https://doi.org/10.33778/kcsa.2024.24.1.069

A White-box ARIA Implementation

Hong Tae Kim*

Abstract

The white-box implementation is a cryptographic technique used to protect the secret key of a cryptographic system. It is primarily employed for digital rights management for music and videos. Since 2002, numerous white-box implementations have been developed to ensure secure digital rights management. These have been applied to AES and DES. ARIA, a 128-bit block cipher with an involution substitution and permutation network (SPN), was selected as a South Korean standard in 2004. In this paper, we propose the first white-box ARIA implementation. Our implementation consists of 7,696 lookup tables, with a total size of 1,984 KB. We demonstrate that it also has considerable white-box diversity and white-box ambiguity from a security perspective.

화이트박스 ARIA 구현

김홍태*

요 약

화이트박스 구현은 암호 시스템의 비밀키를 보호하는데 사용되는 암호화 기술이다. 주로 음악, 비디오 등의 디지털 저작권 관리에 사용된다. 2002년 이후, 안전한 디지털 저작권 관리를 확보하기 위해 많은 화이트박스 구현이 개발되었다. 이는 고급 암호화 표준(AES) 및 데이터 암호화 표준(DES)에 적용되었다. ARIA는 대합(involution) 대입 치환 네트워크(SPN)를 사용하는 128비트 블록 암호로, 2004년에 한국 표준으로 채택되었다. 본 논문에서는 최초의 화이트박스 ARIA 구현을 제안한다. 우리의 구현은 전체 크기가 1,984KB인 7,696개의 조회 테이블로 구성된다. 안전성 측면에서 현저한 화이트박스 다양성과 화이트박스 모호함이 있음을 보인다.

Key words : White-box implementation, ARIA, Block cipher, Lookup table

접수일(2024년 01월 22일), 수정일(1차: 2024년 02월 15일),
게재확정일(2024년 03월 04일)

* 공군사관학교 수학과 교수

1. Introduction

In 2002, Chow et al. introduced the first white-box implementations [4, 5]. They proposed a white-box Advanced Encryption Standard (AES) implementation using a table-based method [4]. They could reduce the storage using a lot of small XOR tables instead of a few big tables. Billet et al. provided an algebraic attack on Chow et al.'s white-box AES implementation with less than 2^{30} computational complexity [3]. Bringer et al. presented a new white-box AES implementation with extra random parts [1]. Some different S-boxes instead of the original AES S-box were used in this scheme. Mulder et al. provided an algebraic attack on Bringer et al.'s white-box AES implementation to get an equivalent key with 2^{17} computational complexity [12]. Kim presented modified White-box AES implementations and attacked them [15, 16].

The white-box Data Encryption Standard (DES) implementation was proposed by Chow et al. [5], and then by Link and Neumann [9]. Those schemes were broken in a few years later [6, 14]. According to those papers, white-box DES implementations presented were broken with 2^{14} computational complexity.

In 2008, Michiels et al. defined a generic class of white-box implementations over general substitution-linear transformation (SLT) cipher and presented a cryptanalysis on white-box implementations for block cipher with this property [11]. They exploited two main techniques, one by Billet et al. [3] and the other by Biryukov et al. [2]. Karroumi presented a white-box AES implementation to enhance the security using 61,200 dual ciphers of AES [7]. These dual ciphers are made of different types of AES original operations and give the same result as the original AES. Lepoint et al. extracted the key from Karroumi's white-box AES implementation with 2^{22} computational complexity [10].

In 2003, Kwon et al. proposed a block cipher called ARIA [8]. The name ARIA comes from the initials of Academy, Research Institute, and Agency which means cooperative efforts of Korean researchers in designing ARIA. It is a Korean standard block cipher which is an involution SPN structure. Also, the Internet Engineering Task Force specifies a set of cipher suites for the Transport Layer Security protocol to support the ARIA encryption algorithm in 2011. We introduce the first white-box ARIA implementation using many lookup tables containing different Exclusive Or (XOR) tables.

The remainder of this paper is organized as follows. In Section 2, we give some notations and introduce the definition of block cipher. The specifications of ARIA are given in Section 3. In Section 4, we give a white-box ARIA implementation. Analysis of the performance and the security is given in Section 5. We end with some remarks in Section 6.

2. Preliminaries

Shannon presented two methods for a secure cipher, confusion, and diffusion, respectively [13]. The method of confusion is to make the relation between the ciphertext and the key complex and involved. The method of diffusion is to make the partial part of the plaintext influence many parts of the ciphertext. By applying these methods iteratively, the cryptosystem can be made more secure. This principle is commonly used in most block ciphers.

