Journal of The Institute of Electronics and Information Engineers Vol.58, NO.12, December 2021 https://doi.org/10.5573/ieie.2021.58.12.19
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2021-58-12-3

JTAG 신호분석을 이용한 상용 MCU 해킹 취약성 연구

(A Study on Hacking Vulnerability for Commercial MCUs using JTAG Signal Analysis)

이 건 하*, 공 원 배**, 정 혜 민**, 전 지 원**, 김 동 규**

(Kunha Lee, Wonbae Kong, Hyemin Jung, Jiwon Chun, and Dong Kyue Kim©)

요 약

4차 산업혁명으로 인한 IoT 및 통신 부문의 급격한 개발로 인해 스마트폰 보급이 증가하고 임베디드 기기들의 수요가 증가하고 있다. IoT 기기 및 스마트폰에서 시스템의 디버깅을 위하여 JTAG을 사용하지만, 포렌식과 같은 JTAG의 취약점을 공격하여 시스템 내부 메모리의 데이터를 얻어내는 공격 방법이 있다. JTAG에 대한 공격 방법을 방지하기 위해서 시스템을 제작하는 주요 Vendor들은 Secure JTAG, Protecion mode 등을 보안 방법으로 제시하였다. 본 논문에서는 JTAG의 신호분석을 이용하여 인증 프로토콜을 분석하는 방법과 허가 되지 않은 사용자가 인증을 획득하는 방법을 제안한다. 주요 vendor 중 하나인 STMicro사의 MCU를 대상으로 제안하는 방법을 적용하여 인증 프로토콜과 인증에 사용되는 비밀키 값을 알아내었다. 또한, 허가되지 않은 사용자가 Control Register를 제어하여 메모리 보호 권한을 수정하는 것을 보여 주었다. 이를 통하여 통신 감청이 고려되지 않은 JTAG 통신과 동일한 비밀키를 반복적으로 사용하는 단순한 인증 프로토콜을 사용하는 MCU들의 해킹 취약성을 알린다.

Abstract

Rapid development of IoT and communication technologies after the 4th industrial revolution, increased demand for smartphones and embedded devices. IoT devices and smartphones often use JTAG for system debugging, but some attacks exploit JTAG vulnerabilities such as forensics to obtain data in the system's internal memory. To prevent JTAG attacks, major vendors who manufacture systems suggested secure JTAG and protection mode as security methods. This paper proposes a method to analyze an authentication protocol using JTAG signal analysis and shows unauthorized users can be authenticated As a result, we could find the authentication protocol and the secret key value used for authentication in STMicro MCUs, one of the major vendors. In addition, we have shown that an unauthorized user can modify the memory protection privileges by revaluing the control register. Through this, we notify the hacking vulnerability of MCUs using a simple authentication protocol that repeatedly uses the same key and JTAG communication that does not consider communication interception.

Keywords : JTAG, Secure JTAG, MCU, ARM, Debug, Signal analysis

Ⅰ. 서 론

JTAG(Joint Test Action Group)은 장치의 내부 리소스에 접근 할 수 있는 기능으로 인해 하드웨어 테스트 및 소프트웨어 디버깅에 널리 사용된다. JTAG의 특징 중 하나는 전체 시스템의 작동과 별개이며 작동에 의존하지 않고 대역 외에서 작동한다는 것이다. 또한, 동시에 서비스 중인 장비를 대규모로 중단시킬 수 있는 고유한 기능을 가지고 있다[1]. 하지만 이러한 특징은 악의적인 사용자가 장치의 백도어로 사용하여 장치 해킹의 초석으로 활용될 수 있다.

JTAG의 악의적인 활용은 스마트폰 및 임베디드 기기에 대한 공격에서 주로 나타난다. 스마트폰 및 임베디드 기기는 각 vendor의 공정에 맞는 프로세서가 내장

*정회원, 한양대학교 정보보안학과(Hanyang University Department of Information Security)
**정회원, 한양대학교 융합전자공학부(Hanyang University Department of Electronic Engineering)
© Corresponding Author(E-mail : dqkim@hanyang.ac.kr)
Received ; October 18, 2021 Revised ; November 26, 2021
Accepted ; December 1, 2021

되어있고 이러한 프로세서는 JTAG을 포함하고 있다. JTAG의 boundary scan 기능을 이용하면 프로세서에 내장된 펌웨어 획득, 메모리에 내장된 데이터 획득 등의 공격[2]이 가능하다. 스마트폰 디지털 포렌식 공격의 방법 중 하나는 JTAG의 boundary scan을 사용하여 훼손된 데이터를 복구[3]하는 것이다. 스마트폰 및 임베디드 기기들은 개인정보, 중요 데이터 등이 저장된 경우가 많기 때문에 더 공격에 주의해야 한다.

