정 보 보 호 학 회 지
제27권 제6호, 2017. 12

경량 사물인터넷 플랫폼 상에서의 대칭키 암호 구현 기술

서 화 정*, 박 태 환**, 이 가 람**

요 약

경량 사물인터넷 플랫폼은 제한적인 연산 성능과 저장 공간을 가진다. 따라서 해당 플랫폼 상에서의 모든 연산들은 효율적으로 구현되어야 한다. 이를 위해 최근에는 경량화된 형태의 대칭키 암호화가 많이 제안되고 있다. 본 논문에서는 경량 사물인터넷 플랫폼 상에서의 효율적인 경량 대칭키 암호화 구현 방안에 대해 확인해 보도록 한다. 먼저 경량 사물인터넷 플랫폼의 특성을 확인해 보며 해당 경량 플랫폼의 특성을 활용하여 경량 대칭키 암호화 연산을 효율적으로 구현하는 방안에 대해 확인해 보도록 한다.

Ⅰ. 서 론

모든 사물이 인터넷을 통해 연결되는 사물인터넷 시대에는 사물 들 간의 안전한 보안 통신을 통하여 서비스의 신뢰성을 보장하는 것이 매우 중요하다. 보안 통신을 위해 사용되는 암호화 알고리즘으로는 크게 대칭키 암호, 공개키 암호, 그리고 해시 함수가 있다. 본 논문에서는 최근에 연구가 활발히 진행되고 있는 대칭키 암호의 경량 사물인터넷 플랫폼 상에서의 효율적인 운용 방안에 대해 확인해 보도록 한다. 본 논문의 구성은 다음과 같다. 2장에서는 경량 사물인터넷 플랫폼에 대해 확인해 본다. 3장에서는 Fair Evaluation of Lightweight Cryptographic Systems (FELICS)에 대해 확인해 보도록 한다. 4장에서는 경량 대칭키 암호화 알고리즘에 대해 알아본다. 5장에서는 경량 사물 인터넷 플랫폼 상에서의 경량 대칭키 암호화 알고리즘 구현 및 성능에 대해 확인해 본다. 6장에서는 본 논문의 결론을 내린다.

Ⅱ. 경량 사물인터넷 플랫폼

2.1. 8-비트 Atmel AVR ATmega128

경량 사물인터넷 플랫폼 중에서 기본 워드의 크기가 가장 작은 Atmel AVR ATmega128은 8-비트 단위로 연산을 수행한다. 하드웨어 아키텍처는 Harvard 구조를 따르며 동작 주파수는 16MHz이다. 또한 32개의 8-비트 일반 목적 레지스터를 탑재하고 있으며 총 81개의 명령어 셋을 가지고 있다. 플래시 메모리는 128KB이며, 4KB의 EEPROM과 4KB의 SRAM을 탑재하고 있다.

2.2. 16-비트 Texas Instruments MSP430F1611

16비트 워드 크기를 가지는 Texas Instruments MSP430F1611은 동작 주파수가 8MHz이며 총 16개의 16-비트 레지스터를 가지며 이중에서 12개 레지스터는 일반 목적 레지스터로 동작하게 된다. 하드웨어 아키텍처는 Von Neumann 구조를 따르며 27개의 명령어 셋을 가지고 있다. 48KB의 플래시 메모리와 10KB의 RAM을 가지고 있다.

2.3. 32-비트 Arduino Due ARM Cortex-M3

32-비트 워드 크기를 가지는 Arduino Due ARM Cortex-M3의 경우 동작 주파수는 84MHz이며 총 16개의 레지스터 중 13개를 일반 목적 레지스터로 활용가능하다. 하드웨어 아키텍처는 Von Neumann 구조를 따르며 56개의 명령어 셋을 가지고 있다. 512KB의 플래시

이 성과는 2017년도 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원과(No. 20152000000170) 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행한 연구 과제입니다(No. NRF-2017R1C1B5075742).
* 한성대학교 IT융합공학부
** 부산대학교 전기전자컴퓨터공학과

메모리와 96KB의 SRAM을 가지고 있다.

