JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 25, No. 4: 588~594, Apr. 2021

블록암호 SIMON의 카운터 모드 사전 연산 고속 구현

권혁동[1] · 장경배[1] · 김현지[2] · 서화정[3*]

The fast implementation of block cipher SIMON using pre-computation with counter mode of operation

Hyeok-Dong Kwon[1] · Kyung-Bae Jang[1] · Hyun-Ji Kim[2] · Hwa-Jeong Seo[3*]

[1]Graduate Student, Department of Information Computer Engineering, Hansung University, Seoul, 02876 Korea
[2]Graduate Student, Department of IT Convergence Engineering, Hansung University, Seoul, 02876 Korea
[3*]Associate Professor, Department of IT Convergence Engineering, Hansung University, Seoul, 02876 Korea

요 약

미국 국가안보국에서 개발된 경량 블록암호 SIMON은 하드웨어 구현에 최적화 된 블록암호 군으로서, 여러 환경에서 효율적으로 동작할 수 있도록 많은 입·출력 규격을 제공한다. 블록암호 카운터 운용모드는 블록암호의 입력 규격보다 더 큰 평문을 암호화할 수 있도록 제공되는 운용모드 중 하나이다. 카운터 운용모드는 입력 값으로 상수 값인 논스와 블록의 번호인 카운터를 사용한다. 이때 논스 부분은 모든 블록이 동일하기 때문에, 다른 상수 값과 연산한다면 항상 동일한 연산 결과를 가진다. 이 특징을 활용한다면 일부 값을 사전 연산하여 라운드 함수의 일부분을 생략하는 것이 가능하다. 일반적인 상황에서 SIMON의 입력 값은 카운터에 영향을 받으나, 8-bit 환경에서는 8-bit 단위로 연산이 되기에 고속 구현이 가능한 부분이 존재한다. 따라서 본 논문에서는 연산 생략이 가능한 지점을 중점적으로 확인하고 기존 SIMON 구현물과 성능 비교를 통해 제안하는 기법의 우수성을 확인한다.

ABSTRACT

SIMON, a lightweight block cipher developed by the US National Security Agency, is a family of block ciphers optimized for hardware implementation. It supports many kinds of standards to operate in various environments. The counter mode of operation is one of the operational modes. It provides to encrypt plaintext which is longer than the original size. The counter mode uses a constant(Nonce) and Counter value as an input value. Since Nonce is the identical for all blocks, so it always has same result when operates with other constant values. With this feature, it is possible to skip some instructions of round function by pre-computation. In general, the input value of SIMON is affected by the counter. However in an 8-bit environment, it is calculated in 8-bit units, so there is a part that can be pre-computed. In this paper, we focus the part that can be pre-calculated, and compare with previous works.

키워드 : 8-bit AVR 마이크로 컨트롤러, 고속 구현, 블록암호 SIMON, 카운터 운용 모드

Keywords : 8-bit AVR microcontroller, Fast implementation, SIMON block cipher, Counter mode of operation

Received 1 February 2021, **Revised** 24 February 2021, **Accepted** 15 March 2021

*** Corresponding Author** Hwajeong Seo(E-mail:hwajeong84@gmail.com, Tel:+82-2-760-8033)
Associate Professor, Department of IT Engineering, Hansung University, Seoul, 02876 Korea

Open Access http://doi.org/10.6109/jkiice.2021.25.4.588 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

5G 기술과 사물 인터넷의 발달로 인해 센서 노드와 같은 초소형 기기들의 활용이 증가하였다. 이러한 초소형 기기는 제한된 자원, 용량 등으로 인해 연산량이 큰 일반적인 암호 운용이 어렵다. 이를 지원하기 위해 제안된 경량 블록암호는 제한된 환경에서 원활하게 연산이 가능하며, 보안성을 제공한다. 경량 블록암호에는 CHAM, SIMON[1]과 같은 암호가 존재한다.

블록암호는 정해진 규격의 평문만 암호화가 가능하기에, 규격보다 더 큰 평문을 암호화하기 위해서 운용모드를 사용한다. 그 중에서 카운터 모드는 블록암호를 스트림암호 형태로 운용하는 모드로, 병렬화에 유리하며, GCM(Galois/Counter Mode) 모드와 같이 활용 범위도 넓기에 널리 사용되는 운용모드이다.