스마트폰, 임베디드 기기의 JTAG에 대한 해킹 위험성을 방지하기 위하여 Secure JTAG이 제안되었다[4]. 일부 Secure JTAG 기술은 vendor와 사용자의 고유 비밀키 값을 사용하는 challenge/response 인증 방법을 사용한다[5]. 메모리의 접근과 제어를 가능하도록 하는 메모리 보호(memory protection)[6] 기능에 JTAG을 이용하여 접근할 때, Secure JTAG의 인증과정을 통하여 권한을 얻을 수 있다. 이러한 방법으로 vendor가 제공하는 허가된 디버거(debugger)의 JTAG을 통해서만 프로세서를 온전히 관찰하고 제어가 가능하게 한다.

본 논문은 허가된 디버거와 MCU 사이의 JTAG 통신을 분석하여 인증 프로토콜과 사용되는 비밀키를 알아내는 방법을 제안한다. 나아가 동일한 키를 반복적으로 사용하는 단순한 인증 프로토콜을 허가되지 않은 사용자가 해킹한 비밀키와 인증 프로토콜을 이용하여 MCU의 제어 권한을 얻는 방법을 제안한다. 본 논문의 기여는 첫째로 신호분석을 이용하여 쉽게 인증 프로토콜과 비밀키를 알아 낼 수 있다는 것을 보여줌으로써 통신 감청이 고려되지 않은 JTAG의 위험성을 알린다. 둘째로 허가되지 않은 사용자가 프로세서 내부 CR(제어 레지스터, Control Register)을 조작 할 수 있다는 것을 보여줌으로써 재사용 공격이 고려되지 않은 인증 프로토콜의 위험성을 알린다.

신호 분석만을 이용한 본 논문의 방식은 기존의 연구들과는 다르게 비교적 쉽게 아무런 흔적 없이 데이터를 추출할 수 있다. 기존의 JTAG에 대한 Sniff attack, Read-out attack[7]은 공격하기 원하는 주체를 두 개의 공격자 프로세서 사이에 두는 형태로 이루어진다. 공격자가 공격자 프로세서를 직접 제어하거나 버스 마스터 역할을 하여 명령을 직접적으로 내려 데이터를 뽑아낸다. 따라서 공격 이후에 흔적을 남기기 쉬우며, 공격 상황에 제약이 크다.

Ⅱ. JTAG 및 Secure JTAG의 개요

본 장에서는 JTAG의 기본 원리 및 Secure JTAG의 기본 원리를 설명한다. 2.1장에서는 프로세서 내부의 값들을 관찰 및 변경할 수 있는 디버깅 장치인 JTAG[8, 9]의 기본 동작 원리에 대해 설명한다. 2.2장에서는 JTAG을 사용하여 허가되지 않은 사용자가 시스템 내부 값들을 관찰 및 변경하지 못하도록 하는 Secure JTAG의 인증 과정 및 동작원리에 대해 설명한다.

2.1) JTAG 표준 프로토콜(IEEE 1149.1)

JTAG의 구조는 그림 1과 같이 크게 세 가지 부분으로 나뉜다. 데이터의 관찰 및 전송을 담당하는 BSC(Boundary Scan Cell), JTAG의 현재 동작설정을 담당하는 세 종류의 레지스터, JTAG의 실제 동작을 이루어지게 하는 TAPC(Test Access Port Controller)로 이루어져 있다.

JTAG의 동작은 TAPC에 연결된 TMS(Test Mode Select) 신호에 의거하여 IR(Instruction Register)과 DR(Data Register)의 상호작용으로 이루어진다. IR은 현재 JTAG의 동작을 결정해주며 요구되는 명령에 맞게 JTAG 채널을 연결시켜 준다. DR은 IR 상태에서 설정된 명령어의 동작 형태에 따라 필요한 레지스터에 연결되어 데이터를 관찰 및 전송 과정을 담당한다. 데이터를 관찰 및 변경하는 원리는 그림 1과 함께 설명한다.

그림 1의 BSC들은 core logic 내부의 필요한 레지스터 및 메모리에 일대일로 연결이 되어있으며 각 셀에는 연결되어 있는 핀(pin)의 정보와 셀 번호가 지정된다. 원하는 데이터가 저장된 메모리에 접근해서 데이터를 읽으려면 연결된 BSC의 제어가 필요하다. 데이터 읽기/쓰기 동작이 수행될 때 전체 BSC는 현 상태의 데이터들을 모두 캡처한 후 원하는 데이터가 저장된 메모리와 연결된 셀의 번호만큼 shift-register 기능을 수행한다. 이후 연결된 wire를 통해 TDO로 원하는 데이터가 출력되도록 shift-register 기능이 수행된다.

