High-speed WAVE Security Chip

Piljoo Choi, Hyun Il Kim, Ryang Ki and Dong Kyue Kim[1]
Department of Electronic Engineering, Hanyang University
E-mail : dqkim@hanyang.ar.kr

Abstract – **Wireless access in vehicular environments (WAVE) has the potential to both improve the quality of traffic services and provide services of convenience. However, these are only guaranteed if WAVE is securely protected from security threats. In order for WAVE to be securely protected, services such as the advanced encryption standard (AES) and elliptic curve cryptosystem (ECC) are required. We implemented a security system on chip that includes the AES and ECC modules, and we then verified. The implemented cryptographic modules have a good performance when compared to other research results.**

I. INTRODUCTION

Wireless access in vehicular environments (WAVE) [1] is an intelligent transport system (ITS) communication standard, which is a modification of the wireless local area network (WLAN) that is designed to be suitable for the vehicular environment. It provides a seamless two-way communication through the antennas installed on the highway with vehicles even at high speeds of 200 km/h. This makes it possible to receive traffic information on traffic jams, constructions on the road, etc., in real time. In addition, it provides convenient services such as internet surfing in vehicles.

Because of WAVE, the quality of the traffic system and users' convenience may improve, but privacy problem and accidents caused by misinformation should also be considered. Security is essential, and so as to prevent these issues, the WAVE standard requires security services such as the advanced encryption standard (AES) [2] and elliptic curve cryptosystem (ECC) [3].

AES is a cryptographic algorithm specified by the National Institute of Standards and Technology (NIST) in 2001. ECC is a public-key cryptographic algorithm proposed by Victor Miller in 1986 and Neal Koblitz in 1987, and is based on the elliptic curve discrete logarithm problem. In the WAVE standard, AES operates in the counter with CBC-MAC (CCM) block mode [4] to ensure the confidentiality and integrity of the data communication. Elliptic curve digital signature algorithm (ECDSA) [5] and elliptic curve integrated encryption scheme (ECIES) [6], which are algorithms based on ECC, are used as a digital signature and public key encryption algorithm. The cryptographic intellectual properties (IPs) of the AES and ECC are essential for the WAVE modem, and they must have high throughput for high-speed communication. ECC requires a lot of computation in particular, so its efficient design is important. Moreover, the WAVE standard requires ECDSA over two NIST prime fields, P256 and P224, so the ECC module within the WAVE modems can perform the finite field operations over two kinds of prime fields.

In this paper, we propose a security system on chip (SoC) for WAVE and high-speed cryptographic IPs. One of the cryptographic IPs, the AES module, is designed to have a high throughput speed between 500 Mbps and 1 Gbps. While the other cryptographic IP, the ECC module, can perform ECDSA and ECIES over P224 and P256. Our SoC includes both these IPs and is implemented using the 180 nm technology library, which produces a better performance by our cryptographic IPs when compared to other research results.

This paper is organized as follows. In Section 2, the security services defined in the WAVE standard, and the details of the AES and ECC algorithms are presented as preliminary information. We describe the design of our SoC and cryptographic IPs in Section 3. Section 4 show the implemented results and experimental environment, and we conclude this paper in Section 5.

II. PRELIMINARY

In this section, we first show the security services defined in WAVE and then present their details.

A. Security services defined in WAVE

A set of security services for confidentiality assurance, authentication, key exchange, etc., is defined in IEEE 1609.2 [1] and is shown in Table I.

TABLE I.
Cryptographic algorithm required in WAVE.

Required function	Cryptographic algorithm	Key length/ finite field
Message encryption	AES-CCM	128-bit
Digital signature generation/verification	ECDSA	P224 & P256
Key exchange	ECIES	P256

a. Corresponding author; dqkim@hanyang.ar.kr

B. AES

AES supports encryption with 128-, 192- and 256-bit key, and WAVE requires an encryption with 128-bit key. Each 128-bit message block is encrypted or decrypted through four transformations, which are shown in Table II.

