High Throughput– Resource Saving Hardware Implementation of AES-CCM for Robust Security Network

Dang Khoa Nguyen, Leonardo Lanante and Hiroshi Ochi
Dept. of Computer Science and Electronics
Kyushu Institute of Technology, Iizuka, Fukuoka, Japan
{khoa, leonardo, ochi}@dsp.cse.kyutech.ac.jp

***Abstract*—This paper presentsa new architecture andASICimplementation of high throughput of Counter with Cipher Block Chaining – Message Authentication Code（CCM) for robust security network such as gigabit wireless IEEE 802.11ac in case considering trade-off between throughput and resource saving. We propose a new architecture of AES-CCM core adopted in parallel which utilizes two separated AES forward cipher cores for MIC calculation in Counter (CTR) Mode and encryption or decryption data in Cipher Block Chaining (CBC) Mode. The implementation of AES-CCM core in Synopsys CMOS SAED90nm process achieves2.69Gbps of throughput at 264MHz clock frequency.The proposed architecture of AES-CCM corereduces latencyby one AES cycle in comparison with conventional architectures. In addition,the AES-CCM core supports bothgeneration-encapsulation and decryption-verifica-tionprocess with symmetrical data processing routine. We also introduce an implementation of reordering AES transformation method comes along with composite Sbox in order to gain maximal period of the composition and saving resources compared to original AES algorithm implementation.**

***Index Terms*-AES, AES-CCM, CCMP, WPA2, 802.11i, Security**

I. INTRODUCTION

Gigabit wireless communication has been the most remarkable requirement forthe legacy IEEE 802.11 standardin recent years[1]. Many studies were published in this trend. Among those gigabit solutions, the IEEE 802.11ac is egressed as a promise to deliver the best end-user experience of gigabit wireless communication. The IEEE 802.11ac WLAN standard offers a maximum PHY data rate just under 7Gbps [2]. To adapt the high speed trafficin MAC mechanism, high throughput of CCMP (Counter with CBC-MAC Protocol) for security in the wireless communication network emerges as a paramount design aspect.

The CCMP is a security protocol which mandated for RSN (Robust Security Network)compliance to provide assurance of the confidentiality, authenticity, integrity, and replay protection. It becomes a popular security protocol for ensuring information data transmission over unsecured WLAN [3]. This protocol simultaneously offers two important features of security service, named data authentication and encryption, by combining the techniques of the Counter (CTR) mode and the Cipher Block Chaining – Message Authentication Code (CBC-MAC) mode[4].In this study, we contribute a high throughput implementation of AES-CCM parallel architecture for IEEE 802.11ac standard.

The remainder of this paper is organized as follow. Section II presents a brief introduction and related works to implement the CCM protocol. Section III and IV provides short description of AES forward cipher algorithm and previous works. Then, the detail of AES-CCM proposed architecture is given in section V. In Section VI, implementation results are summed up and then compared to other similar previous studies. Finally, section VIIprovides conclusion for this paper.

II. CCM PROTOCOL OVERVIEW

CCMPprovides an assurancemechanism for data transmission by using any approved symmetric block cipher algorithm with thefixed size of data block of 128 bits [4]. CCM protocol aims to be used in the packet environmentwhere input data is prerequisite before CCM is applied.It is not designed to support partial processing or stream processing [5]. In theIEEE 802.11 WLANstandard, CCM algorithmis definedin the amendment,named the IEEE 80.211i, whichis specified as a part of security mechanisms for wireless networks based on AES (Advanced Encryption Standard) encryption algorithm. In this paper, the CCMPagreed with thelegacy IEEE 802.11 is discussed.The term AES-CCM is mentioned from now on to mean the CCM protocol scheme that has been designed to use AES algorithmwith 128bits key and 128bitsdata block size as primitiveblock cipher.

