JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.16, NO.3, JUNE, 2016
http://dx.doi.org/10.5573/JSTS.2016.16.3.293
ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

Area-Optimized Multi-Standard AES-CCM Security Engine for IEEE 802.15.4 / 802.15.6

Injun Choi[1] and Ji-Hoon Kim[2]

Abstract—**Recently, as IoT (Internet of Things) becomes more important, low cost implementation of sensor nodes also becomes critical issues for two well-known standards, IEEE 802.15.4 and IEEE 802.15.6 which stands for WPAN (Wireless Personal Area Network) and WBAN (Wireless Body Area Network), respectively. This paper presents the area-optimized AES-CCM (Advanced Encryption Standard – Counter with CBC-MAC) hardware security engine which can support both IEEE 802.15.4 and IEEE 802.15.6 standards. First, for the low cost design, we propose the 8-bit AES encryption core with the S-box that consists of fully combinational logic based on composite field arithmetic. We also exploit the toggle method to reduce the complexity of design further by reusing the AES core for performing two operation mode of AES-CCM. The implementation results show that the total gate count of proposed AES-CCM security engine can be reduced by up to 42.5% compared to the conventional design.**

Index Terms—**AES-CCM, composite field arithmetic, low-complexity, multi-standard**

I. INTRODUCTION

The low complexity security algorithm and its hardware implementation become essential to protect information over WPAN (Wireless Personal Area Network) and WBAN (Wireless Body Area Network) utilized in various areas. The IEEE 802.15.4 standard, one of the well-known standards for WPAN system, provides the security service based on the AES-CCM (Advanced Encryption Standard – Counter with CBC-MAC). Also, in the IEEE 802.15.6, the first international WBAN standard, AES-CCM is adopted in order to guarantee information security.

The AES-CCM consists of the CBC-MAC (Cipher Block Chaining-Message Authentication Code) mode for the MAC calculation and the CTR (Counter) mode for data encryption with the AES (Advanced Encryption Standard) algorithm. As the AES core is dominant hardware element of the AES-CCM implementation, the hardware optimization of the AES core plays a key role to reduce the complexity of AES-CCM security engine.

In this paper, the area-optimized AES-CCM security engine is proposed for IEEE 802.15.4 / 802.15.6 systems where highly limited hardware resources are available for low-cost implementation. To lower the complexity of AES core implementation, which is used as cryptographic primitive for AES-CCM, the folding technique is exploited for small gate count and the S-box without the use of memory is also presented. In addition, we reduce the complexity of the multi-standard AES-CCM security engine with the toggling method where only single AES core is required in two operating mode of AES-CCM.

II. LOW-COMPLEXITY AES CORE

The AES algorithm, which is used in AES-CCM security operation, is one of the symmetric key block cipher algorithm. The AES algorithm interpreted in

Manuscript received Aug. 4, 2015; accepted Jan. 16, 2016
[1] Dept. of EE, Chungnam National University, Daejeon, Korea
[2] Dept. of EIE, Seoul National University of Science and Technology, Seoul, Korea
E-mail : jihoonkim@seoultech.ac.kr

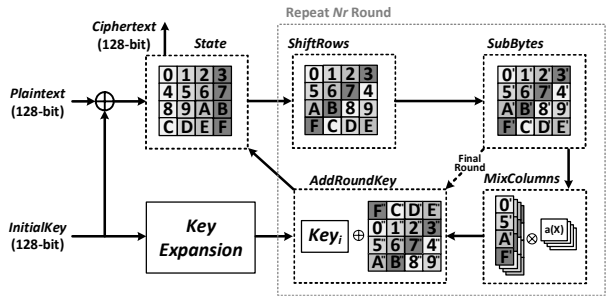


Fig. 1. AES encryption algorithm.

Galois field consists of four sub-algorithms performed during the execution of the encryption. In this section, we introduce the AES algorithm and present a various methodology to reduce the hardware complexity.

