석사학위논문

저사양 마이크로 컨트롤러의 전력 분석을 기반으로 한 ASIC 내성 작업 증명

2021년

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

김 현 준

석 사 학 위 논 문
지도교수 서화정

저사양 마이크로 컨트롤러의 전력 분석을 기반으로 한 ASIC 내성 작업 증명

ASIC-Resistant Proof of Work Based on Power Analysis of Low-End Microcontrollers

2020년 12월 일

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

김 현 준

석 사 학 위 논 문
지도교수 서화정

저사양 마이크로 컨트롤러의 전력 분석을 기반으로 한 ASIC 내성 작업 증명

ASIC-Resistant Proof of Work Based on Power Analysis of Low-End Microcontrollers

위 논문을 공학 석사학위 논문으로 제출함

2020년 12월 일

한 성 대 학 교 대 학 원

I T 융 합 공 학 과

I T 융 합 공 학 전 공

김 현 준

김현준의 공학 석사학위 논문을 인준함

2020년 12월 일

심사위원장 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

국 문 초 록

저사양 마이크로 컨트롤러의 전력 분석을 기반으로 한 ASIC 내성 작업 증명

한 성 대 학 교 대 학 원
I T 융 합 공 학 과
I T 융 합 공 학 전 공
김 현 준

ASIC (Application-Specific Integrated Circuit) 저항 작업 증명 (Proof of Work, PoW)은 현대 암호 화폐에 널리 사용되고 있는 합의 방식이다. ASIC에서 비효율적인 설계를 목적으로 ASIC 저항 PoW는 특수한 함수를 사용한다. 본 논문에서는 저가형 마이크로 컨트롤러를 위한 새로운 ASIC 내성 PoW를 소개한다. 마이크로 컨트롤러에서 특정 입력 값에 대한 암호화 묘듈 실행하여 측정 된 전력 파형을 활용한다. 이 전력 파형은 입력 값에 따라 항상 일정한 특징을 갖으며 ASIC에 의해 예측 및 계산되지 않는다. 따라서 제안 기법은 마이크로 컨트롤러에서 작동 시킬 수 있으며 ASIC 저항을 달성한다.

【주요어】 ASIC저항, 작업증명, 전력 분석, 마이크로컨트롤러, 블록체인

목 차

표 목 차

그 림 목 차

알 고 리 즘 목 차

제 1 장 서론

Bitcoin, Ethereum 및 Monero와 같은 잘 알려진 현대 암호 화폐는 블록의 무결성을 보장하기 위해 작업증명(Proof of Work, PoW) 합의 알고리즘을 사용한다. 작업증명은 네트워크의 노드에서 어려운 수학 문제를 풀고 블록을 생성하고 다른 노드에서 검증을 받는다. 이러한 블록을 생성하는 과정을 채굴이라고 하며 채굴의 보상으로 암호 화폐를 얻는다. 비트코인은 어려운 문제를 해시 연산을 사용한다. 그러나 해시 연산은 ASIC(Application Specific Integrated Circuit)에서 매우 효율적으로 계산 된다. ASIC의 효율적인 해시 연산은 채굴을 독점하여 블록체인 네트워크(i.e. 비트코인)의 탈중앙화를 위협하게 된다. ASIC 저항 알고리즘 기반의 PoW는 ASIC 장치의 독점을 막는 목적으로 개발되었다. ASIC은 특정 목적을 위해 설계된 칩이다. ASIC 저항 알고리즘은 불규칙한 패턴을 사용하여 ASIC 장치가 비효율적으로 설계 되도록 한다.

본 논문에서는 ASIC 및 FPGA와 같은 특수 채굴 장치로 인한 채굴 독점을 방지하는 새로운 기술을 제안한다. 제안 방법에서는 작업증명에 마이크로 컨트롤러에서 암호화 작업을 사용한다. 마이크로 컨트롤러는 암호화 모듈 실행 중에 생성된 전력 파형은 입력 매개 변수에 따라 다른 값을 나타낸다. 이 특징을 사용한 PoW 합의 과정으로 ASIC 저항을 달성한다. 제안기법의 기여는 아래와 같다.

전력 분석을 기반의 새로운 ASIC 내성 PoW 이전의 ASIC 내성 PoW 다중 해시 PoW, 메모리 하드(Memory Hard) PoW 및 프로그래밍 PoW 방식과 다른 전력 분석을 기반으로 한 최초의 ASIC 내성 PoW이다. 마이크로 컨트롤러에서 암호화 연산(e.g. AES) 중 생성되는 전력 파형이 작업증명에 사용된다. 전력 파형은 사용 마이크로 컨트롤러와 암호 연산의 입력값에 따라 다르므로 ASIC 및 FPGA에서 모방할 수 없다. 이러한 이유로 제안 기법은 ASIC 내성을 달성한다.

낮은 비용과 낮은 전력 소비 기존의 ASIC 저항 알고리즘은 CPU와 GPU를 사용하는 상용 컴퓨터를 대상으로 한다. 제안 기법은 IoT에서 사용되는 효율적인 저가형 마이크로 컨트롤러를 대상으로 한다. 낮은 비용과 낮은 전력 소비는 채굴자의 접근성을 낮추어 가상 화폐의 분산화에 기여할 수 있다.

잡음 제거를 위한 후처리 수집되는 전력 파형은 잡음를 포함하고 있다. 작은 잡음는 암호화 작업으로 생성되는 전력 파형의 결과를 크게 바꿀 수 있다. 이 잡음를 필터링하기 위해 FFT(Fast Fourier Transform)를 수행한다. 이 방법은 고주파수를 효율적으로 제거한다. 후 처리 후로 필터링 된 정보가 제안한 PoW에 사용된다.

다양한 블록 암호를 기반한 새로운 PoW에 대한 심층 분석 마이크로 컨트롤러에서 다양한 블록 암호를 사용하여 실험을 수행한다. 결과적으로 제안하는 새로운 PoW 방식이 모든 블록 암호에서 적용될 수 있다는 것을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서 기존연구인 PoW 알고리즘과 ASIC 저항 PoW 알고리즘에 대하여 설명한다. 3장에서는 제안기법의 핵심 아이디어인 전력 분석기반 코드 분석 대한 설명과 전력 분석기반 코드 탐지에 대한 새로운 기법을 제안한다. 그리고 4장에서 본 논문의 제안 기법인 저 사향 마이크로 컨트롤러의 전력 분석을 기반으로 한 ASIC 내성 작업 증명에 대하여 설명한다. 마지막으로 5장에서 결론과 향후 연구 방향에 대하여 논한다.

제 2 장 ASIC 저항 PoW 알고리즘

제 1 절 작업증명(Proof-of Work)

PoW 기반 응용 프로그램들은 여러 문제를 위해 개발되었다. Cynthia D work과 Moni Naor의 PoW 기법은 메시지의 일부 함수를 계산하도록 발신자에게 요청하여 정크 메일을 차단하는 데 사용되었다. 그리고 PoW 기법은 속임수 없는 벤치마크, 미터링 웹 사이트 접근, 복권 체계 및 PoW 퍼즐에 적용되었다. PoW 알고리즘은 1993 년 Cynthia와 Moni가 최초로 제안하고 이후 2009 년에 최초의 암호 화폐인 비트코인에 적용되었다.

성능과 보안을 모두 향상하기 위해 최신 암호 화폐에 여러 해시 함수가 사용된다. 가장 인기 있고 성공적인 암호 화폐 중 하나이며 하나인 비트코인은 1997년 해시 캐시의 아이디어를 기반으로 한다. 해시 캐시의 기본 기능은 해시 연산이다. 비트코인에 새 블록을 추가하려면 SHA-256을 사용하는 값이 일정 난이도로 생성된 목표 값보다 작아야한다. 여기서 SHA-256은 2001 년 NSA에서 설계하고 NIST에서 U.S FIPS로 선택하였다. SHA-256은 비트코인 및 비트코인 캐시에서 사용되는 256 비트 출력을 생성한다. Ethash는 이더리움 합의 프로토콜의 해시 함수이다. Thehash 함수는 Keccak, Hashimoto 및 Dagger 알고리즘의 조합으로 ASIC 내성을 보장한다. Scrypt는 다른 것보다 많은 양의 메모리를 사용하는 암호 기반 키 파생 함수이다. 메모리 하드는 ASIC 내성 기능을 보장한다. 해시 함수는 라이트 코인에서 사용된다. 무작위 메모리 접근 기반 프로토콜 Cryptonote는 Monero의 CryptoNightPoW 알고리즘에서 사용된다. ASIC 내성 PoW에 대한 자세한 내용은 2절에서 다룬다. hashcash 기반 PoW는 효율적이고 모든 플랫폼에서 구현하기 쉽다. 그러나 이 방식은 유효한 블록을 생성하기 위해 엄청난 계산 능력이 필요하다. 비트코인 채굴자들은 세계 전기의 0.33 %를 소비한다. 낭비되는 전기를 절약하기

