JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 23, No. 12: 1609~1617, Dec. 2019

공개키 암호 구현을 위한 경량 하드웨어 가속기

성병윤[1] · 신경욱[2*]

A Lightweight Hardware Accelerator for Public-Key Cryptography

Byung-Yoon Sung[1] · Kyung-Wook Shin[2*]

[1]Engineer, Nextchip Co. Ltd., Gyenoggi-do, 13493 Korea
[2*]Professor, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, 39177 Korea

요 약

ECC (Elliptic Curve Cryptography)와 RSA를 기반으로 하는 다양한 공개키 암호 프로토콜 구현을 지원하는 하드웨어 가속기 설계에 관해 기술한다. NIST 표준으로 정의된 소수체 상의 5가지 타원곡선과 3가지 키길이의 RSA를 지원하며 또한, 4가지 타원곡선 점 연산과 6가지 모듈러 연산을 지원하도록 설계되어 ECC와 RSA 기반 다양한 공개키 암호 프로토콜의 하드웨어 구현에 응용될 수 있다. 저면적 구현을 위해 내부 유한체 연산회로는 32 비트의 데이터 패스로 설계되었으며, 워드 기반 몽고메리 곱셈 알고리듬, 타원곡선 점 연산을 위해서는 자코비안 좌표계, 그리고 모듈러 곱의 역원 연산을 위해서는 페르마 소정리를 적용하였다. 설계된 하드웨어 가속기를 FPGA 디바이스에 구현하여 EC-DH 키교환 프로토콜과 RSA 암호·복호 등작을 구현하여 하드웨어 동작을 검증하였다. 180-nm CMOS 표준 셀 라이브러리로 합성한 결과, 50 MHz 클록 주파수에서 20,800 등가게이트와 28 kbit의 RAM으로 구현되었으며, Virtex-5 FPGA 디바이스에서 1,503 슬라이스와 2개의 BRAM으로 구현되었다.

ABSTRACT

Described in this paper is a design of hardware accelerator for implementing public-key cryptographic protocols (PKCPs) based on Elliptic Curve Cryptography (ECC) and RSA. It supports five elliptic curves (ECs) over $GF(p)$ and three key lengths of RSA that are defined by NIST standard. It was designed to support four point operations over ECs and six modular arithmetic operations, making it suitable for hardware implementation of ECC- and RSA-based PKCPs. In order to achieve small-area implementation, a finite field arithmetic circuit was designed with 32-bit data-path, and it adopted word-based Montgomery multiplication algorithm, the Jacobian coordinate system for EC point operations, and the Fermat's little theorem for modular multiplicative inverse. The hardware operation was verified with FPGA device by implementing EC-DH key exchange protocol and RSA operations. It occupied 20,800 gate equivalents and 28 kbits of RAM at 50 MHz clock frequency with 180-nm CMOS cell library, and 1,503 slices and 2 BRAMs in Virtex-5 FPGA device.

키워드 : 타원곡선 암호, 정보보안, 공개키 암호, RSA, 하드웨어 가속기

Keywords : Elliptic Curve Cryptography, Information Security, Public-key Cryptography, RSA, Hardware Accelerator

Received 29 October 2019, **Revised** 6 November 2019, **Accepted** 12 November 2019

*** Corresponding Author** Kyung-Wook Shin(E-mail:kwshin@kumoh.ac.kr, Tel:+82-54-478-7427)
Professor, School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, 39177 Korea

Open Access **http://doi.org/10.6109/jkiice.2019.23.12.1609** print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

공개키 암호 (public-key cryptography)는 공개키와 개인키 (private key)라는 한 쌍의 서로 다른 키가 사용되는 정보보안의 한 방식이며, 공개키로 암호화하고 특정 개인키(비밀키)로 복호화하는 공개키 암호・복호와 특정 개인키(비밀키)로 암호화되었음을 공개키로 확인하는 공개키 서명으로 나눌 수 있다. 공개키 암호 방식은 전자서명, 인증뿐만 아니라 키교환, ECIES (Elliptic Curve Integrated Encryption Scheme) 등 다양한 정보보안 프로토콜에 사용되고 있으며, 사물인터넷 (IoT)을 비롯하여 스마트 의료 및 헬스기기 보안, 자율주행 자동차와 드론의 통신 보안, 암호화폐 등으로 응용분야가 급속히 확대되고 있다 [1-3].

공개키 암호 방식으로 RSA (Rivest, Shamir, and Adleman) [4]와 타원곡선 암호 (Elliptic Curve Cryptography; ECC) [5-6]가 널리 사용되고 있다. ECC는 RSA에 비해 짧은 키길이를 사용하여 비슷한 수준의 안전성을 얻을 수 있어 경량 하드웨어로 구현될 수 있다는 장점을 가지며, 또한 연산량이 작아 고속 처리가 요구되는 분야에 적합하다. 최근에는 ECC 기반의 공개키 암호 방식이 기존의 RSA 기반 공개키 암호 방식을 급속히 대체하고 있는 추세이다.

최근, 제한된 리소스를 갖는 응용분야의 보안에 적합하도록 경량화된 공개키 암호 프로세서 구현에 관한 연구가 활발하게 이루어지고 있다. NIST에서 정의한 이진체 (binary field) 상의 타원곡선을 지원하는 ECC 프로세서 설계사례 [7-8]와 소수체 (prime field) 상의 타원곡선을 지원하는 ECC 프로세서 설계사례 [9-10] 등이 발표되었으며, 문헌 [11]에는 소수체 상의 타원곡선과 RSA를 지원하는 공개키 암호 프로세서 설계사례가 발표되었다.