TABLE II.
Four transformations in AES.

Transformation	Decryption
ByteSub	Non-linear byte substitution
MixColumn	Combination of each column using linear transformation
ShiftRow	Cyclical shift of each row
KeyAddition	Exclusive or (XOR) operation with the round key

There are various modes of operation for cryptographic block ciphers such as counter (CTR) and cipher block chaining (CBC). Counter with CBC-MAC (CCM) is one of modes and is used for the AES block cipher in WAVE. The CCM mode is a combination of the CTR and CBC modes. The CTR and CBC modes that operate in the CCM mode are used for message encryption and message authentication code (MAC) generation, respectively. In the CTR mode, the encrypted initial vector with a counter is used for encrypting plain texts or decrypting cipher texts. In the CBC mode, only plain texts are used for generating MAC. Thus, the AES module itself is required to only perform encryption in the CCM mode.

C. ECC and cryptographic protocols based on ECC

a. ECC

In WAVE, two NIST prime fields, P224 and P256, are used. The elliptic curve over P224 and P256 consists of points that satisfy the following equation,

$$y^2 = x^3 + ax + b \tag{1}$$

and a point at infinity. Using points on the elliptic curve, EC point addition (ECPA) and EC point doubling (ECDA) are defined. These points can be expressed as follows:

$$\mathrm{P}\,(x_1, y_1) + \mathrm{Q}\,(x_2, y_2) = \mathrm{R}\,(x_3, y_3). \tag{2}$$

If P and Q are different points, the operation is ECPA; otherwise, the operation is ECPD. –R $(-x_3, y_3)$ is another point where a straight line passing through P and Q in ECPA or a tangent on P (= Q) in ECPD meets the elliptic curve.

TABLE III.
EC Point addition (ECPA) and point doubling (ECPD).

Point addition, $\mathrm{P} + \mathrm{Q} = \mathrm{R}(x_3, y_3)$	Point doubling, $2\mathrm{P} = \mathrm{R}(x_3, y_3)$
$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$	$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1$
$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$	$y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1$

Table III shows the calculation details for ECPA and ECPD. The expressions in Table III are performed over a finite field, so the ECC module should be able to calculate modular multiplication (MM), modular inverse (MI), modular addition (MA) and modular subtraction (MS). By repeating ECPD and ECPA, EC point multiplication (ECPM) can be performed and is the main operation of ECDSA and ECIES.

b. ECDSA and ECIES

The ECC-based cryptographic algorithms used in WAVE are ECDSA and ECIES. ECDSA is a public key digital signature algorithm using elliptic curve cryptography and is specified in FIPS 186-3. The details of the signature generation and verification are shown in Table IV.

TABLE IV.
ECDSA digital signature generation and verification.

	Signature Generation	Signature verification
Input	m: message d: private key	m: message Q: public key (r, s): signature
Algorithm	1. Select a random integer k in the interval $(1, n)$ 2. $k\mathrm{G} = (x_1, y_1)$ 3. $r = x_1 \bmod n$ If $r = 0$ then go to step 1 4. $e = \mathrm{H}(m)$ 5. $s = k^{-1} \bmod n$ If $s = 0$ then go to step 1	1. $e = \mathrm{H}(m)$ 2. $w = s^{-1} \bmod n$ 3. $u_1 = ew \bmod n$ $u_2 = rw \bmod n$ 4. $(x_1, y_1) = u_1\mathrm{G} + u_2\mathrm{Q}$ 5. $v = \mathrm{x}_1 \bmod n$
Output	(r, s): signature	1 iif $v = r$, 0 otherwise

ECIES is the public key encryption algorithm based on elliptic curve Diffie–Hellman (ECDH). In WAVE, ECIES is used to securely transmit the secret key, which is used in AES. In ECIES, a point is shared using ECDH, and its x point and some shared parameters are derived as a key using a key derivation function. The derived key is used to encrypt the AES key and to generate its MAC. As a result, while transmitting the AES key using ECIES, its confidentiality and integrity is guaranteed.

