FPGA 역공학을 활용한 S-box 기반의 DES 공격법 구현

임나리, 최소연, *유호영
충남대학교 전자공학과
e-mail : *nrim.cas@gmail.com, sychoi.cas@gmail.com, hyyoo@cnu.ac.kr*

Implementation of S-box based DES Attack using FPGA Reverse Engineering

Nari Im, Soyeon Choi, and *Hoyoung Yoo
Chungnam National University

Abstract

SRAM-based FGPA is one of the representative FPGA used to implement the digital circuits. Since the SRAM-based FPGA requires the external non-volatile memory to store the bitstream, the bitstream can be extracted from the external non-volatile memory. Recently, reverse engineering methods to recover the circuit information from the bitstream have been proposed. In this paper, we obtained the plaintext of DES by attacking S-box based on the FPGA reverse engineering.

I. 서론

FPGA(Field Programmable Gate Array)는 내부 회로 구성을 변경할 수 있어 임베디드 시스템 분야에서 널리 사용되며, 보편적으로 SRAM 기반의 FPGA가 주로 사용된다 [1]. 그러나 SRAM은 휘발성 메모리이기 때문에 FPGA에 구현된 회로 정보를 저장하기 위해서는 외부에 비 휘발성 메모리가 필요하다. FPGA 회로 정보는 비트스트림 형식으로 외부 비 휘발성 메모리에 저장되며, FPGA의 전원이 인가될 때 외부 비 휘발성 메모리에서 FPGA로 전송된다. 이 과정에서 제 3자에 의해서 비트스트림이 추출되어 수정되거나 훔쳐질 수 있다 [1, 2].

추출된 비트스트림으로부터 내부 회로 정보를 복원하는 역공학 방법들이 제시되었으며 [1, 3], 역공학을 기반으로 암호화 알고리즘을 공격하는 방법 또한 제시되었다 [1]. 본 논문은 FPGA 역공학을 활용하여 DES 암호화 알고리즘을 공격하는 것을 구현하였다.

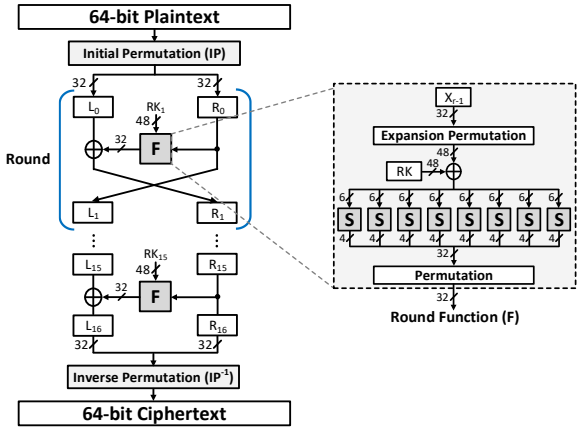

그림 1. DES 암호화 알고리즘

II. 본론

DES는 64비트의 평문(Plaintext)을 64비트 키(Key)를 사용하여 암호문(Ciphertext)으로 생성하는 알고리즘으로 암호화와 복호화에 사용되는 키가 동일한 대칭형 암호이다. DES는 16번의 라운드로 구성되며 암호화 과정은 그림 1과 같고 복호화 과정은 그림 1의 역순으로 수행된다 [4]. 그림 1에 나타난 DES의 라운드 함수(F)는 32-비트를 48-비트로 확장하는 확장 치환(Expansion Permutation), 키를 사용하여 생성된 라운드 키(RK)의 덧셈, 8개의 S-box를 사용하여 48-비트를 32-비트로 대체하는 대체 (Substitution, S), 32-비트의 치환(Permutation)으로 구성

