ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 20-02-14

j.inst.Korean.electr.electron.eng.Vol.24,No.2,468~474,june 2020
http://dx.doi.org/10.7471/ikeee.2020.24.2.468

8가지 블록/키 크기를 지원하는 SPECK 암호 코어

A SPECK Crypto-Core Supporting Eight Block/Key Sizes

양 현 준*, 신 경 욱*★

Hyeon-Jun Yang*, Kyung-Wook Shin*★

Abstract

This paper describes the hardware implementation of SPECK, a lightweight block cipher algorithm developed for the security of applications with limited resources such as IoT and wireless sensor networks. The block cipher SPECK crypto-core supports 8 block/key sizes, and the internal data-path was designed with 16-bit for small gate counts. The final round key to be used for decryption is pre-generated through the key initialization process and stored with the initial key, enabling the encryption/decryption for consecutive blocks. It was also designed to process round operations and key scheduling independently to increase throughput. The hardware operation of the SPECK crypto-core was validated through FPGA verification, and it was implemented with 1,503 slices on the Virtex-5 FPGA device, and the maximum operating frequency was estimated to be 98 MHz. When it was synthesized with a 180 nm process, the maximum operating frequency was estimated to be 163 MHz, and the estimated throughput was in the range of 154 ~ 238 Mbps depending on the block/key sizes.

요 약

IoT, 무선 센서 네트워크와 같이 제한된 자원을 갖는 응용분야의 보안에 적합하도록 개발된 경량 블록 암호 알고리듬 SPECK의 하드웨어 구현에 관해 기술한다. 블록 암호 SPECK 크립토 코어는 8가지의 블록/키 크기를 지원하며, 회로 경량화를 위해 내부 데이터 패스는 16-비트로 설계되었다. 키 초기화 과정을 통해 복호화에 사용될 최종 라운드 키가 미리 생성되어 초기 키와 함께 저장되며, 이를 통해 연속 블록에 대한 암호화/복호화 처리가 가능하도록 하였다. 또한 처리율을 높이기 위해 라운드 연산과 키 스케줄링이 독립적으로 연산되도록 설계하였다. 설계된 SPECK 크립토 코어를 FPGA 검증을 통해 하드웨어 동작을 확인하였으며, Virtex-5 FPGA 디바이스에서 1,503 슬라이스로 구현되었고, 최대 동작 주파수는 98 MHz로 추정되었다. 180 nm 공정으로 합성하는 경우, 최대 동작 주파수는 163 MHz로 추정되었으며, 블록/키 크기에 따라 154 Mbps ~ 238 Mbps의 처리량을 갖는다.

Key words : SPECK, Block Cipher Algorithm, Symmetric Key Algorithm, Feistel, Information Security

* School of Electronic Engineering, Kumoh National Institute of Technology
★ Corresponding author
Email : kwshin@kumoh.ac.kr, Tel : +82-54-478-7427
※ Acknowledgment
- This research was supported by the KIAT (Korea Institute for Advancement of Technology) grant funded by the Korea Governmen (MOTIE : Ministry of Trade Industry and Energy). (No. N0001883, HRD Program for Intelligent semiconductor Industry)
- Authors are thankful to IDEC for supporting EDA software.

Manuscript received May. 29, 2020; revised Jun. 14, 2020; accepted Jun. 17, 2020.

Ⅰ. 서론

사물 인터넷(Internet of Things)과 무선 센서 네트워크 기술의 응용분야가 확대됨에 따라 이들 시스템을 구성하는 종단 장치의 종류와 수가 기하급수적으로 증가하고 있다. 이에 따라 종단 장치와 시스템 간 네트워크를 통해 수집되고 유통되는 정보에 대한 보안의 필요성과 중요성이 큰 이슈로 부각되고 있다. 정보보안 시스템은 대칭키(symmetric-key) 암호, 공개키(public-key) 암호, 해시(hash) 함수, 무작위 난수 생성 등을 기반으로 정보에 대한 기밀성, 무결성 검증, 사용자 및 장치 인증, 전자서명 생성 및 검증, 키교환 등 다양한 보안 프로토콜들이 구현된다[1, 2].