III. PROPOSED SECURITY SOC FOR WAVE

In this section, we show the structure of the proposed security SoC and explain the design of our AES module and ECC module.

A. Structure of Security SoC for WAVE

The security SoC includes the AES and ECC modules, which are connected through a bus to the core, memories, and peripherals. The structure of the proposed security SoC for WAVE is shown in Fig. 1.

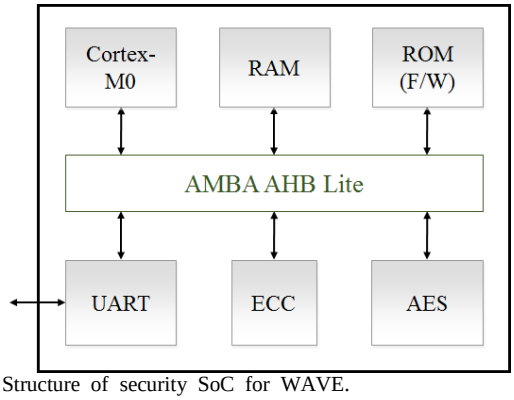


Fig. 1. Structure of security SoC for WAVE.

Cortex-M0 and AMBA AHB are used as a core and a bus, respectively. The universal asynchronous receiver and transmitter (UART) module is used to communicate with the personal computer (PC) that acts as a server for SoC verification. The ROM has a firmware for the AES and ECC modules operation. When the AES and ECC module are practically used for WAVE, an SoC similar to Fig. 1 may be connected to a WAVE modem chip through other communication methods instead of a UART module, or only cryptographic IPs may be directly embedded and connected with the bus in a WAVE chip.

B. Structure of AES module

The structure of the AES module in Fig. 1 is shown in Fig. 2.

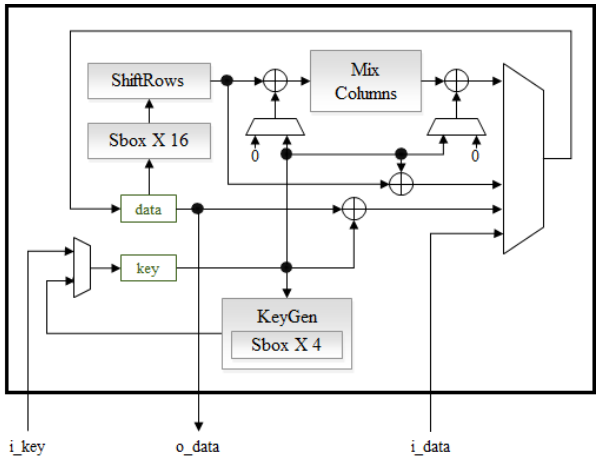


Fig. 2. Structure of AES module.

A total of 20 Sboxes are fully used, thus only 13 clock cycles are consumed to encrypt a single message block. The AES module for WAVE operates only in CCM mode, thus only encryption is necessary as explained in Section 2-B. As a result, the AES module in Fig. 2 does not include inverse Sboxes, inverse MixColumns, etc. When the AES module is connected to the bus, an AMBA AHB slave wrapper is added to the interface of the AES module.

C. Structure of the ECC module

We first show the entire structure of the ECC module, and then explain the design of the MM module, which has an influence on the entire performance of the ECC module.

a. Structure of ECC module

The structure of the ECC module in Fig. 1 is shown in Fig. 3. The ECC module consists of registers, control logic, and three submodules for modular operations. The main registers are as follows: D0 and D1 store the inputs of the submodules; X0, Y0, X1, and Y1 store the two points on the elliptic curve; and T0, T1, and T2 are the temporary storage required for computing ECPA and ECPD. By controlling the inputs and outputs of the submodules, the control logic computes ECPA and ECPD, which are repeated to compute ECPM.