본 논문에서는 ECC와 RSA 기반의 공개키 암호 프로토콜의 하드웨어 구현을 위한 공개키 암호 하드웨어 가속기 코어 PKCC (Public-Key Cryptography Core)를 설계하였다. ECC를 위한 점 연산과 모듈러 멱승, 모듈러 곱셈, 모듈러 나눗셈, 모듈러 역원 등의 유한체 연산기능을 가져 다양한 공개키 암호 프로토콜의 경량 하드웨어 구현에 적합하도록 설계하였다. 2장에서는 ECC와 RSA 공개키 암호 방식에 대해 간략히 소개하고, 3장에서는 PKCC의 구조와 동작모드에 관해 설명한다. 4장에서는 PKCC를 FPGA에 구현하여 공개키 암호 프로토콜의 하드웨어 동작을 검증한 결과와 성능평가를 기술하고, 5장에서 결론을 맺는다.

Ⅱ. ECC와 RSA 공개키 암호

ECC는 Neil Koblitz [5]와 Victor Miller [6]에 의해 1985년에 제안된 공개키 암호 방식이며, 소수체 또는 이진체 상의 타원곡선 군 (group)을 기반으로 한다. 소수체 상의 타원곡선은 식 (1)과 같이 정의되며, 유한체 종류와 크기에 따라 생성점, 생성점의 위수 (order), 타원곡선 계수 (a, b) 등의 도메인 파라미터가 NIST FIPS PUB 186-3 [12], SEC 2 [13]에 정의되어 있다.

$$GF(p) : y^2 = x^3 + ax + b, \ (4a^3 + 27b^2 \neq 0) \quad (1)$$

타원곡선 암호는 식 (2)로 표현되는 점 스칼라 곱셈 (Point Scalar Multiplication; PSM)을 기반으로 공개키 생성, 암호화 및 복호화가 이루어진다.

$$Q = kP \quad (2)$$

식 (2)에서 임의의 정수 k는 개인키로 사용되며, 타원곡선 상의 임의의 두 점 P와 Q를 알더라도 비밀키 k를 알아내기 어렵다는 타원곡선 군의 이산대수 문제 (discrete logarithm problem)에 안전성 기반을 갖는다.

식 (2)의 PSM은 점 덧셈 (Point Addition; PA)과 점 두배 (Point Doubling; PD) 연산의 반복으로 계산되며, 점 연산은 타원곡선 상의 점을 표현하는 좌표계에 따라 하위계층의 유한체 (finite field) 연산이 달라진다. 타원곡선 상의 점을 표현하기 위해 2차원 좌표계인 아핀 (affine) 좌표계 또는 3차원 좌표계인 투영 (projective) 좌표계가 사용된다. 타원곡선 PA과 PD 연산은 좌표계에 따라 표 1과 같은 유한체 연산 수식으로 계산된다 [14]. PA과 PD를 아핀 좌표계에서 연산하면 모듈러 나눗셈 연산이 포함되어 복잡한 하드웨어와 많은 클록 사이클이 소요된다. 반면에, 투영 좌표계의 한 형태인 자코비안 좌표계를 이용하면 모듈러 나눗셈 연산 없이 모듈러 덧셈, 뺄셈, 곱셈, 제곱 연산으로 구현된다. 본 논문에서는 자코비안 좌표계를 적용하여 나눗셈 연산 없이 PA과 PD이 계산되도록 하였다.

Table. 1 Equations for computing point addition and point doubling operations according to the coordinate system

Coordinate	Point addition	Point doubling
Affine	$\lambda = (y_1 - y_0)/(x_1 - x_0)$ $x_2 = \lambda^2 - x_0 - x_1$ $y_2 = \lambda(x_0 - x_2) - y_0$	$\lambda = (3x_0^2 + a)/(2y_0)$ $x_2 = \lambda^2 - 2x_0$ $y_2 = \lambda(x_0 - x_2) - y_0$
Jacobian	$U_0 = X_0 Z_1^2,\ U_1 = X_1 Z_0^2$ $S_0 = Y_0 Z_1^3,\ S_1 = Y_1 Z_0^3$ $H = U_1 - U_0,\ R = S_1 - S_0$ $X_2 = -H^3 - 2U_0 H^2 + R^2$ $Y_2 = -S_1 H^3 + R(U_0 H^2 - X_2)$ $Z_2 = Z_0 Z_1 H$	$M = 3X_0^2 + aZ_0^4$ $T = -2S + M^2$ $S = 4X_0 Y_0^2$ $X_2 = T$ $Y_2 = -8Y_0^4 + M(S - T)$ $Z_2 = 2Y_0 Z_0$

공개키 (e, N)를 이용한 평문 M의 RSA 암호화는 식 (3)과 같이 모듈러 (modular) 멱승으로 표현되며, 암호문 C가 얻어진다. 암호문 C를 개인키 (d, N)로 RSA 복호화하면 평문 M이 얻어지며, RSA 복호화는 (4)와 같이 표현된다.

$$C = M^e \bmod N \tag{3}$$

$$M = C^d \bmod N \tag{4}$$

소수체 상의 ECC와 RSA는 하위계층의 유한체 연산에 유사성을 갖는다. ECC의 점 연산은 소수체상의 덧셈, 뺄셈, 곱셈, 제곱 연산으로 계산되며, RSA는 소수체상의 곱셈과 제곱 연산으로 모듈러 멱승 연산이 계산된다. 따라서 다양한 키길이에 대해 곱셈이 가능하도록 모듈러 곱셈기를 설계하면, ECC와 RSA를 통합하여 단일 하드웨어 구조로 구현할 수 있다 [11].

Ⅲ. 공개키 암호 하드웨어 가속기 코어

3.1. 전체 구조

공개키 암호 하드웨어 가속기 코어 PKCC는 NIST FIPS PUB 186-4에 정의되어 있는 P-192, P-224, P-256, P-384, P-521의 다섯 가지 소수체 타원곡선과 1,024 비트, 2,048 비트, 4,096 비트 키길이의 RSA를 지원한다. PKCC의 내부 구성도는 그림 1과 같으며, 두 개의 싱글 포트 메모리 (ModP_Mem, ERA_Mem), 유한체 연산회로 (GFp_Alu), 동작모드설정 레지스터 (OP_Reg), 그리고 제어블록 (Cntl_FSM)으로 구성된다.