Ⅲ. Fair Evaluation of Lightweight Cryptographic Systems (FELICS)

2015년에 룩셈부르크 대학에서는 2장에서 명시한 3개의 경량 사물인터넷 플랫폼 상에서 대칭키 암호화 알고리즘 최적화 구현 결과를 평가하였다 [1, 2]. 평가에 사용된 시나리오는 총 3개이다. 첫 번째로 “Cipher Operation”에서는 암호화 및 복호화를 통해 테스트 벡터 결과값을 도출하도록 하였다. 두 번째는 “Communication Protocol”로써 128 바이트 메시지를 Cipher Block Chaining (CBC) 모드로 암호화하여 전송하도록 하였다. 세 번째는 “Challenge-Handshake Authentication Protocol”로써 사물인터넷 디바이스는 라운드 키를 보관하고 있고 이를 Counter (CTR) 모드로 128-비트 암호화하도록 한다. 성능 비교 척도로는 프로그램 코드 크기, RAM 사용량, 그리고 연산 속도를 사용하였다. 그 결과 첫 번째 라운드에서는 128-비트 LEA가 1등을 차지하였으며 두 번째 라운드에서는 128-비트 HIGHT가 1등을 차지하였다.

Ⅳ. 대칭키 암호화 알고리즘 구현

대칭키 암호화는 크게 Addition Rotation eXclusive-or (ARX) 구조와 Substitute Permutation Network (SPN) 구조로 나뉘어 생각해 볼 수 있다. ARX 구조는 덧셈, 회전, 그리고 XOR 연산으로 구성되는 대칭키 암호화 알고리즘으로써 본 논문에서는 LEA, HIGHT, CHAM, SIMON, 그리고 SPECK에 대해 확인해 보도록 한다. SPN 구조는 S-box와 Permutation 연산을 통해 대칭키 암호를 구현하며 본 논문에서 살펴볼 SPN 대칭키 암호화는 PRESENT와 GIFT이다.

4.1. LEA

NSR에 의해 개발되어 WISA’13에서 발표된 LEA 블록암호화는 ARX 구조에 기반을 둔 경량 암호화 기법이다 [3]. 지원하는 암호화 비트는 128, 192, 그리고 256-비트이며 블록의 크기는 128-비트이다. 특히 ARX 연산자는 32-비트 단위로 동작한다. 경량화된 연산자는 8-비트 AVR 프로세서와 32-비트 ARM 프로세서 상에서 효율적이며 특히 최신 병렬화 엔진인 SSE, AVX, 그리고 NEON를 활용하면 높은 성능을 기대할 수 있다.

4.2. HIGHT

CHES’06에서는 경량 블록암호화 HIGHT가 제안되었다 [4]. HIGHT 암호화는 8-비트 기반 ARX 연산으로 설계되었다. 핵심 연산은 8-비트 회전 연산과 XOR 연산으로 구성된 F0 그리고 F1 함수이다. HIGHT 암호화는 하드웨어 상에서 초경량 구현이 가능하다. 블록의 크기는 64-비트이며 암호화 키는 128-비트를 사용한다.

4.3. CHAM

ICISC’17에서는 초경량 블록암호화 CHAM이 NSR에 의해 제안되었다 [5]. LEA 블록 암호화 알고리즘이 32-비트 ARM 프로세서 상에서는 효율적이었지만 8-비트/16비트 프로세서 상에서는 효율적이지 못한 결과를 보여주었다. 새로운 블록 암호화 CHAM의 경우 8-비트/16-비트 프로세서 상에서도 높은 성능을 나타낼 수 있도록 경량 설계되었다. 제공하는 블록암호화는 CHAM-64/128, CHAM-128/128, 그리고 CHAM-128/256이다. 이중에서 CHAM-64/128은 16-비트 워드 기반 연산자, CHAM-128/128 그리고 CHAM-128/256의 경우에는 32-비트 워드 기반 연산자로 구성된다. 소프트웨어 그리고 하드웨어 상에서의 경량 구현에 매우 효율적인 장점을 가지고 있다.

4.4. SIMON 그리고 SPECK

NSA에 의해 개발된 SIMON 그리고 SPECK 블록암호화는 DAC’15에서 발표되었다 [6]. 두 블록암호화 모두 ARX 연산을 기반으로 하고 있다. 두 블록암호화 알고리즘의 차이점으로는 SIMON은 하드웨어 상에서의 최적화 구현이 가능하며, SPECK은 소프트웨어 상에서 효율적이라는 것이다. 해당 블록암호화의 큰 특징으로는 다양한 블록과 키 크기를 제공한다는 것이다. 제공하는 블록 크기와 키 크기의 목록은 다음과 같다 (32/64, 48/72, 48/96, 64/96, 64/128, 96/96, 96/144, 128/128, 128/192, 128/256).