본 논문에서는 블록암호 운용모드인 카운터 모드의 특성을 활용하여 중간 값의 일부를 선행 연산하여 연산 속도를 향상시킨 SIMON 암호를 제안 및 구현한다. 구현은 8-bit 마이크로컨트롤러인 ATmega128상에서 진행한다. 또한 제안하는 기법과 기존 연구 결과를 비교하여 성능 개선 정도를 확인한다.

본 논문의 구성은 다음과 같다. 2장에서 카운터 운용모드와 블록암호 SIMON의 구성 및 다른 블록암호에 적용된 카운터 고속 구현물에 관해 확인한다. 3장에서는 제안하는 기법의 구현에 관해 상세히 서술한다. 4장에서 제안하는 기법과 기존 구현물의 성능 비교를 진행하며, 5장에서 본 논문의 결론을 맺는다.

Ⅱ. 관련 연구 동향

2.1. 블록암호 카운터 운용모드

블록암호는 정해진 길이의 평문을 암호화하는 알고리즘이다. 따라서 입력된 평문이 정해진 크기보다 크다면 암호화를 진행할 수 없다. 이를 해소하기 위해서 블록암호 운용모드가 제안되었다. 운용모드에는 ECB (Electronic Codebook), CBC(Cipher Block Chaining), CFB(Cipher Feedback), OFB(Output Feedback), 그리고 CTR(Counter) 등이 있다.

이 중에서 카운터 모드는 블록암호를 스트림암호 형태로 변경하는 모드이다. 따라서 카운터 모드는 일반 평문 대신 특별한 입력 값을 사용하여 이를 암호화 한 뒤, 평문과 XOR 연산을 진행하는 것으로 암호문을 생성한다.

카운터 모드의 입력 값은 상수 값인 논스(Nonce)와 블록의 번호를 뜻하는 카운터(Counter)로 분리된다. 이 중에서 논스는 모든 블록이 동일한 값을 지니고 카운터 값만 다르게 된다. 이러한 특징으로 인해 카운터 모드는 각각의 블록들의 연산 결과가 서로 영향을 주지 않으므로 병렬화가 가능하다. 또한 암·복호화를 같은 알고리즘으로 구성이 가능하다[2]. 카운터 모드의 구조와 동작은 그림 1에서 확인이 가능하다.

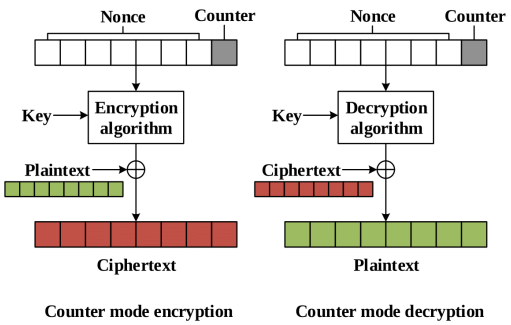


Fig. 1 Schemes of Counter mode of operation.

2.2. 경량 블록암호 SIMON

경량 블록암호 SIMON은 경량 블록암호 SPECK과 함께 발표된 블록암호로, 2013년 미국 국가안보국(National Security Agency, NSA)에서 개발하였다. 2018년에는 RFID 에어 인터페이스 표준으로 선정되어 ISO/29167-21 표준 번호를 보유하고 있다[3]. SIMON은 하드웨어 구현에 최적화 되어있으며, 다양한 환경을 지원하기 위해서 표 1과 같이 매우 많은 종류의 암호 규격을 제공한다.

일반적으로 ARX 암호 알고리즘은 modular Addition, Rotation, XOR로 세 종류의 연산자를 사용한다. 다만 SIMON은 ARX 구조를 채택했지만, modular Addition 연산자 대신 And 연산자를 사용한다. SIMON의 구조는 Feistel 구조로서, 암·복호화를 진행할 때는 입력한 평문을 2개의 블록으로 나누어서 그림 2와 같이 반복적으로 라운드 함수를 진행한다.

Table. 1 Parameters for the lightweight block cipher SIMON.