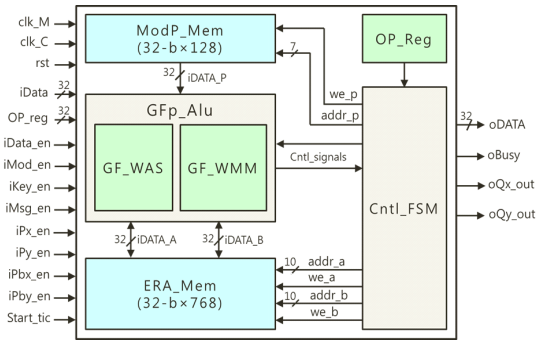


Fig. 1 Architecture of PKCC

메모리 ModP_Mem은 유한체 연산에 사용되는 모듈러 값 N을 저장하며, 초기에 모듈러 값 N이 저장되면 ECC와 RSA 연산이 완료될 때까지 읽기동작만 수행된다. 메모리 ERA_Mem은 ECC, RSA 연산에 필요한 초기 데이터와 유한체 연산의 중간결과 및 최종결과 값을 저장한다. 유한체 연산회로 GFp_Alu는 ECC의 PSM과 RSA의 모듈러 멱승 연산을 위한 모듈러 곱셈, 모듈러 제곱, 모듈러 가산/감산 연산을 수해하며, 워드기반 몽고메리 곱셈기 (GF_WMM), 모듈러 가산기/감산기 (GF_WAS), 내부 제어블록으로 구성된다. Cntl_FSM 블록은 ECC의 점 스칼라 곱셈, RSA의 모듈러 멱승, 그리고 모듈러 가산/감산/곱셈/나눗셈/역원 연산을 위해 GFp_ALU 블록과 전체 동작을 제어하는 유한상태머신 (FSM)으로 구성된다.

3.2. 유한체 연산회로 (GFp_Alu)

유한체 연산회로 GFp_Alu는 PKCC의 산술연산을 담당하며, 내부 구조는 그림 2와 같다. 워드기반 모듈러 가산기/감산기 (GF_WAS), 워드기반 몽고메리 곱셈기 (GF_WMM), 계수기 (Mul_Counter, AS_Counter), Alu_reg 등으로 구성된다. iStart_AS 신호에 의해 GF_WAS에서 모듈러 가산 또는 감산이 연산되며, iStart_Mul 신호에 의해 GF_WMM에서 모듈러 곱셈연산이 수행된다. Alu_reg는 워드기반 모듈러 연산의 중간 결과 값을 임시 저장한다.

GF_WAS 블록은 32 비트 모듈러 가산 또는 감산 연산을 선택적으로 수행하며, 반복 연산을 통하여 32 비트 배수의 모듈러 가산 또는 감산을 연산한다. 2개의 32 비트 캐리선택 (carry-select) 가산기/감산기와 캐리저장

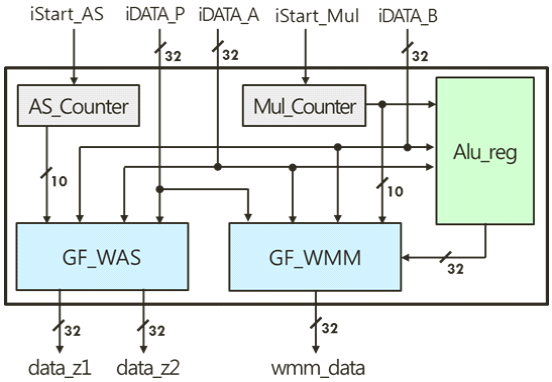


Fig. 2 GFp_Alu block

플립플롭, 모듈러 축약 (modular reduction) 연산을 결정하는 로직으로 구성된다.

GF_WMM 블록은 워드 기반 몽고메리 곱셈 알고리듬 [11]을 기반으로 구현되었으며, 32 비트와 33 비트 캐리선택 가산기, 32 비트 곱셈기, 캐리저장 플립플롭 등으로 구성된다. 모듈러 곱셈을 32 비트 단위로 반복 연산하여 32 비트 배수의 모듈러 곱셈을 연산한다.

3.3. 동작 모드

PKCC에서 수행되는 연산은 크게 나누어 타원곡선상의 점 연산과 유한체 연산으로 구분되며, OP_Reg에 동작모드 설정 데이터가 저장된다. OP_Reg 설정 값은 공개키 암호 알고리듬 (ECC, RSA)과 키길이, 연산모드 등의 선택과 메모리 리셋을 결정하여 Cntl_FSM 블록의 동작에 필요한 신호를 제공한다. OP_reg는 32 비트의 레지스터로 구성되며, 레지스터 맵은 그림 3과 같다. OP_reg의 최상위 비트인 OP_reg[31]에 의해 ECC 또는 RSA가 선택된다. ECC 모드로 설정된 경우, OP_reg [28:26]의 3 비트에 의해 다섯 가지 타원곡선 중 하나가 선택된다. RSA 모드의 경우, OP_reg[23:22]에 의해 RSA의 키길이가 선택된다. OP_reg[19:16]에 의해 PKCC에서 제공되는 9가지 동작모드 중 하나가 선택된다. RSA는 모듈러 멱승만 지원하므로, OP_reg[31]과 OP_reg[23:22]를 제외한 나머지 비트들은 0으로 설정된다. OP_reg의 최하위 비트가 OP_reg[0]=1로 설정되고 나머지 비트들은 0이면, 내부 메모리가 모두 0으로 초기화된다.