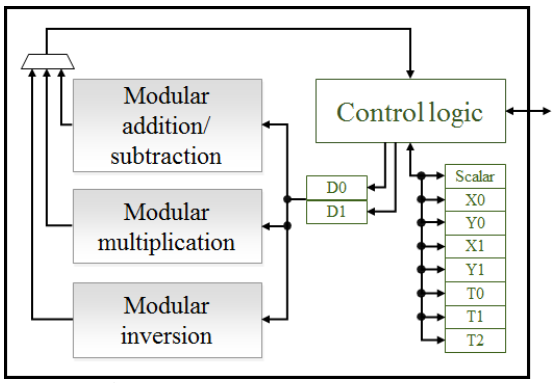


Fig. 3. Structure of ECC module.

Among all modular operations, both MM and MI require a high number of computations. Our MI module is based on the modified right-shift binary inverse algorithm in [7], and our MM module is based on the word-based Montgomery product [8], [9]. If the processing speed of an MM module is faster than that of an MI module, the MI can be replaced with several MMs. However, the processing speed of our MM module is about four times faster than that of our MI module. Under such conditions, the replacement method is not efficient. As a result, the structure of our ECC module is simpler than an ECC with the replacement method.

Our MM module is designed to be simpler than other MM modules based on the word-based Montgomery product, by utilizing the characteristics of the modulus p over P224 and P256. We explain the details of our MM module in the next subsection. Similar to the AES module, an AMBA AHB slave wrapper is added to the ECC module in Fig. 3 to connect with the bus.

b. Structure of Modular Multiplication module

The structure of the MM module is shown in Fig. 4.

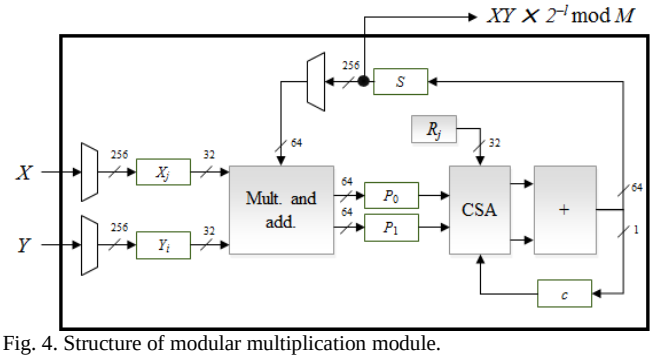


Fig. 4. Structure of modular multiplication module.

Using a single 32-bit × 32-bit multiplier, the multiplication and reduction is repeated word-by-word. Its detailed operations are shown in Algorithm 1 where w is the number of words for both input and output data. The w for P224 and P256, are seven and eight, respectively. The P_0 and P_1 registers that were inserted for pipelining in Fig. 4 are omitted in Algorithm 1 for the sake of simplicity.

Algorithm 1 MontMult(X, Y)

Input: $X = \{X_{w-1}, ..., X_1, X_0\}$, $Y = \{Y_{w-1}, ..., Y_1, Y_0\}$

Output: $X \times Y \times 2^{-n} \bmod M$

1 : $S = \{S_{w-1}, ..., S_1, S_0\} = 0$
2 : $A = 0$
3 : $c = 0$
4 : for $i = 0$ to $w - 1$
5 : $\{c, S_1, A\} = X_0 \times Y_i + \{S_1, S_0\}$;
6 : for $j = 1$ to $w - 2$
7 : $\{c, S_{j+1}, S_{j-1}\} = X_j \times Y_i + \{S_{j+1}, S_j\} + \{c, R_j\}$
8 : end for
9 : $\{S_{w-1}, S_{w-2}\} = X_{w-1} \times Y_i + \{R_w, S_{w-1}\} + \{c, R_{w-1}\}$
10 : if($M = n$)
11 : $A' = A \times n' \bmod 2^{32}$ // $n' \times n = -1 \pmod{2^{32}}$
12 : $\{c, S_0, T\} = n_0 \times A' + \{S_0 \ll 32\}$
13 : for $j = 1$ to $w - 1$
14 : $\{c, S_j, S_{j-1}\} = n_j \times A' + \{S_j, S_{j-1}\}$
15 : end if
16 : end for
17 : Return S