JTAG은 명령어에 따라서 여러 가지 기능이 가능하고 기능에 따라서 필요한 레지스터가 연결되어 수행된다. 이러한 JTAG의 동작은 내부 core logic의 동작과 별개로 BSC에서 캡처 과정을 통해 동작이 이루어지기 때문에 시스템에서 사용하는 데이터에 대한 공격이 가능하다.

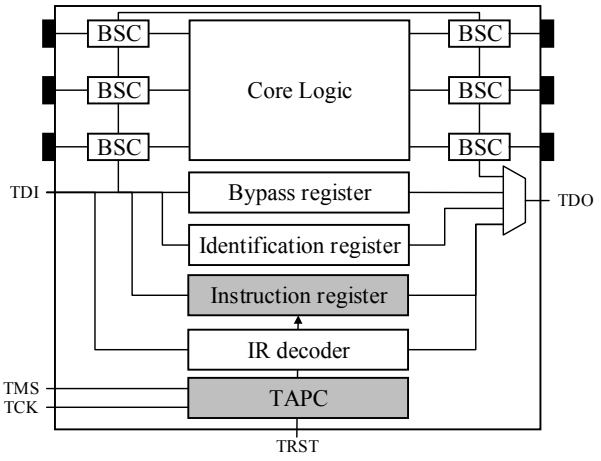

그림 1. JTAG의 구조
Fig. 1. JTAG's structure.

2.2) Secure JTAG의 동작원리

JTAG으로 시스템 내부 데이터에 대한 공격이 가능하기 때문에 그에 대한 해결책 중 하나로 Secure JTAG이 연구되고 있다. Secure JTAG은 각각의 고유한 인증 프로토콜을 통해 허가된 사용자만 JTAG 기능을 사용 가능하게 하여 내부 데이터를 보호한다.

Secure JTAG의 일반적인 구조는 JTAG 통신에 있어서 SoC 자원에 접근 가능한 부분과 접근하지 못하는 부분으로 나뉜다. Secure JTAG의 secure 모드가 활성화되어 있다면, JTAG으로 SoC 자원에 접근하지 못한 상태로 테스트 제어만을 진행할 수 있게 된다. 이때 Secure JTAG이 요구하는 인증 과정을 진행 할 수 있고 인증에 성공하게 되면 SoC 자원에 접근이 가능하게 되는 것이 기본 원리[10, 11]이다.

위에서 설명한 Secure JTAG의 인증과정은 각각 다른 인증 프로토콜을 사용하며 일반적으로는 challenge /response 인증방식을 사용한다. Challenge/response 인증방식은 그림 2와 같이 동작하며 사용자와 인증 주체 간의 비밀 키 연산 및 암호화된 상호작용을 통해 인증이 이루어진다.

Challenge/response 인증 프로토콜은 challenge 키값과 response 키값의 교환 방식이나 사용하는 암호 알고리즘에 따라 다양한 방식이 존재한다. 그 예로는 사용자와 서버가 서로 동일한 비밀키를 사용하는 대칭키 암호를 사용하는 방법과 사용자와 서버가 서로 다른 키와 공개키 암호를 사용하는 방법 그리고 일회용 키를 생성하여 사용하는 방법 등이 있다[12, 13].

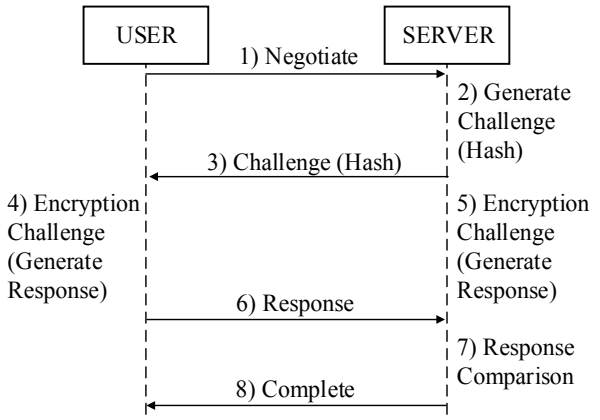

그림 2. Challenge/Response 인증 메커니즘
Fig. 2. Challenge/Response authentication mechanism.