위해 PoW를 통한 유용한 결과의 PoW 측면이 제시되었다(예 : Bread Pudding Protocols). 하나의 예는 과학적 문제를 해결하는 것으로 2013 년에 Cunning-ham 소수 체인에 대한 PoW 프로토콜이 제안되었다. 그리고 PoW 프로토콜은 DNA 및 RNA 시퀀싱과 같은 다른 문제를 위해 설계되었다. 또한, MicroMint 및 데이터 처리에도 많은 연구가 이루어졌다. PoW 프로토콜에서 채굴자는 먼저 문제를 해결한 다음 새 블록을 브로드 캐스트한다. 블록이 유효하면 채굴자는 일정한 보상을 받게 되며, 위의 원칙에 따라 프로토콜은 대규모 분산 네트워크를 신뢰하지 못하도록 구성한다. 그러나 예상치 못한 상황으로 인해 문제가 발생할 수 있다. 이 문제를 해결하기 위해 포크 블록을 선택하는 특정 기준이 있다. 먼저 가장 오래된 포크 블록이 선택된다. 둘째, 가장 긴 포크 블록이 선택된다. 셋째, 가장 많은 컴퓨팅 전력을 사용하는 체인을 선택한다. PoW가 널리 사용됨에 따라 많은 공격이 수행되었다. PoW 프로토콜을 사용하는 블록 체인에 대한 가장 잘 알려진 공격은 51 % 공격이다. 이 공격은 악의적인 사용자가 네트워크 컴퓨팅 성능의 절반 이상을 제어할 때 방아쇠 될 수 있다. 이 공격은 2018 년에 BTC Gold 및 ZenCash에서 실제로 발생했다. 또한, 이기적인 채굴 공격, 네트워크 수준 공격, 풀 관련 공격, 골드 핑거 공격과 같은 PoW 프로토콜에 대한 수많은 공격이 조사되었다. 위의 PoW 프로토콜을 사용하여 블록체인 기술을 탐색함에 따라 많은 새로운 공격이 발생하였다. 본 논문에서는 저가형 사물 인터넷을 위한 ASIC 내성 PoW에 중점을 둔다.

제 2 절 ASIC 저항 PoW 알고리즘

최근에는 PoW 계산을 위해 설계된 ASIC (Application-Specific Integrated Circuits)이 합의를 지배하여 블록체인의 탈중앙화를 위협하고 있으며, ASIC 접근을 방지하기 위해 ASIC 내성 PoW 알고리즘이 도입되었다. 대표적으로 다중 해싱 PoW, 메모리 하드 PoW, 프로그래밍 방식 PoW가 있다.

다중 해싱 PoW는 PoW 알고리즘의 약점 중 하나인 단일 해싱 함수에 대한 종속성을 해결하기 위해 연속단계로 함께 연결된 다중 해싱 함수를 사용한다. 여기에는 Quarkcoin1 및 LYRA2RE2가 포함된다. 그러나 다중 해시 PoW 메커니즘은 ASIC 기반 채굴을 방지하기에 충분하지 않다. 다중 해시 PoW ASIC 저항성은 FPGA (Field-Programmable Gate Ar-ray)에서 다중 해시 PoW를 구현하여 분석되었다. 그 결과 다중 해시 pow는 ASIC를 차단할 수 없는 PoW 메커니즘과 유사한 수준의 ASIC 저항을 보여주었다.

메모리 하드 PoW는 증명자가 증명을 위해 많은 메모리 용량을 사용하고 검증자가 작업을 확인하는 데 적은 양의 메모리 공간과 시간을 사용하는 PoW 알고리즘이다. ASIC에서 대량의 메모리를 늘리는 것은 비효율적이며 채굴자는 메모리의 양을 늘리면 되므로 계산 속도를 높이기 위해 ASIC을 사용하는 것은 의미가 없다. Ethereum의 Ethash3 및 CryptNight 가 여기에 해당한다. 그리고 메모리를 사용한 방식인 메모리 바운드 PoW는 큰 난수 테이블에서 임의 경로를 통과하여 다중 메모리 접근을 만든다. 메모리 하드와 유사한 메모리 바인딩 된 PoW는 암호화 해싱에서 메모리 접근 형태를 보일 위험이 있으며 네트워크를 통해 증명자와 검증자 사이에 큰 임의의 테이블을 보내야 하는 단점이 있다. 그러나 메모리 하드 PoW 알고리즘을 대상으로 하는 ASIC 시스템이 도입되어 ASIC 저항성을 잃었다.

프로그래밍 방식 PoW는 임의의 코드를 실행하여 계산의 다양성을 높이는

새로운 방법이다. 무작위로 생성된 프로그램이 작업의 일부로 포함한다. ASIC에서 각 계산 작업을 대상으로 하는 특수 하드웨어 모듈을 구축하는 것은 비현실적이다. 따라서 ASIC는 임의의 프로그램에서 범용 컴퓨터보다 덜 사용된다. RandomX는 이 방법을 활용하였다.

제 3 장 전력 분석 기반 코드 탐지

1996년 폴 코처(Paul C. Kocher)가 제시한 타이밍 공격을 시작으로 장치 내의 암호 모듈이 동작하면서 물리적인 정보인 연산의 소요 시간, 소비전력 및 전자파와 같은 부채널 정보를 이용하는 공격을 말한다. 전력 분석은 부채널 분석 기법 중 가장 강력한 방법이다. 단순전력분석(Simple Power Analysis, SPA), 차분전력분석(Differential Power Analysis, DPA),상관전력분석(Correlation Power Analysis, CPA) 등이 있다. 전자 장치에서 동작하는 동안 0과 1의 변화를 처리하는 동안 소비 전력이 다르게 발생한다. 이를 모델링하여 분석하는데 모델링 기법으로 Hamming distance나 Heamming waight 등을 사용한다. 그림 3-1은 전압 레벨로 전환된 비트 수를 추적 할 수 있음을 보여준다. 이 결과는 처리 된 데이터가 전압 레벨에 미치는 영향을 보여준다. 9개의 오버레이 된 파형은 로드 (LDA) 명령으로 서로 다른 데이터에 액세스 할 때 전력 트레이스에 해당된다.

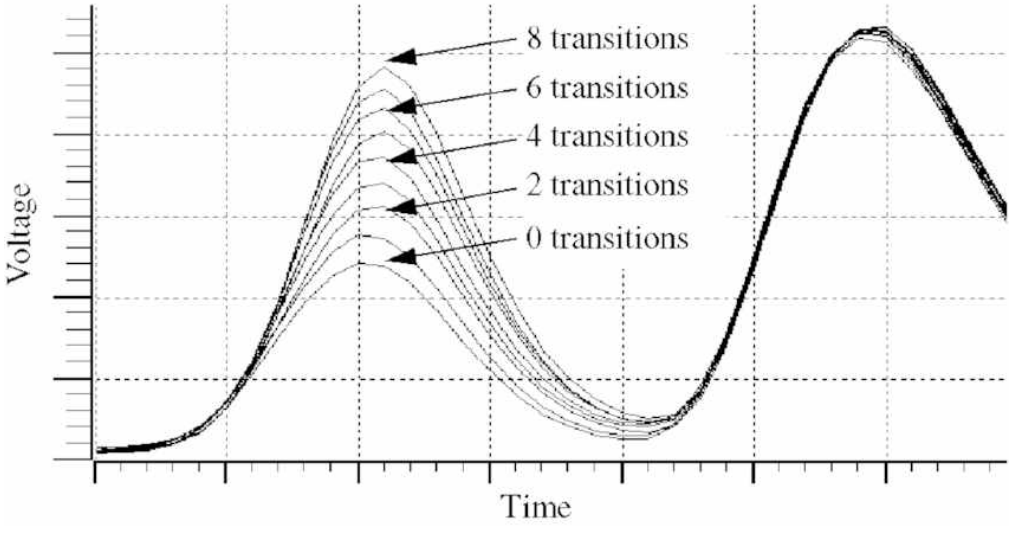

그림 3-1 전압 레벨로 비트 전환 수 추적 [Thomas S, al.(2002)]

이러한 부 채널 정보 기술을 사용한 코드 워터마킹 방법이 연구되고 있으며 부 채널 정보가 프로그램 코드에 접근하지 않고 감지할 수 있는 점을 이용하여 임베디드 시스템에서 작동하는 소프트웨어의 IP 불법 복제를 방지한다.

본 논문의 전력 분석 기반의 ASIC 저항 알고리즘은 이러한 부 채널 정보를 기반 기술의 응용이다. 본 장에서는 전력 분석 기반 IP 탐지에 대한 소개와 새로운 전력 분석 기반 IP 탐지 기법을 제안한다. 해당 기법은 단일 파형에 대한 상관계수가 높게 나타나는 점을 사용한다. 제안 기법은 애플리케이션을 보호하기 위해 추가로 워터마킹 할 필요가 없이 코드를 실행하는 마이크로 컨트롤러의 유출데이터를 워터마크로서 사용할 수 있다. 두 가지 다른 구현의 각각 하나의 부채널 파형에 대한 절대 상관 계수를 기반으로 분석한다. 어셈블리 언어로 작성된 다양한 테스트 응용 프로그램을 사용 Xmaga128 마이크로 컨트롤러에서 평가하였다. 제안 기법은 어셈블리 코드를 수정하는 공격자에게도 강력하며 코드에 대한 정보와 입력에 대한 접근이 불가능하여도 탐지가 가능하다.