An n-bit block cipher is a deterministic function mapping n-bit plaintext blocks to n-bit ciphertext blocks. The block cipher consists of the encryption function E_k and the decryption function D_k. The encryption function E_k is given as follows:

$$E_k : \{0, 1\}^m \times \{0, 1\}^n \rightarrow \{0, 1\}^n \quad (1)$$

where $E_k(P)=C$ for m-bit key k, n-bit plaintext P, and n-bit ciphertext C. The decryption function D_k is given as

$$D_k : \{0, 1\}^m \times \{0, 1\}^n \rightarrow \{0, 1\}^n \quad (2)$$

where $D_k(C)=P$ for m-bit key k, n-bit ciphertext C, and n-bit plaintext P. Two functions must have the property that $D_k(E_k(P))=P$ for all $k \in \{0, 1\}^m$.

We use notations as follows.

- $GF(2^8)$: Finite field with order 2^8 (or $\{0, 1\}^8$)
- Z_2 : Group of integers modulo 2
- A_i : 8×8 invertible matrix of $GL(8, Z_2)$ where $GL(8, Z_2)$ is a general linear group over Z_2
- $\cdot$: Multiplication of two operands, matrix and vector, or two matrices
- $\oplus$: A bitwise XOR operation
- S_i : $GF(2^8) \rightarrow GF(2^8)$ defined by $S_i(x) = A_i \cdot x^{-1} \oplus a_i$ where $A_i \in GL(8, Z_2)$ and $a_i \in GF(2^8)$
- $\ggg n$: Right circular rotation of operand by n bits
- $\lll n$: Left circular rotation of operand by n bits

A byte b can be considered as a polynomial $b_7x_7 + b_6x_7 + \cdots + b_0$ where $b = (b_7b_6 \cdots b_0)_2$ and $b_i \in \{0, 1\}$ for $i = 0, 1, \cdots, 7$.

3. ARIA

We focus on the explanation of the block cipher ARIA with 12-round.

3.1 Key schedule

The key schedule of ARIA uses 128-, 196- or 256-bit master key for 12-, 14- or 16-round, respectively. It consists of two processes which are initialization and round key generation, respectively. We omit these and you can get the concrete processes in [8]. The decryption round keys are derived from the encryption round keys. Let B be the diffusion layer of ARIA in Section 3.4. The decryption round keys for ARIA with 12-rounds are given by

$$DK_1 = EK_{13},\ DK_2 = B \cdot EK_{12},$$
$$DK_3 = B \cdot EK_{11},\ \cdots,$$
$$DK_{12} = B \cdot EK_3,\ DK_{13} = EK_1,$$

where DK_i and EK_i are the i-th round decryption key and encryption key for $i = 1, 2, \cdots, 12$, respectively. DK_{13} and EK_{13} are the last round decryption key and encryption key of the final round, respectively.

3.2 Key addition

This is done by bitwise XOR operation with 128-bit round key.

3.3 Substitution layer

There exist two s-boxes and their inverses. The s-box S_1 : $GF(2^8) \rightarrow GF(2^8)$ is defined by $S_1(x) = A_1 \cdot x^{-1} \oplus a_1$ where

$$A_1 = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \text{ and } a_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

The s-box S_2 : $GF(2^8) \rightarrow GF(2^8)$ is defined by $S_2(x) = A_2 \cdot x^{247} \oplus a_2$ where

$$A_2 = \begin{pmatrix} 0&1&0&1&1&1&1&0 \\ 0&0&1&1&1&1&0&1 \\ 1&1&0&1&0&1&1&1 \\ 1&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&0&0 \\ 1&0&0&0&0&0&0&1 \\ 0&1&0&1&1&1&0&1 \\ 1&1&0&1&0&0&1&1 \end{pmatrix} \text{ and } a_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

The inverse functions of S_1 and S_2 are denoted by S_1^{-1} and S_2^{-1}, respectively.

3.4 Diffusion layer

The diffusion layer $B : GF(2^8)^{16} \to GF(2^8)^{16}$ is defined by

$$B \cdot \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{15} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{15} \end{pmatrix}$$

where

$$B = \begin{pmatrix}
0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\
0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\
0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\
1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\
1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\
0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\
1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\
0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\
1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\
1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\
0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\
0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\
0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\
1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\
1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\
0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1
\end{pmatrix}$$

and x_i, y_i are in $GF(2^8)$ for $i = 0, 1, \cdots, 15$. For example, we get

$$y_0 = x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14}$$

from the above. This linear map B has an involution structure, i.e. B^2 = identity where B is a 16×16 matrix with coefficients in {0, 1}.