Ⅲ. JTAG을 이용한 시스템 분석 방법

본 장에서는 ADI(ARM Debug Interface)[14]를 사용하는 시스템과 허가된 디버거 사이의 JTAG 통신을 분석하여 시스템 정보를 취득하는 방법을 제안한다. 3.1절에서는 시스템과 허가된 디버거 사이의 JTAG 통신을 로직 분석기를 이용하여 분석하는 방법과 분석을 통해서 알아낼 수 있는 JTAG 정보들에 대해서 설명한다. 3.2절에서는 ADI 프로토콜에 대해 설명하고 신호분석을 통해 알아낸 JTAG 정보를 기반으로 알아 낼 수 있는 ADI 시스템의 정보들에 대해 설명한다.

3.1) 로직 분석기를 사용한 JTAG 분석

허가된 JTAG 디버거를 사용하여 프로세서를 디버깅하거나 내부 메모리 값을 읽고 쓰는 동작을 수행할 때 디버거 내부의 JTAG 동작은 사용자에게 공개되지 않는다. 하지만 JTAG 통신 케이블에 연장 핀을 연결하면 통신 신호와 로직 분석기를 사용하여 동작과정을 알아낼 수 있다.

그림 3은 JTAG의 상태도이고 그림 4는 JTAG 동작과정을 로직 분석기로 검출한 결과이다. 이는 2.1에서 설명한 JTAG 동작의 IR, DR의 설명을 바탕으로 그림 3의 상태도와 그림 4의 관찰된 값을 비교할 수 있다. 그림 3의 상태도의 IR, DR은 현재 받아들일 데이터가 둘 중에 어떠한 것인지를 뜻하며 그림 4의 JTAG이 IR 상태이므로 JTAG이 명령어 설정 단계에 있는 것을 확인할 수 있다. 같은 방법으로 JTAG의 모든 상태와 각 상태에서 사용하는 데이터들을 확인 할 수 있다.

JTAG 신호를 검출 및 분석하여 얻을 수 있는 정보들은 다음과 같다. JTAG의 명령어, 레지스터 및 메모

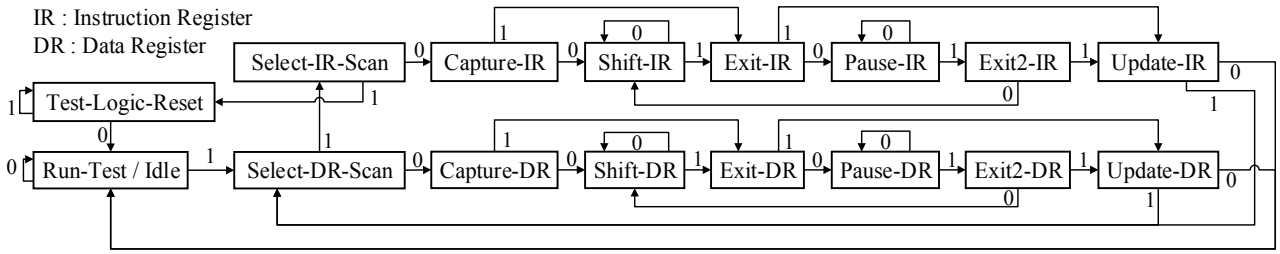

그림 3. JTAG의 동작 상태도
Fig. 3. JTAG state diagram.

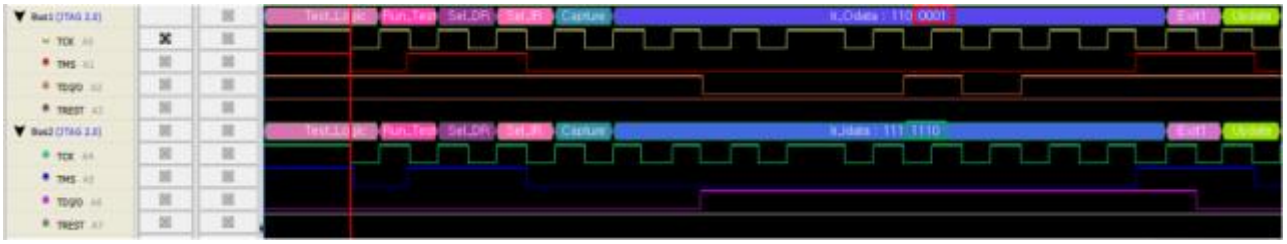

그림 4. 로직 분석기 분석 결과
Fig. 4. Logic analyzer analysis results.