1. AES Algorithm

The AES algorithm is a symmetric block cipher that processes 128-bit data blocks arranged as a 4x4 matrix of bytes called a ***State***. All bytes in the AES algorithm are regarded as elements of the Galois Field. The Galois Field elements can be added and multiplied by using the mathematical concepts different from standard arithmetic [1]. The symbol "⊕" and "⊗" are used to denote the addition and the multiplication in the Galois Field, respectively. Galois Field Arithmetic allows mathematical operation to encrypt data easily and effectively. As shown in Fig. 1, the AES algorithm consists of four sub-algorithms including ***ShiftRows***, ***SubBytes***, ***MixColumns***, and ***AddRoundKey***. In the encryption of the AES algorithm, each round except the final round requires all of the above sub-algorithms, while the final round does not have the ***MixColumns***. The AES algorithm takes the initial key and performs a ***KeyExpansion*** to generate *RoundKey* (***Key***$_i$) used in *i-th* round. For the AES-128 used in the IEEE 802.15.4, the number of rounds is represented by ***Nr***, where ***Nr*** = 10 [1].

In the ***ShiftRows***, the bytes in each row of the round data cyclically shifted over different numbers of bytes. The ***SubBytes*** is a non-linear substitution that operates independently on each byte. The ***MixColumns*** operates on the ***State*** column by column and the bytes of each column are mixed together. In the ***AddRoundKey***, the ***State*** is added to the *RoundKey*, which generated from the ***KeyExpansion*** with one *InitialKey* in each round.

In the ***SubBytes*** step, each byte of the ***State*** is replaced by new byte using the substitution table (S-box) constructed by composing two transformations. The S-box is the multiplicative inverse of a Galois field GF (2^8) with the irreducible polynomial, followed by an affine transformation. The ***SubBytes*** can be described by matrix form as Eq. (1), where ***M*** is an 8x8 binary matrix, and ***C*** is 8-bit vector {01100011}.

$$S' = M \cdot S^{-1} + C \tag{1}$$

Most of the operations are implemented using a chain of XORs except for the ***SubBytes*** that performs multiplicative inverse operation. The ***SubBytes*** is the most complex operation which dominates the hardware cost of the AES core implementation and performance of the AES algorithm. Therefore, the optimization of the ***SubBytes*** is critical to the low complexity AES design.

2. Proposed AES Core

To reduce hardware complexity, the folded

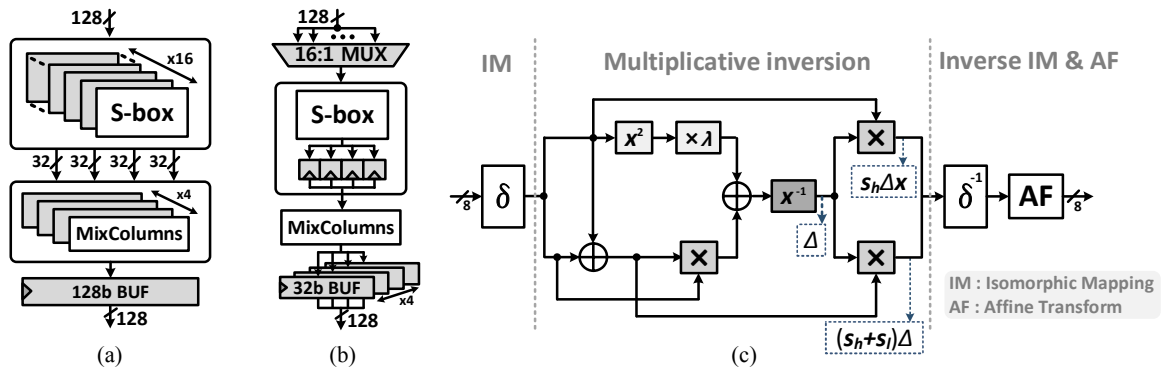


Fig. 2. (a) 128-bit block wide AES, (b) Proposed 8-bit folded AES, (c) S-box employing fully combinational logic.

architecture is adopted with the S-box based on CFA (Composite Field Arithmetic). In resource constrained systems, it is not suitable using the traditional LUT-based S-box which increases the total memory requirement. We exploit the on-the-fly S-box based on CFA instead of the traditional S-box. The AES core is a strong candidate for folding, since the AES algorithm is a repetition of four sub-algorithms. Folding techniques reusing computational units have been exploited in order to reduce hardware complexity.