OP_Reg [31:0]	Function	Description
[31]	OP_Reg[31] 1: ECC, 0: RSA	Select PKC Algorithm
[30:29]		Reserved
[28:26]	OP_Reg[28:26] 000: P-192, 001: P-224, 010: P-256 011: P-384, 100: P-521	Select Elliptic Curve
[25:24]		Reserved
[23:22]	OP_Reg[23:22] 00: 1024-bit 01: 2048-bit 10: 4096-bit	Select RSA Key Length
[21:20]		Reserved
[19:16]	OP_Reg[19:16] 0000: EC Point Scalar Multiplication(ECPSM) 0001: EC Point Addition(ECPA) 0010: EC Point Doubling(ECPD) 0011: Modular Addition(MA) 0100: Modular Subtraction(MS) 0101: Modular Multiplication(MM) 0110: Modular Division(MD) 0111: Modular Inversion(MI) 1000: EC Point Subtraction(ECPS)	Select Mode
[15:1]		Reserved
[0]	OP_Reg[0] 1; Reset Memory	Reset Memory

Fig. 3 Register map of OP_reg

그림 4는 PKCC의 동작모드 중, ECC의 타원곡선 점 스칼라 곱셈 (ECPSM)과 RSA의 모듈러 멱승 연산의 동작 타이밍도를 보인 것이다.

(1) ECPSM 연산

ECPSM 연산의 동작 타이밍은 그림 4-(a)와 같으며, ① 데이터 입력 (Din) → ② $R^2 \bmod N$과 $-n_0^{-1} \bmod 2^{32}$ 생성 및 매핑 (GM) → ③ ECPSM 연산 → ④ 좌표계 변환 (TA) → ⑤ 리매핑 (RM) → ⑥ 데이터 출력 (Dout)의 과정으로 연산된다.

Start_tic 신호와 함께 OP_reg가 설정되어 ECPSM 모드가 선택된 후, 연산에 필요한 모듈러 값 N, 정수 K 그리고 시작점의 x-좌표 값 X와 y-좌표 값 Y가 그림 4-(a)의 타이밍으로 입력된다. iData_en 신호 후의 iXx_en 신호 (iMod_en, iKey_en iPx_en, iPy_en)에 의해 iData 포트로 입력되는 데이터가 구별되며, 데이터는 32-비트 단위로 최하위 워드부터 순차적으로 입력된다. 파라미터와 데이터 입력이 완료되면, $R^2 \bmod N$과 $-n_0^{-1} \bmod 2^{32}$의 계산, 몽고메리 도메인으로 매핑, 좌표계 변환 과정이 진행된다.

$R^2 \equiv (R \times R) \bmod N$ (단, $R = 2^{klen}$)로 정의되는 R^2 값은 피연산자를 몽고메리 도메인으로 변환하기 위해 사

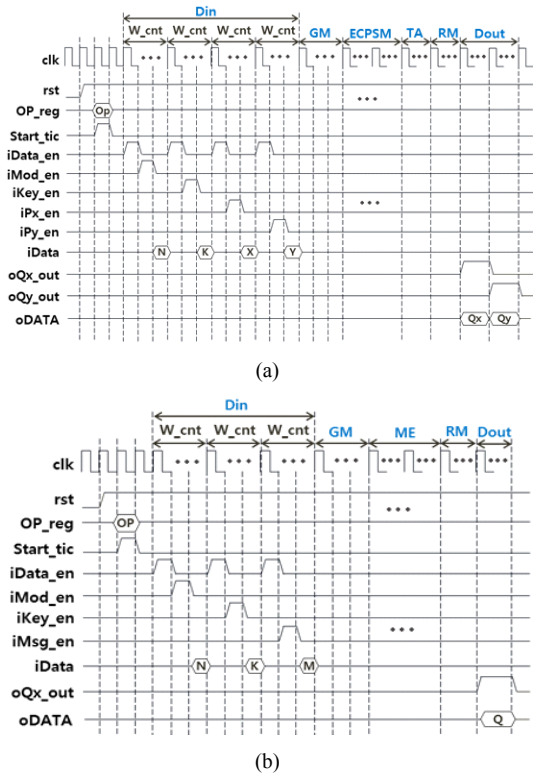


Fig. 4 Timing diagram of PKCC (a) ECPSM for ECC, (b) Modular exponentiation for RSA

용된다. 몽고메리 곱셈 알고리듬으로 R^2을 연산하면 $MM(R,R,N)=R$이 되어 R^2 값을 생성할 수 없으므로, 모듈러 가산·감산 연산을 적용하여 R^2 값을 생성해야 한다. 본 논문에서는 left-to-right 이진 방법에 의한 모듈러 곱셈 알고리듬을 적용하여 $R^2 \bmod N$ 값을 생성했다. $-n_0^{-1} \bmod 2^{32}$은 워드기반 몽고메리 곱셈을 위해 사용되며, 모듈러 값 N의 최하위 워드 n_0의 모듈러 곱의 역원의 마이너스 값이다.

PKCC에서 ECPSM 연산이 이루어지는 과정을 슈도코드로 나타내면 그림 5와 같다. 슈도코드의 단계-1은 점 (X, Y)를 몽고메리 도메인으로 변환하기 위해 X와 Y에 각각 R^2를 곱하는 $WMM(X,R^2,N)=XR$, $WMM(Y,R^2,N)=YR$을 연산한다. $WMM(X,R^2,N)$은 X와 R^2의 워드기반 몽고메리 곱셈을 나타내며, XR는 X가 몽고메리 도메인으로 변환된 값을 나타낸다. 슈도코드의 단계-2에 의해 $WMM(1,R^2,N)=R$이 연산되고, 몽고메리 도메인으로 변환된 타원곡선 상의 점은 자코비안 좌표계의 점

$$
\begin{aligned}
&\textit{Input}: && X=\{x_{m-1},\cdots,x_1,x_0\}_w\\
&&& Y=\{y_{m-1},\cdots,y_1,y_0\}_w\\
&&& K=\{k_{n-1},\cdots,k_1,k_0\}_2 \ \text{ for } k_{n-1}\neq 0\\
&&& N=\{n_{m-1},\cdots,n_1,n_0\}_w\\
&\textit{Output}: && Q=kP
\end{aligned}
$$

