for Bit-Parallel Computation in Galois Fields", Ph.D. thesis, Institute for Experimental Mathematics, University of Essen, Essen, Germany, Jun. 1994.

[4] 문상국, "타원곡선 암호용 프로세서를 위한 고속 VLSI 알고리즘의 연구와 구현", 연세대학교 대학원 박사학위 논문집, 2002.

[5] L. Song, "Low-Power VLSI Architectures for Finite Field Applications", Ph.D. thesis, UMI Microform 9935004, 1999.