대칭키 암호는 암호화와 복호화에 동일한 키가 사용되는 방식이며, 스트림 암호와 블록 암호가 있다. 블록 암호는 입력 평문/암호문을 고정된 크기의 블록으로 분할하여 암호화/복호화를 하며, Feistel 구조와 SPN(Substitution Permutation Network) 구조의 두 가지 방식으로 구분된다. 오늘날 널리 사용되고 있는 대표적인 블록 암호 알고리듬으로는 미국 표준인 AES[3]와 우리나라 표준인 ARIA[4] 등이 있다.

IoT, 무선 센서 네트워크 종단 장치와 같이 제한된 자원을 갖는 응용분야에서는 AES, ARIA와 같은 범용 블록 암호가 적합하지 않으며, 적은 게이트로 구현이 가능한 경량(lightweight) 블록 암호 알고리듬들이 제안되어 사용이 확대되고 있다[5, 6]. 대표적인 경량 블록 암호 알고리듬으로는 TWINE [7], HIGHT[8], PRESENT[9], Piccolo[10], SPECK [11, 12] 등이 있다. 경량 블록 암호는 범용 블록 암호에 비해 보안 성능이 다소 떨어져 하드웨어 복잡도와 보안 성능 사이의 균형이 중요하며, 응용분야의 요구 사양을 고려하여 알고리듬, 블록 및 키 길이 등을 선택해야 한다.

본 논문에서는 경량 블록 암호 SPECK의 8가지 블록/키 길이를 지원하도록 하드웨어를 설계하고, FPGA 구현을 통해 하드웨어 동작을 검증하였다. II장에서는 SPECK 알고리듬에 관해 간략히 소개하고, III장에서는 하드웨어 설계를 기술한다. IV장은 설계된 SPECK 크립토 코어의 FPGA 검증 결과와 성능 분석 결과를 기술하고, V장에서 결론을 맺는다.

Ⅱ. 블록 암호 SPECK[11, 12]

SPECK은 미국 국가안보국(National Security Agency) 연구소에서 개발한 경량 블록암호 알고리듬으로 Feistel 구조의 블록 암호이다. Feistel 구조란 평문/암호문 블록을 상위 워드와 하위 워드로 나누어 연산을 적용한 후, 상위 워드와 하위 워드의 위치를 서로 교환하는 방식으로 라운드 연산이 수행되는 블록 암호 방식이다. 매 라운드 연산에는 라운드 키가 사용되며, 암호화/복호화 연산의 라운드 함수는 키 값이 달라지는 것을 제외하면 동일한 연산이 반복되는 구조이다.

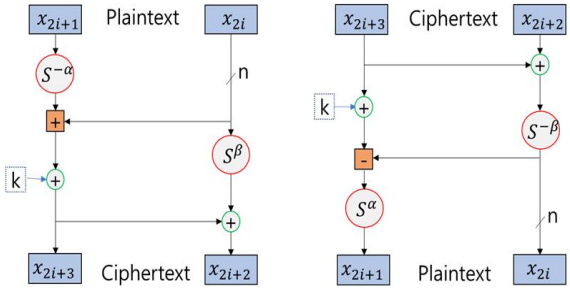
SPECK의 암호화와 복호화 라운드 함수는 그림 1과 같다. SPECK 2n의 암호와 과정은 n-비트에서 3가지 연산을 사용하며, 여기서 n은 워드 크기이다. 기호 "⊕"는 비트 XOR를 나타내며, 기호 "⊞"는 모듈러 덧셈 (modular addition)을 의미한다. S^j와 S^{-j}는 각각 j-비트만큼 왼쪽과 오른쪽 순환이동(cyclic shift)을 나타낸다. 키 값 k는 $k \in GF(2)^n$이며, SPECK 2n의 라운드 함수 R_k: $GF(2)^n \times GF(2)^n$는 식 (1)과 같이 표현되고, 블록크기가 32-비트인 경우의 순환이동 크기는 $\alpha = 7$, $\beta = 2$이며, 그 외의 블록크기에 대해서는 $\alpha = 8$, $\beta = 3$이다.