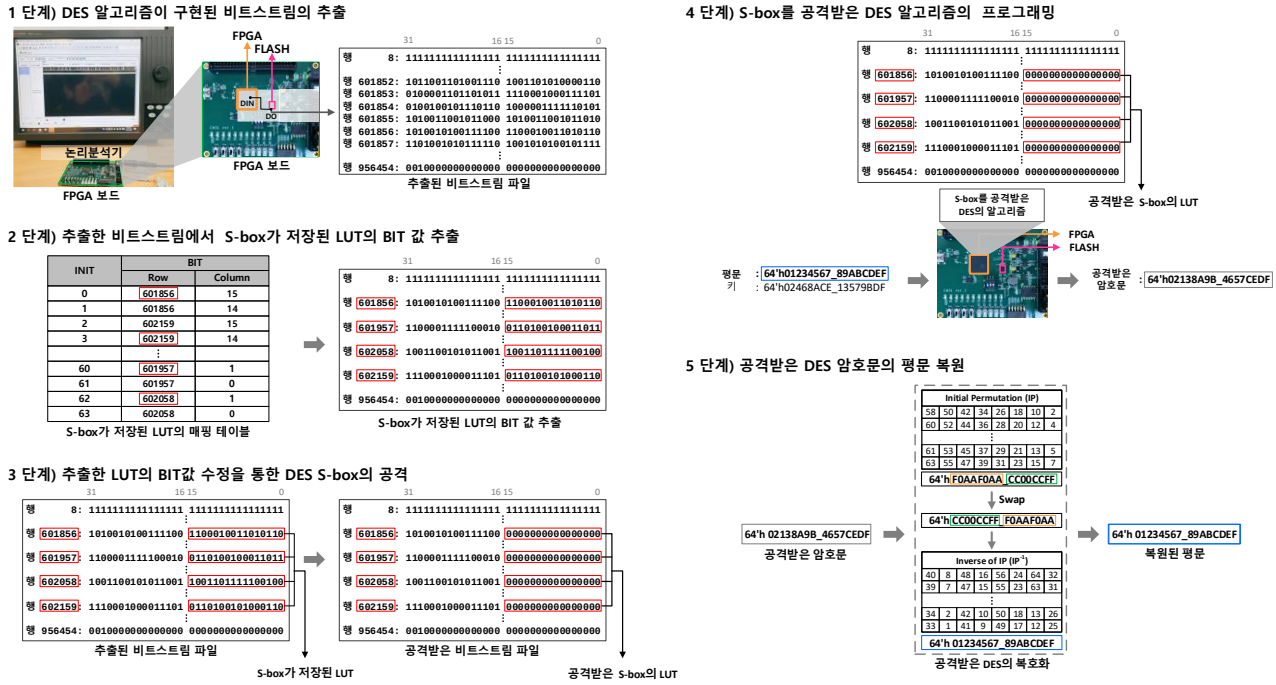

그림 2. FPGA 역공학을 활용한 S-box 기반의 DES 공격 과정

된다. 대체에 사용되는 S-box는 6-비트 입력에 대한 4-비트 출력 값을 저장하며, 8개의 서로 다른 값의 S-box를 사용하여 대체를 수행한다 [4].

DES 알고리즘을 FPGA에 구현할 때 대체에 사용되는 S-box를 LUT(Look-Up Table)에 저장되도록 구현할 수 있다. 6-입력 LUT에 S-box를 구현하면 S-box의 출력 1-비트 값이 하나의 LUT에 저장된다. 따라서 하나의 S-box를 LUT에 구현하기 위하여 6-입력 LUT 4개가 사용되고, 8개의 S-box를 구현하기 위하여 32개의 LUT이 사용된다. S-box가 LUT에 구현된 DES 알고리즘을 LUT의 역공학을 활용하여 공격할 수 있다.

S-box 공격에 사용되는 LUT의 역공학은 매핑테이블을 생성하고 [1], 매핑테이블에 나타난 LUT의 비트스트림 값(BIT) 순서대로 비트를 배열하여 LUT을 복원하는 과정으로 이루어진다. 매핑테이블은 LUT의 초기 값(INIT)이 64'h00000000_00000000인 베릴로그 파일을 one-hot으로 수정한 64개의 베릴로그 파일을 생성하고 비트스트림 파일로 변환한 후, 64개의 비트스트림 파일을 서로 비교하여 생성할 수 있다 [1]. 생성한 매핑테이블은 6-입력 LUT이 저장하고 있는 2^6 비트가 비트스트림에 나타나는 위치 정보를 나타낸다. 매핑테이블을 바탕으로 LUT의 비트스트림 값 순서대로 비트를 배열하여 비트스트림에서 S-box가 저장된 LUT을 찾을 수 있다.

S-box가 구현된 32개의 LUT은 비트스트림에서 추출한 비트를 매핑테이블의 INIT 값 순서대로 재배열한 값과 S-box를 저장한 LUT의 초기 값이 일치하는지 비교하여 찾을 수 있다. 이 과정에서 LUT의 6-비트 입력 전위에 대한 가능성 6!(=720)을 고려해야 한다 [1]. 비트스트림에서 DES의 S-box를 저장하는 32개의 LUT을 모두 찾은 후 LUT 값의 수정을 통해 DES 알고리즘을 공격할 수 있다.

S-box 기반의 DES의 공격법은 대체에 사용되는 8개의 S-box가 입력되는 값에 상관없이 항상 0을 출력하도록 무력화시키는 것이다 [1, 5]. S-box의 출력이 0이 되면 그림 1의 치환의 출력도 0이 되어 라운드 함수(F)의 출력이 0이 된다. 라운드 함수(F)의 출력이 0이 되면 그림 1의 알고리즘에서 XOR의 의미가 없어진다. 따라서 S-box를 공격받은 DES의 암호화 알고리즘은 초기 치환(IP), 15번의 교환(Swap), 역 초기 치환(IP^{-1})으로 나타낼 수 있다. 이 때 15번의 교환은 1번의 교환과 같은 결과를 내므로 초기 치환, 교환, 역 초기 치환으로 단순화될 수 있다. 공격받은 DES의 알고리즘은 키를 사용하지 않게 되므로 키에 대한 정보 없이 암호문을 평문으로 복원 수 있다.