Block Size 2n(bit)	Key size mn(bit)	Number of rounds
32	64	32

Block Size 2n(bit)	Key size mn(bit)	Number of rounds
48	72	36
	96	36
64	96	42
	128	44
96	96	52
	144	54
128	128	68
	192	69
	256	72

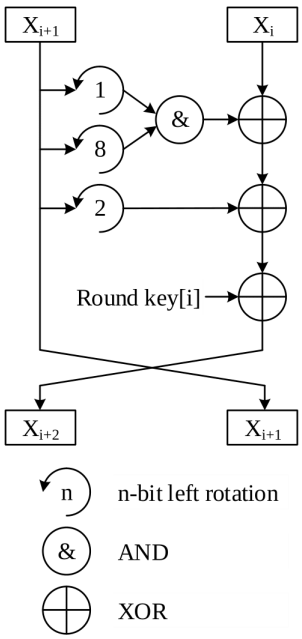


Fig. 2 Round function structure of the SIMON cipher.

2.3. 카운터 모드 고속 구현

카운터 모드는 입력한 평문 대신 논스와 카운터의 조합을 사용하기 때문에, 모든 입력 블록은 카운터 부분을 제외하고는 동일한 값을 지닌다. 따라서 암호 알고리즘의 라운드 함수를 진행하는 동안 논스 부분이 카운터 부분에 영향 받지 않으며, 상수 값과 연산을 진행한다면 해당 연산 결과는 모두 동일하게 된다.

이러한 특징을 사용하여 카운터 모드 상에서 사전 연산을 통해 일부 라운드를 생략하는 기법이 제안되었다. [4]는 AES를 대상으로 한 고속 구현으로 2라운드까지의 연산을 생략하며, [5]는 이를 확장하여 8-bit 마이크로컨트롤러를 대상으로 3라운드까지 연산을 생략한다. [6]은 국산 경량 블록암호 CHAM의 revised 버전을 구현한 것으로 8라운드까지 일부 연산을 생략한다. [7]은 [6]을 확장 구현한 구현물로, 키가 갱신되는 환경 상에서도 사전연산을 지원한다. [8]은 CHAM의 카운터 모드 사전 연산 구현물 중 CHAM-64/128을 대상으로 구현한 것으로, 미사용 레지스터에 일부 라운드 키를 미리 로드하여 더욱 빠른 속도로 동작하도록 하였다. [9]는 국산 블록암호 LEA, HIGHT에 대한 카운터 모드 사전 연산을 구현한 것이다. 제안하는 기법에서 LEA는 3라운드, HIGHT는 4라운드 중에 일부 연산을 생략 가능하다.

Ⅲ. 제안 기법

카운터 모드 사전 연산 고속 구현에서 가장 중요한 부분은 논스 블록이 카운터 블록에 영향을 받는지 유무이다. ARX 구조 암호 알고리즘에서 사용되는 라운드 키, 라운드 카운터 등의 값은 각 라운드별로 지정된 상수 값이므로 논스와 연산할 시 연산 결과가 동일함이 보장된다. 하지만 카운터 값은 모든 블록이 서로 다르기 때문에 연산 결과를 예상할 수 없게 된다. SIMON은 입력 평문을 동일한 크기의 두 개로 나누어서 연산을 진행하는 Balanced feistel 구조를 취하고 있다. 따라서 그림 2의 라운드 함수 구조 상, 모든 블록이 1라운드 중에 카운터 값의 영향을 받으므로 사전 연산이 불가능하다.

하지만 8-bit 환경에서는 일부 구간에서 사전 연산이 가능하다. 이는 8-bit 단위로 연산되는 특성상 논스 값이 있는 레지스터와 카운터 값이 있는 레지스터를 분리해서 볼 수 있기 때문이다. 본 장에서는 각각의 평문 길이별로 사전연산이 가능한 구간과 생략되는 명령어의 구체적인 내역을 확인한다.

3.1. 평문 길이 32-bit SIMON

평문 길이가 32-bit일 때는 16-bit 길이의 블록 두 개가 생성된다. 일반적인 상황에서 카운터 모드는 32-bit 카운터를 사용하지만, 이 경우 모든 블록이 카운터가 되므로 논스가 존재하지 않게 된다. 그러므로 32-bit 입력의 경우에만 예외로 16-bit 카운터를 사용한다. 평문 길이 32-bit 규격의 SIMON 라운드 함수 구조는 그림 3과 같다.