AES-CCMconsists of two related processes: generation-encryption and decryption-authentication.AES-CCMcan be considered as an operation mode of AES cipher by integrating two operation modes of AES cipher which are CTR mode and CBC mode associated with encrypting/decrypting data and forming MIC (Message Integrity Check) function. The CTR and CBC operation are illustratedin Fig. 1 and Fig. 2 respectively. MIC is used for authentication and verification purpose.The input

Manuscript received November 30, 2012; revised February 4, 2013.

doi: 10.12720/joace.1.3.250-254

data for each process include: payload, AAD (Additional Authentic Data), and a unique number NONCE.In the IEEE 802.11ac, the maximum length of payload data in MPDU is 16383 bytes [2], AAD length is varied from 22 to 30 bytes depend on the presence of A4 and QC field in the MPDU header, NONCE length is 13 bytes.Those data are formatted into a 128-bit data blocks denotedas B_0, $B_1 \ldots B_m$, before processing.The formatting functions for B_0, $B_1 \ldots B_m$, are clearly specified in the NIST SP800-38C [4].B_0 block is constructed from NONCE. The next two blocks B_1, $B_2$are created from AAD data. The other blocks are formed from payload. Zero-padding at the lower significant bits is needed incase of empty bits are presented in the data block. The following are intended to explain those processes in detail.

In generation-encryption process, CBC mode is applied to NONCE, AAD data, payloadin order to generate 8-byte MIC message; then CTR mode is applied to payload and 8-byte MIC message to form cipher text data. Thus, in the IEEE 802.11ac, the output MPDU from AES-CCM generation-encryption is expanded in comparison with the size of the input MPDU by 8 additional bytes of encrypted MIC. The expanded MPDU is show in the Fig. 3.

Whereas decryption-verification process is different from encryption-decryption process, CTR mode is applied to encrypted-MPDU to recover MIC and corresponding payload. Then, CBC mode is applied to NONCE, AAD data and decrypted payload to reproduce MIC. The MIC in CBC mode is compared with the encrypted MIC in CTR mode in order to determine whether the received payload is correct or not. This step is called verification. Verification successfully provides authentication for the MPDU data.

CCM is designed to provide stronger assurance for authenticity than a checksum or error detection codes, such as parity check or CRC (Cyclic Redundancy Check) code [4]. CCM is not only detecting accidental modifications of the data, but also discovering intentional, unauthorized modifications from the opponents[4].In addition, AES-CCM does not require the reverse cipher of AES algorithm, which is more complex than AES forward cipher, in order to decrypt the ciphertext.It is a better usage of resources.

The inputs, outputs, and pseudo code of the two CCM processes are explained in detail in Table. I. and Table. II., respectively. We note that Plen, Clen, and Tlencorrespond to the length in bits of payload, ciphertext and MIC.

TABLE I. INPUTS AND OUTPUT OF CCM PROCESSES

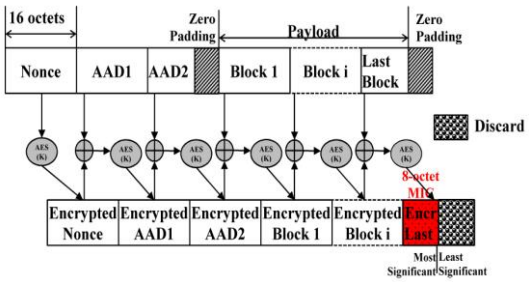
Process	Input	Output
Generation-Encryption	NONCE AAD data Payload P	Ciphertext C of the purported payload P and MIC message
Decryption-Verification	NONCE AAD data Ciphertext C Cipher MIC	Either payload P or INVALID

III. AES FORWARD CIPHER ALGORITHM

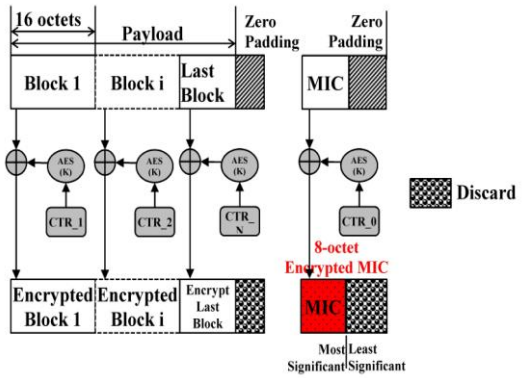
AES, a symmetric-block cipher, was officially published in 2001 as a new American encryption standard [6]. AES has been selected to replace the old standard DES (Data Encryption Standard) because it was designed from the ground up rapidly, secure and be able to support vary computing devices.