Algorithm 1 shows the iteration of adding $X \times Y_i$ to S and right-shifting S by 32-bit where X_i is the (i+1)-th word of X, and X_0 is the least significant word. Before the 32-bit right shift, we need to find q that satisfies $R = A + q \times M = 0 \pmod{2^{32}}$, where M is a modulus and $A = S_0 + X_0 \times Y_i \bmod 2^{32}$. There are two moduli p and n, in the NIST elliptic curve. Depending on which modulus is used, the method to find q is different.

[Case 1: the modulus is p] Over P224, $p = 2^{224} - 2^{96} + 1$, so $q = 2^{32} - A \bmod 2^{32}$, and over P256, $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$, so $q = A$. Since R_0 is always zero, only R_j for $0 < j \leq w$ is required.

TABLE V.
R_j ($1 \leq j \leq$ w) for non-zero A.

M \ j	8	7	6	5	4	3	2	1
p (P256)	$A - 1$	$\sim A + 1$	A	0	0	A	0	1
p (P224)	-	~A	$2^{32} - 1$	$2^{32} - 1$	$2^{32} - 1$	A	0	1
n	0	0	0	0	0	0	0	1

Table V shows R when A is not zero, where $\sim A$ is the one's complement of A. When A is zero, R is zero as well. Lines 10-14 of algorithm 1 are for modulus n, hence they are not executed.

[Case 2: the modulus is n] According to the reduction method of the normal Montgomery production, $q = A' = A \times n' \bmod 2^{32}$ where $n' \times n = -1 \pmod{2^{32}}$. q is expressed as A' in Algorithm 1 as q is stored in the register for A. We can see that $A' \times n + A = (A \times n') \times n + A = \mathrm{A} \times (-1) + A \pmod{2^{32}} = 0 \pmod{2^{32}}$. Before adding $A' \times n$ to S, S is already right-shifted, so the least significant word of $A \times n'$, T is ignored. When A is not zero, $A + T = 2^{32}$, i.e., $A + T$ is not zero. To consider the carry from $A + T$, $R_1 = 1$ and $R_j = 0$ ($1 < j \leq w$) for non-zero A, and $R = 0$ for $A = 0$.

A single clock cycle is consumed in each of line 5, 7, 9, 11 and 14 of Algorithm 1. Since lines 7 and 14 are repeated $w - 2$ and $w - 1$ times, respectively, Case 1 and Case 2 consume w^2 and $w \times (2w+1)$ clock cycles, respectively. In Fig. 4, P_0 and P_1 are inserted to reduce the critical path. Including the clock cycles for this pipelining and two cycles for the start and end signals of the MM module, the total consumed clock cycles are shown in Table VI.

TABLE VI.
Clock cycles for MM.

	Modulus is p ($w^2 + 3$)	Modulus is n ($w \times (2w + 2) + 3$)
P256	67	147
P224	52	115

IV. IMPLEMENTATION AND CHIP VERIFICATION

A. Implemented results

Using the 180 nm technology library, we implemented our security SoC for WAVE.

TABLE VII.
Performance of AES and ECC module @ 100 MHz.

	AES	ECC (P224)	ECC (P256)
Area	21 k GE	91 k GE	
Throughput	0.98 Gbps	625 ECPM/s	526 ECPM/s

Table VII shows the area and throughput of the AES and ECC modules when they are synthesized at 100 MHz. As the AES module needs 13 clock cycles to encrypt a message block, its throughput is about 1 Gbps. In WAVE, AES operates in the CCM mode, which is a combination of the CBC and CTR modes that halves the throughput into about 0.5 Gbps. This occurs only when a single AES module is included. When two AES modules are used, each message block can be processed in the CBC and CTR modes in parallel without having to halve the throughput.