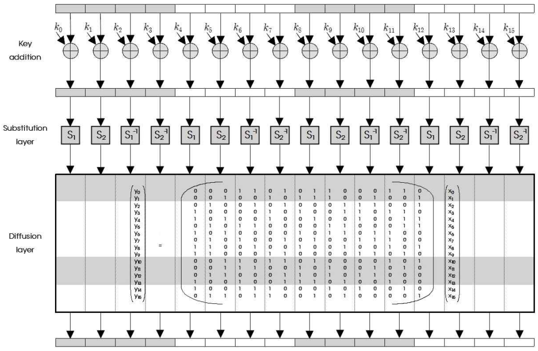
3.5 The cipher

We explain mainly 12-round ARIA in this paper. The i-th round function($1 \leq i \leq 11$) is given by $DL \circ SL \circ KA$, where KA is the key addition, SL is the substitution layer and DL is the diffusion layer. Figure 1 shows the process of $DL \circ SL \circ KA$. The final round function is given by $KA \circ SL \circ KA$ and the figure of this is given similarly. The decryption process is the reverse of the encryption process and uses different round keys compared to encryption round keys as Section 3.1. The i-th round function($1 \leq i \leq 11$) is given by $KA \circ SL \circ DL$ and the final round function is given by $KA \circ SL \circ KA$.

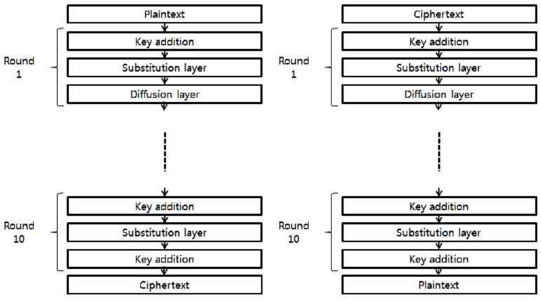
The encryption and decryption processes are given in Figure 2. Both processes have eleven same round functions and a different round function, respectively.

We can get the ARIA implementation using the tables in Table 1.

<Fig. 1> The i-th round of ARIA (i = 1, 2, ⋯ , 11)


k_0 k_1 k_2 k_3 k_4 k_5 k_6 k_7 k_8 k_9 k_{10} k_{11} k_{12} k_{13} k_{14} k_{15}
Key addition
Substitution layer
S_1 S_2 S_1^{-1} S_2^{-1}
Diffusion layer

<Fig. 2> Encryption and Decryption processes of ARIA



Encryption Decryption

4. White-box ARIA Implementation

We only present a white-box ARIA implementation for the encryption process. White-box ARIA implementation for the decryption process is given by a similar method and we omit this.

<Table 1> The number and size of the ARIA implementation

Processes	Number of tables	Size (KB)
$DL \circ SL \circ KA$ (i = 1, 2, ⋯, 11)	176	704
$KA \circ SL \circ KA$	16	64
XOR	1	64
Total	193	832

4.1 Blocking method

White-box implementation for a block cipher was made by adding extra information to the block cipher. After dividing a block cipher into some parts, we hide information about the original part using extra information with linear and nonlinear components. Each part of these is implemented by many input/output tables. There exist 2 types (8-bit input/128-bit output tables, 8-bit input/4-bit output tables) or 3 types (8-bit input/128-bit output tables, 8-bit input/4-bit output tables, 8-bit input/32-bit output tables) of tables according to the implementation method.

4.2 Application to ARIA-128

We divided each round of ARIA as Section 3.5. After this, we suggest a white-box ARIA implementation dividing each round of ARIA into two blocks. Let U_i be the i-th round function of white-box ARIA implementation(i = 1, 2, ⋯ , 12). Then we can make a white-box ARIA implementation as follows:

$$U_i = (Q_i \circ R_i) \circ (R_i^{-1} \circ DL \circ SL \circ KA \circ P_i), \quad \text{where } i = 1, 2, \cdots, 11 \qquad (3)$$

where P_i, Q_i(i = 1, 2, ⋯ , 11) are composed of 16 matrices in $GL(8, Z_2)$ and R_i(i = 1, 2, ⋯ , 11) is a matrix in $GL(128, Z_2)$. The 12-th round function U_{12} of white-box ARIA implementation is given as follows:

$$U_{12} = (Q_{12} \circ R_{12}) \circ (R_{12}^{-1} \circ KA \circ SL \circ KA \circ P_{12}) \qquad (4)$$