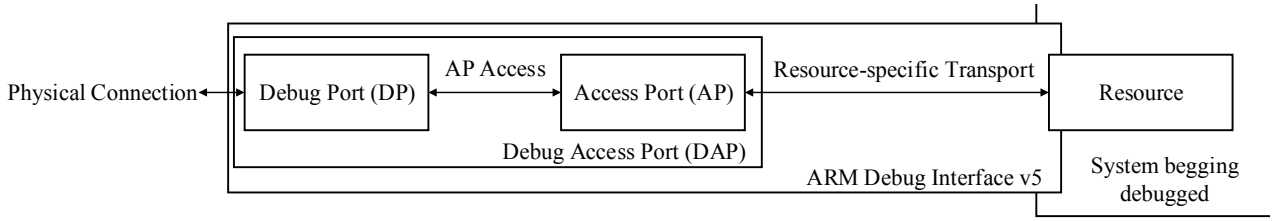

그림 5. ARM Debug Interface의 일반적인 구조
Fig. 5. ARM Debug Interface Structure.

리 주소, 읽기/쓰기 데이터 등의 정보이다. 위의 정보들을 이용하면 고성능의 디버거로도 확인하지 못 하는 프로세서가 의도적으로 숨겨놓은 동작 또한 분석 가능하다. 신호 분석을 통해 상용 MCU 내부의 숨겨진 동작을 실제로 분석하는 것은 4장에서 설명한다.

3.2) ADI 분석 방법

ARM core를 기반으로 하는 시스템은 모두 ARM cpu 디버그 프로토콜을 따르며 이는 ADI 구조로 설계되어 있다. 그림 5는 ADI의 구조를 나타낸 그림이고 논리적으로 ADI는 DP(Debug Port), AP(Access Port), DAP(Debug Access Port)와 같이 세 가지 부분으로 구성된다.

그림 5에서 보이는 것과 같이 ADI는 JTAG이 동작할 때 DAP에 의해 DP와 AP가 서로 상호작용을 하면서 데이터를 주고받게 된다. DP의 역할은 JTAG port를 통해 들어오는 데이터에 대하여 식별을 하고 DP와 AP의 접근 여부를 생성한다. AP의 역할은 디버그 구성 요소간의 인터페이스를 제공한다. AP에는 프로세스의 각 자원에 연결된 주소가 존재하여 실제 메모리 지정(memory mapped) 자원에 접근하여 데이터의 읽기/쓰기 기능을 수행한다.

DP와 AP간에 상호작용하여 자원에 접근하는 과정을 간단히 설명하면 다음과 같다. DP에 접근하여 디버그 상태를 활성화하고 DP를 통해 접근할 AP를 수신한다. 수신된 AP를 통해 원하는 자원의 주소로 접근한 뒤, 접근한 자원에 데이터 읽기/쓰기를 한다. JTAG의 내부동작 과정을 통해 프로세스 내부의 특정 자원에 접근하여 데이터를 읽기/쓰기 하는 과정을 분석하기 위해서는 DP의 읽기 버퍼 레지스터와 AP의 TAR(Transfer Address Register)를 분석해야 한다.

AP의 TAR은 32bit or 64bit로 이루어진 액세스할 메모리의 주소를 저장하는 레지스터이다. 또한 DP의 읽기 버퍼 레지스터는 JTAG 동작 과정 중 TAR 레지스터에

의해 접근한 주소에 읽기/쓰기 한 데이터의 값이 저장되는 레지스터이다. 따라서 위와 같은 ADI 내부 동작과정을 통해 접근하는 자원 주소와 데이터를 분석함으로써 숨겨놓은 디버깅 명령어의 동작을 읽어낼 수 있다[15].

Ⅳ. 상용 MCU 대상 분석 결과

본 장에서는 3장의 분석 방법을 토대로 디버거를 사용한 상용 MCU의 CR 제어과정을 로직 분석기로 분석한 결과를 보여준다. 또한 분석으로 알아낸 정보를 이용하여 허가되지 않은 사용자가 상용 MCU의 CR을 제어하는 것을 보인다. 4.1절에서는 STMicro사의 MCU와 J-Link 디버거로 허가된 사용자가 메모리 보호를 해제하는 것을 로직 분석기를 이용한 신호 분석으로 인증 프로토콜을 알아내는 과정과 결과를 보인다. 4.2절에서는 4.1절에서 얻어낸 정보를 바탕으로 JTAG 장치를 이용해 허가되지 않은 사용자가 MCU의 option bytes를 수정하는 과정과 결과를 보인다.