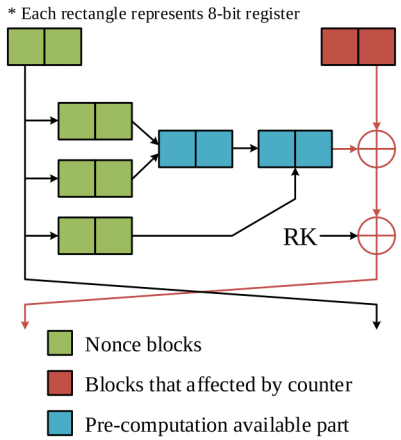


Fig. 3 Round function structure of SIMON-32/m.

입력 블록 중, 논스 부분의 연산은 And, Rotation, XOR의 연산이 진행되는 것을 확인할 수 있다. 해당 부분에는 다른 입력 값이 존재하지 않고 오로지 논스만 연산에 참여한다. 모든 입력 값이 상수이기 때문에 연산 결과가 항시 동일함을 알 수 있다. 결과적으로 마지막 And 연산 부분의 결과물을 사전 연산하는 것으로, 이전 단계를 모두 생략이 가능하다. 구현물 코드 상에서는 MOVW, LSL, ROL, ADC, AND, EOR 명령어가 각각 2개씩 생략된다.

3.2. 평문 길이 48-bit SIMON

평문 길이가 48-bit인 경우, 24-bit 길이의 블록 두 개가 생성된다. 카운터는 32-bit를 사용하므로, 하나의 블록은 모든 블록이 카운터이며 다른 하나는 8-bit가 카운터가 된다. 일반적인 상황에서는 두 개의 블록 모두가 카운터의 영향을 받고 있기 때문에 사전 연산이 불가능하다.

하지만 8-bit 단위로 연산을 하는 환경에서는 논스 부분만 계산을 진행할 수 있기에 부분적으로 사전 연산이 가능하다. 그림 4는 48-bit SIMON의 라운드 함수를 묘사한 것이다. 카운터에 영향 받지 않는 논스 부분을 확인할 수 있다. 해당 부분은 사전 연산이 가능하다. 구현 결과, AND, EOR 명령어가 각각 1개씩 생략된다.

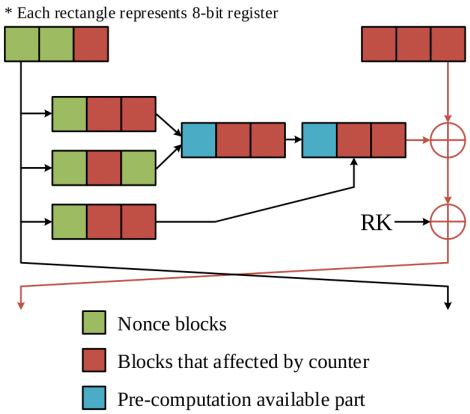


Fig. 4 Counter value flow of SIMON-48/m.

3.3. 평문 길이 64-bit SIMON

평문 길이가 64-bit인 경우, 두 개로 나뉜 각각의 블록은 32-bit이며 카운터는 32-bit를 사용하므로 하나의 블록 전체를 카운터로 사용한다. 그림 5는 64-bit 평문을 사용하는 SIMON의 카운터 흐름을 묘사한 것이다. 전체적인 흐름이 32-bit 평문을 사용하는 SIMON과 동일한 것을 알 수 있다. 실제 구현에서는 기존 SIMON에 비해 4개의 MOVW, 2개의 LSL, 6개의 ROL, 2개의 ADC, 4개의 AND, 그리고 4개의 EOR 명령어를 생략할 수 있었다.

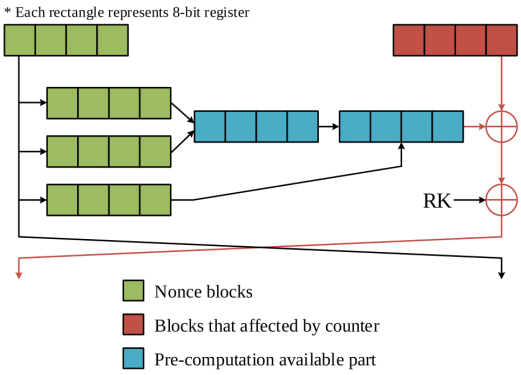


Fig. 5 First round of SIMON-64/m.