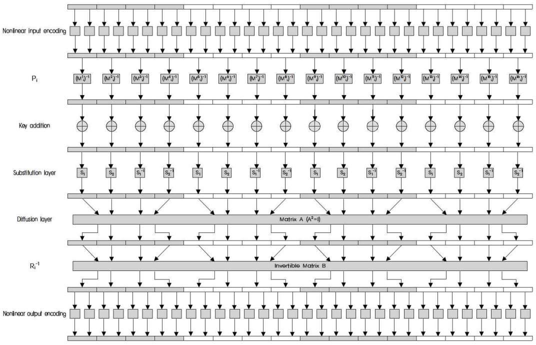
where P_{13}, Q_{12} are composed of 16 matrices in $GL(8, Z_2)$ and R_{12} is a matrix in $GL(128, Z_2)$.

In Equation (3) and Equation (4), Q_i, P_{i+1} have the relation $P_{i+1} = Q_i^{-1}$ for i = 1, 2, ⋯ , 11. We add nonlinear input encodings in the previous position of P_i, R_i and nonlinear output encodings in the latter position of R_{i-1}, Q_i to improve the security of white-box ARIA implementation. Both nonlinear input encodings and nonlinear output encodings are composed of 32 4-bit input/4-bit output nonlinear functions, respectively. The i-th round functions $R_{i-1} \circ DL \circ SL \circ KA \circ P_i$($i$ = 1, 2, ⋯ , 11) and $Q_i \circ R_i$(i = 1, 2, ⋯ , 11) of white-box ARIA implementation including nonlinear encodings are shown in Figure 3 and Figure 4, respectively. Input encodings $R_{i-1} \circ DL \circ SL \circ KA \circ P_i$($i$ = 1, 2, ⋯ , 11), $Q_i \circ R_i$(i = 1, 2, ⋯ , 12) and $R_{12}^{-1} \circ KA \circ SL \circ KA \circ P_{12}$ are composed of 8-bit input/128-bit output tables. There exist 8-bit input/4-bit output tables (XOR tables) between them.

4.3 A variant for efficiency improvement

We can change into 8-bit input/32-bit output tables instead of 8-bit input/128-bit output tables in $Q_i \circ R_i$ for i = 1, 2, ⋯ , 11. Since we need fewer numbers of XOR tables in this case, we can reduce the storage of the system.

<Fig. 3> $R_i^{-1} \circ DL \circ SL \circ KA \circ P_i$ (i = 1, 2, ⋯ , 11) of white-box ARIA implementation


Nonlinear input encoding
P_i
$(M^1_i)^{-1}$ $(M^2_i)^{-1}$ $(M^3_i)^{-1}$ $(M^4_i)^{-1}$ $(M^5_i)^{-1}$ $(M^6_i)^{-1}$ $(M^7_i)^{-1}$ $(M^8_i)^{-1}$ $(M^9_i)^{-1}$ $(M^{10}_i)^{-1}$ $(M^{11}_i)^{-1}$ $(M^{12}_i)^{-1}$ $(M^{13}_i)^{-1}$ $(M^{14}_i)^{-1}$ $(M^{15}_i)^{-1}$ $(M^{16}_i)^{-1}$
Key addition
Substitution layer
S_1 S_2 S_1^{-1} S_2^{-1} S_1 S_2 S_1^{-1} S_2^{-1} S_1 S_2 S_1^{-1} S_2^{-1} S_1 S_2 S_1^{-1} S_2^{-1}
Diffusion layer
Matrix A (A^2=I)
R_i^{-1}
Invertible Matrix B
Nonlinear output encoding

5. Analysis

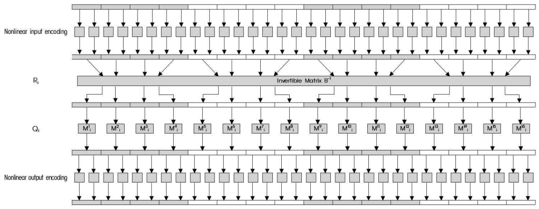
5.1 Size and performance

We need many tables and sizes for the white-box ARIA implementation in Section 4.2 as Table 2. Since we need different XOR tables for each calculation, there are many XOR tables to do this implementation.