$$R_k(x,y) = ((S^{-\alpha}x + y) \oplus k, S^{\beta}y \oplus (S^{-\alpha}x + y) \oplus k) \quad (1)$$

복호화는 암호화의 역 (inverse) 연산이 역순으로 수행되며, 식 (2)와 같이 표현된다. 기호 "⊟"는 모듈러 뺄셈(modular subtraction)을 의미한다.

$$R_k^{-1}(x,y) = (S^{\alpha}((x \oplus k) - S^{-\beta}(x \oplus y)), S^{-\beta}(x \oplus y)) \quad (2)$$

라운드 함수에 사용되는 키 값은 키 스케줄 연산



(a) Encryption (b) Decryption

Fig. 1. Round function of SPECK block cipher.
그림 1. SPECK 블록 암호의 라운드 함수

Table 1. Algorithm parameters of SPECK block cipher.
표 1. 블록 암호 SPECK의 알고리듬 파라미터

block size $2n$	key size mn	word size n	key words m	rot α	rot β	# of rounds T
32	64	16	4	7	2	22
48	72	24	3	8	3	22
	96		4			23
64	96	32	3			26
	128		4			27
96	96	48	2			28
	144		3			29
128	128	64	2			32
	192		3			33
	256		4			34

에 의해 만들어진다. 키 스케줄 연산은 라운드 함수와 동일하며 키 값 대신 라운드 계수를 사용하는 것만 다르다. $K=(l_{m-2},...,l_0,k_0)$으로 표현되며, m은 키 크기에 따라 {2, 3, 4} 중 하나로 결정되고, $l_i,k_0 \in GF(2)^n$이다. 블록 암호 SPECK의 알고리듬 파라미터 값은 표 1과 같으며, 블록/키 쌍을 하나의 동작모드로 보았을 때 총 10가지 모드가 있다. 본 논문에서는 블록크기 48-비트를 제외한 총 8가지의 블록/키 크기의 암호, 복호를 지원하도록 설계하였다.

III. SPECK 크립토 코어 설계

본 논문에서 구현한 SPECK 크립토 코어는 그림 2와 같이 라운드 연산을 수행하는 Rnd 블록, 키 스케줄 연산을 수행하는 Keysch 블록, 그리고 제어 블록인 Cntl로 구성된다.

Keysch는 키 스케줄 연산을 수행하는 블록으로 외부에서 입력되는 키 값으로부터 매 라운드에 사

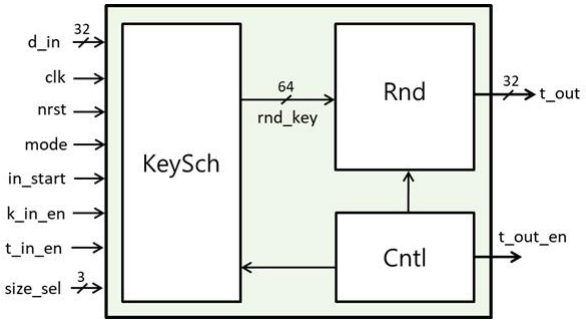


Fig. 2. Block diagram of SPECK crypto-core.
그림 2. SPECK 크립토 코어의 블록도

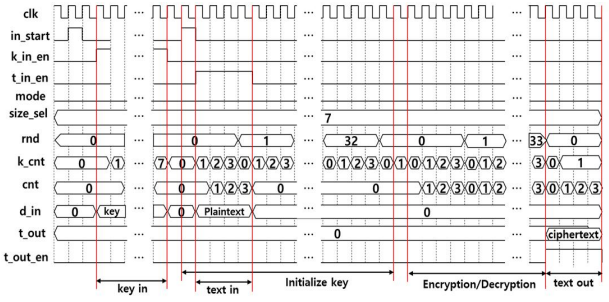


Fig. 3. Timing diagram of SPECK crypto-core.
그림 3. SPECK 크립토 코어의 동작 타이밍도

용되는 라운드 키 값을 생성한다. Rnd 블록은 라운드 연산을 수행하는 블록으로 평문 (또는 암호문) 그리고 라운드 키 값을 받아 암호화 (또는 복호화) 라운드 연산을 수행한다. Cntl는 회로 전체의 동작을 제어하는 블록으로 FSM, 제어신호 발생기, 계수기 등으로 구성된다.