The S-box that can be implemented without the use of memory has been proposed in [2] and further optimization for minimizing the hardware complexity of the S-box has been proposed in [3]. The S-box in [4] is implemented in composite field and employs circuit sharing between S-box and S-box^{-1}. These low-complexity S-box relies on CFA that offers efficient implementation for operations such as multiplication and inversion. In order to exploit CFA, the elements in original field GF (2^8) should be mapped to an isomorphic composite field. The field mapping is performed by isomorphic mapping function that is decided by the field polynomials of GF (2^8) and its composite fields. The transformation matrix $\boldsymbol{\delta}$ and its inverse $\boldsymbol{\delta^{-1}}$ that are used in isomorphic mapping can be obtained by the exhaustive search algorithm [3]. The matrix $\boldsymbol{\delta}$ and $\boldsymbol{\delta^{-1}}$ are shown Eq. (2).

$$\delta = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}_2 \quad \delta^{-1} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}_2 \tag{2}$$

After isomorphic mapping, the elements that have been arranged to composite field can be described as $\boldsymbol{s_h x+s_l}$, where $\boldsymbol{s_h}$ and $\boldsymbol{s_l}$ are elements of GF (2^4) and $\boldsymbol{x}$ is a root of composite field polynomial. Using Extended Euclidean algorithm, the multiplicative inverse can be computed as Eq. (3).

$$\begin{aligned} &(s_h x + s_l)^{-1} = s\Delta x + (sh + sl)\Delta, \\ &\quad where \quad \Delta = (s_h^2 \lambda + s_l(s_h + s_l))^{-1} \end{aligned} \tag{3}$$

According to (3), the multiplicative inversion can consist of operations in sub field GF (2^4) such as multiply, addition, squaring and multiplicative inversion that can be implemented by fully combinational logic. The operations in sub field can significantly reduce the hardware complexity of S-box. Fig. 2(c) shows the low complexity S-box using composite field arithmetic. The squarer, multiplication and constant multiplication are illustrated by $\boldsymbol{x^2}$ block, × block and $\times\boldsymbol{\lambda}$ block. The $\boldsymbol{x^{-1}}$ block is the multiplicative inversion that is further decomposed to GF ((2^2)^2).

The AES is a symmetric encryption algorithm that has a fixed block size of 128-bit. A typical block-wide AES structure illustrated in Fig. 2(a) takes only one clock cycle with using 16 S-boxes and 4 ***MixColumns*** in each round. Since the block-wide AES focuses on high performance, it simultaneously uses many duplicated units. Since the block-wide structures require numerous hardware resources, it is not proper for WPAN application. To utilize the resources efficiently, the folded AES structure that reuses the hardware resources has been proposed in [2] as shown in Fig. 2(b). We exploit the folding techniques to reduce hardware complexity by reusing only one S-box and one ***MixColumns*** unit with the low overhead compared to the sub-algorithm unit reduction. Because the AES decryption operation is unnecessary in the AES-CCM mode, we use the AES encryption core that except the AES decryption operation.

III. AES-CCM FOR IEEE 802.15.4 / 802.15.6 STANDARDS

In this section, we present the AES-CCM algorithm and the security operations of the IEEE 802.15.4 and IEEE 802.15.6 standards. The AES-CCM designed for use with the AES provides both the confidentiality and the authentication of the data. The AES-CCM is composed with combinations of the CBC-MAC (Cipher Block Chaining-Message Authentication Code) mode and the CTR (Counter) mode. These combinations provide strong assurance of data integrity. The security operation of the IEEE 802.15.6 standard based on the AES-CCM defines the initial block construction that contains control information of the AES-CCM mode. The AES-CCM* used in the IEEE 802.15.4 standard includes

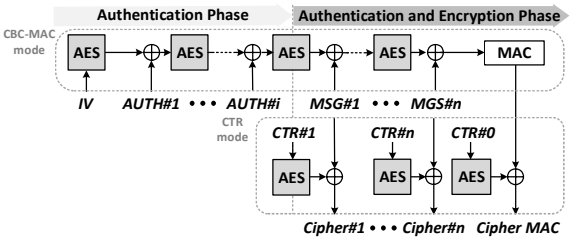


Fig. 3. Authentication and encryption process for the AES-CCM algorithm.

the features of the AES-CCM and additionally offers the various security option.