<Table 2> The number and size of the white-box ARIA implementation

Processes	Number of tables	Size (KB)
Input encoding	16	64
$R_i^{-1} \circ DL \circ SL \circ KA \circ P_i$ (i = 1, 2, ⋯ , 11)	176	704
$Q_i \circ R_i$ (i = 1, 2, ⋯ , 11)	176	704
$R_{12}^{-1} \circ KA \circ SL \circ KA \circ P_{12}$	16	64
$Q_{12} \circ R_{12}$	16	64
XOR	11,520	1,440
Total	11,920	3,040

<Fig. 4> $Q_i \circ R_i$ (i = 1, 2, ⋯ , 11) of white-box ARIA implementation



The result of the variant of the white-box ARIA implementation in Section 4.3 is given in Table 3. This white-box ARIA implementation is 2.4 times larger than the original ARIA implementation. Since there are many XOR tables to do this white-box ARIA implementation, it needs 39.9 times tables more than the original ARIA implementation.

<Table 3> The number and size of the variant of the white-box ARIA implementation

Processes	Number of tables	Size (KB)
Input encoding	16	64
$R_i^{-1} \circ DL \circ SL \circ KA \circ P_i$ (i = 1, 2, ⋯ , 11)	176	704
$Q_i \circ R_i$ (i = 1, 2, ⋯ , 11)	176	176
$R_{12}^{-1} \circ KA \circ SL \circ KA \circ P_{12}$	16	64
$Q_{12} \circ R_{12}$	16	64
XOR	7,296	912
Total	11,920	1,984

5.2 Security

5.2.1 Strength against known attacks

ARIA consists of two s-boxes and a diffusion layer using a 16×16 matrix. Billet et al.'s attack is applied to an implementation using one s-box and a 4×4 matrix for each round. Our implementation is secure against the original Billet et al.'s attack.

5.2.2 White-box diversity and ambiguity

There are two types of measures for white-box cryptography security. These are white-box diversity and white-box ambiguity, respectively. White-box diversity is the total number of existing implementations. White-box ambiguity is the number of the same implementations for a given implementation. There are four types of tables in our implementation. They are input encoding tables, i-th round function $R_{i-1} \circ DL \circ SL \circ KA \circ P_i$($i$ = 1, 2, ⋯ , 11) tables, i-th round function $Q_i \circ R_i$(i = 1, 2, ⋯ , 11) tables, and XOR tables.

White-box Diversity

White-box diversity measures the number of distinct implementations for a given type. We have white-box diversity as follows.

- Input decoding tables
  : $(16!)^2 \times 20160^{64} \times (16!)^{32} \approx 2^{2419.7}$
- $R_{i-1} \circ DL \circ SL \circ KA \circ P_i$($i$ = 1, 2, ⋯ , 11) tables : $(16!)^2 \times 256 \times 2^{62.2} \times 2^{256} \times (16!)^8 \approx 2^{768.1}$
- $Q_i \circ R_i$(i = 1, 2, ⋯ , 11) tables

: $(16!)^2 \times 2^{256} \times (16!)^8 \approx 2^{698.5}$

- XOR tables : $(16!)^2 \times 16! \approx 2^{132.8}$

White-box Ambiguity

White-box ambiguity measures the number of alternative implementations for a given table. We have white-box diversity as follows.

- Input decoding tables : $(16!)^2 \times 20160^{32} \approx 2^{546.1}$
- $R_{i-1} \circ DL \circ SL \circ KA \circ P_i (i = 1, 2, \cdots, 11)$ tables : $(16!)^2 \times 15! \approx 2^{128.8}$
- $Q_i \circ R_i (i = 1, 2, \cdots, 11)$ tables : $(16!)^2 \times 20160^2 \approx 2^{117.1}$
- XOR tables : $16! \times 16 \approx 2^{48.3}$

6. Conclusion

The ARIA is a block cipher designed by Korean researchers and widely utilized. It has been designated as a standard cryptographic technique by the Korean Agency for Technology and Standards. In this paper, we presented the first white-box implementation for the ARIA block cipher utilizing a form of obfuscation. This technique is similar to the white-box AES implementation and involves the use of tables. While this implementation necessitates more tables than Chow et al.'s AES implementation, it assures ample security concerning white-box diversity and white-box ambiguity. White-box cryptography is commonly employed in digital rights management to prevent the unauthorized distribution of data such as music and videos. It functions by encrypting the data and furnishing various information types accessible solely to authorized devices. Our white-box ARIA implementation ensures that the data remains secure and only legitimate users can access it. Nevertheless, there is still room for more efficient white-box ARIA implementations and adequate attacks against this white-box ARIA implementation.