4.1) CR 제어 과정 분석 결과

MCU의 CR 제어는 허가된 JTAG 디버거만 가능하다. MCU 제작 vendor에서는 허가된 디버거와 특정 소프트웨어로만 CR 제어가 가능하도록 하고 동작과정은 비공개하는 경우가 많다. 예로 STMicro의 MCU에는 메모리 보호 기능이 있다. 메모리 보호 기능의 설정/해제는 일반 JTAG 디버거로는 불가능하며, 허가된 JTAG 디버거만이 인증과정을 거쳐서 설정/해제가 가능하다. 이때의 구체적인 인증과정은 공개되어 있지 않다. 본 절에서는 허가된 JTAG과 MCU사이의 인증과정을 관찰하고 분석하는 방법을 설명한다.

로직 분석기를 이용하여 JTAG Interface 입출력 값을 읽고 이를 통해 JTAG 프로토콜의 IR, DR값을 설정하거나 확인하는 방법을 3.1에서 설명하였다. IR, DR의 제어를 통하여 시스템의 ADI를 제어하도록 되어있기 때문에 JTAG 신호 분석을 통하여 관찰되는 IR, DR의 값으로 ADI의 동작을 알아 낼 수 있다. 표 1은 ADI의 TAR을 읽는 과정에 해당하는 IR과 DR의 값을 나타낸 것이다. TAR을 선택하기 위해서 DR을 SELECT 레지스터에 연결하여 TAR의 주소값을 받아와 이후 TAR에 연결하여 저장된 주소값을 읽는다. 위와 같은 동작 과정의 ADI의 값을 표 1에 표기하였고 위를 통해 TAR에 저장된 키의 주소값 알 수 있다.

JTAG 신호분석을 이용한 ADI의 동작을 알아낼 수 있고 ADI의 동작방식은 3.2절에서 설명하였다. AP와 DP의 register에 따라서 접근하는 ADI 자원을 확인 할 수 있고 그 중 TAR을 확인하여 MCU 내부의 어떠한 레지스터에 접근하는지 확인할 수 있다. 그리고 이후의 RDBUFF를 통하여 접근하는 레지스터의 값을 확인할 수 있다. 표 2는 키의 값을 확인할 수 있는 TAR과 RDBUFF 값을 보여준다. 먼저 TAR에 키의 주소값이 설정되면 이후에 RDBUFF에 저장되는 값이 키 값에 해당된다. 이와 같은 방식을 통하여 인증 프로토콜에 사용되는 데이터들과 프로토콜을 알아 낼 수 있다.

표 1. JTAG 신호 분석을 통한 ADI 동작 분석
Table 1. ADI analysis through JTAG signal analysis.

TDI		동작
IR	DR	
0xA	0x00000000 + 3'b100	DP TAR 선택
0xB	0x40022004 + 3'b010	AP TAR 읽기

표 2. ADI 동작 분석을 통한 데이터 분석
Table 2. Data analysis through ADI analysis.

	AP	DP
ADI reigster	TAR	RDBUFF
value	0x4002 2004	0x4567 0123
mean	Key1 address	Key1 data

CR을 제어하는 전체 동작과정의 신호를 추출하여 동작을 분석하였다. 분석 결과 STMicro MCU의 option bytes 수정 권한을 획득하는 과정과 수정하는 과정을 알아 낼 수 있었고 이는 각각 그림 6, 그림 7과 같다. 이 때 각 과정에서는 디버거와 MCU의 두 개의 키 값(키1, 키2)을 알아 낼 수 있었다.

메모리 보호를 수정하기 위한 첫 번째 과정으로 수정 권한을 획득하는 과정은 그림 6과 같다. 수정 권한 획득을 위해서는 MCU 메모리 내의 키 레지스터 공간에 키값이 올바르게 작성되어야 한다. 과정 1), 2)는 키1, 키2를 MCU 메모리 내의 키 레지스터에 작성하는 과정이며, 과정 3), 4)는 option bytes 수정 권한 획득을 위한 추가 과정을 나타낸다. 해당 키 입력 과정과 같이 부분적 Secure JTAG 방법으로 메모리 접근 권한 획득을 성공적으로 완료하면 option bytes 수정이 가능해진다.

그림 7은 분석 결과 알아낸 두 번째 과정으로 권한 획득 이후 option bytes 기존 값을 삭제한 후, 원하는 값으로 수정하는 과정이다. 그림 8은 플래시 메모리의 프로그래밍 동작을 관리하는 CR 구조이며, 해당 레지

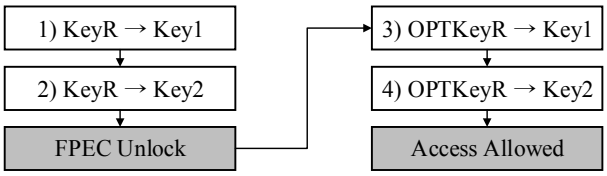

그림 6. Option bytes 권한 획득 과정
Fig. 6. Option Bytes Permission Acquisition Process.