TABLE II. THE PSEUDO CODE OF CCM PROCESSES

Process	Pseudo code
Generation-Encryption	Generation scheme: 1. Apply formatting function to payload $B_0,\ldots B_m$ 2. Set $Y_0=AES_k(B_0)$ 3. For i=1 ← m, do $Y_i=AES_K(B_i \oplus Y_{i-1})$ 4. Set $MIC=MSB_{32}(Y_m)$ Encryption scheme: 5. Set $r=roundup(Plen/128)$ 6. For j=0 ← r, do $S_j=AES_K(Ctr_j)$ 7. Set $S=concatenate(S_1,S_2,\ldots S_r)$ 8. Return $C=(P \oplus MSB_{Plen}(S)) \mid\mid (MIC \oplus MSB(S_0))$
Decryption-Verification	Decryption scheme: 1. Set $r=roundup(P_{len}/128)$ 2. For j=0 ← r, do $S_j=AES_K(Ctr_j)$ 3. Set $S=concatenate(S_1,S_2,\ldots S_r)$ 4. Set $P=MSB_{Clen-Tlen}(C) \oplus MSB_{Clen-Tlen}(S)$ 5. Set $MIC\_dec=LSB_{Tlen}(C) \oplus MSB_{Tlen}(S_0)$ Verification scheme: 6. Apply formatting function to P to obtain $B_0,\ldots B_m$ 7. Set $Y_0=AES_k(B_0)$ 8. For i=1 ← m, do $Y_i=AES_K(B_i \oplus Y_{i-1})$ 9. Set $MIC=MSB_{32}(Y_m)$ 10. If (MIC_dec≠ MIC) VALID = 0 else VALID = 1 and return P



AEC-CBC mode



AES CTR mode

In IEEE 802.11i, AES encryption for CCMP is standardized on the plaintext block and cipher key of 128 bits length. The original transformation structure of AES forward cipher algorithm is depicted in Fig. 4. The AES encryption treats 128 bits input block as group of 16 bytes organized in a 4x4 matrix named State. The AES cipher of 128 bits key length consists of an initial transformation, nine iterations, and last round. Initial transformation applies Add-Round-Key (ARK) function only. On the other hand, iteration includes of four transformation functions: Byte-Substitution (BS), Shift-Row (SR), Mix-

Column (MC), and ARK, while the last round executes three functions BS, SR and ARK. We notice that round-key is different from each other for each ARK function in initial round, iterations, and last round as well. Number of round-keys for AES forward cipher with 128-bit key length are 11. The extra 10 round-keys are obtained on-the-fly by applying key generation process called Key-Scheduling.

In this study, we have implemented AES encryption with a reorderingBS transformation and SR transformationin the iterations of AES encryption algorithm, shown in Fig. 5. The combinationof BS and SR provides maximal period value of the composition, approximately 2^{19}, compared to original method is about 2^{18} [7].

In additional, we chose to implement AES cipher with composite Sbox instead of LUT or ROM. As you may know that, SB function in AES cipher is a nonlinear transformation with compute multiplicative inverse in the finite field GF(2^8) followed by an affine transformation which presents as the matrix AT(a) below[7]. Thus, multiplicative inverse computation is an intensive effort as well as consuming resources if we do directly with GF(2^8) arithmetic operation. In order to simplify the Sbox, we will figure the multiplicative inverse of the GF(2^8) in its composite field GF((2^4)^2)[8].