본 장의 구성은 다음과 같다. 1절에서는 전력 분석 기반 IP 탐지에 대한 소개 2절에서는 제안하는 기법에 관해 설명한다. 3절에서는 제안 기법에 대한 실험을 통해 성능을 평가한다.

제 1 절 전력 분석 기반 IP 탐지

임베디드 시스템에서 작동하는 소프트웨어는 지적 재산권으로 보호받을 수 있으며 일반적으로 개발자가 지적 재산권 (IP) 코어라고 하는 타사 디자인을 구매하고 사용한다. Marketsandmarkets의 연구에 따르면 임베디드 시스템은 2020년 865억 달러에서 2025년 1,110억 달러로 성장할 것으로 예상한다. 2020 년에서 2025년까지 연평균 6.1 % 증가할 것으로 예상하다. 이와 함께 IP 코어 판매에 대한 시장도 커지며 IP 도용 또한 중요한 위협이 되었다. IP는 리버스 엔지니어링을 통해 해킹하고 디자인을 불법 복제하여 판매하거나 사용할 수 있다. 이런 문제는 제품 수익에 큰 영향을 끼치는 만큼 임베디드 시스템 제조업체의 큰 관심 사항이며 IP 불법 복제를 방지하기 위해 다양한 기술이 제안되었다.

이러한 IP 불법 복제를 방지하는 방법으로 워터마킹 기술을 활용하는 방법이 있다. 소프트웨어 워터마크는 공격자가 프로그램을 복사하는 것을 막지는 않지만, 복제를 탐지하는 데 사용된다. 워터마킹은 리버스 엔지니어링 분석보다 저렴한 비용으로 불법 복제물을 탐지할 수 있다.

워터마크 기법에는 감지하는 소프트웨어나 실행 중인 메모리에 대한 접근이 필요하다. 그러나 코드 복사가 관련 위협이 있는 시스템의 경우에서는 대부분 메모리가 무단 읽기 작업으로부터 보호되기 때문에 임베디드 환경에서는 의심스러운 코드에 접근하기가 쉽지 않다. 이 문제를 해결하기 위해 최근 연구에는 프로그램 코드에 접근하지 않고 감지할 수 있는 부 채널 정보 기술을 사용한 워터마킹 방법이 제안되고 있다. 관련된 가장 최근의 연구에서는 공격자가 IP 표절 탐지를 막기 위해 마이크로 컨트롤러 상에서 코드가 작동할 때 나타나는 전력 파형을 비교하여 표절 여부를 판단한다. 이때 분석을 위해 10만 개 이상의 파형과 같은 입력 값을 사용하나 입력이 알려지지 않거나 입력 데이터를 조작할 수 없는 경우에는 이러한 분석이 불가능하다. 본 논문

에서는 다른 입력값에서도 적용되며 하나의 파형으로도 분석 가능한 새로운 기법을 제안한다. kocher의 부채널 공격에 대한 개척 이후 마이크롤러 컨트롤러에서 코드가 실행되는 동안 전력 소비, 전자기 방출에서 부채널 누설을 유발한다는 것이 잘 알려졌다. 마이크로 컨트롤러에서 수행되는 명령어들은 각각의 다른 전력 소비를 가진다. 제한 기법에서는 IP 보호를 구현된 프로그램 코드와 구현된 미지의 의심스러운 코드의 전력 파형을 사용한다. 두 코드의 파형의 상관 계수를 계산하면 두 코드의 유사도가 높을수록 상관계수가 높게 나타난다. 만약 공격자가 코드를 복사할 경우 코드 상에서 같은 부분은 높은 상관계수를 나타낼 것이다. 제안 기법에서는 원본 파형을 일정 길이의 구간으로 나누어 원본 파형이 드러나는 부분을 탐색하여 IP 탐지에 적용하였다.

제 2 절 제안기법

kocher의 부채널 공격에 대한 개척 이후 마이크롤러 컨트롤러에서 코드가 실행되는 동안 전력 소비, 전자기 방출에서 부 채널 누설을 유발한다는 것이 잘 알려졌다. 마이크로 컨트롤러에서 수행되는 명령어들은 각각의 다른 전력 소비를 가진다. 제한 기법에서는 IP 보호를 구현된 프로그램 코드와 구현된 미지의 의심스러운 코드의 전력 파형을 사용한다. 두 코드의 파형의 상관 계수를 계산하면 두 코드의 유사도가 높을수록 상관계수가 높게 나타난다. 만약 공격자가 코드를 복사할 경우 코드상에서 같은 부분은 높은 상관계수를 나타낼 것이다. 제안 기법에서는 원본 파형을 일정 길이의 구간으로 나누어 원본 파형이 드러나는 부분을 탐색하여 IP 탐지에 적용하였다.

1) 공격

공격은 두 가지의 경우로 나눌 수 있다. 첫 번째로는 공격자가 도난 한 IP를 그대로 사용하는 경우이다. 두 번째는 공격자가 도난 한 IP를 수정하여 사용하는 방법이다. 첫 번째의 경우는 일반적인 워터마킹의 기법으로 IP 복제의 여부를 판단할 수 있지만 두 번째의 경우에는 워터마킹으로 사용된 코드 부분을 수정하여 IP 탐지를 방해할 수 있다. 제한 기법에서는 공격자가 이 두 가지의 공격에 대한 방어를 중점으로 두었다.

2) 추출

데이터의 추출은 정품의 프로그램과 알려지지 않은 프로그램의 파형을 수집하여야 한다. 정품 코드의 파형을 워터마킹으로 사용하며 대상장치에서 사용되는 입력값을 선택할 수 없다고 가정하에 해당 코드는 알려지지 않은 입력에 대한 입력을 사용한다. 측정은 유사성을 비교하기 쉽도록 두 프로그램은

같은 측정 설정을 사용한다. 전처리 같은 파형이라도 잡음으로 인해 변경될 수 있다. 데이터의 압축과 잡음에 대한 필터링을 위해 FFT (Fast Fourier Transform)를 적용한다.

3) 탐지

탐지는 추출한 두 개의 데이터에서 4개의 행렬을 생성한다. 이 4개의 행렬을 사용하여 상관계수를 계산하고 최댓값으로 사영하고 이 값을 비교한다.

(가) 추출 데이터 기반으로 행렬 생성

추출한 정품코드의 파형 데이터를 N의 길이로 나눈 N x M1 행렬 $T_{genuine}$과 추출한 의심코드의 파형 데이터를 N의 길이로 나눈 N x M2 행렬 $T_{unknown}$을 생성한다. M은 각 데이터 샘플의 길이를 N으로 나눈 값이다. 추출한 코드의 파형 데이터의 길이 l을 N만큼 추출한 값을 모든 구간에서 추출하여 FFT 적용하여

$$T' = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-2} & a_{n-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{l-n} & a_{l-n+1} & \cdots & a_{l-n+1} & a_{l-1} \end{pmatrix}$$

T'를 생성한다. 정품 파형의 데이터로 N x M1` 행렬 $T`_{genuine}$과 의심 파형의 데이터로 N x M2` 행렬 $T`_{unknown}$을 만든다.

(나) 피어슨 상관계수 계산

제안기법에서는 전력 기반 암호 분석 분야에서 많이 사용되고 있는 피어슨 상관관계를 사용한다. 정품 코드와 미지의 코드에 데이터 누출 간의 유사성을 계산하기 위해 행렬 $T_{genuine}$와 $T`_{unknown}$의 상관계수로 이루어진 M1 x M2` 행렬 S1 그리고 행렬 $T_{unknown}$ 와 $T`_{genuine}$의 상관계수로 이루어진 M2 x M1` 행렬 S2를 구한다.

(다) Maximum Projection

구간으로 나눈 파형이 비교하는 파형과 유사한 지점이 있다면 해당 구간은 그 지점에서 높은 상관계수를 갖는다. 이러한 값을 찾기 위해 S1와 S2의 열에서 최댓값으로 이루어진 $p1_{col}$와 $p2_{col}$를 구한다. 이 $p1_{col}$와 $p2_{col}$의 평균치로 표절 여부를 판단하고 각 구간에 대한 값을 사용하여 해당 구간에 대한 유사성을 판단하여 표절 후 수정 여부를 평가한다.

제 3 절 실험

실험에서 장치의 수집은 입력이 알려지지 않은 단일 파형이 사용되며 소스 코드 또한 알려지지 않았다. 측정에는 동일하게 ChipWhispererLite XMEGA (8-bit processor)를 사용하며 7.38 MS/s 샘플링으로 모두 동일한 설정을 사용한다.

1) 같은 암호에 다른 구현 비교

같은 알고리즘에도 구현에 따라 속도와 구조가 다르다. IP의 구분에 있어서도 서로 다른 IP로 간주해야 한다. 또한 코드의 일부가 유사하거나 동일하다는 것을 확인하기 위해 3가지의 AES 암호화 구현에 대한 테스트를 진행한다.