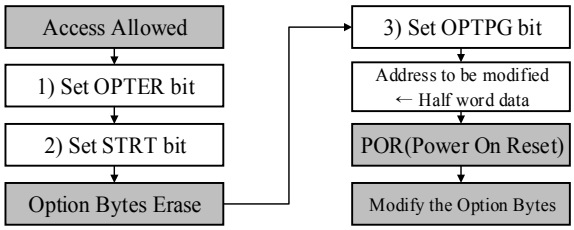

그림 7. Option bytes 수정 과정
Fig. 7. Option Bytes Modification Process.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	~		8	7	6	5	4	3	~					0
Reserved		~		Res.	LOCK	STRT	OPTER	OPTPG	Res.	~					PG
					rw	rw	rw	rw							rw

그림 8. Flash Memory Control Register 구조
Fig. 8. Flash Memory Control Register Structure.

스터의 수정을 통해 두 번째 과정이 진행된다. 그림 7의 과정 1), 2)는 option bytes 기존 값을 삭제하는 과정이며, 3)은 option bytes 레지스터를 수정하는 과정이다. 이후 사용자가 option bytes 레지스터에 메모리 보호를 해제하기 위한 16-bit 크기의 특정 값을 작성한 후 전원을 재인가하면 정상적으로 해제된다.

4.2) Option Bytes 조작 실험 결과

STM32F103RCT6 MCU는 SEGGER사의 J-Link 디버거를 사용하면 소프트웨어의 명령어 기능을 이용하여 원하는 메모리 주소에 데이터를 읽기/쓰기 할 수 있다. 4,1절의 분석 결과로 STMicro 메모리 보호 프로토콜과 해당 프로토콜의 과정에서 필요한 키를 얻은 상태이기 때문에 해당 디버거를 사용하여 직접 조작이 가능하다. 본 논문은 J-Link 소프트웨어를 사용하여 허가되지 않은 사용자가 MCU의 CR을 조작 하였고 조작을 통해 메모리 보호가 해제되는 것을 확인 할 수 있었다. 다음의 그림 9, 10은 조작 과정과 결과를 보여준다.

그림 9은 메모리 보호를 설정한 MCU에서 JTAG을 이용해 메모리를 읽는 명령어를 통해 플래시 메모리의

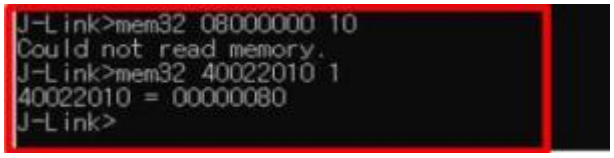

그림 9. Memory Protection 실험 결과
Fig. 9. Memory Protection Experiment Result.

그림 10. Option Bytes 조작 실험 결과
Fig. 10. Option Bytes Modification Experiment Result.

데이터를 읽는 명령어를 실행한 결과이다. 하지만 메모리 보호를 적용했기 때문에 메인 플레시 메모리의 데이터는 읽을 수 없는 것을 확인할 수 있다. 따라서 4.1절의 신호 분석을 바탕으로 얻어낸 결과인 그림 6, 7의 과정을 통해서 메모리 보호를 해제할 수 있다. TAR, 읽기 버퍼 레지스터의 분석으로 얻어낸 option bytes의 주소와 키1, 키2의 정확한 값으로 메모리 보호를 해제할 수 있다. 그림 10에서 허가되지 않은 사용자가 메모리 보호를 해제하고 그 결과 정상적으로 메모리 읽기가 가능해진 것을 확인 할 수 있다.

V. 결 론

본 논문은 허가된 디버거와 MCU사이의 JTAG 통신을 분석하는 방법과 분석된 결과를 이용하여 인증에 필요한 정보를 탈취하는 방법을 제시하였다. 실제 상용 MCU 칩인 STMicro 사의 STM32F103RCT6 제품과 SEGGER 사의 J-Link 디버거 사이의 JTAG 통신을 로직 분석기를 이용하여 분석하였고 분석 결과 STMicro 사의 MCU 칩 내부 CR을 조작하기 위해 필요한 인증 프로토콜과 비밀 키를 알아낼 수 있었다. 이를 통하여 감청이 고려되지 않은 통신 채널인 JTAG의 위험성을 경고하였다. 나아가 알아낸 인증 프로토콜과 비밀 키를 이용하여 허가되지 않은 디버거가 MCU의 제어 권한을 탈취하는 것을 실제 STM32F103RCT6 제품 내부의 CR을 조작하여 메모리 보호 기능을 해제하는 것으로 보였다. 이는 부분적으로 Secure JTAG이 적용되어 사용자 인증 프로토콜을 갖춘 상용 MCU 제품을 공격하여 권한을 탈취 한 것이다. 실험한 STMicro사의 제품 외에도 일반적인 MCU 제품 대부분이 비슷한 방식을

사용한다. 이러한 반복적으로 동일한 비밀 키를 사용하는 인증 프로토콜이 재사용 공격에 취약하다는 것을 보이며 경고하였다.

