SIMECK-64/128 블록암호 알고리듬의 하드웨어 구현

김민주 · 정영수 · 신경욱*

금오공과대학교

A Hardware Implementation of SIMECK-64/128 Block Cipher Algorithm

Min-Ju Kim · Young-su Jeong · Kyung-Wook Shin*

Kumoh National Institute of Technology

E-mail : m0325j@kumoh.ac.kr / 20161101@kumoh.ac.kr / kwshin@kumoh.ac.kr

요 약

본 논문에서는 적정한 보안 강도를 가지면서 경량 하드웨어 구현이 가능한 SIMECK 블록암호 알고리듬의 하드웨어 설계를 기술한다. 빠른 암호화와 복호화를 진행할 수 있도록 동작 라운드 수를 줄이는 two-stage 방식을 이용하여 구현하였다. 설계된 SIMECK 암호 코어를 Arty S7-50 FPGA 디바이스에 구현하고, Python을 이용한 GUI와 결합하여 암호화·복호화의 하드웨어 동작을 검증하였다.

ABSTRACT

In this paper, we describe a hardware design of the SIMECK block cipher algorithm that can be implemented in lightweight hardware with appropriate security strength. To achieve fast encryption and decryption operations, it was designed using two-step method that reduces the number of operation rounds. The designed SIMECK cryptographic core was implemented in Arty S7-50 FPGA device and its hardware operation was verified with a GUI using Python.

키워드

SIMECK, lightweight block cipher, information security, FPGA

Ⅰ. 서 론

정보화 사회가 되면서 개인정보 보안의 중요성이 커지고 있고 IoT (Internet of Things) 기술이 발전함에 따라 전자기기와 네트워크 환경이 다양해지며 IoT 기술에 대한 보안 필요성이 높아지고 있다. IoT, WSN (Wireless Sensor Network) 등 제한된 하드웨어 자원을 갖는 분야의 정보보안에 적합한 경량 (lightweight) 블록암호 알고리듬으로 SIMON, PICCOLO, SPECK, SIMECK 등이 제안되고 있다. 본 논문에서는 임베디드 시스템에 적합한 경량 블록암호 SIMECK 알고리듬을 two-stage 방식의 하드웨어 설계로 설계하고 Arty S7-50 FPGA 디바이스에 구현을 하여 암호화·복호화 동작을 검증하였다.

II장은 SIMECK 알고리듬에 대해 소개하고 III장은 SIMECK 하드웨어 설계 및 RTL 시뮬레이션과 GUI로 암호화·복호화 동작 결과를 보이며, IV장에서 결론을 맺는다.

* corresponding author

Ⅱ. SIMECK 경량 블록암호 알고리듬

SIMECK 블록암호 알고리듬은 SIMON 알고리듬의 블록 단위 암호 방식과 SPECK 알고리듬의 키 생성 방식에서 아이디어를 얻어 만든 Feistel 방식의 암호이다.[1] SIMECK 알고리듬은 SIMECK-32/64, SIMECK-48/96, SIMECK-64/128를 지원하며, 본 논문에서는 SIMECK-64/128를 하드웨어로 구현했다. SIMECK-64/128은 64-비트의 메시지 블록을 128-비트 키를 이용하여 암호화와 복호화 한다.

SIMECK-64/128의 암호화 방식은 먼저 64-비트 메시지 블록을 상위 32-비트와 하위 32-비트로 나눈다. 이를 $R_{k_i}(l_i, r_i) = (r_i \oplus f(l_i) \oplus k_i, l_i)$ 연산을 이용하여 44번의 라운드(round) 연산을 수행한다. 여기서 l_i와 r_i는 각각 64-비트 블록의 상위 32-비트