$$AT(a) = C_{8\times8}(1,1,1,1,1,0,0,0)\times(a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7)^T \oplus (0,1,1,0,0,0,1,1)^T \quad (1)$$

where C_{8x8} is circulant matrix

Every element in the GF(2^8) is represented as a polynomial with the most power is seven. For example:

$$\{10100111\}_2 = \{A7\}_H = x^7 + x^5 + x^2 + x + 1 \quad (2)$$

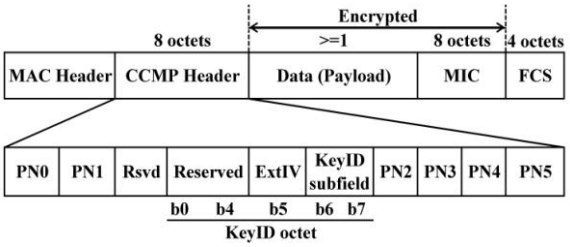
Come up with the ideal to downgrade the complexity of computing multiplicative inverse in the GF(2^8) which expressed in [8], we can represent an element in the GF(2^8) as an element in its composite field GF((2^4)^2) as (bx+c) with an appropriate irreducible polynomial x^2+x+λ, where b is the most 4 significant bits and c is the least significant bits of the element in the GF(2^8). Thus, the multiplicative inverse value can be obtained in the GF((2^4)^2) with the belowformula[9]

$$(bx+c)^{-1} = b\left(b^2B + bcA + c^2\right)^{-1}x + (c+bA)\left(b^2B + bcA + c^2\right)^{-1} \quad (3)$$

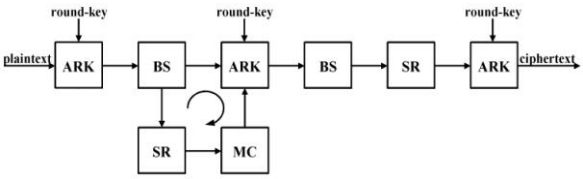
In case of this work, we selected x^2+x+λ as irreducible polynomial A=1 and B=λ. Since then the formula above can be simplified as:

$$(bx+c)^{-1} = b\left(b^2\lambda + bc + c^2\right)^{-1}x + (c+b)\left(b^2\lambda + bc + c^2\right)^{-1} \quad (4)$$

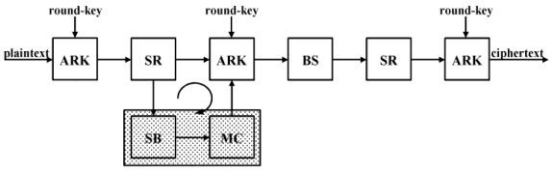
The multiplicative inversion of an element in the GF(2^8) now can be computed in the composite field GF((2^4)^2). We note that all the arithmetic operations of the formula below, multiply (X), addition (⊕), multiply with λ (xλ), squaring (x^2) and multiplicative inversion (x^{-1}) are done in the GF(2^4) instead of GF(2^8). Fig. 6depicts the multiplicative inversion block diagram in the composite fieldGF((2^4))^2.



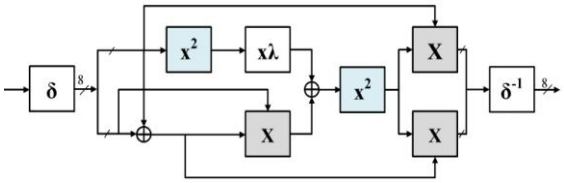
Expanded MPDU after generation-encryption process



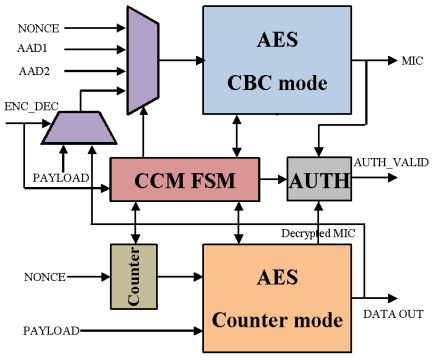
AES forward cipher process



AES forward cipher with modified process



GF(2^8) Multiplicative inversion block diagram



AES-CCM proposed architecture

IV. PREVIOUS RELATED WORKS

There are many hardware implementation approaches of AES-CCM in different design aspects e.g. AES-CCM sequential architecture [10] or interleave architecture [11], [12], [13] and [14] which uses one AES encryption core for both CTR and CBC mode for resources saving. Hence, its throughput is low. Another report for the Wireless Sensor Network (WSN) IEEE 802.15 considered the important design factor was energy [15]. Because of saving power, the core was operated at low clock