그림 3은 블록크기 128-비트인 경우의 동작 타이밍도이다. 키 값이 입력되면, 키 초기화가 진행된 후에 라운드 연산이 진행된다. 키 초기화가 끝나면, 34회의 라운드 연산이 136 클록 동안 수행된다. 라운드 블록과 키 스케줄 블록이 16-비트 데이터 패스로 설계되어 한 라운드에 4 클록 주기가 소요된다. 라운드 연산이 완료되면 32-비트의 출력포트를 통해 4 클록 주기 동안 암호/복호 결과가 출력된다.

Rnd 블록은 그림 4와 같이 구성되며, 64-비트 레지스터 두 개와 8-비트, 3-비트, 1-비트 레지스터를 각각 하나씩 갖는다. Rnd 블록의 회로 경량화를 위해 내부 연산회로를 16-비트 데이터 패스로 구현하였으며, 16-비트 캐리선택 가산기 두개, 16-비

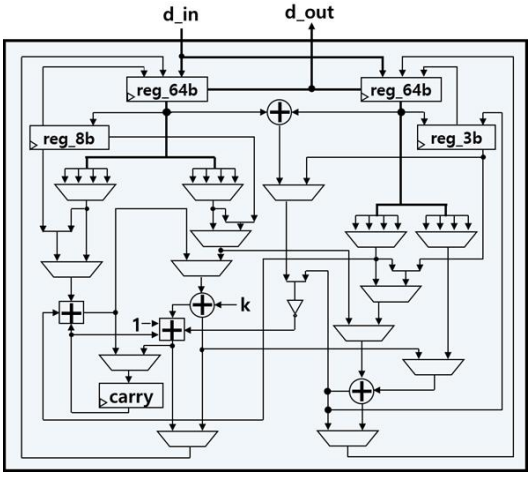


Fig. 4. Internal block diagram of Rnd block.
그림 4. Rnd 블록의 내부 구성도

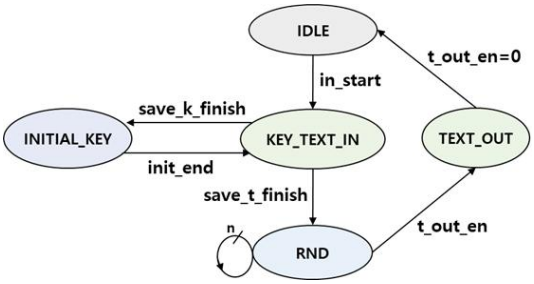


Fig. 5. FSM of control block.
그림 5. 제어블록의 유한상태머신

트 XOR 게이트 두 개 그리고 3-비트 XOR 게이트로 구성하였다. Keysch 블록은 Rnd 블록과 구조가 동일하며, Rnd 블록에서 사용되는 키 값 대신 라운드 상수 값이 사용되는 점과 256-비트 레지스터 세 개가 사용되는 점만 다르다.

키 스케줄과 라운드 연산의 동작은 그림 5의 유한상태머신에 의해 제어된다. in_start 신호를 통해 IDLE에서 KEY_TEXT_IN 상태로 천이되며, size_sel 입력에 의해 블록/키 크기가 결정되고 키 값이 저장된다. 일단, 초기키 값이 저장되면 INITIAL_KEY 상태로 천이되어 키 초기화 과정이 진행된다. 키 초기화는 복호과정에 사용될 마지막 라운드 키 값을 생성하는 과정이며, 키 초기화 과정 중에도 평문/복호문이 입력되어 내부 레지스터에 저장될 수 있다. 초기화 과정이 끝나면, 첫 번째 라운드 키 값과 마지막 라운드 키 값 모두를 가진 상태가 되며, init_end 신호에 의해 KEY_TEXT_IN 상태로 천이된다. 저장된 데이터가 없다면 계속 대기 상태에 머물고, 저장된 데이터가 있다면, save_t_finish의 신호에 의해 RND 상태로 천이하여 mode 신호에 따라 암호 또는 복호 연산이 진행된다. 마지막 라운드 연산이 끝나면 t_out_en 신호가 출력되면서 TEXT_OUT 상태로 넘어가 암호문/복호문이 출력되며, 출력이 완료되면 다시 IDLE 상태로 돌아간다.