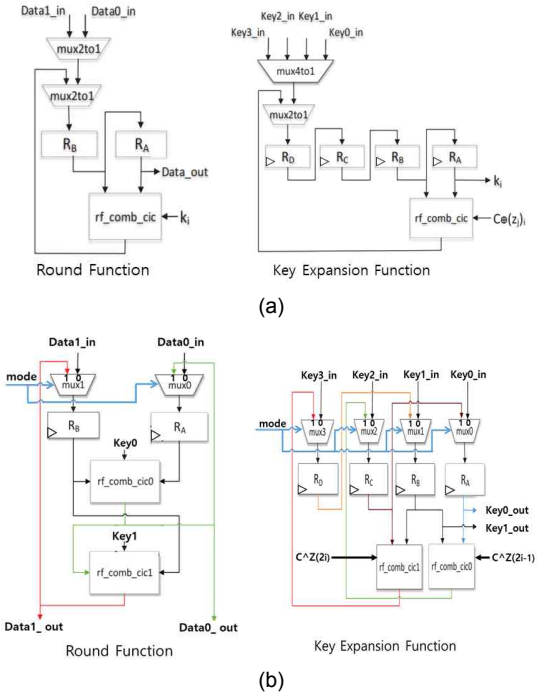

그림 1. SIMECK-64/128 블록암호 알고리듬의 하드웨어 구현 방식. (a) basic iterative architecture 방식과 (b) two-stage 방식의 라운드와 키 확장 모듈

와 하위 32-비트에 해당한다. 함수 $f()$는 $f(x)=(x\odot(x\lll 5))\oplus(x\lll 1)$로 비트 이동하고, AND 연산과 XOR 연산하는 함수를 의미한다. k_i는 i-번째 32-비트 키 값으로 입력받은 128-비트 키를 32-비트씩 나누어 상위 32-비트부터 t_2,t_1,t_0,k_0라 했을 때, k_i는 $k_{i+1}=t_i$, $t_{i+3}=k_i\oplus f(t_i)\oplus C\oplus(z_j)_i$에 의해 생성된다. 이때, $C=2^n-4$이고, n은 워드 크기이다. 본 논문에서 구현하는 SIMECK-64/128는 워드 크기가 32이므로 C=0xFFFFFFFC 이다. z_j는 LFSR (Linear Feedback Shift Register)에서 나오는 1-비트 값에 해당한다. SIMECK-64/128를 구현하므로 z_1에 해당하는 LFSR 다항식 X^6+X+1을 이용하여 난수를 발생시키고 얻어진 값을 대입한다.

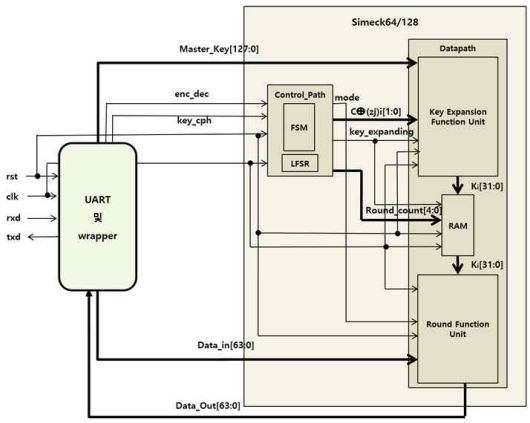

그림 2. SIMECK-64/128 코어의 블록도

복호화는 Feistel 방식의 암호이기 때문에 암호화와 연산 방식이 동일하지만 상위 32-비트와 하위 32-비트를 나눈 블록을 암호화와 반대로 입력해야 하고, 키 또한 역순으로 입력하는 차이점이 있다.

III. SIMECK 하드웨어 설계 및 검증

SIMECK-64/128 알고리듬은 그림 1에서 보는 바와 같이 basic iterative architecture 방식과 two-stage 방식으로 구현할 수 있다. 본 논문에서는 임의의 길이의 메시지를 빠르게 암호화·복호화할 수 있도록 한 클록에 1 라운드씩 처리하는 basic iterative architecture 방식 대신에 two-stage 방식으로 키 확장과 암호화·복호화 모듈을 구성하여 한 클록에 2 라운드씩 수행하도록 설계하였다 [2]. 그림 2는 설계된 SIMECK-64/128 블록암호 코어의 블록도이며, 키 확장, 암호화, 복호화를 위해 유한 상태머신 (finite state machine)을 이용하여 동작 신호를 제어하였다. two-stage 방식의 키 확장을 위해 LFSR 신호도 한 클럭에 두 번 작동한 신호가 나올 수 있도록 X^6+X+1의 다항식을 갖는 LFSR에 조합회로를 이용하여 설계하였으며, 또한 확장된 키는 RAM에 저장해서 암호화·복호화에 사용되도록 하였다.