frequency. Consequently, throughput is low. In [16], the author used two AES encryption engines for CBC mode and CTR mode, but throughput was moderate. Those implementations may not be suitable for gigabit WLAN due to throughput results.

In addition, those studies gave a heed to design AES-CCM with generation-encryption process and did not pay attention on supporting both generation-encryption and decryption-verification. Therefore, the demand of high throughput AES-CCM aimed to Wi-Fi Protected Access 2 (WPA2) for the IEEE 802.11ac is pointed out. The next section, we provides discussion on AES-CCM implementation for gigabit communication which is capable of operating in both generation-encryption and decryption-verification modes.

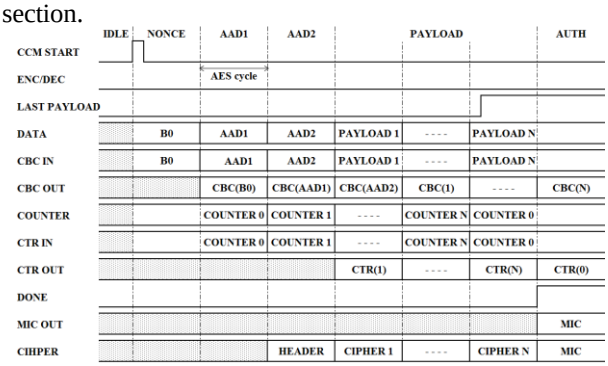
V. AES-CCM Proposed Architecture

The proposed architecture of AES-CCM for IEEE 802.11ac is shown in the Fig. 7.The AES-CCM core uses two AES forward cipher for CTR mode and CBC mode to enhance throughput.

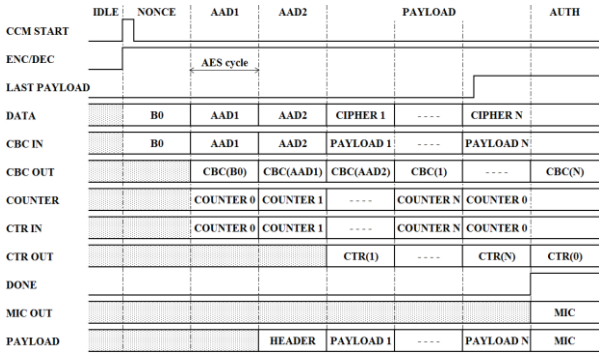
Counter block is intended to increase by one for each payload block. The initial value of counter is set as counter generation function which defined by a combination of {Flag field, NONCE, Index}[4].In IEEE 802.11ac, Flag field is set to 01_H. Index starts counting from 0000_H and counted up by one until it reaches to the number of blocks in MPDU data. Index zero is reserved for encryption or decryption MIC. The number of blocks in MPDU is calculated by rounding up quotient of the variable described length in bytes of MPDU divides 8.4to1 MUX is a multiplexer which controlled by CCM FSM. 4to1 MUX and 2to1 MUX are employed to choose the suitable data input for CBC mode corresponding to each state of CCM process. The block called AUTH provides authentication for the MPDU data when the core performs encryption-verification process. The AUTH_VALID signal is asserted if and only if the encrypted MIC from CTR mode equals to the retrieved MIC from CBC mode.

The block named CCM FSM is a finite state machine. It provides control to AES CTR, AES CBC, Counter, AUTH block as well as 4to1 MUX. This FSM is designed to achieve the symmetrical in data processing for generation-encryption and decryption-verification procedure. The timing diagram of the proposed architecture of AES-CCM is illustrated in Fig. 8 and Fig. 9 for generation-encryption and decryption-verification processes, respectively.