1. AES-CCM Algorithm

The AES-CCM consists of the CBC mode for the MAC generation and the CTR mode for data encryption. The CBC-MAC generates the MAC which provides strong assurance of authenticity to the overall payload by applying cipher block chaining to the associated data. The CTR generates the ciphertext that is encrypted from the message and the MAC which is generated from CBC-MAC.

In the CBC-MAC, the previous encrypted data block is XORed with each successive plaintext block to create block chaining. As each block depends on the previous block by chain of blocks, the MAC that depends on the overall payload ensures the data integrity. In the CTR, each counter block that consists of the nonce data and the counter sequence is encrypted by the AES encryption algorithm and then the resulting block is XORed with the message block to produce the ciphertext. Since the AES encryption algorithm is also used in the decryption of the AES-CCM, the AES decryption process is not necessary in the AES-CCM. Therefore, we exploit the encryption only AES hardware in order to reduce the hardware resources.

The authentication and encryption process of the AES-CCM algorithm are shown in Fig. 3. The ***IV*** (Initial Vector), ***MSG*** (Message) and the ***AUTH*** (Authentication data) are formatted according to the building blocks mechanisms in [5]. Both ***MSG*** blocks and ***AUTH*** blocks are used in the CBC-MAC to generate a MAC. On the other hand, the CTR uses only ***MSG*** blocks for encryption.

2. Security Operations for IEEE 802.15.4 / 802.15.6 Standards

The IEEE 802.15.6 standard provides the security service based on the AES-CCM for message encryption and authentication. The block ***IV*** defined in [5] contains control information such as the flag, the MAC size and the octet length, denoted as Q, of the message payload size. In the case of the IEEE 802.15.6 standard, the flag is whether or not to encrypt the message payload and the MAC size is four octets, denoted as MIC-32 (Message Integrity Codes-32), and the Q is two octets [5].

The AES-CCM* that is adopted security operation of the IEEE 802.15.4 standard is based on the AES-CCM. In contrast to the AES-CCM, the AES-CCM* provides the eight levels of security option. These security options can be classified four modes: unsecured, encryption only, authentication only, and encryption and authentication [6]. Moreover, these security options offer varying levels of data authenticity such as MIC-32, MIC-64 and MIC-128, that offers stronger data authenticity than that of the IEEE 802.15.6 standard.

IV. AREA-OPTIMIZED MULTI-STANDARD AES-CCM SECURITY ENGINE

There are several approaches to implement the area-optimized AES-CCM security engine that generates the MAC for the data integrity and the ciphertext for the confidentiality. The CBC-MAC and the CTR use the same AES core to encrypt the plaintext. So, the structure of the AES-CCM security engine hugely depends on the AES core management method, such as a sequential method, a parallel method, and a toggle method. The AES core management methods are shown in Fig. 4.

In the sequential method, since the RST (response time) increases linearly according to the overall payload size which contains both the authentication data and the message, the sequential method has a longer response time than others [7]. In the parallel method, the security engine has high data throughput and short response time by processing the CBC-MAC and the CTR at the same time, but the engine using parallel method [8, 9] requires more hardware resources than sequential method.

The security engine that is designed by the toggle method operates the CBC-MAC and the CTR with only