	400	200	100	50
Furious	0.99213	0.99170	0.99287	0.99486
	0.99216	0.99194	0.99320	0.99507
Fast	0.83707	0.88271	0.92129	0.96302
	0.88861	0.92829	0.95482	0.98099
Fantastic	0.84027	0.88468	0.91259	0.94188
	0.88577	0.91963	0.94888	0.96943

표 3-1 Furious와 다른 AES암호구현과의 제안기법을 사용한 n에 따른 절대 상관계수

표 3-1 은 모든 테스트 AES 구현에 대한 평균 절대 상관 계수를 보여

준다. 라운드의 반복으로 같은 구간에 대한 비교로 유사한 파형이 반복됨을 알 수 있다. (그림 1)과 같이 같은 구현을 비교할 경우 N의 값이 작아지더라도 0.99 이상의 비교적 높은 상관계수를 나타낸다. 다른 구현의 경우에는 N의 값이 클수록 높은 상관계수를 나타내며 N의 작을 경우 (그림 2)와 같이 동일한 코드의 부분에서는 높은 상관계수를 나타내고 그렇지 않은 부분에서는 낮은 상관계수를 드러낸다. 그래프를 통해 코드의 어느 부분이 유사하거나 동일한 지 알 수 있다.

2) 표절 후 수정된 구현 비교

코드를 적극적으로 조작하는 공격자를 시뮬레이션하기 위해 Furious를 실제 구현으로 선택하고 6개의 방식으로 수정한다.

주소변경 구현에서 사용되는 레지스터와 SRAM 주소를 변경한다.

명령어변경 가능한 경우 클럭 사이클의 변경 없이 명령의 순서나 명령어를 바꾼다.

주소와 명령어 변경 구현에서 사용되는 레지스터와 SRAM 주소 그리고 명령의 순서를 바꾼다.

NOP 더미 추가 더미로 NOP명령어를 삽입한다. 실험에는 570 사이클을 추가 하였다.

Smart 더미 추가 더미로 NOP 명령어를 사용 시에는 전력 소비가 적기 때문에 구별이 쉽다. 전력소비를 보다 정교하게 보이기 위해 실제 연산과 유사한 무의미한 연산을 하는 Smart 더미를 사용한다. 실험에는 570 사이클을 추가하였다.

주소와 명령어 변경과 Smart 더미 추가 레지스터와 SRAM 주소 그리고 명령의 순서를 변경하고 Smart 더미를 추가한다.

	400	200	100	50
addr	0.98174	0.98092	0.98204	0.98700
	0.98230	0.98124	0.98272	0.98786
swap	0.99026	0.98946	0.98998	0.99198
	0.99023	0.98943	0.99021	0.99218
addr+swap	0.98123	0.98073	0.98243	0.98707
	0.98168	0.98176	0.98338	0.98773
dummy (nop)	0.88323	0.92220	0.93959	0.96058
	0.88191	0.92733	0.96737	0.98681
dummy (smart)	0.78183	0.85451	0.91320	0.95578
	0.81031	0.91686	0.96328	0.98745
addr+swap+ dummy(smart)	0.79846	0.85621	0.91140	0.95135
	0.81913	0.91282	0.95799	0.98227

표 3-2 실제 구현과 수정한 코드의 제안기법을 사용한 n에 따른 절대 상관계수

표 3-2는 수정한 코드의 평균 절대 상관 계수를 보여준다. 레지스터 수정과 명령어 교환에서는 수정과 무관하게 높은 평균의 상관계수를 나타내 표절하여도 탐지할 수 있다. 더미를 추가 하였을 때는 더미의 추가가 상관계수를 낮추는 것을 알 수 있다. 그러나 더미 추가의 경우 N의 크기를 줄이면 (그림 3), (그림4)와 같이 수정 전 코드구간을 수정코드에 탐색할 때는 전체적으로 높은 상관계수를 나타내지만 반대의 경우는 더미를 추가한 구간의 상관계수가 낮게 나와 이를 구분할 수 있다. 그림과 같이 유사성을 나타내는 그래프에서 차이로 더미를 추가하였음을 알 수 있어 더미 코드를 추가 하였어도 탐지할 수 있음을 확인할 수 있다.

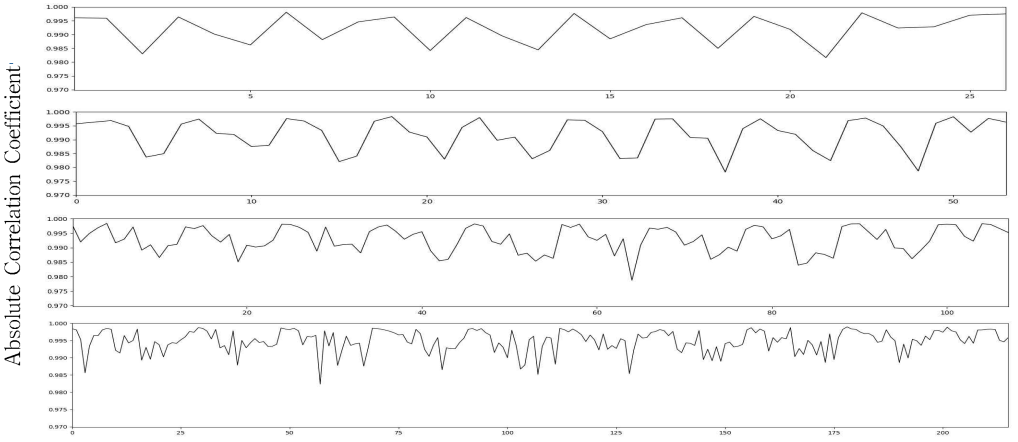

그림 3-2 Furious와 Furious의 n=400, 200, 100, 50에서 p1col 그래프. n의 개수와 상관없이 높은 상관계수를 나타낸다.

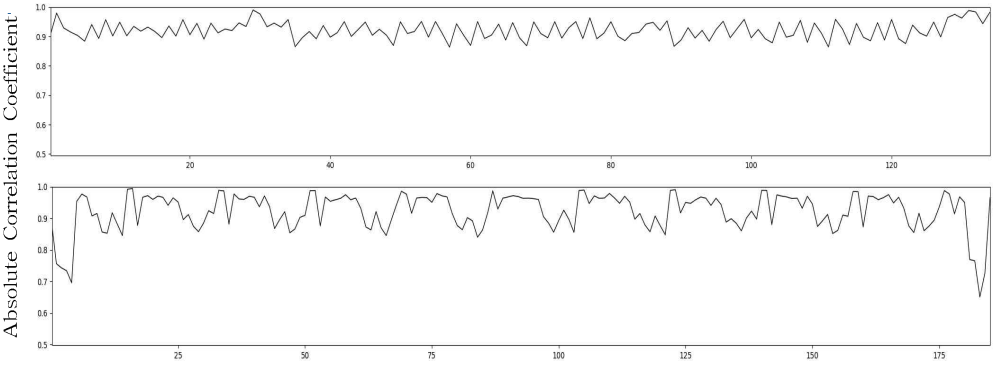

그림 3-3 Furious와 Fast의 n=100에서 p1col 그래프와 Furious와 Fantastic의 n=100에서 p1col의 출력 그래프. Furious와 Furious 비교 보다 낮은 상관계수를 나타낸다.

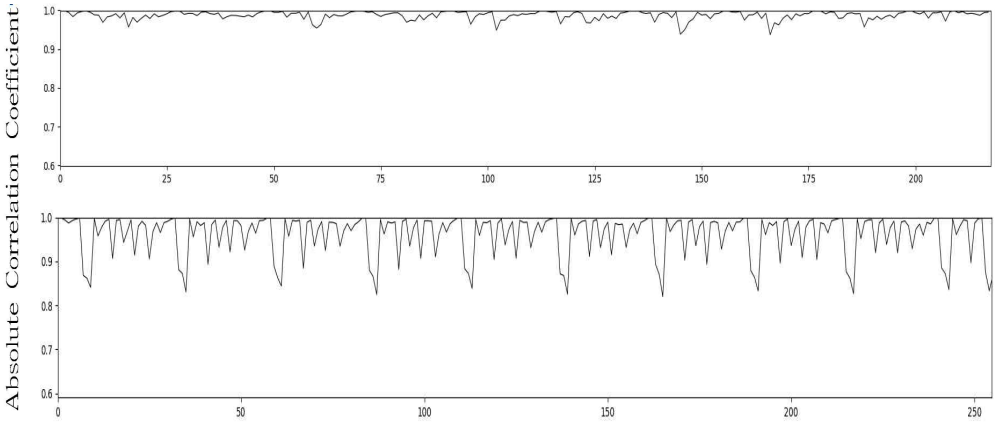

그림 3-4 Furious와 dummy(nop)을 추가한 Furious의 n=50에서 p1col 그래프와 p2col 그래프. p1col 그래프와 다르게 p2col 그래프는 추가된 더미 코드 부분에서의 상관계수가 높게 나오는 것을 확인 가능하다.