본 논문은 그림 2의 5 단계로 FPGA 역공학을 활용한 S-box 기반의 DES 공격법을 구현하여 키에 대한 정보 없이 평문을 복원했다.

1) DES 알고리즘이 구현된 비트스트림의 추출
2) 추출한 비트스트림에서 S-box가 저장된 LUT의 BIT 값 추출
3) 추출한 LUT의 BIT 값 수정을 통한 DES S-box의 공격

4) S-box를 공격받은 DES 알고리즘의 프로그래밍
5) 공격받은 DES 암호문의 평문 복원

먼저, FPGA에 전원이 인가될 때 외부 비 휘발성 메모리에서 FPGA로 전송되는 DES가 구현된 비트스트림을 추출한다. 추출한 비트스트림에서 LUT의 역공학을 이용하여 S-box가 저장된 32개의 LUT을 찾는다. 비트스트림으로부터 추출한 32개 LUT의 BIT값을 모두 0으로 수정하여 S-box의 출력이 항상 0이 되도록 공격한 DES의 비트스트림을 생성한다. S-box를 공격받은 DES의 암호화 알고리즘을 다시 FGPA의 외부 비 휘발성 메모리에 저장하여 공격받은 암호문을 출력한다. 마지막으로 공격받은 암호문은 공격받은 DES의 알고리즘의 복호화를 통해 평문으로 복원된다.

Ⅲ. 구현

본 논문은 DES를 구현한 비트스트림에서 S-box를 공격하여 평문을 복원하는 과정을 검증하였다. 실험에 사용한 FPGA는 Xilinx Artix-7의 XC7A100T이며 외부 Flash 메모리의 MT25-QL와 연결된 보드를 사용하여 실험을 진행하였다. DES 암호화 알고리즘의 암호문은 UART(Universal Asynchronous Receiver /Transmitter)로 출력되도록 설계하였으며, 논리 분석기를 사용하여 DES 알고리즘이 구현된 비트스트림을 추출하였다.

먼저, 그림 2의 1단계와 같이 논리 분석기로 외부 Flash 메모리의 데이터 출력 핀(DO)에서 FPGA의 데이터 입력 핀(DIN)으로 전송되는 비트스트림을 Flash 메모리의 핀을 측정하여 추출하였다. 추출한 비트스트림 파일에서 LUT의 역공학을 이용하여 S-box를 저장하는 32개의 LUT을 찾고 그림 2의 2단계에 나타난 것과 같이 LUT의 BIT 값을 추출하였다. 다음으로 그림 2의 3단계에서 볼 수 있듯이 S-box를 저장하는 32개 LUT의 BIT 값을 0으로 수정하여 공격받은 DES의 비트스트림 파일을 생성하였다. 다음으로 그림 2의 4단계와 같이 공격받은 DES의 비트스트림 파일을 Vivado Design Suite을 사용하여 외부 Flash 메모리에 프로그램하였으며, UART를 통해 공격받은 암호문을 출력했다. 마지막으로 그림 2의 5단계의 과정과 같이 공격받은 암호문을 초기 치환, 교환, 역 초기 치환을 통해 평문으로 완벽하게 복원하였다.

Ⅳ. 결론 및 향후 연구 방향

본 논문에서 FPGA에 구현된 DES의 S-box를 공격하여 암호화 키에 대한 정보 없이 평문을 복원하였다. 이는 S-box를 포함하는 다른 암호화 알고리즘 또한 공격에 노출될 수 있음을 의미하므로, 암호화 알고리즘의 S-box 공격 가능성 및 대응책에 대한 추가적인 연구가 필요하다.

ACKNOWLEDGEMENTS

이 논문은 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. 2021R1I1A3055806)

참고문헌

[1] P. Swierczynski , M. Fyrbiak, P. Koppe, and C. Paar, "FPGA Trojans through detecting and weakening of cryptographic primitives," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 8, pp. 1236- 1249, Aug. 2015

[2] R. Chakraborty, I. Saha, A. Palchaudhuri and G. Naik, "Hardware Trojan insertion by direct modification of FPGA configuration bitstream", IEEE Des. Test, vol. 30, no. 2, pp. 45-54, Apr. 2013, doi: 10.1109/MDT.2013.2247460.

[3] S. Choi, H. Yoo, "Fast Logic Function Extraction of LUT from Bitstream in Xilinx FPGA", Electronics MDPI, vol.9, no.7, 2020, doi: 10.3390/electronics9071132.

[4] "Data Encryption Standard", FIPGs Pub. 46, NSA, U.S. Dep. Of Commerce, Jan. 1977.

[5] T. Kerins and K. Kursawe, "A cautionary note on weak implementations of block ciphers", Proc. 1st Benelux Workshop Inf. Syst. Security (WISSec), 2006, pp. 12.