Ⅳ. FPGA 구현 및 성능평가

1. RTL 기능 검증

설계된 SPECK 크립토 코어를 RTL 시뮬레이션으로 기능 검증을 하였다. 그림 6은 블록/키 크기가 128-비트/256-비트인 경우의 시뮬레이션 결과 중 일부이다. 키 값은 '1f1e_1d1c_1b1a_1918_1716_1514_1312_1110_0f0e_0d0c_0b0a_0908_0706_0504_0302_0100'이고, 첫 번째 평문은 문헌 [11]의 테스트 벡터 값 '6573_6f68_7420_6e49_202e_7265_6e6f_6f70'을, 두 번째 평문은 '0123_4567_8910_1112_1415_1617_1819_2021'을 사용하였다. 시뮬레이션 결과로 암호문 '4109_0104_05c0_f53e_4eee_b48d_9c18_8f43'와 '0c5d_3928_6250_2dc1_f707_9cc7_a6d9_c5e6'가 출력되었으며, 두 암호문에 대한 복호 연산 결과로 평문 '6573_6f68_7420_6e49_202e_7265_6e6f_6f70'와 '0123_4567_8910_1112_1415_1617_1819_2021'이 출력되어 암호화와 복호화 연산이 정상 동작함을 확인하였다.

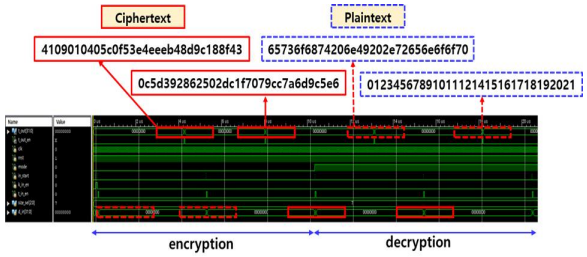
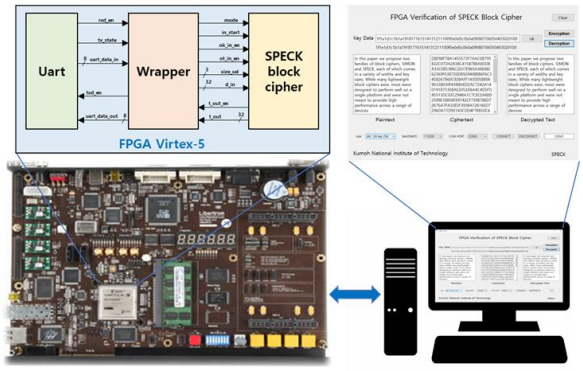
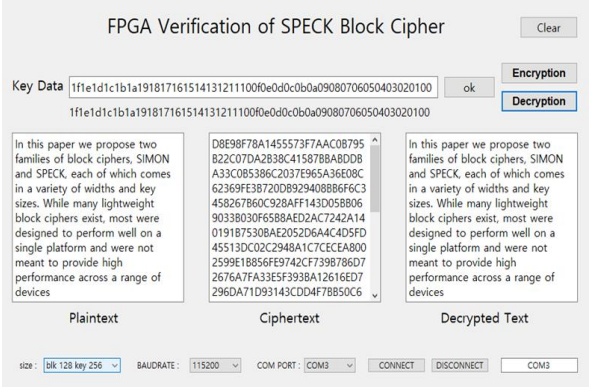


Fig. 6. RTL simulation results of SPECK crypto-core.
그림 6. SPECK 크립토 코어의 RTL 시뮬레이션 결과

2. FPGA 검증



(a) FPGA verification platform



(b) Screenshot of FPGA verification results

Fig. 7. FPGA verification of SPECK crypto-core.
그림 7. SPECK 크립토 코어의 FPGA 검증

Table 2. Required clock cycles depending on block-key sizes.
표 2. 블록/키 크기에 따른 소요 클록 사이클 수

size (blk=32m)	# of Round	Key Initialization [cycles]	Encryption/ Decryption [cycles]	Throughput @163 MHz [Mbps]
b32/k64 (m=1)	22	21	22	238
b64/k96 (m=2)	26	50	52	201
b64/k128 (m=2)	27	52	54	194
b96/k96 (m=3)	28	81	84	187
b96/k144 (m=3)	29	84	87	180
b128/k128 (m=4)	32	124	128	163
b128/k192 (m=4)	33	128	132	158
b128/k256 (m=4)	34	132	136	154