4.5. PRESENT

CHES'07에서 발표된 경량 블록 암호화 PRESENT는 SPN 구조를 가진다 [7]. 블록 크기는 64-비트이며 키크기는 80-비트와 128-비트 두 가지 모드를 지원한다. S-box의 크기는 4-비트 기반으로 설계되어 초경량 디바이스 상에서 매우 효율적인 구현이 가능하다. Permutation 연산의 경우 하드웨어 상에서는 회로 연결을 통해 추가적인 연산없이 구현 가능하다.

4.6. GIFT

CHES'17에서는 PRESENT 암호화 연산을 보다 경량화한 블록암호화 GIFT가 제안되었다 [8]. GIFT는 PRESENT와 동일하게 SPN 구조로 설계되었다. GIFT가 지원하는 모드는 GIFT-64/128과 GIFT-128/128이다. PRESENT와 유사하게 4-비트 기반 S-box를 사용하였지만 Permutation 연산의 경우 PRESENT와는 상이한 형식으로 설계하여 암호화 연산을 경량화함과 동시에 보안 강도는 유지하였다.

Ⅴ. 성능평가

LEA 블록 암호화는 32-비트 단위의 ARX 연산으로 구성되어 있다. 해당 연산은 32-비트 ARM 프로세서 상에서는 효율적으로 구현이 가능하다. 하지만 8-비트 AVR 그리고 16-비트 MSP 프로세서 상에서는 다중 연산을 통해 32-비트 연산을 구현 가능하다. 덧셈의 경우 carry 연산을 수행하며 XOR의 경우 32-비트 단위에 맞춰 워드 연산을 수행하도록 한다. ARX연산 중에서 특히 성능에 영향을 미치는 연산은 회전연산으로써 8-비트 그리고 16-비트 프로세서 상에서는 8-비트 그리고 16-비트 범위 안의 회전 연산은 생략이 가능한 특징을 활용할 수 있다 [9]. 이와 반대로 HIGHT 블록암호화의 경우에는 8-비트 단위로 연산이 수행되는 특징을 가진다. 해당 연산자들은 8-비트 AVR 프로세서 상에서 매우 효율적으로 연산이 가능한 특징을 가진다. 반면에 16-비트 MSP 프로세서와 32-비트 ARM 프로세서 상에서는 비효율적인 특징을 가진다. [10]에서는 16-비트 MSP 프로세서 상에서는 바이트 단위 연산으로 HIGHT 블록 암호화를 수행하였다. 이와 더불어 16-비트 레지스터를 효율적으로 활용하기 위해 2 개의 인자를 하나의 레지스터에 저장하여 활용하는 방안을 적용하였다. 32-비트 ARM 프로세서의 경우에는 SIMD 연산자가 지원되지 않는 모델에서도 SIMD와 유사한 연산이 가능하도록 하는 Pseudo SIMD 연산을 활용하였다. 2 개의 블록암호화를 한 번에 수행가능하도록 8-비트 연산자 입력값 두 개와 더불어 패딩값을 활용하여 연산 효율성을 높였다. 최근에 제안된 경량 암호화 CHAM의 경우 연산 워드의 크기가 모드에 따라 상이하게 나타나게 된다 [5]. 먼저 CHAM-64/128의 경우에는 16-비트 워드 단위 연산이 가능한 특징을 가진다. 이는 8-비트 AVR과 16-비트 MSP 프로세서 상에서 매우 효율적으로 구현이 가능한 특징을 가진다. 반면에 32-비트 ARM 프로세서 상에서는 덧셈과 회전 연산이 비효율적으로 구현되는 특징을 가진다. 그 외의 연산 모드인 CHAM-128/128과 CHAM-128/256의 경우에는 32-비트 워드 단위의 연산이 가능한 특징을 가진다. 두 모드의 경우에는 32-비트 ARM 프로세서 상에서 효율적으로 구현 가능하다. SIMON과 SPECK의 경우 기존의 ARX 연산과 유사한 형태로 성능 향상이 가능하다 [11]. 그 중에서도 8-비트 AVR 상에서의 회전 연산을 기존의 ROR/ROL과 같은 기본 연산자 대신 곱셈 연산자 (MUL)을 이용하여 효율적으로 계산하는 방법을 제시하였다. 이는 3-비트 회전 연산을 14 사이클과 20 바이트의 플래시 메모리로 구현 가능한 특징을 가진다. [12]에서는 8-비트 AVR을 포함하여 16-비트 MSP 그리고 32-비트 ARM 프로세서 상에서도 매우 효율적인 연산 결과 도출이 가능함을 증명하였다.