In the CCM algorithm, the data processing flows for generation-encryption and decryption-verification are different. To obtain the symmetry, counter is handled to increase by 1 right after the AAD1 phase. Counter is reset to index zero at the phase of processing last payload block. This counter value with index zero is supplied to CTR mode for calculating MIC message. We note that the DATAOUT of CTR mode is encrypted payload in the generation-encryption process or decrypted payload if the AES-CCM core operates in decryption-verification mode. The implementation results and comparisons of the proposed AES-CCM core are men-tionedin the next section.



Proposed architecture timing diagram for generation-encryption



Proposed architecture timing diagram for decryption-verification

TABLE III. Comparison Table of Different AES-CCM Implementation

Author	Target	Resource	Architecture	Latency	Throughput (Mbps)
E. Lopez-Trejo[13]	Spartan3 3s4000	Slices: 2154 RAMs: 106	Parallel	N/A	1051Mbps @100MHz
D. Bae [8]	Stratix	Logic cells: 5606	Sequential	44 clock cycles	285Mbps @50MHz
J. D. Ji [7]	Spartan3 A 3s700a	Slices: 1803 ROMs: 4	Sequential	N/A	588Mbps @105MHz
A. Aziz [10]	Spartan3 3s50pq	Slides: 487 RAMs: 4	Sequential	N/A	687Mbps @247MHz
Helion [15]	ASIC	<19K gates	N/A	N/A	>800Mbps
Our design	CMOS SAED90	20.5K gates	Parallel	36 clock cycles	2690Mbps @264MHz

VI. Mplementation Results and Comparisions

The design of CCM-AES core was written in Verilog HDL language, simulated in Modelsim SE6.6a with several test vectors [17] and synthesized in Synopsys Design Compiler target CMOS SAED90nm process.

Table. III. summarizes some design results, such as the use of resources, throughput, implementation target, and latency, of our design in comparison with the other studies as well as commercial product, such as AES-CCM core from Helion[18]

We assumed a 1020-byte payload data to AES-CCM core under clock frequency 264MHz for computing throughput. With that input data, AES-CCM core needs to process 67 blocks of 128-bit (3 additional blocks are NONCE and AAD data). In our design, the number of clock cycles for processing one block data, which called AES cycle, is 12. We obtained throughput by (5). Where, N is the number of blocks of payload data.

$$Throughput = \frac{N \times 128}{(N+3) \times 12} \times 264 \qquad (5)$$

It can be seen that our design accomplishes the requirement of operation with gigabit WLAN communication in IEEE 802.11ac. In comparison with other reports, our design performs high throughput and short latency. However, we use more resources than the commercial product [18].That can be explained by our proposed AES-CCM core was designed in parallel architecture.

VII. Conclusions

This paper has proposed thenovelhigh throughput parallel architecture of AES-CCM forthe IEEE 802.11ac standard.We also introduce the symmetricalarchitecture of AES-CCM which supports both generation-encryption and decryption-verification processesin an equivalent data processing routine.Moreover, we have employed AES forward cipher with a reordering transformation to enhance period of the composition compared to original algorithm.The hardware implementation results confirm that our design achieves throughput of 2.69Gbps and takes only 36 clock cycles to response to the input payload. For those advantages, the AES-CCM core is well suitedand ready to integrateto VLSI gigabit wireless communication applications, such as the VHT WLANs IEEE 802.11ac chip.

References

[1] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Costa, and B. Walke, "The {IEEE} 802.11 universe," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 62–70, Jan. 2010.

[2] E. H. Ong, "IEEE 802.11ac: Enhancements for Very High Throughput WLANs," 2011.

[3] "IEEE Standard for Information Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements," *IEEE Std 802.11i-2004*, pp. 1 –175, 2004.

[4] "Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality." NIST SP 800-38C, 2004.