1: $XR \leftarrow WMM(X,R^2,N)$; $YR \leftarrow WMM(Y,R^2,N)$;
2: $R \leftarrow WMM(1,R^2,N)$; $P \leftarrow (xR,yR,R)$;
3: $P_0 \leftarrow P$; $P_1 \leftarrow 2P$
4: **for** $i=n-2$ **down to** 0 **do**
5: **if** $k_i=1$ **then**
6: $P_0 \leftarrow P_0+P_1$; $P_1 \leftarrow 2P_1$;
7: **else**
8: $P_1 \leftarrow P_0+P_1$; $P_0 \leftarrow 2P_0$;
9: **end if**
10: **end for**
11: $P_0 \leftarrow (QxR,QyR,zR)$
12: $K \leftarrow N-2$; $M \leftarrow zR$; $T \leftarrow WMM(1,R^2,N)$;
13: **for** $i=n-1$ **down to** 0 **do**
14: $T \leftarrow WMM(T,T,N)$;
15: **if** $k_i=1$ **then**
16: $T \leftarrow WMM(T,M,N)$;
17: **end if**
18: **end for**
19: $QXR \leftarrow WMM(QxR,T^2,N)$; $QYR \leftarrow WMM(QyR,T^3,N)$;
20: $QX \leftarrow WMM(QXR,1,N)$; $QY \leftarrow WMM(QYR,1,N)$;
21: **Return** Q $\leftarrow (QX,QY)$

Fig. 5 Pseudo code for ECPSM operation

$P=(xR,yR,R)$로 매핑된다. 몽고메리 도메인과 자코비안 좌표계로 매핑된 점 $P=(xR,yR,R)$을 K번 더하는 연산으로 ECPSM이 계산되며, 슈도코드의 단계-3~단계-10에 해당된다. 스칼라 곱셈을 위해 몽고메리 래더 알고리듬이 사용되었으며, 정수 K의 최상위 비트가 1일 때 단계-3이 수행된 후, 단계-4~단계-10이 순차적으로 수행된다.

ECPSM 연산이 완료된 결과 (QxR,QyR,zR)는 자코비안 좌표계에서 아핀 좌표계로 변환된다. 자코비안 좌표계의 점 (QxR,QyR,zR)를 아핀 좌표계의 점 $(QX/Z^2, QY/Z^3)$로 변환은 그림 5의 슈도코드에서 단계-12~단계-19에 의해 이루어진다. 이를 위해 필요한 Z의 곱의 역원인 Z^{-1}은 페르마의 소정리 (Fermet's little theorem)를 기반으로 계산된다. 페르마의 소정리는 유한체 상에서 모듈러 값만큼 모듈러 멱승 연산을 수행하면 다시 자신의 값이 된다는 특성을 정의한 것이며, $Z^{(N-2)} \bmod N = Z^{-1}$의 특성을 이용하여 단계-12~단계-18에 의해 연산된다. Z^{-1}으로부터 Z^{-2}, Z^{-3}을 연산하고 이로부터 X/Z^2, Y/Z^3의 연산은 슈도코드의 단계

-19에 해당된다. 아핀 좌표계로 변환된 ECPSM 결과는 몽고메리 도메인에서 정수 도메인으로 리매핑을 통해 최종 ECPSM 연산 결과가 얻어진다. 몽고메리 도메인의 (QXR, QYR)를 $WMM(QXR, 1, N) = QX$와 $WMM(QYR, 1, N) = QY$의 몽고메리 곱셈을 통해 정수 도메인으로 역변환이 이루어지며, 슈도코드의 단계-20에 해당한다.

(2) RSA의 모듈러 멱승 연산

RSA의 모듈러 멱승 연산의 동작 타이밍도는 그림 4-(b)와 같으며, ① 데이터 입력 (Din) → ② $R^2 \bmod N$과 $-n_0^{-1} \bmod 2^{32}$ 생성 및 매핑 (GM) → ③ RSA 연산 (ME) → ④ 리매핑 (RM) → ⑤ 데이터 출력 (Dout)의 과정으로 연산된다. ECPSM 연산과 비교하여 좌표계 변환 (TA) 과정을 거치지 않는다.

Start_tic 신호와 함께 OP_reg가 설정되어 RSA 모드가 선택된 후, 연산에 필요한 모듈러 값 N, 공개키 또는 개인키 K 그리고 메시지 M이 그림 4-(b)의 타이밍으로 입력된다. iData_en 신호 후의 iMod_en, iKey_en iMsg_en 신호에 의해 iData 포트로 입력되는 데이터가 구별되며, 각 데이터는 32-비트 단위로 최하위 워드부터 순차적으로 입력된다. 파라미터와 데이터 입력이 완료되면, $R^2 \bmod N$과 $-n_0^{-1} \bmod 2^{32}$의 계산과 몽고메리 도메인으로 매핑 과정이 진행된다.

PKCC에서 RSA 연산이 이루어지는 과정을 슈도코드로 나타내면 그림 6과 같다. 슈도코드의 단계-1은 메시지 M을 몽고메리 도메인으로 변환하기 위해 $WMM(M, R^2, N) = MR$을 연산한다.