SPECK 크립토 코어는 그림 7-(a)의 FPGA 검증 플랫폼을 사용하여 하드웨어 동작을 검증하였다. 검증 플랫폼은 FPGA 보드, PC 그리고 구동 소프트웨어로 구성되며, FPGA 디바이스에 구현된 SPECK 크립토 코어의 동작결과는 PC의 GUI 화면을 통해 표시된다. Xilinx Virtex-5 FPGA 디바이스가 사용되었으며, SPECK 크립토 코어와 통신을 위한 Uart 그리고 Wrapper 회로가 FPGA에 구현되었다. FPGA의 동작 주파수는 50 MHz로 설정하였고, 문헌 [11]에 제시된 테스트 벡터와 임의의 값을 사용하여 검증하였다. FPGA 검증 결과는 그림 7-(b)와 같으며, 설계된 SPECK 크립토 코어가 정상 동작함을 확인하였다.

3. 성능 평가

블록/키 크기에 따라 암호화/복호화에 소요되는 클록 수는 표 2와 같다. 키 초기화와 암호화/복호화에 소요되는 클록 수는 각각 $(m-1)\times rnd$, $m\times rnd$로 표현되며, rnd는 라운드 연산 횟수, m은 블록 사이즈를 32-비트로 나눈 몫이다.

표 3은 본 논문에서 설계된 SPECK 크립토 코어를 문헌 [11]의 사례와 비교한 것이다. 문헌 [11]에는 단일 블록/키 크기의 암호 연산만 지원하는 설계 결과가 제시되어 있으며, 표 3에는 블록크기 128-비트, 키 길이 256-비트의 경우를 참조하였다. 본 논문의 SPECK 크립토 코어는 32-비트, 64-비트, 96-비트, 128-비트의 4가지 블록크기와 64-비트~256-비트 범위의 6가지 키 길이의 조합으로 총 8가지의 블록/키 크기를 지원하며, 또한 암호와 복호를 모두 지원한다. 문헌 [11]의 사례와 비교하여, 본 논문의 크립토 코어는 처리 성능 (throughput)이 우수하나, 등가 게이트 수가 약 4.9배 크다. 그 이유로는 암호화뿐만 아니라 복호화도 지원하기 위해 키 초기화 결과를 저장하는 256-비트 레지스터가 필요하고, 연속 블록의 암호화/복호화를 지원하기 위해 초기 키 값을 저장하는 256-비트 레지스터가 필요하며, 또한 총 8가지의 블록/키 크기를 지원하기 위한 부가적인 하드웨어가 필요하기 때문이다. 비교 대상인 문헌 [11]의 사례는 블록크기 128-비트, 키 길이 256-비트의 암호화 기능만 가지며, 또한 매 평문 블록마다 키 값을 입력해야 하는 단점을 갖는다. 본 논문의 SPECK 크립토 코어는 요구되는 보안성능에 따라 블록크기와 키 길이의 선택이 가능하고, 연속 블록의 암호화/복호화가 가능하여 유용성이 우수하고 다양한 분야의 보안에 사용될 수 있다는 장점을 갖는다.

Table 3. Comparison of SPECK cores.
표 3. SPECK 코어의 비교

	[11]	This paper
Technology	130 nm	180 nm
Block size [bit]	128	32, 64, 96, 128
Key size [bit]	256	64, 96, 128, 144, 192, 256
Operation mode	Encryption only	Encryption & Decryption
Throughput* [kbps] (@100 kHz)	88.9	94.1
Area* [GE] (@100 kHz)	2,872	14,098
Maximum Frequecy	-	163 MHz

* block size: 128-bit

V. 결론

8가지 블록/키 크기의 암호화/복호화를 지원하는 SPECK 크립토 코어를 16-비트 데이터 패스로 구현하고, FPGA 구현을 통해 하드웨어 동작을 검증하였다. 0.18-μm CMOS 셀 라이브러리로 합성한 결과, 14,098 등가 게이트로 구현되었으며, 최대 동

작 주파수는 약 163 MHz로 예측되었다. Virtex-5 FPGA 디바이스로 합성한 결과, 1,503 슬라이스가 사용되었으며, 최대 동작 주파수는 98 MHz로 예측되었다.