References

[1] J. Bringer, H. Chabanne and E. Dottax, "White box cryptography: Another attempt", Cryptology ePrint Archive, Report 2006/468, 2006, http://eprint.iacr.org/.

[2] A. Biryukov, C. De Canni`ere, A. Braeken and B. Preneel, "A Toolbox for Cryptanalysis: Linear and Affine Equivalence Algorithms", EUROCRYPT 2003, LNCS Vol. 2656, pp. 33-50. Springer, Heidelberg, 2003.

[3] O. Billet, H. Gilbert and C. Ech-Chatbi, "Cryptanalysis on a white box AES implementation", SAC 2004, LNCS Vol. 3357, pp. 227-240. Springer, Heidelberg, 2004.

[4] S. Chow, P. Eisen, H. Johnson and P. C. van Oorschot, "White-box cryptography and an AES implementation", SAC 2002, LNCS Vol. 2595, pp. 250-270. Springer, Heidelberg, 2003.

[5] S. Chow, P. Eisen, H. Johnson and P. C. van Oorschot, "A White-Box DES Implementation for DRM Applications", DRM 2002, LNCS Vol. 2696, pp. 1-15. Springer, Heidelberg, 2003.

[6] L. Goubin, J. M. Masereel and M. Quisquater, "Cryptanalysis on white box DES implementations", SAC 2007, LNCS Vol. 4876, pp. 278-295. Springer, Heidelberg, 2007.

[7] M. Karroumi, "Protecting White-Box AES with Dual Ciphers", ICISC 2010, LNCS Vol. 6829, pp. 278-291, Springer, Heidelberg, 2011.

[8] D. Kwon, J. Kim, S. Park, S. H. Sung, Y. Sohn, J. H. Song, Y. Yeom, E. Yoon, S. Lee, J. Lee, S. Chee, D. Han and J. Hong, "New Block Cipher: ARIA", ICISC 2003, LNCS 2971, pp.432-445, Springer, Heidelberg, 2004.

[9] H. E. Link and W. D. Neumann, "Clarifying obfuscation: Improving the security of whitebox DES", International Conference on Information Technology: Coding and Computing, Vol. I, pp. 679-684, IEEE Computer Society Press, Washington, DC, USA, 2005.

[10] T. Lepoint, M. Rivain, Y. D. Mulder, P. Roelse and B. Preneel, "Two Attacks on a White-Box

AES Implementation", SAC 2013, LNCS Vol. 8282, pp. 265-185. Springer, Heidelberg, 2013.

[11] W. Michiels, P. Gorissen and H. D. L. Hollmann, "Cryptanalysis on a Generic Class of White-Box Implementations", SAC 2008, LNCS Vol. 5381, pp. 414-428. Springer, Heidelberg, 2009.

[12] Y. D. Mulder, B. Wyseur and B. Preneel, "Cryptanalysis on a Perturbated White-box AES Implementation", INDOCRYPT 2010, LNCS Vol. 6498, pp. 292-310. Springer, Heidelberg, 2010.

[13] C. E. Shannon, "Communication Theory of Secrecy Systems", Bell System Technical Journal, Vol. 28, No. 4, pp. 656-715, 1949.

[14] B. Wyseur, W. Michiels, P. Gorissen and B. Preneel, "Cryptanalysis on white-box DES implementations with arbitrary external encodings", SAC 2007, LNCS Vol. 4876, pp. 264-277. Springer, Heidelberg, 2007.

[15] H. T. Kim, "Attacks of Modified White-box AES Implementations", Journal of Social Convergence Studies, Vol. 5, No. 2, pp. 1-13, 2021.

[16] H. T. Kim, "On Conditions to Satisfy White-Box Cryptography", Journal of Security Engineering, Vol. 11, No. 2, pp. 155-164, 2014.

〔저자소개〕

김 홍 태 (Hong Tae Kim)
2003년 2월 서울대 수리과학부 학사
2006년 2월 서울대 수리과학부 석사
2013년 2월 서울대 수리과학부 박사
2006년 2월 ~ 현재 공군사관학교
수학과 수학교수
email : yeskafa@gmail.com