모듈러 멱승은 몽고메리 도메인으로 매핑된 메시지 MR을 K번 곱하는 연산으로 $MR^K \bmod N$이 계산되며, 슈도코드의 단계-2 ~ 단계-7에 해당한다. 모듈러 멱승 연산에는 Left-to-Right 이진 알고리듬이 사용되었다. 모듈러 멱승 연산 결과 QR은 $WMM(QR, 1, N) = Q$의 리매핑을 통해 몽고메리 도메인에서 정수 도메인으로 역변환 되며, 슈도코드의 단계-9에 해당한다.

```
Input:  M = {m_{m-1}, ..., m_1, m_0}_w
        K = {k_{n-1}, ..., k_1, k_0}_2
        N = {n_{m-1}, ..., n_1, n_0}_w
Output: Q = {t_{m-1}, ..., t_1, t_0}_w

1: T ← WMM(1, R^2, N); MR ← WMM(M, R^2, N);
2: for i = n − 1 down to 0 do
3:     T ← WMM(T, T, N);
4:     if k_i = 1 then
5:         T ← WMM(T, MR, N);
6:     end if
7: end for
8: QR ← T
9: Q ← WMM(QR, 1, N);
10: Return Q
```

Fig. 6 Pseudo code for modular exponentiation

Ⅳ. 기능검증 및 성능평가

4.1. FPGA 구현을 통한 하드웨어 동작 검증

그림 7과 같은 FPGA 검증 플랫폼을 사용하여 PKCC의 하드웨어 동작을 검증하였다. Virtex-5 FPGA 소자가 사용되었으며, UART 인터페이스를 통해 테스트 벡터를 FPGA에 구현된 PKCC로 로딩하고, RSA 또는 ECC 연산을 수행한 결과를 PC로 전송하여 GUI 화면에 표시되도록 하였다.

그림 8-(a)는 PKCC의 ECC 모드를 이용하여 EC-DH (Elliptic Curve Diffie-Hellman) 키교환 프로토콜을 구현한 검증 결과 화면이며, 다음의 과정으로 동작한다.

① GUI 화면에서 좌측 상단에서 타원곡선을 선택하면, 선택된 타원곡선의 도메인 파라미터가 로딩된다. 우측 상단의 1. Generation 버튼을 클릭하면 Alice의 개인키 k_a와 Bob의 개인키 k_b가 각각 A_Key와 B_Key 영역에 생성된다.

② 2. Calculation 버튼을 클릭하면 Alice의 개인키와 타원곡선 파라미터가 FPGA에 로딩된다. 생성점 $G(Gx, Gy)$와 개인키 k_a의 ECPSM 연산을 통해 Alice의

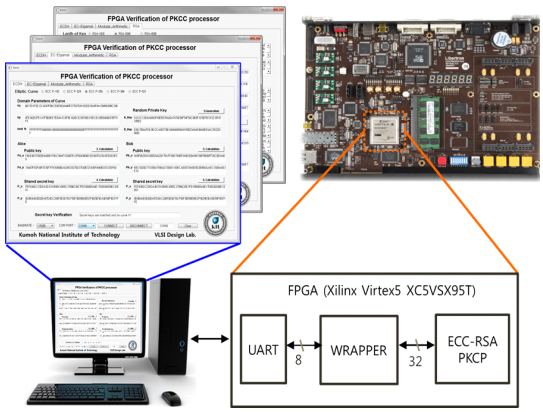
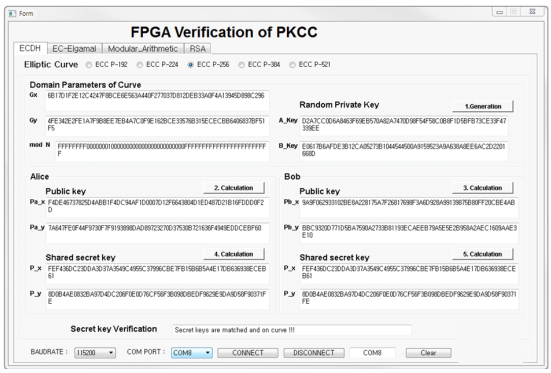
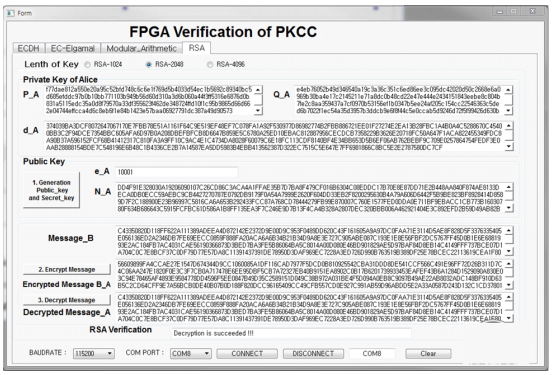


Fig. 7 FPGA verification platform



(a)



(b)

Fig. 8 Screenshots of FPGA verification results
(a) EC-DH key exchange protocol using P-256 EC
(b) RSA encryption/decryption using 2048-bit key

공개키 $P_a = k_a G$가 생성되어 버튼2 아래 영역에 표시되고, 상대방 Bob에게 전송된다.

③ 3. Calculation 버튼을 클릭하면 Bob의 개인키와 타원곡선 파라미터가 FPGA에 로딩된다. 생성점 $G(Gx, Gy)$와 개인키 k_b의 ECPSM 연산을 통해 Bob의 공개키 $P_b = k_b G$가 생성되어 버튼3 아래 영역에 표시되고, 상대방 Alice에게 전송된다.

④ 4. Calculation 버튼을 클릭하면 Bob의 공개키 P_b와 타원곡선 파라미터가 FPGA에 로딩된다. 시작점 $P_b(P_b x, P_b y)$와 Alice 개인키 k_a의 ECPSM 연산을 통해 공유키 $P = k_a P_b$가 생성되고, 버튼4 아래 영역에 표시된다.

⑤ 5. Calculation 버튼을 클릭하면 Alice의 공개키 P_a와 타원곡선 파라미터가 FPGA에 로딩된다. 시작점 $P_a(P_a x, P_a y)$와 Bob 개인키 k_b의 ECPSM 연산을 통해 공유키 $P = k_b P_a$가 생성되고, 버튼5 아래 영역에 표시된다.