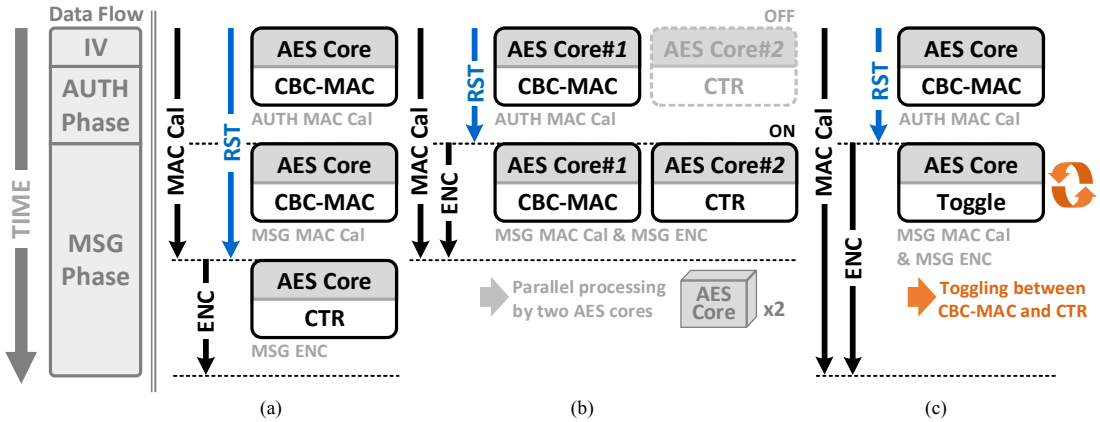


Fig. 4. AES Core management methods for AES-CCM (a) sequential method, (b) parallel method, (c) toggle method.

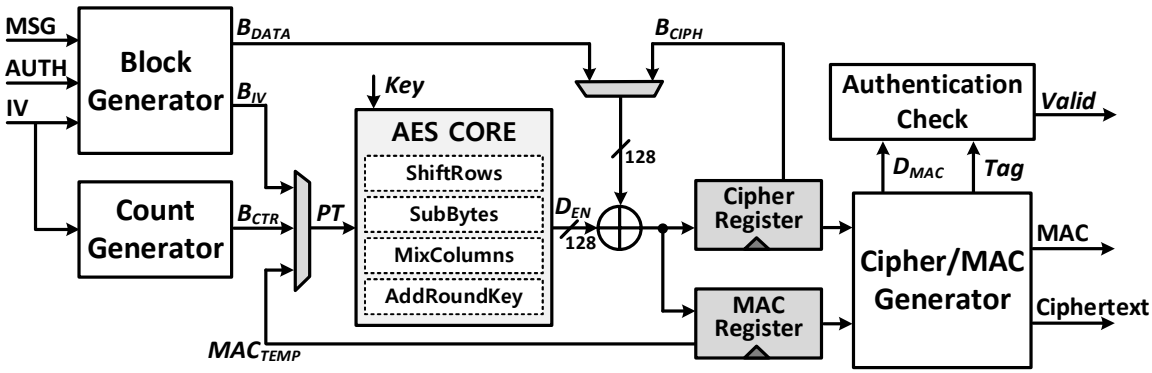


Fig. 5. Structure of the area-optimized AES-CCM security engine based on the toggle method.

one AES core by turns [10]. During the authentication phase, the security engine performs only the MAC data calculation in the CBC-MAC. Then, during the message phase, the security engine needs to operate both the CBC-MAC and the CTR. The MSG data blocks are used for the MAC calculation in CBC-MAC and encrypted to the ciphertext in the CTR by turns. The toggle method reduces the hardware resource by using only one AES core with the same response time in the parallel structure as shown in Fig. 4. The IEEE 802.15.4 and IEEE 802.15.6 standards which intend to design a type of WPAN and WBAN provide the low cost wireless communication with moderate data rate. Since the both WPAN and WBAN focus on the low-cost, the low complexity design is more optimum solution than high data rate design accompanied by parallel processing. So we employ the toggle method that provides a short response time and a low cost design using only one AES core for the area-optimized AES-CCM security engine.

Fig. 5 shows the structure of the area-optimized AES-CCM security engine that is implemented by using the toggle method. The Block Generator and the Count Generator are formatting the input data blocks, such as $\boldsymbol{B_{DATA}}$, $\boldsymbol{B_{IV}}$, and $\boldsymbol{B_{CTR}}$, that used in the CBC-MAC and the CTR. The ***PT*** (plaintext) that is selected by current state is encrypted to encrypted data block $\boldsymbol{D_{EN}}$ in the AES core. The $\boldsymbol{D_{EN}}$ is XORed with the data block $\boldsymbol{B_{DATA}}$ or the cipher block $\boldsymbol{B_{CIPH}}$, and then the result is stored in the **Cipher Register** or **MAC Register** according to the AES-CCM operation modes. The Authentication Check unit provides the ***Valid*** (validate signal) by comparing the ***TAG*** with the $\boldsymbol{D_{MAC}}$ in the verification state.