3.4. 평문 길이 96-bit SIMON

96-bit 평문을 사용할 때는, 48-bit 크기의 블록 두 개가 생성된다. 카운터의 크기가 32-bit이므로 하나의 블록은 논스 만을 가지며, 다른 하나의 블록은 16-bit의 논스를 보유한다. 그림 6은 96-bit 평문을 사용하는 SIMON의 라운드 함수 구조를 묘사한 것이다.

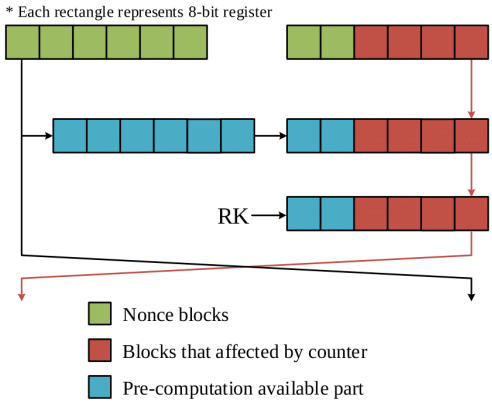


Fig. 6 Round function structure of SIMON-96/m.

다른 SIMON 알고리즘에 비해 크게 차이를 보이는 부분은 두 개의 블록 간 XOR 연산을 진행하는 부분이다. 이전 규격에서는 해당 단계에서 카운터 부분과 XOR을 진행하기 때문에 사전 연산이 불가능했다. 하지만 본 규격에서는 16-bit의 논스가 존재하기 때문에, 해당 부분에서는 사전 연산이 가능하다. 마찬가지로 라운드 키와 XOR하는 부분도 논스 부분에 한해서 사전 연산이 가능하다. 이와 같이 구현한다면, 6개의 MOVW, 2개의 LSL, 10개의 ROL, 2개의 ADC, 6개의 AND, 10개의 EOR, 그리고 2개의 LD 명령어가 생략된다.

그림 6에서 라운드 종료 후, 일부 논스 부분이 여전히 남아있는 것을 확인할 수 있다. 따라서 다음 라운드 중에 사전 연산을 행할 수 있으나, 해당 부분 사전 연산을 통해 얻을 수 있는 이득이 매우 작기에 구현에서는 제외한다.

3.5. 평문 길이 128-bit SIMON

마지막으로 128-bit의 경우에는 하나의 블록은 64-bit가 된다. 카운터의 크기는 32-bit이므로 블록 하나를 절반 차지하며, 나머지 절반인 32-bit는 논스가 된다. 96-bit SIMON과 마찬가지로 두 블록 간의 XOR하는 단계 및 라운드 키를 추가하는 부분에서 약간의 논스 값이 남아있다. 따라서 해당 부분에 한해서 사전 연산이 가능하다. 그림 7은 128-bit SIMON의 라운드 함수 구조를 도식화한 것이다.

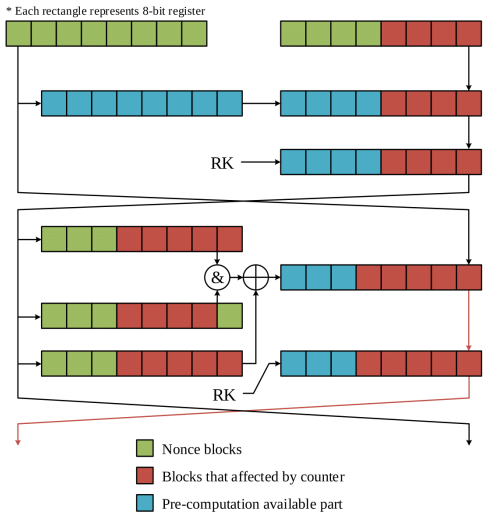


Fig. 7 Counter value flow of SIMON-128/m during two rounds.

128-bit 평문을 사용하는 SIMON은 이전 규격에 비해 논스 부분이 더 많이 포함되어있다. 따라서 1라운드 이후, 다음 라운드에서도 사전 연산을 통해 일부 명령어를 추가로 생략할 수 있다. 이와 같이 구현한 결과 생략된 명령어는 MOVW 8개, LSL 2개, ROL 14개, ADC 3개, AND 11개, EOR 25개, 그리고 LD 7개이다.