SPN 구조를 가지는 PRESENT의 경우에는 경량 프로세서 상에서의 구현보다는 고성능 SIMD 프로세서 상에서 bitslicing기법을 활용하여 효율적인 연산이 가능함을 증명하였다 [13]. bitslicing 기법은 기존의 워드 단위의 연산 체계가 아니라 워드를 비트 단위로 나눈 다음 해당 비트 단위로 연산을 수행함으로써 연산 성능을 향상시키는 기법이다. bitslicing 기법이 효과적으로 적용되기 위해서는 한 번에 많은 암호화 연산을 수행해야 하는 경우를 선정해야 한다. 하지만 마이크로 프로세서는 매우 한정적인 레지스터 저장 공간을 가지고 있기 때문에 많은 연산을 모두 레지스터에 저장하는 것이 불가능하였다. 이를 효과적으로 개선하기 위해 최근에 CHES'17에서 발표된 논문에서는 Permutation과 S-box

연산의 순서를 교환하여 암호화 연산을 적용하는 기술이 제안되었다 [14]. 해당 기법은 32-비트 ARM 프로세서 상에서 구현 시 16-비트 단위 연산 구현 시 두 개의 인자를 하나의 32-비트 워드에 넣어 구현하는 방법을 취했다. 그리고 S-box 연산을 기존 Look-Up Table (LUT)가 아닌 연산의 조합으로 풀어서 구현하는 방법을 취했다. Permutation을 먼저 수행하게 되는 경우 워드 상의 값들이 S-box를 효율적으로 수행할 수 있는 형태로 변경되는 특징을 가진다. 따라서 S-box를 Permutation과 순서를 달리 하여 구현 시 높은 성능 달성이 가능하다. GIFT 블록 암호화의 경우 PRESENT로부터 파생된 블록암호화로써 [14]에서 제안된 기법이 적용가능할 것으로 보인다. 하지만 Permutation 과정이 PRESENT의 경우 S-box를 bitslicing하기 좋도록 배열하는 특징이 있다면 GIFT의 경우 S-box에 효과적인 형태로의 배열을 제공하지 않는다. 따라서 Permutation과 S-box의 순서를 변경하는 기법이 바로 적용되기는 어렵다.

경량 암호화 연산에 대한 성능 평가 지표는 매우 다양하다. 저장 공간인 ROM과 RAM의 소모를 생각해 볼 수 있으며 그와 Trade-off 관계에 있는 연산 성능을 확인해 볼 수 있다. 본 논문에서는 그 중에서도 현재까지 발표된 결과 중 Electronic Codebook (ECB) 혹은 CTR과 같은 운영 모드로 동작을 시켰을 때 가장 빠르게 연산을 수행하는 경우를 기준으로 비교해보도록 하겠다. [표 1]에서는 8-비트 AVR 프로세서 상에서의 성능을 나타낸다. ARX 기반 블록 암호화의 경우 매우 높은 성능을 나타냄을 확인할 수 있다. 특히 SPECK-64/128의 경우 122 cycle/byte로 가장 높은 성능을 나타냄을 확인할 수 있다. SIMON과 비교 시 SPECK이 소프트웨어 성능이 높게 나옴을 확인할 수 있다. 국내 암호화 중에서는 CHAM-128/128이 가장 높은 성능인 148 cycle/byte를 나타내고 있다. LEA와 HIGHT의 경우에도 168 그리고 160 cycle/byte라는 높은 성능을 나타냄을 확인할 수 있다. SPN 구조를 따르는 블록 암호화인 PRESENT와 GIFT의 경우 8-비트 프로세서 상에서의 최적화 연산 구현이 아직 많이 연구되어 있지 않은 시점이다. 따라서 이를 효과적으로 구현하기 위한 방안 모색이 필요하다.