The processing time for ECPM is 1.6 ms and 1.9 ms over P224 and P256, respectively. Hence, the ECC module can compute 625 and 526 ECPM per second over P224 and P256, respectively, as shown in Table VII. According to the finite field (P256 or P224), only the iteration number of some lines in Algorithm 1 and the values of R_j ($1 \leq j \leq w$) are different, thus the overhead area is very small.

TABLE VIII.
Performance comparison of our ECC module.

	Area	Time for ECPM (ms)	Freq. (MHz)	Technology (or device)
This work	91 k GE	1.9	100	180 nm
	112 k GE	0.95	200	
[10]	120 k GE	2.68	137.7	130 nm
[11]	122 k GE	1.01	556	130 nm
[12]	34.6 LUTs	2.26	160	Virtex 5
[13]	197 k GE	1.21	208	130 nm
[14]	189 k GE	1.45	316	55 nm

Table VIII compares the synthesized results of our ECC module with other research. Normally, as the area increases, the maximum operating clock frequency decreases after layout (place and route). For a fair comparison and as our module is synthesizable at 200 MHz, the results at 200 MHz are included and are shown to be better than the results of other ECCs in Table VIII. Since the ECC module in [10], [12], [13], [14] can compute ECPM over dual-field, their area may become smaller if the logic to compute ECPM over binary field is removed. Even after taking this into consideration, the area of the ECC module in [13], [14] is still 1.69-1.76 times larger than ours, and the processing time of the ECC module in [10] is more than twice as large as ours. The ECC module in [11] has a slightly longer processing time and a slightly larger area than ours, but its operating clock frequency is faster, thus its dynamic power may be significantly higher compared to our module. The ECC module in [12] is implemented in FPGA, so we are unable to directly compare its area to our module, but its processing time is more than twice as large as ours.

B. Chip verification

To verify our implemented chip, we used the test board shown in Fig. 5.

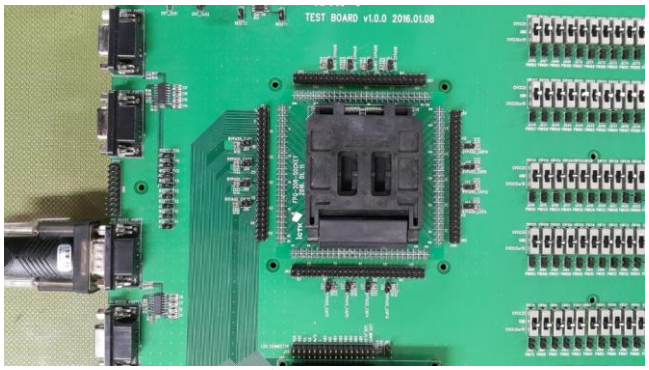


Fig. 5. Test board for chip verification.

The chip on the test board is connected to the PC through a UART, and we verified that it operates normally at 100 MHz.

V. CONCLUSION

We designed and implemented a security SoC that includes the AES and ECC modules, which is required for WAVE. The AES and ECC modules embedded in the security SoC are speed-oriented designed. Our AES module can encrypt a message block in the CCM mode with a speed of 0.5 Gbps, which can be doubled when two AES modules are used in parallel if required. Our ECC module can perform ECDSA and ECIES over the NIST prime fields, P224 and P256, and has a better performance compared to other ECC modules. The operation of the implemented chip is verified at 100 MHz. It is expected that our security SoC or only the cryptographic IPs are used to support security functions required for WAVE.

ACKNOWLEDGMENT

This work is supported by IDEC.

REFERENCES

[1] ITS Committee, "IEEE standard for wireless access in vehicular environments-security services for applications and management messages," *IEEE Vehicular Technology Society 1609.2*, Apr. 2013.