⑥ FPGA를 통해 연산된 공유키 결과는 ECDH 프로토콜의 소프트웨어 구현 결과와 비교된다. 두 결과의 일치 및 타원곡선 상의 존재 여부가 GUI 화면 하단부에 표시되어 FPGA에 구현된 PKCC에서 ECPSM 연산이 올바로 수행되었는지 확인할 수 있다.

그림 8-(b)는 2048 비트 키길이의 RSA를 이용한 메시지 암호와 복호 동작을 검증한 결과 화면이며, 다음과 같이 4 단계로 실행된다.

① 그림 8-(b) GUI 화면의 P_A, Q_A 영역에 2048 비트의 소수 P와 Q를 입력한 후, 1. Generation Public Key 버튼을 클릭하면 Alice의 공개키 e와 개인키 d, 모듈러 값 N, 그리고 Bob의 메시지 M이 생성된다.

② 2. Encrypt Message 버튼을 클릭하면 모듈러 값 N, Bob의 메시지 M, Alice의 비밀키 e가 UART를 통해 FPGA 디바이스로 전송되고, RSA 모드 연산을 통해 모듈러 멱승연산이 수행되면, 암호화 메시지가 Encrypted Message B_A 영역에 표시된다.

③ 3. Decrypt Message 버튼을 클릭하면 암호화된 메시지 Encrypted Message B_A와 Alice의 개인키 d가 FPGA로 전송되고, RSA 모듈러 멱승 연산을 통해 복호화가 수행되면, Bob이 전송한 메시지 M이 복원되어 Decrypted Message_A 영역에 표시된다.

④ FPGA를 통해 연산된 RSA 암호·복호 결과를 입력 메시지와 비교한 결과가 화면 하단에 표시되며, FPGA에 구현된 PKCC의 RSA 동작모드의 올바른 동작 여부를 확인할 수 있다.

4.2. 성능 평가

설계된 PKCC를 180-nm CMOS 표준셀 라이브러리로 합성한 결과, 동작 주파수 50 MHz에서 20,800 GEs (gate equivalences)와 28 kbit의 RAM으로 구현되었다. 본 논문의 PKCC와 문헌에 발표된 ECC, RSA 공개키 암호 프로세서 사례를 표 2에 비교하였다. 문헌 [15]의 사례는 P-192, P-224, P-256의 세 가지 소수체 상의 타원곡선을 지원하며, 본 논문의 PKCC에 비해 높은 97.7 kbps의 처리율을 가지나, 동일한 Virtex-5 FPGA 디바이스에서 약 20배 많은 슬라이스를 필요로 한다. 문헌 [16]의 사례는 소수체 상의 타원곡선 P-256, P-384, P-521와 이진체 상의 타원곡선 B-163, B-233, B-571을 지원하는 듀얼 필드 ECC 프로세서이며, 본 논문의 PKCC와 유사한 점 연산과 유한체 연산을 지원하나, 점 뺄셈과 모듈

Table. 2 Comparison of public-key cryptography processors

		This paper	[15]	[16]	[17]
Algorithm		ECC, RSA	ECC	Dual-field ECC	RSA
Key size of RSA [bits]		1024, 2048, 4096	-	-	512, 1024, 2048, 3072
Field size of ECC [bits]		192, 224, 256, 384, 521	192, 224, 256	163, 233, 256, 384, 521, 571	-
Operation modes		ECPSM, ECPA, ECPD ECPS, MA, MS, MM MD, MI, ME	ECPSM	ECPSM, ECPA, ECPD, MA, MS, MM, MD	ME
Throughput [kbps] @100 MHz		ECPSM(P-256): 20.0 RSA(2048-b): 7.54	ECPSM(P-256): 97.7 @73 MHz	ECPSM(P-256): 176.5 @316 MHz	RSA(2048-b): 7.87
Technology / FPGA device		180 nm / Virtex-5	Virtex-5	55 nm / Virtex-4	180 nm
Hardware complexity	Gate Equivalent	25,800 GEs + 28 kbit RAM	-	189,000 GEs	10,670 GEs + 18.5 kbit RAM
	FPGA	1,503 slices + 2 BRAMs	31,431 slices	24,003 LUTs + 2 BRAMs	-

러 역원 및 모듈러 멱승 연산을 지원하지 않아 구현 가능한 공개키 암호 프로토콜에 제약을 갖는다. 100 MHz의 동작 주파수로 환산했을 때, 본 논문의 PKCC에 비해 약 2.8배의 연산 성능을 가지나 약 7.3배 많은 게이트를 필요로 한다. 문헌 [17]의 사례는 512, 1024, 2048, 3072 비트의 키길이를 지원하는 RSA 프로세서이며, 본 논문의 PKCC와 유사한 RSA 처리율을 갖지만, ECC를 지원하지 않는다. 이들 사례와 비교할 때, 본 논문의 PKCC는 소수체 상의 다섯 가지 타원곡선에 대한 점 스칼라 곱셈, 점 가산, 점 감산, 점 두배의 연산을 지원하며, 또한 세 가지 키길이의 RSA에 대한 모듈러 멱승 연산과 함께 모듈러 곱셈, 나눗셈, 역원 등 다양한 모듈러 연산 기능을 가져 다양한 공개키 암호 프로토콜을 구현할 수 있으므로, 유용성 측면에서 우수하다.