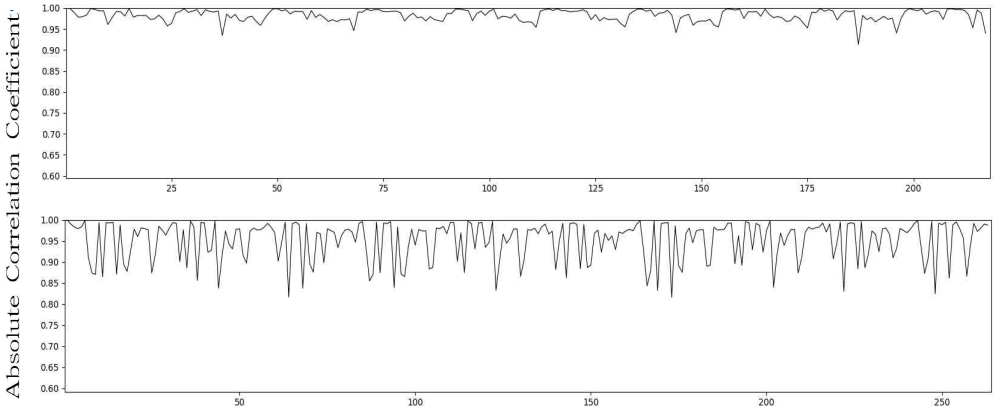

그림 3-5 Furious와 Dummy(smart)을 추가한 Furious의 n=50에서 p1col 그래프와 p2col 그래프. p1col 그래프와 다르게 p2col 그래프는 추가된 더미 코드 부분에서의 상관계수가 높게 나오는 것을 확인 가능하다.

제 4 장 전력 분석 기반 PoW

본 장에서는 3장의 전력 분석 기반 코드 분석을 바탕으로 전력 파형을 작업증명 방식에 사용하는 전력 분석 기반 PoW 방법을 제안한다. 전력 분석 기반 코드 분석은 마이크로 컨트롤러에서 발생하는 소비전력은 명령어에 따라 다르게 발생하는 점을 사용한다. 2장에서 살펴본 바와 같이 작업 증명 기반의 합의 알고리즘은 풀기는 어렵고 검증은 쉬운 문제를 풀도록 한다. ASIC에서는 이러한 문제를 빠르게 풀 수 있으며 AISC 저항 알고리즘은 ASIC의 최적화를 줄이려는 방법을 사용한다. 제안 기법은 어려운 문제로 비트코인과 같은 목푯값 이하의 해시값을 찾는 문제를 사용한다. 그리고 ASIC의 최적화를 줄이려는 방법으로 채굴 과정에 마이크로 컨트롤러 암호화 모듈을 수행하는 과정을 추가한다. 블록 헤더의 정보를 입력값으로 마이크로 컨트롤러에서 암호화를 수행하며 발생한 전력소비 파형을 블록 헤더에 추가한다. 검증은 암호화 프로그램은 블록값에 저장된 키 및 임시 값을 사용하여 마이크로 컨트롤러에서 작동한다. 실행 중에 전력 소비 파형이 생성되어 캡처된다. 그런 다음 FFT 기능을 사용하여 전력 소비 파형을 필터링한 후 필터링 된 파형과 유효 파형 간의 상관 계수를 계산한다. 값이 임곗값을 초과하면 검증이 확인되며 이는 검증을 위해 전력 파형을 활용하는 새로운 접근 방식이다. 본 장에서는 1절에서 제안기법에 대한 설명과 ASIC 저항 달성에 대하여 논한다. 그리고 2절에서 실험을 통해 제안 기법을 검증한다.

제 1 절 제안 기법

1) 블록생성

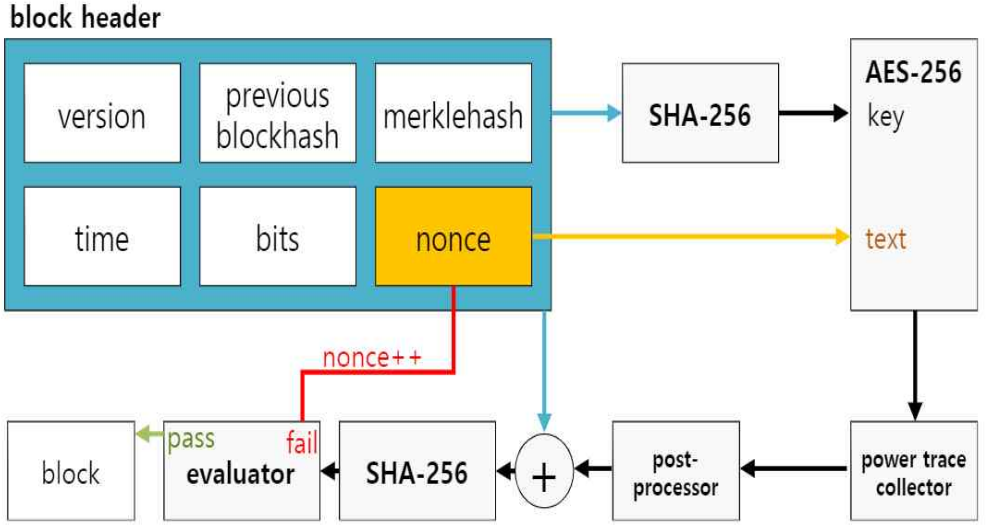

그림 4-1 전력 파형 기반 작업증명 과정

제안한 방법의 블록 생성 과정은 그림 1과 같다. 먼저, 블록 헤더 정보는 버전(version), 이전 블록 해시(previous blockhash), 머클 해시(merklehash), 시간(time), 비트(bits) 및 임시 값(nonce)으로 채워진다. 그 후 해시 기능 (예 : SHA-256)이 블록 헤더에서 수행된다. 출력은 암호화 모듈 (예 : AES)의 키 값에 사용된다. 텍스트 부분의 경우 임시값이 사용된다. 이러한 입력 값을 사용하여 암호화 모듈이 대상 마이크로 컨트롤러에서 실행된다. 마이크로 컨트롤러는 암호화를 수행하여 특정 전력 파형을 생성하며 이러한 전력 파형에는 잡음이 포함될 수 있다. 잡음을 제거하기 위해 후처리가 수행된다. 그다음 전력 파형과 블록 헤더가 함께 추가된다. 덧셈의 출력은 해시 함수의 입력 값으로 사용된다(예 : Sha256). 마지막으로 검증자는 해시 함수의 출력이 목푯

값을 충족하는지 확인한다. 목푯값을 만족하면 유효한 블록이 생성된다. 그렇지 않으면 nonce가 증가하고 루틴이 다시 수행된다. 제안한 PoW 방법은 Algorithm 1과 같다. 3단계에서 임시 값을 업데이트한다. 4 단계와 5단계에서는 블록 헤더와 nonce의 SHA-256 출력을 각각 AES-256 암호화의 키와 텍스트에 사용하고, 6단계와 7단계에서는 전력 소비량을 측정하고 사후 처리한다. 8단계에서 블록 헤더와 전력 파형이 함께 추가된다. 마지막으로 정보는 목푯값 생성에 사용된다. 값이 목푯값 보다 작으면 블록이 성공적으로 생성된다. 그렇지 않으면 nonce 값을 늘린 후 계산이 다시 수행된다.

Algorithm 1 Proposed PoW algorithm

1: **procedure** PROPOSED POW(*Block*)
2: **do**
3: $Block.header.nonce \leftarrow Block.header.nonce + 1$
4: $AES256.key \leftarrow SHA256(Block.header)$
5: $AES256.text \leftarrow Block.header.nonce$
6: $Block.header.trace \leftarrow PowerCapture(AES256)$
7: $Block.header.trace \leftarrow PostProcessor(Block.header.trace)$
8: $Block.header \leftarrow Block.header + Block.header.trace$
9: $H \leftarrow Sha256(Block.header)$
10: **while** $H < Target$
11: **return** *Block*
12: **end procedure**

알고리즘 4-1 제안하는 PoW 알고리즘

가) 블록헤더

비트코인의 PoW에서 블록의 해시값은 생성 시간, 버전, 비트, 루트 해시, 이전 블록의 해시, 임시 값인 임시 값을 조합하여 생성된다. 새로 생성된 해시값이 특정 목푯값보다 작으면 새로운 블록이 생성되어 기존 블록체인 네트워크에 연결된다. 제안한 방법에서는 블록 생성 시간, 버전, 비트, 루트 해시, 이전 블록 해시, 임시 값, 전력 소비 파형을 결합하여 블록 해시값을 생성하

며, 이 단계는 고유한 전력 파형을 예측할 수 없도록 하기 위해 사용된다.

나) 마이크로 컨트롤러에서 전력 파형 수집

이 단계에서는 키와 임시 값으로 AES-256 암호화가 수행된다. 전력 파형의 출력은 블록 헤더에 저장된다. 전력 파형은 입력과 알고리즘에 따라 달라지므로 전력 파형의 출력에 키 및 임시 정보가 포함되어 있는지 확인한다. 전력 파형은 사후 처리 된 다음 해시 함수 (즉, Sha256)의 입력 값에 사용된다. 전력 파형 정보는 고유하므로 이전 단계 없이는 해시 기능을 수행할 수 없다. 이것은 마이크로 컨트롤러를 사용하여 위의 단계를 통해 전력 추적을 생성하도록 하는 주요 목적이다.