To lower the complexity of the security engine, we propose four techniques to reduce requirement of the hardware resource. First, we exploit the composite field arithmetic to reduce the hardware complexity of S-box that dominates the AES hardware cost. Second, the AES data path bit-width is restricted to 8-bit by using folding techniques to reuse the S-box and ***MixColumns*** unit. Third, to reduce unnecessary hardware resources, we use the AES encryption core that except the AES decryption operation. Finally, as the toggle method is adopted in the

security engine, the security engine is implemented with only one AES encryption core.

The proposed AES-CCM security engine can be operated with the various security modes. The security engine is processing the authentication and the encryption and constructing the MAC according to the security mode. The security mode is selected by the ***IV*** (Initial Vector) for the various combinations of the security options of IEEE 802.15.4 and IEEE 802.15.6 standards. Our security engine optimized with the four techniques can be compatible with multi-standard by offering the various security options.

V. IMPLEMENTATION RESULTS

The area-optimized AES-CCM security engine was described in Verilog-HDL and synthesized with a 65nm standard CMOS process. The proposed security engine runs at 6 MHz and 35 MHz for IEEE 802.15.4 and 802.15.6, respectively. The proposed security engine is designed to be working in conjunction with the IEEE 802.15.4-compatible ZigBee modem based on O-QPSK modulation that runs at the operating frequency of 6 MHz. Table 1 and 2 denote the implementation results of the proposed designs and the conventional designs.

Table 1 shows the comparison results between the proposed AES core and the one in [8]. In the proposed AES core, the total gate count can be dramatically reduced up to 88.1% in comparison with the 128-bit architecture AES core in [8] by exploiting 8-bit folded architecture and the CFA which leads to the low-complexity S-box implementation. Table 2 compares the proposed security engine with other designs published in [8, 9]. To summarize, [9] is the parallel structure using the 128-bit AES core; [8] exploits the parallel structure using the 8-bit AES core with the LUT-based S-box; this work proposes the area-optimized AES-CCM security engine that adopts the toggle method with the low-complexity 8-bit AES encryption core.

As denoted in Table 2, the gate count of the proposed security engine can be reduced by up to 42.5% in comparison with [9], while the data rate of 250 kbps that is required by the IEEE 802.15.4 standard. Moreover the proposed security engine can be operated at over a clock frequency of 35 MHz. The data rate of 10 Mbps required by the IEEE 802.15.6 standard is satisfied in the proposed security engine at the operating frequency of 35 MHz without the additional hardware cost. Also as shown in Fig. 5, the proposed security engine has the dedicated Authentication Check block which can check the correctness of the $\boldsymbol{D_{MAC}}$ for the payload by comparing with ***TAG***, while the previous works are based on software-based Authentication Check process which leads to high performance requirements in CPU / DSP core [8, 10]. Since the WPAN and WBAN devices mainly focus on area efficient implementation, this paper proposes the area-optimized AES-CCM security engine that is compatible with security services for IEEE 802.15.4 and IEEE 802.15.6 standard.

Table 1. Synthesis results of the AES Core

	[8]			Proposed
Technology/ Frequency	250 nm/ 10 MHz			65 nm/ 6 MHz
S-box Structure	LUT			CFA
Bit Width	128 bits	32 bits	8 bits	8 bits
AES Core Gate Count	15.98 K (100%)	7.73 K (48.3%)	4.02 K (25.1%)	3.50 K (21.9%)

Table 2. Synthesis results of the AES-CCM security engine

	[9]	[8]	Proposed
Technology/ Frequency	90 nm/ 264 MHz	250 nm/ 10 MHz	65 nm/ 6 MHz
Bit Width	128 bits	8 bits	8 bits
Structure	Parallel	Parallel	Toggle
AES Core Gate Count	N/A	4.02 K	3.50 K
Total Gate Count	20.5 K (100%)	14.9 K (72.6%)	11.8 K (57.5%)
Standard(s)	IEEE 802.11.ac	IEEE 802.15.4	IEEE 802.15.4 & 802.15.6