Ⅳ. 성능 평가

본 장에서는 제안하는 사전연산을 활용한 SIMON CTR 고속 구현 기법과 기존 구현물의 성능 비교를 진행한다. 개발은 Atmel Studio 7.0을 사용한다. 구현은 ATmega128 프로세서를 대상으로 진행하며, 속도 측정은 최적화 옵션 -O2를 적용한다. 비교 대상 구현물은 [10]의 구현물이다. [10]에서는 8-bit 환경에서의 SIMON, SPECK 블록암호의 최적구현을 제시하며, 본 논문의 구현물과 동일하게 ATmega128 프로세서를 대상으로 한 다양한 최적화 구현물을 제시한다. [10]에서 제안된 최적화 구현물은 ROM 최적화, RAM 최적화, 그리고 속도 최적화이다.

비교를 위한 속도 측정은 [10]과 동일한 프로세서 상인 ATmega128 상에서 이루어진다. 비교 대상은 [10]의 구현물 중, 속도 최적화 구현물을 대상으로 한다. 비교

단위는 바이트 처리에 소요되는 클록의 수를 의미하는 cpb(Clockcycle per byte)를 사용한다. 구체적인 성능 측정 및 비교 결과는 표 2에서 확인할 수 있다.

Table. 2 Performance comparison result (Unit: cpb)

Type	[10]	This work (Fixed-key)	This work (Variable-key)
32/64	172	**162**	**164.8**
48/72	191	**186**	**188**
48/96	191	**186**	**188**
64/96	209	**201**	**206.8**
64/128	221	**214.8**	**216.4**
96/96	253	**249.3**	**250.5**
96/144	264	**258.6**	**260**
128/128	337	**319.6**	**320.8**
128/192	339	**325.2**	**325.4**
128/256	357	**338.1**	**339.3**

제안 기법의 구현물은 고정키, 가변키로 총 두 가지가 존재한다. 각 구현물의 차이는 키의 갱신 여부를 고려한다. 가변키 구현물은 키가 갱신되는 것을 고려하여, 사전연산을 다시 시행하는 부분이 포함된다. 반대로 고정키 구현물은 재연산하는 부분이 존재하지 않는다.

전체적으로 기존 구현물에 비해 속도 향상이 있음을 확인할 수 있다. 입·출력 규격에 따라 성능 개선 정도는 다르지만, 최소 1.5%에서 최대 5.3%의 성능 향상을 확인할 수 있다. 특히, 평문 길이 128-bit를 사용하는 경우에서 성능 향상이 가장 큰 것을 확인할 수 있다. 이는 다른 알고리즘은 1라운드 내에서 사전 연산이 종료되는 것에 반해, 128-bit 길이의 평문을 사용할 때, 2라운드의 일부 명령어까지 생략이 가능하기 때문이다. 가변키 구현물이 고정키 구현물에 비해 동작 속도가 다소 느리게 측정되는 이유는 사전 연산 테이블에서 가져오는 시간이 추가되기 때문이다.

Ⅴ. 결 론

본 논문에서는 경량 블록암호 SIMON의 카운터 모드 사전 연산 최적화에 대해서 그 구조와 원리를 서술하며 구현물의 제안하였다. 제안된 구현물은 기존 구현물에 비해 최대 5.3%의 빠른 속도로 동작하는 것을 확인할 수 있었다. 또한 고정키 상태뿐만 아니라 키가 변동되는 상황인 가변키 상황을 고려하여 두 가지 형태의 알고리즘을 제안하였다. 이것으로 더 다양한 환경에서 효율적으로 사용할 수 있게 하였다.

특히 카운터에 직접적으로 영향 받는 일반적인 환경에서는 제안된 기법 적용이 어렵지만, 8-bit라는 특수한 환경에 맞춘다면 구현이 가능한 것을 확인할 수 있었다. 이와 같이 통상적으로 최적화 기법 적용이 어렵더라도, 특정 환경에서 알고리즘을 분석하는 것으로 기법 적용이 가능한 구간이 존재하는 것을 알 수 있다.

경량 블록암호는 자원이 극도로 제한된 환경에서 원활한 암호화를 제공할 수 있도록 개발되었으며 다양한 운용모드와 결합하는 것으로 안전한 환경을 형성한다. 특히 카운터 운용모드는 GCM 모드에도 활용되어 암호화 뿐만 아니라 인증 기능도 제공할 수 있다. 따라서 카운터 모드를 활용한 더 많은 최적화 기법 개발의 연구를 후속 과제로 제시한다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1F1A1048478). This research was financially supported by Hansung University for Hwajeong Seo.