(1) 소비 전력 파형

마이크로 컨트롤러에서 프로그램을 작동하여 생성된 소비 전력 파형은 각기 다른 특성을 나타낸다. 이는 명령어 수행 시 발생하는 소비 전력의 차이 때문이다. 또한, 명령어의 매개변수 값에 따라 소비전력의 차이를 보인다. 제안 기법에서는 이 독특한 소비 전력의 특성을 작업을 증명하는 데 사용 한다. 소비 전력은 공격자가 고속 마이닝을 수행하기 위해 코드를 빠르게 변경하거나 실행하지 않고 임의 추적에 진입하는 것을 방지한다. 이 마이크로 컨트롤러의 소비 전력 파형은 잡음이 있을 수 있다. 잡음으로 인한 오 탐을 방지하기 위해 FFT (Fast Fourier Transform)가 소비 전력 파형의 잡음 제거에 적용된다. FFT는 획득된 소비 전력 파형에 적용되어 고주파수를 제거한 다음 주파수에서 신호로 다시 변환된다. 그림 4-2의 상단과 하단에는 각각 잡음이 있는 소비 전력 파형과 잡음이 없는 소비 전력 파형이며, 잡음이 있는 소비 전력 파형은 없는 경우보다 더 명확한 정보를 보여준다.

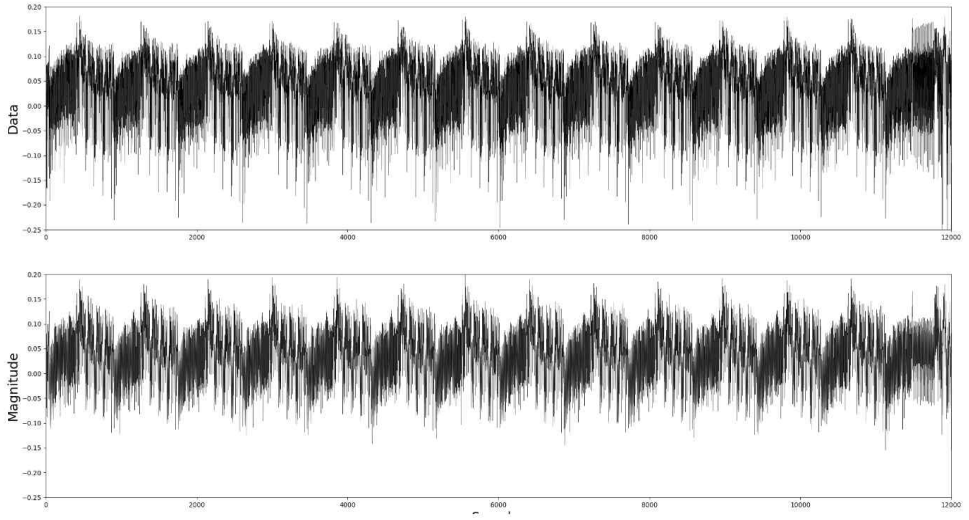

그림 4-2 후처리 전 전력파형(상단)과 후처리 후 전력파형(하단)

(2) 입력 값에 대한 높은 엔트로피

마이크로 컨트롤러에서는 AES-256 암호화가 수행된다. AES-256 암호는 각각 32바이트의 키와 16바이트의 평문을 사용한다. 제안 기법에서 사용하는 AES-256의 켜는 블록 헤더를 해싱(즉, SHA-256)한 값을 사용한다. 블록 헤더의 임시 값은 변경되므로 키는 해시 함수로 인해 임의성을 가진다. 그리고 평문은 16바이트 임시 값을 사용한다. 키와 일반 텍스트는 각각 256bit와 128bit이므로 키와 일반 텍스트의 복잡성은 2384이다. 이런 이유로 무차별 대입 공격을 수행하고 사전에 사전 계산을 통해 테이블을 만드는 것은 불가능하다. 이러한 임의성은 공격자가 전력 소비 추적을 생성하기 위해 미리 마이크로 컨트롤러를 작동하는 것을 허용하지 않는다. 소비 전력 파형은 마이크로 컨트롤러가 동작하는 기능과 입력 값에 따라 다르게 출력되기 때문에 전력 파형 값을 예측하기 어렵다.

2) 검증

블록 체인에서 생성된 블록은 다른 노드로 전파되고 해시값이 조건에 일치하는지 검증한다. 제안한 방법에서는 블록 작성자가 올바른 코드와 입력 값으로 마이크로 컨트롤러에서 작업했는지 확인하기 위해 기존 방법과 함께 추가로 전력 소비 파형을 확인한다.

(가) 파형 검증

마이크로 컨트롤러에서 같은 프로그램 코드와 입력 값을 사용하더라도 수집 시 발생하는 잡음으로 인해 같은 소비전력 파형이 나타나지 않는다. 따라서 전력 소비 파형의 값을 비교하는 것만으로는 전력 소비 파형이 같은 코드와 입력 값을 사용하는지 확인할 수 없다. 이러한 이유로 잡음을 제거하기 위해 전력 파형에서 FFT를 사용한 필터링이 수행된다. 이후 같은 입력 값은 다른 입력 값에 비해 높은 상관 계수를 갖는 특징을 사용한다. 그림 3에서와같이 FFT를 소비 전력 파형에 적용했을 때 같은 입력 값에 대한 상관 계수의 분포가 좁다. 이로 FFT 기능을 사용한 필터링이 제대로 작동함을 확인하였다. 상관 계수의 차이는 실험 중에 확인된다. 제안한 추적 검증 방법에 대한 자세한 설명은 알고리즘 2에 나와 있다. 2 단계에서 AES256 암호화 중에 전력 파형이 캡처된다. 3 단계에서는 전력 파형이 사후 처리된다. 4 단계에서는 정제된 전력 파형과 유효 전력 파형 간의 상관 계수가 계산된다. 5～8단계에서는 상관관계를 특정 임곗값과 비교한다. 값이 임곗값을 초과하면 검증이 올바른 블록으로 판단하여 완료된다.

Algorithm 2 Power Consumption Trace Verification algorithm

```
1: procedure TRACE_VERIFICATION(key, nonce, validate_trace)
2:     trace ← Power_capture(AES256(key, nonce))
3:     refined_trace ← Post_processing(trace)
4:     col ← Correlation_coefficient(refined_trace, valid_trace)
5:     if col > Threshold then
6:         return True
7:     end if
8:     return False
9: end procedure
```

알고리즘 4-2 전력 소비 파형 검증 과정

3) ASIC 저항성

가) PoW 요구 사항

PoW는 Sybil Attack과 같은 블록체인 네트워크 공격을 방지하도록 설계되었다. 안전한 작업 증명을 달성하기 위한 몇 가지 요구 사항이 있다. ThePoW는 증명하기 어렵다. 그러나 검증은 제공자에게 쉬운 문제 여야 한다. 비트코인은 이러한 요구 사항을 충족하기 위해 해시를 사용한다. 그러나 비트코인은 ASIC 공격에 취약하다. 제안한 방법은 기존과 같은 해시 함수를 사용하여 비대칭성을 만족하게 한다. PoW는 특정 조건에 따라 최적화가 불가능해야 한다. 누군가 최적화 방법을 찾으면 이점을 취할 것이다. 이는 블록체인 네트워크의 중앙 집중화 문제를 일으키며, 비트코인은 SHA256을 사용하여 ASIC과 같은 빠른 계산을 수행할 수 있는 채굴자이기 때문에 일반 채굴자들이 참여하기 어려운 암호 화폐이다.

나) ASIC 저항

이전 PoW 방식들은 ASIC로 최적화할 수 있다. 제안한 방법에서는 마이크로 컨트롤러가 같은 코드와 입력 값을 수행한 검증 단계를 추가하여 마이크로 컨트롤러에서 지정된 코드를 사용하는 단계를 수행해야 한다. 따라서 A

SIC를 사용한 최적화가 불가능하므로 ASIC는 우리 시나리오에서 의미가 없다. 출력 전력 소비 추적 정보는 해시 계산 (SHA-256)의 입력 값으로 사용되기 때문에 병렬 처리를 사용한 해시 계산이 불가능하다. 비 최적화 및 병렬성 제약으로 인해 제안한 방법은 ASIC 저항을 달성했다.

다) 작업 시간

제안한 작업 증명 실행 시간은 마이크로 컨트롤러에 암호 모듈의 실행 시간과 해시 작업 시간의 합계에 반복 횟수를 곱한 값이다. 반복 횟수를 조정할 수 없으므로 마이크로 컨트롤러의 암호화 모듈 실행 시간이 고정된다. 병렬 작업이 불가능하므로 실행 시간을 줄이는 유일한 방법은 마이크로 컨트롤러에서 해시 알고리즘을 빠르게 계산하는 것이다. 그러나 대상 마이크로 컨트롤러에는 특정 자원 제한 기능이 있으므로 계산 속도를 향상하게 시킬 수 없다. 따라서 고가로 특수 채굴 장비를 구매할 이유가 없으므로 채굴자들은 적절한 속도로 해싱할 수 있는 범용 마이크로 컨트롤러를 사용할 것으로 예상한다.