[표 2]에서는 16-비트 MSP 프로세서 상에서의 구현 결과가 나타나 있다. 여기서도 최고의 성능은 SPECK-64/128에서 89 cycle/byte로 확인할 수 있었다. 이는 SPECK이 소프트웨어 상에서 보다 효율적으로 구현할 수 있는 경량 연산자들로 구성되어 있기 때문이다. 국산 암호 중에서는 CHAM-64/128이 118 cycle/byte로 가장 높은 성능을 나타내었다. CHAM-64/128의 경우 16-비트 단위의 연산자로 구성되어 있어 16-비트 MSP 프로세서 상에서 보다 효율적인 구현이 가능하다. 마찬가지로 PRESENT와 GIFT의 경우 16-비트 프로세서 상에서의 최적화 방안에 대한 모색이 필요하다.

[표 3]에서는 32-비트 ARM 프로세서 상에서의 성능을 나타내고 있다. LEA의 경우 32-비트 워드 단위 연산에 매우 효율적이기 때문에 가장 높은 성능을 나타냄을 확인할 수 있다. 반면에 8-비트 단위 연산이 주를 이

〔표 1〕 Performance evaluation of block cipher on 8-bit AVR processors where (C/B) is cycle per byte.

Algorithm	Encryption (C/B)
LEA-128 〔10〕	168
HIGHT-128 〔10〕	160
CHAM-64/128 〔5〕	172
CHAM-128/128 〔5〕	148
SIMON-64/128 〔11〕	221
SIMON-128/128 〔11〕	337
SPECK-64/128 〔11〕	122
SPECK-128/128 〔11〕	143
PRESENT	-
GIFT	-

〔표 2〕 Performance evaluation of block cipher on 16-bit MSP processors where (C/B) is cycle per byte.

Algorithm	Encryption (C/B)
LEA-128 〔10〕	129
HIGHT-128 〔10〕	222
CHAM-64/128 〔5〕	118
CHAM-128/128 〔5〕	125
SIMON-64/128 〔12〕	153
SIMON-128/128 〔12〕	379
SPECK-64/128 〔12〕	89
SPECK-128/128 〔12〕	101
PRESENT	-
GIFT	-

〔표 3〕 Performance evaluation of block cipher on 32-bit ARM processors where (C/B) is cycle per byte.

Algorithm	Encryption (C/B)
LEA-128 〔10〕	31
HIGHT-128 〔10〕	258
CHAM-64/128 〔5〕	134
CHAM-128/128 〔5〕	70
SIMON-64/128	-
SIMON-128/128	-
SPECK-64/128 〔2〕	53
SPECK-128/128	-
PRESENT 〔14〕	264
GIFT	-

루는 HIGHT의 경우 258 cycle/byte가 필요함으로써 성능이 많이 저하됨을 확인할 수 있다. CHES'17에서 발표된 PRESENT에 대한 최적화 구현의 경우 PRESENT가 하드웨어와 고성능 프로세서 상에서는 효과적이지만 저전력 마이크로 프로세서 상에서는 264 cycle/byte가 필요한 것으로 볼 때 여전히 다른 ARX 기반 암호화에 비해 성능이 떨어짐을 확인할 수 있다. GIFT의 경우에는 PRESENT와 유사하지만 Permutation에서 복잡도가 상승하기 때문에 이를 효과적으로 구현하는 방안에 대한 연구가 필요할 것으로 보인다.

Ⅵ. 결 론

본 논문에서는 경량 사물인터넷 플랫폼 상에서 암호화 연산을 효율적으로 구현할 수 있는 다양한 방안들에 대해 확인해 보았다. 현재 경량 암호화 구현 분야는 매우 많은 연구가 진행되고 있으며 앞으로도 보다 경량의 특징을 가지지만 높은 보안성을 확보할 수 있는 대칭키 암호화 방안에 대한 연구와 이를 경량화된 사물인터넷 플랫폼 상에서 최적화하여 구현하는 방안에 대한 연구가 활발히 진행될 것으로 보인다.