References

[1] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, “The SIMON and SPECK lightweight block ciphers,” in *52nd ACM/EDAC/IEEE Design Automation Conference*, San Francisco, pp. 1-6, 2015.

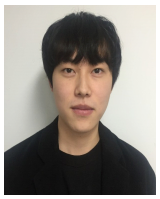
[2] D. H. Kim and K. W. Shin, “An Efficient Hardware Implementation of ARIA Block Cipher Algorithm Supporting Four Modes of Operation and Three Master Key Lengths,” *The Korea Institute of Information and Communication Engineering*, vol. 16, no. 11, pp. 177-184, Nov. 2012.

[3] ISO/IEC Std. 29167-21, *Information technology – Automatic identification and data capture techniques – Part 21: Crypto suite SIMON security services for air interface communications*, ISO/IEC, Geneva, 2018.

[4] J. H. Park and D. H. Lee, “FACE: Fast AES CTR mode Encryption Techniques based on the Reuse of Repetitive Data,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 3, no. 3, pp. 469-499, Aug. 2018.
[5] K. H. Kim, S. J. Choi, H. D. Kwon, Z. Liu, and H. J. Seo, “FACE – LIGHT: Fast AES – CTR Mode Encryption for Low-End Microcontrollers,” in *International Conference on Information Security and Cryptology*, Seoul, pp. 102-114, 2020.
[6] H. D. Kwon, H. J. Kim, S. J. Choi, K. B. Jang, J. H. Park, H. J. Kim, and H. J. Seo, “Compact Implementation of CHAM Block Cipher on Low-End Microcontrollers,” *Information Security Applications*, pp. 127-141, Dec. 2020.
[7] H. D. Kwon, S. W. An, Y. B. Kim, H. J. Kim, S. J. Choi, K. B. Jang, J. H. Park, H. J. Kim, S. C. Seo, and H. J. Seo, “Designing a CHAM Block Cipher on Low-End Microcontrollers for Internet of Things,” *Electronics*, vol. 9, no. 9, pp. 1548, Sep. 2020.
[8] H. D. Kwon, K. B. Jang, J. H. Park, and H. J. Seo, “ High-Speed Implementation to CHAM-64/128 Counter Mode with Round Key Pre-Load Technique,” *Korea Institute of Information Security and Cryptology*, vol. 30, no. 6, pp. 1217-1223, Dec. 2020.
[9] Y. B. Kim, H. D. Kwon, S. W. An, H. J. Seo, and S. C. Seo, “Efficient Implementation of ARX-Based Block Ciphers on 8-Bit AVR Microcontrollers,” *Mathematics*, vol. 8, no. 10, pp. 1837, Oct. 2020.
[10] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, “The Simon and Speck Block Ciphers on AVR 8-Bit Microcontrollers,” in *International Workshop on Lightweight Cryptography for Security and Privacy*, Istanbul, pp. 3-20, 2014.

권혁동(Hyeok-Dong Kwon)

2018년 2월: 한성대학교 정보시스템공학과 공학 학사 졸업
2020년 2월: 한성대학교 IT융합공학부 석사 졸업
2020년 3월~현재: 한성대학교 정보컴퓨터공학과 박사과정
※관심분야 : 정보보안, 암호구현



장경배(Kyung-Bae Jang)

2019년 2월: 한성대학교 IT융합공학부 공학 학사 졸업
2021년 2월: 한성대학교 IT융합공학부 석사 졸업
2021년 3월~현재: 한성대학교 정보컴퓨터공학과 박사과정
※관심분야 : 양자 컴퓨터, 정보보안

김현지(Hyun-Ji Kim)

2020년 2월: 한성대학교 IT융합공학부 공학 학사 졸업
2020년 3월~현재: 한성대학교 IT융합공학부 석사과정
※관심분야 : 정보보안, 인공지능

서화정(Hwa-Jeong Seo)

2010년 2월 부산대학교 컴퓨터공학과 학사 졸업
2012년 2월 부산대학교 컴퓨터공학과 석사 졸업
2012년 3월~2016년 1월: 부산대학교 컴퓨터공학과 박사 졸업
2016년 1월~2017년 3월: 싱가포르 과학기술청
2017년 4월~현재: 한성대학교 IT 융합공학부 조교수
※관심분야 : 정보보호, 암호화 구현, IoT