라) 마이크로 컨트롤러에서 암호화 모듈의 유연성

마이크로 컨트롤러에서 작동하는 암호화 모듈은 변경 가능하다. AES-256 암호화 모듈을 사용하는 이유는 국제 표준으로 구현이 간편하고 안전성이 보장됐기 때문이다. 또한, 입력 값이 같으면 중간 값과 출력값이 같다. 이 함수로 전력 소비 추적을 확인할 수 있다. 제안한 방법은 마이크로 컨트롤러에서 AES-256을 사용하지만 좋은 암호화와 효율성이 있는 경우 이 과정에 다른 블록 암호를 사용하여도 좋다.

마) 저가형 프로세서 사용

기존 AISC 저항 알고리즘은 상용 컴퓨터에서 효율적으로 작동한다. 상용 컴퓨터의 작동은 ASIC에서 구현될 수 있으므로 기존 방법은 알고리즘이 ASIC에서 실행될 때 개발 비용이 성능 향상보다 높을 것이라는 예측에 의존한다. 그러나 소비전력 파형의 고유한 특성으로 인해 제안한 방법은 해당 마이크로 컨트롤러에서 계산이 필요하다. 따라서 ASIC 구현이 불가능하다. 또한, 제안한 기법에서는 여러 장치를 사용하여 해시를 계산할 수 있다. 이 해싱 작업은 전체 연산 과정에서 차지하는 비율이 적고 이전 단계의 마이크로 컨트롤러에서 작업한 후에 발생하므로 해싱 연산이 빠르게 작동하더라도 많은 효율성을 달성하지 못한다. 따라서 이 해시 연산은 임베디드 컨트롤러를 사용할 때 효율적이다. 전력 소비를 측정하는 마이크로 컨트롤러와 해시 작업을 수행하는 장치가 같은 작업을 수행한다고 가정하면 다른 장치에서 작업하는 것이 로우 엔드 프로세서 장치에서 사용하는 컨트롤러에서 작업하는 것보다 더 효율적이지 않다. 마이크로 컨트롤러에서 작업이 클수록 효율성이 떨어진다. 이러한 장점은 채굴자의 접근성을 높여 암호 화폐의 탈중앙화를 촉진하는 데 이바지할 수 있다.

제 2 절 성능 평가

1) 실험 환경

제안 방법은 전력 파형 정보에 의존한다. 이러한 이유로 고품질 전력 추적 정보가 중요하다. ChipWhispererLite XMEGA (8 비트 프로세서)를 통해 마이크로 컨트롤러의 전력 소비량을 수집하였다. 샘플링 속도는 초당 7.38 million (MS / s)이며 이 수치는 대상 프로세서의 작동 주파수이다. 마이크로 컨트롤러의 소스 코드는 C 언어로 작성되고 AVR-GCC로 컴파일된다. 전력 추적 수집을 위해 ChipWhispererLite API 버전 5.1을 사용했다. 실험 코드는 Python으로 작성되었다.

2) ASIC 저항성

마이크로 컨트롤러에서 AES-256 암호화 모듈을 실행하여 같은 입력 값에서 생성된 전력 소비 파형을 수집하고, 암호화 모듈이 입력 변수에 따라 고유한 특성을 나타내는지 확인한다. 키와 평문 둘 다 하나를 고정하여 사용하여 수집한 소비 전력 파형, 키는 고정한 값을 사용하고 평문은 임의의 값을 사용한 소비 전력 파형, 키는 임의의 값 평문은 고정한 값을 사용, 키와 평문 임의의 값을 사용하는 경우 이 4가지 경우에 대하여 각각 2,000개의 파형을 수집하고 1,000개의 그룹으로 나누어 상관계수를 계산한다. 표 2는 결과의 통계치를 보여준다. 키와 입력 모두 다른 경우보다 상대적으로 높은 상관 계수를 나타냈다. 특히 키와 입력을 고정했을 때의 최솟값은 키만 임의의 값을 사용했을 때의 최댓값 보다 큰 차이를 보인다. 이것은 그림 4에서 분명하게 확인할 수 있다. 같은 입력 값과 다른 입력 값에 대한 공통 값은 없다. 같은 경우의 최솟값과 다른 경우의 최댓값 사이에는 큰 차이가 있다. 제안 방식의 AES 뿐만 아니라 LEA, ARIA 및 SEED에 대해 같은 실험을 수행하여 같은 중간값과 출력 값을 가진 다른 암호도 이러한 특징을 가지고 있음을 확인하였다.

Measurement(key/plaintext)	Maximum	Minimum	Mean	First Quartile	Third Quartile
AES					
Same key / Different plaintext	0.99656	0.99433	0.99549	0.99519	0.99578
Different key / Same plaintext	0.99619	0.99406	0.99416	0.99488	0.99542
Same keys / Same plaintext	0.99965	0.99820	0.99884	0.99859	0.99903
Different key/Different / plaintext	0.99605	0.99361	0.99493	0.99461	0.99524
LEA					
Same key / Different plaintext	0.99495	0.99093	0.99288	0.99236	0.99337
Different key / Same plaintext	0.99502	0.99114	0.99283	0.99231	0.99331
Same key / Same plaintext	0.99980	0.99670	0.99809	0.99756	0.99854
Different key / Different plaintext	0.99471	0.99093	0.99271	0.99221	0.99317
ARIA					
Same key / Different plaintext	0.99612	0.99417	0.99505	0.99483	0.99526
Different key / Same plaintext	0.99611	0.99409	0.99498	0.99476	0.99518
Same key / Same plaintext	0.99987	0.99872	0.99919	0.99900	0.99937
Different key / Different plaintext	0.99594	0.99401	0.99492	0.99469	0.99514
SEED					
Same key / Different plaintext	0.99222	0.98817	0.99034	0.98990	0.99080
Different key / Same plaintext	0.99263	0.98849	0.99050	0.99005	0.99097
Same key / Same plaintext	0.99993	0.99834	0.99901	0.99875	0.99927
Different key / Different plaintext	0.99218	0.98801	0.99023	0.98976	0.99068

표 4-1 각 블록 암호들의 소비전력파형간의 상관계수

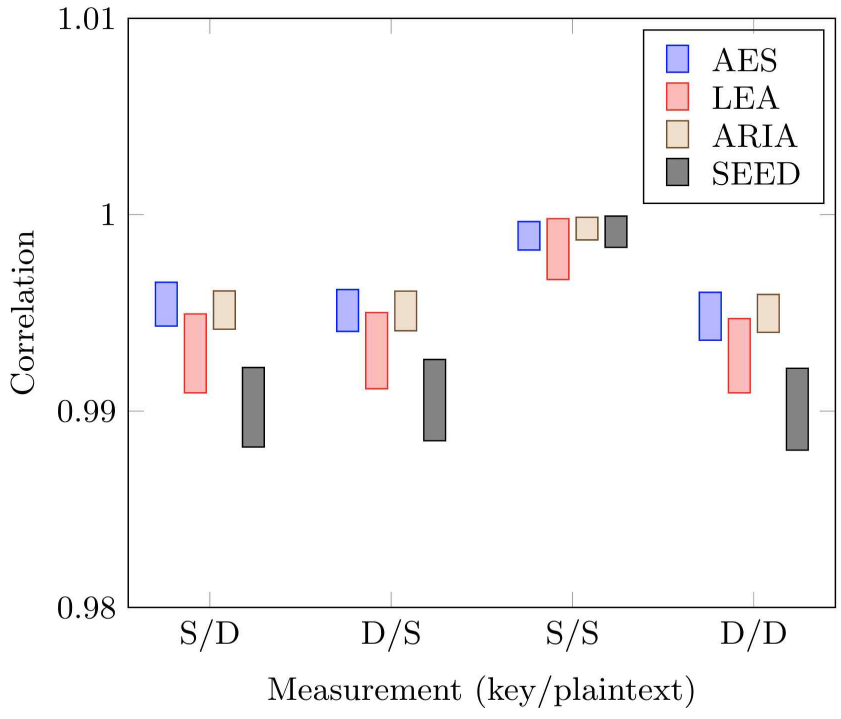

그림 4-3 각 암호에서 발생한 전력 소비 트레이스의 상관계수 비교 결과, S / D, D / S, S / S 및 D / D 간의 상관 계수는 각각 동일한 키 / 다른 일반 텍스트, 다른 키 / 동일한 일반 텍스트, 동일한 키 / 동일한 일반 텍스트 및 다른 키 / 다른 일반 텍스트를 나타냅니다.