REFERENCES

[1] “Standard for Test Access Port and Boundary-Scan Architecture”, IEEE 3 Park Avenue New York, NY 10016-5997 USA, 13 May 2013
[2] Kurt Rosenfeld, Ramesh Karri,“Attacks and Defenses for JTAG”, IEEE CS and the IEEE CASS, 2010
[3] George Grispos, William Bradley Glisson, Tim Storer, “Recovering Residual Forensic Data from Smartphone Interactions with Cloud Storage Providers” in The Cloud Security Ecosystem, Pages 347-382, 2015
[4] R.F. Buskey and B.B. Frosik. “Protected JTAG”, International Conference on Parallel Processing Workshops, 2006., pages 8 pp. - 414, 2006.
[5] NXP, AN12419, “Secure JTAG for i.MXRT10xx” Rev. 1, November 2019
[6] Emett Witchel, Josh Cates, and Krste Asanovic, “Mondrian Memory Protection”, MIT Laboratory for Computer Science, Cambridge, MA 02139
[7] Kurt Rosenfeld, Ramesh Karri,“Attacks and Defenses for JTAG”, IEEE CS and the IEEE CASS, 2010
[8] Heo Kyungchul, Park Hyungbae, Jung Seungpyo, Park Joosung, “Modified JTAG design for debugger of RISC processor”, Journal of the Electronic Engineering Association, Volume 48, SD, No. 7, pages 65-75, 2011
[9] Yoon Yeon-sang, Kim Seung-yeol, Kwon Soon-yeol, Park Jin-seop, Kim Yong-dae, Yoo Young-gap, "PIDM for Improving the Performance of JTAG-based Tests," Journal of the Electronic Engineering Society, Volume 41, SD, No. 8, 697-704, 2004
[10] Emanuele Valea1, Mathieu Da Silva1, Marie-Lise Flottes1, Giorgio Di Natale2, Bruno Rouzeyre1 “Encryption-Based Secure JTAG”, 22nd International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2019
[11] Keunyoung Park, Sang Guun Yoo, Taejun Kim , Juho Kim, “JTAG Security System Based on Credentials”, Springer Science+Business Media, LLC , 2010
[12] “Secure JTAG Implementation Using Schnorr Protocol”, Springer Science+Business Media, New York, 2013
[13] “Multi-Level Secure JTAG Architecture“, IEEE 17th International On-Line Testing Symposium, 2011
[14] ARM, IHI 0031C, “ARM Debug Interface Architecture Specification ADIv5.0 to ADIv5.2” 2013
[15] Zhenyu Ning, Fengwei Zhang, “Understanding the Security of ARM Debugging Features”, Department of Computer Science Wayne State University, 2019

저 자 소 개

이 건 하(정회원)
2020년 호서대학교 정보통신공학과 학사 졸업

<주관심분야 : 보안 SoC, 보안 프로세서, 암호 가속기, Secure JTAG>

공 원 배(정회원)
2015년 한양대학교 융합전자공학부 학사 졸업.
2019년 한양대학교 융합전자공학과 석·박사 수료.

<주관심분야 : 보안 SoC, 보안 프로세서, 암호 가속기>

정 혜 민(정회원)
2021년 한양대학교 융합전자공학부 학사 졸업.

<주관심분야 : 보안 SoC, 보안 프로세서, 암호 가속기, 메모리 암호화>

전 지 원(정회원)
2021년 단국대학교 전자전기공학부 학사 졸업.

<주관심분야 : 보안 SoC, 보안 프로세서, Secure JTAG, 메모리 암호화>

김 동 규(정회원)
1992년 서울대학교 컴퓨터공학 학사 졸업.
1994년 서울대학교 컴퓨터공학 석사 졸업.
1999년 서울대학교 컴퓨터공학 박사 졸업.
2000년~2005년 부산대학교 조교수
2006년~현재 한양대학교 융합전자공학부 교수
<주관심분야 : 보안 SoC, 보안 프로세서, 암호 가속기, 정보 보안 시스템, 차량용 기능안전 및 보안>