경량 블록 암호는 제한된 자원을 갖는 IoT 디바이스나 무선 센서 네트워크의 종단 보안에 적합하도록 적은 수의 게이트로 구현이 가능하다. 그러나 IoT 디바이스나 센서 네트워크에서 수집된 정보가 모이는 상위 시스템에서는 다양한 블록/키 크기의 암호화/복호화 기능과 고성능이 요구된다. 본 논문에서 설계된 SPECK 크립토 코어는 8가지 블록/키 크기의 암호화/복호화를 연속블록으로 처리할 수 있어 요구되는 보안성능에 따라 다양한 분야에 응용될 수 있을 것이다.

References

[1] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol.38, pp.8-27, 2018.
DOI: 10.1016/j.jisa.2017.11.002

[2] Q. Jing, A.V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the Internet of Things: perspectives and challenges," *Wireless Netw* 20, pp.2481-2501, 2014. DOI: 10.1007/s11276-014-0761-7

[3] NIST Std. FIPS-197, Advanced Encryption Standard, National Institute of Standard and Technology (NIST), 2001.

[4] KS X 1213, 128 bit Block Encryption Algorithm ARIA, Korean Agency for Technology and Standards, 2004.

[5] B. J. Mohd and T. Hayajneh, "Lightweight Block Ciphers for IoT: Energy Optimization and Survivability Techniques," *IEEE Access*, vol.6, pp.35966-35978, 2018.
DOI: 10.1109/ACCESS.2018.2848586

[6] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A Survey of Lightweight-Cryptography Implementations," *IEEE Design & Test of Computers*, vol.24, no.6, pp.522-533, 2007. DOI: 10.1109/MDT.2007.178

[7] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "Twine: A lightweight, versatile block cipher," *ECRYPT Workshop on Lightweight Cryptography*, pp.146-169, 2011.

[8] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee., "HIGHT: A new block cipher suitable for low-resource device," *Cryptographic Hardware and Embedded Systems-CHES 2006*, vol.4249 of LNCS, pp.46-59, 2006.
DOI: 10.1007/978-3-540-74735-2_31

[9] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," *Cryptographic Hardware and Embedded Systems-CHES 2007*, vol. 4727 of LNCS, pp.450-466, 2007. DOI: 10.1007/978-3-540-74735-2_31

[10] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: an ultra-lightweight block cipher," *Cryptographic Hardware and Embedded Systems-CHES 2011*, vol, 6917 of LNCS, pp 342-357, 2011.
DOI: 10.1007/978-3-642-23951-9_23

[11] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers," *Cryptology ePrint Archive, Report 2013/404*, 2013.
DOI: https://eprint.iacr.org/2013/404

[12] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," *2015 52nd ACM/ EDAC/IEEE Design Automation Conference (DAC)*, pp.1-6, 2015.

BIOGRAPHY

Hyeon-Jun Yang (Member)

2020 : BS degree in Electronic Engineering, Kumoh National Institute of Technology.
2020~ : Graduate student, Kumoh National Institute of Technology

Kyung-Wook Shin (Member)

1984 : BS degree in Electronic Engineering, Korea Aerospace University
1986 : MS degree in Electronic Engineering, Yonsei University
1990 : Ph.D. degree in Electronic Engineering, Yonsei University
1990~1991 : Senior Researcher, Semiconductor Research Center, Electronics and Telecommunications Research Institute (ETRI)
1991~ : Professor in School of Electronic Engineering, Kumoh National Institute of Technology
1995~1996 : University of Illinois at Urbana- Champaign (Visiting Professor)
2003~2004 : University of California at San Diego (Visiting Professor)
2013~2014 : Georgia Institute of Technology(Visiting Professor)