3) 이전 연구와 비교

이전 방법은 상용 CPU와 GPU에서 동작을 목표로 ASIC에 저항하는 어려운 문제를 제안하였다. 제안한 방법은 마이크로 컨트롤러, 암호화 모듈 및 입력 값과 관련된 다양한 전력 파형 기능에서 기반하는 문제이기 때문에 저가형 IoT 장치에서 효율적으로 사용할 수 있다. 제안 기법의 핵심 아이디어인 마이크로 컨트롤러에서 동작 시 발생하는 소비 전력 파형은 에뮬레이션하기 어렵다. 따라서 제안기법은 저가형 IoT 장치를 위한 최초의 실용적인 PoW 방법이다.

제 5 장 결론 및 향후 개선 방안

본 논문에서는 저가현 마이크로 컨트롤러의 전력 분석을 기반으로 한 새로운 ASIC 내성 PoW를 제시하였다. 이전 연구와 다르게 CPU와 GPU 동작의 목표가 아닌 저가형 IoT 장치를 위한 최초의 기법이다. 따라서 낮은 비용과 낮은 전력소비의 특징을 가지며 채굴자의 접근성을 낮추어 가상 화폐의 분산화에 기여한다. 이 접근 방식은 마이크로 컨트롤러에서 암호화 모듈을 수행할때 발생하는 고유한 전력 소비 패턴을 기반으로한다. 전력 파형은 PoW 검증시 높은 품질을 위해 FFT로 후처리되고 블록생성에 사용된다. 제안 기법의 실용성을 확인하기 위해 저사양 마이크러컨트롤러에서 기법을 구현하고 4개의 다른 블록암호를 평가하였다. 결과적으로 제안하는 기법은 대상 마이크로 컨트롤러에서 효율적으로 작동하는 것을 확인하였다. 향후 작업으로 공격 시나리오와 관련하여 전력 추적을 위한 ASIC 기반의 에뮬레이션을 조사하고 제안된 방법을 공격하기 위해 딥 러닝 기반 에뮬레이션에 대하여 함께 조사할 것이다.

참 고 문 헌

1. 국외문헌

Nakamoto, S., (2008). Bitcoin: A peer-to-peer electronic cash syste m.

Woodet, G. al., (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, vol. 151, n o. 2014, pp. 1-32

Mackenzie, A. and Noether, S., (2016). Ring confidential transactions. vol. 1, pp. 1-18.

Dwork, C. and Naor, M., (1992). Pricing via processing or comba tting junk mail. inAnnual International Cryptology Conferenc e, Springer, pp. 139-147.

Cai, J. Y., Lipton, R. J., Sedgewick R. and Yao, A.C., (1993). Toward s uncheatable bench-marks. in[1993] Proceedings of the Eigth Annual Structure in Complexity TheoryConference, IEEE, pp. 2 -11.

Ar, S. and Cai, J. Y., (1994). Reliable benchmarks using numerical ins tability. in SODA, pp. 34-43.

Franklin, M. K. and Malkhi, D., (1997). Auditable metering with light weight security. inInternational Conference on Financial Crypto graphy, Springer, pp. 151-160.

Goldschlag, D. M. and Stubblebine, S. G., (1998). Publicly verifiable l otteries: Applicationsof delaying functions. inInternational Conference on Financial Cryptography, Springer, pp. 214-22 6.

Rivest, R. L. , Shamir, A., and Wagner, D.A., (1996). Time-lock puzz les and timed-releasecrypto.

Dwork, C. and Naor, M., (1992). Pricing via processing or comba tting junk mail. inAnnual International Cryptology Conferenc e, Springer, pp. 139-147.

Backet,A. al., (2002). Hashcash-a denial of service counter-measure.

Bertoni G., Daemen, J., Peeters, M. and Assche, G. V., (2013). Ke ccak. inAnnualinternational conference on the theory and appli cations of cryptographic techniques,pp. 313-314, Springer.

Dryja, T., (2009). Hashimoto: I/o bound proof of work.

Buterin, V., (2013). Dagger: A memory-hard to compute, memory-ea sy to verify scryptalternative. tech. rep., Technical Report, 2013. URL http://www. hashcash.org/papers/dagger.

Percival, C., (2009). Stronger key derivation via sequential memory-h ard functions.

Saberhagen, N. V, (2013). Cryptonote v 2.0.

Vries, A. D., (2018). Bitcoin's growing energy problem,"Joule, vol. 2, no. 5, pp. 801-805.

Primecoin, K. S., (2013) Cryptocurrency with prime number proof -of-work{J}, July7th.

Rivest, R. L. and Shamir, A., (1996). Payword and micromint: Two si mple micropaymentschemes. inInternational workshop on s ecurity protocols, Springer, pp. 69-87.

Miller, A., Juels, A., Shi, E., Parno, B. and Katz, J., (2014). Permacoi n: Repurposing bitcoinwork for data preservation. in2014 IEE E Symposium on Security and Privacy, IEEE, pp. 475-49 0.

Sompolinsky, Y. and Zohar, A., (2015). Secure high-rate transaction processing in bitcoin. inInternational Conference on Financial C ryptography and Data Security, Springer, pp. 507-527.

Jang, J. and Lee, H.N. , (2019). Profitable double-spending attac ks. arXiv preprintarXiv:1903.01711.

Eyal, I. and Sirer, E. G. , (2014). Majority is not enough: Bitcoin min ing is vulnerable. inInternational conference on financial crypto graphy and data security, Springer, pp. 436-454.

Heilman, E., Kendler, A., A. Zohar, and S. Goldberg, (2015). Eclipse attacks on bitcoin'speer-to-peer network. in24th{USENIX}Sec urity Symposium ({USENIX}Secu-rity 15), pp. 129-144.

Daian, P., Eyal, I., Juels, A., and Sirer, E. G., (2017). (short paper) p iecework: Generalizedoutsourcing control for proofs of work. in

International Conference on FinancialCryptography and Data Security, Springer, pp. 182-190.

Kroll, J. A., Davey, I. C., and E. W. Felten, (2013) "The economics of bitcoin mining, orbitcoin in the presence of adversaries," inProceedings of WEIS, vol. 2013, p. 11.

Abadi, M., Burrows, M., Manasse, M. and Wobber, T., (2005). Moderately hard, memory-bound functions. ACM Transactions on Internet Technology (TOIT), vol. 5, no. 2,pp. 299-327.

Dwork, C., Goldberg, A., and Naor, M., (2003). On memory-bound functions for fight-ing spam. inAnnual International Cryptology Conference, Springer, pp. 426-444.

Dwork, C., Naor, M., and Wee, H., (2005). Pebbling and proofs of work. inAnnual Inter-national Cryptology Conference, Springer, pp. 37-54.

Ren, L. and Devadas, S., (2017). Bandwidth hard functions for ASIC resistance. inTheoryof Cryptography Conference, Springer, pp. 466-492.

Bradley, W. F., (2014). Superconcentration on a pair of butterflies. arXiv preprintarXiv:1401.7263.

Cook, S. A., (1973). An observation on time-storage trade off. inProceedings of the fifthannual ACM symposium on Theory of computing, pp. 29-33.

Kocher, P., Jaffe, J. and Jun, B., (1999). Differential power analysis. inAnnual Interna-tional Cryptology Conference, Springer, pp. 388-397.

Durvaux, F., Gerard, B., and Kerckhof, S., (2012). Intellectual property protection forintegrated systems using soft physical hash functions. inInternational Workshopon Information Security Applications, Springer, pp. 208-225.

Samarin, P. and Lemke-Rust, K., (2017). Detecting similar code segments through sidechannel leakage in microcontrollers. inInternational Conference on InformationSecurity and Cryptology, Springer, pp. 155-174.

Kwon, D.,et al., (2003). New block cipher: ARIA. inInternational Co

nference onInformation Security and Cryptology, Springer, pp. 432-445.

Hong, D.,et al., (2013). LEA: A128-bit block cipher for fast encrypt ion on common processors. inInternationalWorkshop on Informa tion Security Applications, Springer, pp. 3-27.

Park, J., Lee, S., Kim, J., and Lee, J., (2005). The SEED encryption algorithm.

Thomas S. Messerges, Ezzat A. Dabbish, Robert H. Sloan, (2002). Exami ng Smart-Card Security under the Threat of Power Analysis Atta cks. IEEE transactions on computer, vol. 51, no. 5.

ABSTRACT

Improved security of lock pattern using random vibration intensity

Kim, Hyun-Jun

Major in IT Convergence Engineering

Dept. of IT Convergence Engineering

The Graduate School

Hansung University

ASIC (Application-Specific Integrated Circuit) Resistance Proof of Work (PoW) is widely adopted in modern cryptocurrencies. In an ASIC, the ASIC resistor PoW is inefficient in design due to its special function. The paper introduces a new ASIC-resistant PoW for low-cost microcontrollers. The microcontroller executes an encryption module for a specific input value and utilizes the measured power waveform. This power waveform always has a certain characteristic depending on the input value and the scheme is not predicted and calculated by the ASIC. Thus, the scheme can be operated on a microcontroller and achieves ASIC resistance.

【Keyword】 ASIC-Resistant, Proof of Work, Power Analysis, Microcontroller, Blockchain