Ⅴ. 결 론

ECC와 RSA 기반의 공개키 암호 프로토콜의 하드웨어 구현을 위한 공개키 암호 하드웨어 가속기 코어 PKCC를 설계하였다. 본 논문의 PKCC는 네 가지 타원곡선 점 연산과 모듈러 멱승을 포함한 여섯 가지 유한체 연산기능을 가져 Diffie-Hellman (DH) 키교환, DSA 등 RSA 기반 다양한 공개키 암호와 EC-DH, EC-DSA, ECIES 등의 타원곡선암호 기반 공개키 암호 프로토콜의 구현에 폭넓게 사용될 수 있다. 180-nm CMOS 셀 라이브러리로 합성한 결과, 50 MHz 클록주파수에서 20,800 GEs와 28 kbit RAM의로 경량 하드웨어로 구현되었다. 100 MHz 클록 주파수로 동작하는 PKCC의 연산 처리율은 P-256 타원곡선의 점 스칼라 곱셈 연산은 20.0 kbps이고, 2048 비트 RSA 복호 연산은 7.54 kbps로 평가되었다. 본 논문의 PKCC는 유한체 연산회로, 메모리 등 하드웨어 자원의 공유를 통해 저면적으로 구현되어 하드웨어 리소스가 제한된 분야의 공개키 암호 시스템 구현에 적합한 것으로 평가되며, 보안용 시스템온칩 (System-on-Chip)에 내장되는 공개키 암호의 하드웨어 가속기로 사용될 수 있다.

ACKNOWLEDGEMENT

- This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(No. 2017R1D1A3B03031677)
- This research was supported by the KIAT(Korea Institute for Advancement of Technology) grant funded by the Korea Government(MOTIE: Ministry of Trade Industry and Energy). (No. N0001883, HRD Program for Intelligent semiconductor Industry)
- Authors are thankful to IDEC for supporting EDA software.

REFERENCES

[1] C. Paar and J. Pelzl, "Introduction to Public-Key Cryptography," Chapter 6 in *Understanding Cryptography, A Textbook for Students and Practitioners*, Springer, 2009.

[2] O. Toshihiko, "Lightweight Cryptography Applicable to various IoT Devices," *NEC Technical Journal*, vol. 12, no. 1, pp. 67-71, Oct. 2017,

[3] J. Athenaorcid and V. Sumathy, "Survey on Public Key Cryptography Scheme for Securing Data in Cloud Computing," *Circuits and Systems*, vol. 8, no. 3, pp. 77-92, Aug. 2017.

[4] R. Rivest, A. Shamir and L. Adleman, "A method for obtaining Digital Signatures and Public-Key Crypto-systems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, Feb. 1978.

[5] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203 – 209, Jan. 1987.

[6] V.S. Miller, "Use of elliptic curve in cryptography," CRYPTO85: *Proceedings of the Advances in Cryptology*, Springer-Verlag, 1986, pp. 417 – 426.

[7] Z.U. Khan and M. Benaissa, "High-Speed and Low-Latency ECC Processor Implementation Over GF (2^m) on FPGA," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* vol. 25, no. 1, pp. 165-176, Jan. 2017.

[8] S.H. Lee and K.W. Shin, "A Lightweight Hardware Implementation of ECC Processor Supporting NIST Elliptic Curves over $GF(2^m)$," *Journal of Institute of Korean Electrical and Electronics Engineers*, vol. 23, no. 1, pp. 58-67, Mar. 2019.

[9] B.G. Park and K.W. Shin, "A Lightweight ECC Processor Supporting Elliptic Curves over NIST Prime Fields," *Journal of The Institute of Electronics and Information Engineers*, vol. 55, no. 9, pp. 1107-1115, Sep. 2018.

[10] K.C.C. Loi and S.B. Ko, "Scalable elliptic curve cryptosystem FPGA processor for NIST prime curves," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 11, pp. 2753-2756, Nov. 2015.

[11] B.Y. Sung, "A Lightweight Public-key Cryptography Processor Integrating ECC and RSA into a Unified Hardware," Thesis, Kumoh National Institute of Technology, Korea, 2019.

[12] NIST Std. FIPS PUB 186-2, *Digital Signature Standard (DSS)*, National Institute of Standard and Technology (NIST), Jan. 2000.

[13] Standards for Efficient Cryptography Group, *SEC 2: Recommended Elliptic Curve Domain Parameters*, version 2.0, Jan. 2010. Available: https://www.secg.org/sec2-v2.pdf

[14] A.P. Zele and A.P. Wadhe, "Comparatively Study of ECC and Jacobian Elliptic Curve Cryptography," *International Journal of Science and Research (IJSR)*, pp. 2086-2089, 2013.

[15] K. Javeed, X. Wang and M. Scott, "High performance hardware support for elliptic curve cryptography over general prime field," *Microprocessors and Microsystems*, vol. 51, pp. 331-342, Jun. 2017.

[16] Z. Liu, D. Liu and X. Zou, "An Efficient and Flexible Hardware Implementation of the Dual-Field Elliptic Curve Cryptographic Processor," IEEE Transactions on Industrial Electronics, vol. 64, No. 3, Mar. 2017.

[17] W.L. Cho and K.W. Shin, "Scalable RSA public-key cryptography processor based on CIOS Montgomery modular multiplication Algorithm," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 22, no. 1, pp. 100-108, Jan. 2018.

성병윤(Byung-Yoon Sung)

2015: BS degree in School of Electronic Engineering, Kumoh National Institute of Technology.

2019: MS degree in Department of Electronic Engineering, Kumoh National Institute of Technology

2019~Present: Nextchip Inc.

※관심분야 : 정보보호 SoC 설계

신경욱(Kyung-Wook Shin)

1984 : BS degree in Electronic Engineering, Korea Aerospace University

1986 : MS degree in Electronic Engineering, Yonsei University

1990 : Ph. D. degree in Electronic Engineering, Yonsei University

1990-1991 : Senior Researcher in Semiconductor Research Center, Electronics and Tele-communication Research Institute (ETRI)

1991-Present : Professor in School of Electronic Engineering, Kumoh National Institute of Technology

1995-1996 : University of Illinois at Urbana-Champaign (Visiting Professor)

2003-2004 : University of California at San Diego (Visiting Professor)

2013-2014 : Georgia Institute of Technology (Visiting Professor)

※ 관심분야 : 정보보호 SoC 설계, 통신 및 신호처리용 SoC 설계, 반도체 IP 설계