[2] Standard, NIST-FIPS "Announcing the advanced encryption standard (AES)," *Federal Information Processing Standards Publication 197*, vol. 197, pp. 1-51, Nov. 2001.

[3] Miller, Victor S. "Use of elliptic curves in cryptography," *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417-426, 1985.

[4] Morris Dworkin, "*Recommendation for Block Cipher Modes of Operation: The CCM mode for authentication and confidentiality*," US Department of Commerce, Technology Administration, National Institute of Standards and Technology, May 2004.

[5] Kerry, C. F and C. R, "Digital signature standard (DSS)," *Federal Information Processing Standards Publication 186-4*, July 2013.

[6] IEEE Std 1363a, "IEEE Standard Specifications for Public-Key Cryptography-Amendment 1: Additional Techniques," Mar. 2004.

[7] Piljoo Choi, Dong Kyue Kim, "Design of efficient modular inversion module using resource sharing," *Conference Proceedings MITA 2015*, pp. 298-299, Tashkent, Uzbekistan, Jun. 2015.

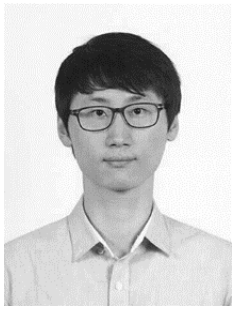
[8] Peter L. Montgomery, "Modular multiplication without trial division." *Mathematics of computation*, vol. 44, pp. 519-521, 1985.

[9] Hyun Il Kim, Dong Kyue Kim, "Design of hardware ECC for various environments," *Master's Thesis Hanynag University*, Feb. 2017.

[10] A. Satoh, K. Takano, "A scalable dual-field elliptic curve cryptographic processor, " *IEEE Transaction on Computer*, vol.52, no.4, pp. 449-460, Apr. 2003.

[11] Gang Chen, et al, "A High-Performance Elliptic Curve Cryptographic Processor for General Curves Over GF(p) Based on a Systolic Arithmetic Unit," *IEEE Transactions on Circuit and Systems*, vol. 54, no. 5, pp. 412-416, May 2007

[12] Hamad Marzouqi, et al, "A high-speed FPGA implementation of an RSD-based ECC processor," *IEEE Transaction on VLSI System*, vol. 24, no. 1, pp. 151-164, Jan. 2016.
[13] Jyu-Yuan Lai, et al, "A highly efficient cipher processor for dual-field elliptic curve cryptography," *IEEE Transactions on Circuit and Systems*, vol. 56, no. 5, pp. 394-398, May 2009.
[14] Zilong Liu, et al, "An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor," *IEEE Transaction on Industrial Electronics*, vol. 64, no. 3, pp. 2353-2362, Mar. 2017.



Piljoo Choi received the B.S. and M.S. degrees in Electronic Engineering from Hanyang University in 2010 and 2012, respectively. He is currently a Ph.D. candidate in the Department of Electronic Engineering at Hanyang University, Korea. His research interests are in the areas of security SoC (System on Chip), crypto-coprocessors, and information security.

Hyun Il Kim received the B.S. and M.S. degrees in Electronic Engineering from Hanyang University in 2015 and 2017. He is currently a master's student in the Department of Electronic Engineering at Hanyang University, Korea. His research interests are in the areas of crypto-coprocessors.

Ryang Ki received the B.S. degrees in Electronic Engineering from Hanyang University in 2017. She is currently a master's student in the Department of Electronic Engineering at Hanyang University, Korea. Her research interests are in the areas of crypto-coprocessors, and functional safety.

Dong Kyue Kim received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Seoul National University in 1992, 1994, and 1999, respectively. From 1999 to 2005, he was an assistant professor in the Division of Computer Science and Engineering at Pusan National University. He is currently a full professor in the Department of Electronic Engineering at Hanyang University, Korea. His research interests are in the areas of security SoC (System on Chip), crypto-coprocessors, and information security.