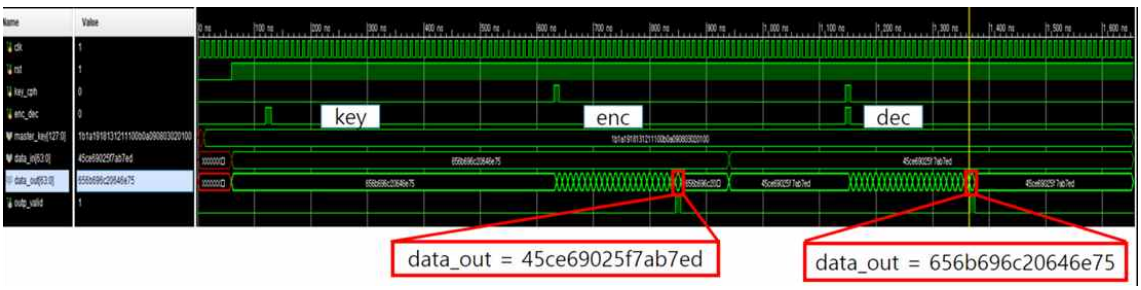

그림 3. 설계된 SIMECK-64/128 코어 RTL 기능검증 결과

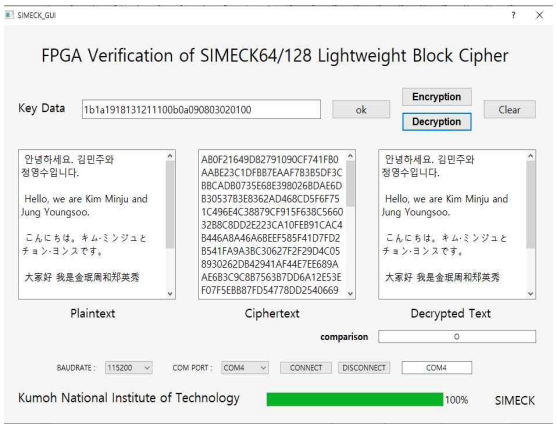

그림 4. 설계된 SIMECK-64/128 코어의 FPGA검증 결과

그림 3은 그림 2의 블록도를 바탕으로 설계된 SIMECK-128/64 코어의 RTL 기능검증 결과이다. 키 “0x1b1a1918131211100b0a090803020100”을 입력하고 평문 “0x656b696c20646e75”과 암호화 신호를 인가하면 암호화된 값 “0x45ce69025f7ab7ed”이 출력이 된다. 키가 입력된 상태에서 복호화 하려는 암호문 “0x45ce69025f7ab7ed”와 복호화 신호를 인가하면 복호화된 평문 “0x656b696c20646e75”이 출력되는 것을 볼 수 있다.

설계한 SIMECK-64/128를 Arty S7-50 FPGA 디바이스에 구현한 후, PC와 통신하여 암호화•복호화 동작의 하드웨어 동작을 검증한 결과는 그림 4와 같다. 동일한 키 값에 대해 암호화한 text를 복호화하였을 때 처음 입력한 문장으로 잘 복호화되는 결과를 확인하여 Arty S7-50 FPGA 디바이스에 구현된 SIMECK- 64/128 코어가 올바로 동작함을 확인하였다.

Ⅳ. 결 론

SIMECK-64/128 코어는 긴 평문을 빠르게 처리할 수 있도록 설계하였고, Arty S7-50 FPGA 디바이스에 구현하여 하드웨어 동작을 검증했다. LUT 1,177개와 FF 2,187개의 하드웨어 자원이 사용되었으며, 100 MHz 클록으로 동작을 확인하였다.

References

[1] G. Yang, B. Zhu, V. Suder, M. D. Aagaard, and G. Gong, “The Simeck Family of Lightweight Block Ciphers,” *Proceedings of 17th International Workshop on Cryptographic Hardware and Embedded Systems-CHES 2015*, Saint-Malo, France, Sep. 2015.

[2] S. Limnaios, N. Sklavos and O. Koufopavlou, “Lightweight Efficient Simeck32/64 Crypto-Core Designs and Implementations, for IoT Security,” *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*, 2019, pp. 275-280, doi: 10.1109/VLSI-SoC.2019.8920349.