참 고 문 헌

[1] D. Dinu, Y. L. Corre, D. Khovratovich, L. Perrin, J. Großschädl, A. Biryukov, “Triathlon of lightweight block ciphers for the internet of things,” *IACR ePrint archive*, 2015.

[2] D. Dinu, A. Biryukov, J. Großschädl, D. Khovratovich, Y. L. Corre, L. Perrin, “FELICS – Fair Evaluation of LIghtweight Cryptographic Systems,” *In NIST Workshop on Lightweight Cryptography*, vol. 128, 2015.

[3] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, D. G. Lee, “LEA: A 128-bit block cipher for fast encryption on common processors,” *In International Workshop on Information Security Applications (WISA’13)*, pp. 3-27, 2013.

[4] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, S. Chee, “HIGHT: A new block cipher suitable for low-resource device,” *Conference on Cryptographic Hardware and Embedded Systems (CHES’06)*, vol. 4249, pp. 46-59, 2006.

[5] B. Koo, D. Roh, H. Kim, Y. Jung, D. Lee, D. Kwon, “CHAM: A Family of Lightweight Block Ciphers for Resource-Constrained Devices,” *International Conference on Information Security and Cryptology (ICISC'17)*, 2017.

[6] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, L. Wingers, “The SIMON and SPECK lightweight block ciphers,” *In Design Automation Conference (DAC‘15)*, pp. 1-6, 2015.

[7] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, C. Vikkelsoe, “PRESENT: An ultra-lightweight block cipher,” *Conference on Cryptographic Hardware and Embedded Systems (CHES'07)*, vol. 4727, pp. 450-466, 2007.

[8] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, Y. Todo, “GIFT: a small PRESENT,” *Conference on Cryptographic Hardware and Embedded Systems (CHES'17)*, pp. 321-345, 2017.

[9] H. Seo, Z. Liu, J. Choi, T. Park, H. Kim,

"Compact implementations of LEA block cipher for low-end microprocessors," I*n International Workshop on Information Security Applications (WISA'15)*, pp. 28-40, 2015.

[10] H. Seo, I, Jeong, J. Lee, W. H. Kim, "Compact Implementations of ARX Based Block Ciphers on IoT Processors," *ACM Transactions on Embedded Computing Systems (ACM-TECS)*, accepted for publication.

[11] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "The SIMON and SPECK block ciphers on AVR 8-bit microcontrollers," I*n International Workshop on Lightweight Cryptography for Security and Privacy*, pp. 3-20, 2014.

[12] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "SIMON and SPECK: Block Ciphers for the Internet of Things," *IACR Cryptology ePrint Archive*, 2015.

[13] R. Benadjila, J. Guo, V. Lomné, T. Peyrin, "Implementing lightweight block ciphers on x86 architectures," *In International Conference on Selected Areas in Cryptography (SAC'13)*, pp. 324-351, 2013.

[14] T. Reis, D. Aranha, J. López, " PRESENT Runs Fast: Efficient and Secure Implementation in Software," *Conference on Cryptographic Hardware and Embedded Systems (CHES'17)*, 2017.

〈저 자 소 개〉

서 화 정 (Hwa-jeong Seo)
종신회원

2010년 2월 : 부산대학교 컴퓨터공학과 학사 졸업
2012년 2월 : 부산대학교 컴퓨터공학과 석사 졸업
2016년 2월 : 부산대학교 컴퓨터공학과 박사 졸업
2015년 4월~5월 : 싱가포르 난양공대 인턴쉽
2016년 1월~2017년 3월 : 싱가포르 과학기술청 연구원
2017년 4월~현재 : 한성대학교 조교수
관심분야 : 정보보호, 암호화 구현, IoT

박 태 환 (Tae-hwan Park)
학생회원

2013년 2월 : 부산대학교 정보컴퓨터공학부 학사 졸업
2013년 3월~현재 : 부산대학교 전기전자컴퓨터공학과 석, 박사 통합과정
관심분야 : 암호화 구현, IoT 디바이스 보안, 양자 내성 암호

이 가 람 (Ga-ram Lee)
학생회원

2016년 2월 : 부산대학교 정보컴퓨터공학부 학사 졸업
2016년 3월~현재 : 부산대학교 전기전자컴퓨터공학과 석사과정
관심분야 : SW 암호 최적화 구현, IoT 보안, 역공학, 임베디드 보안, 머신러닝