VI. CONCLUSIONS

In this paper, the design of the area-optimized AES-CCM security engine is described. The proposed security engine is compatible with IEEE 802.15.4 and IEEE 802.15.6 standards for WPAN and WBAN. The AES-CCM security engine is organized around the 8-bit AES encryption core that exploits the composite field arithmetic in order to reduce hardware complexity of the S-box. We use the encryption only AES core excepting the AES decryption operation that is unnecessary in the AES-CCM. By exploiting the toggle method that operates the CBC-MAC mode and the CTR mode by

turns with the proposed AES core, the total gate count of the security engine can be reduced by up to 42.5%. Implementation results show that our proposed design can effectively reduce the area cost while satisfying the data rate required by the IEEE 802.15.4 and IEEE 802.15.6 standards.

ACKNOWLEDGMENTS

This work was supported by the Center for Integrated Smart Sensors funded by the Ministry of Science, ICT & Future Planning as Global Frontier Project (CISS-2011-0031860) and by the National GNSS Research Center program of Defense Acquisition Program Administration and Agency for Defense Development

REFERENCES

[1] National Institute of Standards and Technology (NIST): FIPS-197: Advanced Encryption Standard (2001), Nov., 2001.

[2] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-Box optimization," in Proc. ASIACRYPT 2001, pp. 239-254, Dec., 2000.

[3] X. Zhang, K. K. Parhi, "High-speed VLSI architectures for the AES algorithm ," *Very Large Scale Integration Systems, IEEE Transactions on*, Vol. 12, Issue. 9, pp.957-967, Sep., 2004.

[4] Shen-Fu Hsiao, Ming-Chih Chen, and Chia-Shin Tu, "Memory-Free Low-Cost Designs of Advanced Encryption Standard Using Common Subexpression Elimination for Subfunctions in Transformations," *Circuits and Systems I, IEEE Transactions on*, Vol. 53, No. 3, pp. 615-626, Mar., 2006.

[5] IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks (2012), Feb., 2012.

[6] IEEE Standard for Local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) (2011), Sep., 2011.

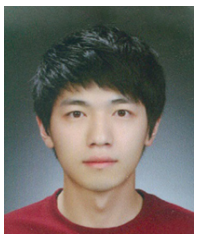
[7] A Aziz, N Ikram, "Hardware implementation of AES-CCM for robust secure wireless Network," *Annual ISSA Information Security Conference*, Vol. 5, pp. 44-51, 2005.

[8] L. Huai, X. Zou, Z. Liu, and Y. Han, "An Energy-Efficient AES-CCM Implementation for IEEE802.15.4 Wireless Sensor Networks," NSWCTC 2009, pp. 394-397, Apr., 2009.

[9] D. Nguyen, L. Lanante and H. Ochi, "High Throughput– Resource Saving Hardware Implementation of AES-CCM for Robust Security Network," *Journal of Automation and Control Engineering*, Vol. 1, No. 3, pp. 250-254, Sep., 2013.

[10] D. Bae, G. Kim, J. Kim, S. Park, O. Song, "An Efficient Design of CCMP for Robust Security Network ," ICISC 2005, Vol. 3935, pp. 352-361, 2006

[11] IP Cores Inc.: CCMZ1/CCMZ2 IEEE 802.15.4 (ZigBee™) CCM* AES Cores (2015), Available at http://www.ipcores.com/zigbee_802.15.4_aes_ccm



Injun Choi received the B.S. degree in the Department of Electronics Engineering in 2014, from Chungnam National University, Daejeon, Korea, where he is currently working toward the M.S degree. His interests include low-power digital VLSI circuit and SoC design.

Ji-Hoon Kim received the B.S. (*summa cum laude*) and Ph.D. degrees in electrical engineering and computer science from KAIST, Daejeon, Korea, in 2004 and 2009, respectively. His Ph.D. work focused on design of high performance CPU and baseband modem for mobile hand-held devices. In 2009, he joined Samsung Electronics, Suwon, Korea, where he worked on the SoC architecture design for next generation cellular modem. He was a faculty member of Chungnam National University. In 2016, he joined the faculty of the Department of Electrical and Information Engineering at Seoul National University of Science and Technology, Seoul, Korea, and now is an associate professor. His current interests include CPU/DSP, communication modem, and low power SoC design.