[5] M. Leon, R. Aldeco, and S. Merino, "Performance analysis of the confidentiality security service in the IEEE 802.11 using WEP, AES-CCM, and ECC," in *2nd International Conference on Electrical and Electronics Engineering*, 2005, pp. 52– 55.

[6] C. K. Koc, *Cryptographic Engineering*, C. K. Koc, Ed, 1st ed. Springer, 2008.

[7] E. Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic," 2007.

[8] V. Rijmen. (2000). Efficient Implementation of the Rijndael S-box. [Online]. Available: http://www.researchgate.net/publication

[9] X. Zhang and K. K. Parhi, "On the optimum constructions of composite field for the aes algorithm," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 10, pp. 1153–1157, Oct 2006.

[10] J. D. Ji, S. W. Jung, E. A. Jun, and J. Lim, "Efficient sequential architecture for the aes ccm mode in the 802.16e standard," in *Second International Conference on Intelligent Networks and Intelligent Systems*, September 2009, pp. 253–256.

[11] D. Bae, G. Kim, J. Kim, S. Park, and O. Song, "An efficient design of ccmp for robust security network," in *Information Security and Cryptology*, vol. 3935, D. H. Won and S. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 352–361.

[12] I. A. Badillo, C. F. Uribe, R. Cumplido, and M. M. Sandoval, "FPGA implementation and performance evaluation of aes-ccm cores for wireless networks," in *International Conference on Reconfigurable Computing and FPGAs, 2008. ReConFig '08*, 2008, pp. 421–426.

[13] A. Aziz and N. Ikram, "An fpga-based aes-ccm crypto core for ieee 802.11 I architecture," *International Journal of Network Security International Journal of Network Security*, vol. 5, no. 2, pp. 224–232, 2007.

[14] Z. C. Rossan and M. R. Doomun, "Performance of interleaved cipher block chaining in CCMP," in *Novel Algorithms and Techniques in Telecommunications and Networking*, T. Sobh, K. Elleithy, and A. Mahmood, Eds. Dordrecht: Springer Netherlands, 2010, pp. 53–58.

[15] L. Huai, X. Zou, Z. Liu, and Y. Han, "An energy-efficient AES-CCM implementation for IEEE802.15.4 wireless sensor networks," in *International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC '09*, 2009, vol. 2, pp. 394–397.

[16] E. López-Trejo, F. Rodríguez-Henríquez, and A. Díaz-Pérez, "An FPGA implementation of CCM mode using AES.," in *ICISC*, 2005, vol. 3935, pp. 322–334.

[17] S. Leffler. (2004).Test Vectors for CCMP IEEE 802.11. [Online]. Available: http://fossies.org/linux/misc/madwifi-0.9.4.tar.gz:a/madwifi-0.9.4/regression/ccmp/test_ccmp.c.

[18] Helion AES-CCM Core. [Online]. Available: http://www.heliontech.com/aes_ccm.htm.

Dang Khoa Nguyen received the B.S. in electronicsform University of Techno-logyin 2008 and M.S in microelectronics from University of Science in 2012, both in Ho Chi Minh city, Vietnam. He is currently pursuing PhD degree in computer science in Kyushu Institute of Technology, Fukuoka, Japan. His re-search interests include wireless security and wireless systems.

Leonardo Lanantereceived theB.S and M.S. in electrical engineering both from University ofthe Philippines in 2005 and 2007 respectively. He received his PhD in computerscience in Kyushu Institute of Technology in 2011. He is currently pursuingpost-doctoral research also in KyushuInstitute of Technology. His researchinterests include synchronization algorithms and wireless signal processing for MIMO OFDM.

Hiroshi Ochireceived Ph.D. degree in Tokyo Metropolitan University in 1991 and MBA degree from Kyushu University in 2007. He is currently with Kyushu Institute of Technology as a professor in computer and electronics engineering department from 1999. His current researches include wireless signal processing, VLSI chip design and MOT education. He is also a CEO of aventure company namedRadrixco.ltd.