Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.24, NO.1, Feb. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.1.241

안전하고 신뢰성 있는 PUF 구현을 위한 가이드라인*

이 동 건,[1†] 이 연 철,[1] 김 경 훈,[1] 박 종 규,[1] 최 용 제,[2] 김 호 원[1‡]
[1]부산대학교, [2]한국전자통신연구소

Guidelines for Safe and Reliable PUF Implementation*

Donggeon Lee,[1†] Yeonchoel Lee,[1] Kyunghoon Kim[1], Jong-gyu Park,[1]
Yong-je Choi,[2] Howon Kim[1‡]
[1]Pusan National University,
[2]Electronics and Telecommunication Research Institute

요 약

PUF는 디바이스 식별을 위한 기술로써, 인간의 지문이나 홍채와 같은 생체 정보처럼 다른 디바이스들로부터 특정 디바이스를 구별하기 위해 사용되는 기술이다. 지난 10여 년간 PUF를 구현하기 위한 다양한 방법이 많은 연구자들에 의해 연구되었으며, 식별뿐만 아니라, 키 분배 및 인증, 난수 생성 등 PUF를 활용하는 다양한 방법도 연구되었다. 하지만, PUF를 대상으로 하는 다양한 공격들은 PUF의 도입을 저해하는 주요 원인이 되고 있으며, 여전히 PUF의 안전성을 높이고자 하는 다양한 기술들이 연구되고 있다. 본 논문에서는 PUF 및 PUF를 대상으로 하는 다양한 공격들에 대하여 살펴보고, 안전하게 PUF를 구현하기 위한 가이드라인을 제시하고자 한다. 본 연구에서 제안하는 가이드라인은 향후 신뢰성 있고 안전한 PUF를 구현하기 위한 밑거름이 될 것으로 기대한다.

ABSTRACT

A PUF is a technology for distinguishing a device from other devices like biological information such as humans' iris or fingerprints. Over the past decade, many researchers studied various methods for implementing PUFs and utilizing them in identification, random number generation, key distribution and authentication. However, various attacks on the PUFs are the major reason to inhibiting the proliferation of PUF. For the reasons, various technologies are being studied to enhance safety of PUFs. In this paper, we will see several PUF implementations and various attacks on PUFs, and suggest guidelines for securely implementing PUFs. We expect our guidelines would be the foundation for implementing the secure and reliable PUFs.

Keywords: Physically Unclonable Function; Implementation Guideline;

I. 서 론

최근 IT 기술의 급격한 발전으로 인해 빠른 속도로 새로운 최첨단 디바이스들이 등장하고 있지만, 불법 복제에 대한 피해와 위조로 인한 경제적, 산업적 손실이 날이 갈수록 커져가고 있는 실정이다. 이러한 문제를 해결하고자 Physical Unclonable Function (PUF)라는 새로운 기술이 등장하였다. PUF는 마치 인간의 지문이나 홍채와 같은 생체 정보처럼 각각의

접수일(2014년 1월 2일), 수정일(2014년 2월 5일), 게재확정일(2014년 2월 8일)
* 본 논문은 한국전자통신연구원의 기초연구과제로 수행한 연구결과입니다.
† 주저자, guneez@pusan.ac.kr
‡ 교신저자, howonkim@pusan.ac.kr (Corresponding author)

디바이스가 고유의 특성을 가질 수 있는 기술로써, PUF는 동일한 공정으로 만들어진 디바이스라 할지라도, 다른 특성을 가질 수 있도록 하는 기술이다. 즉, 아무리 똑같은 방법으로 디바이스를 만들어도 절대로 그 고유한 특성만큼은 복제할 수 없는 기술이다. PUF라는 이름이 만들어 지기 이전에도 특정 장치를 식별할 수 있는 많은 기술들이 소개되었지만[1-3], 지난 2002년 디지털 회로로 PUF를 구현할 수 있는 기술[4]이 처음 소개된 이후, PUF에 대한 가능성에 많은 사람들이 주목하게 되었고, 향후 10여 년간 PUF를 구현할 수 있는 많은 방법들이 연구 되었다. 하지만, 이후 PUF를 대상으로 하는 많은 공격들이 제시되었다. PUF의 출력을 예측하는 모델링 기법을 비롯하여, PUF를 복제할 수 있는 공격까지 등장하면서, PUF의 안전성이 크게 위협받기도 하였지만, 그에 따른 방어 기법에 대한 연구도 많이 이루어지고 있다. 하지만, 현재까지 PUF에 대한 공격을 완벽하게 차단할 수 있는 기술은 제시되지 않았으며, 따라서 PUF를 안전하게 구현하고 사용할 수 있는 가이드라인이 요구되고 있다. 이와 같은 요구에 의해 본 논문에서는 지난 10여 년 동안 연구되어 왔던 PUF를 구현하기 위한 기술과, PUF의 다양한 활용 사례들을 소개하고, PUF에 대한 다양한 공격 사례들을 소개하고, PUF를 안전하게 구현하기 위한 가이드라인을 제시하고자 한다.

본 논문은 디지털 회로를 이용해서 구현할 수 있는 PUF에 대해서만 다루고 있으며, 본 논문의 구성은 다음과 같다. 2절에서는 PUF 구현 기술 연구 동향에 대하여 소개하고, 3절에서는 PUF를 활용하는 다양한 사례에 대해서 소개하고자 한다. 4절에서는 PUF 및 PUF를 활용하는 프로토콜을 대상으로 하는 공격 사례들을 소개하며, 5절에서는 그 동안의 PUF에 대한 연구 내용들을 바탕으로 하여 PUF를 안전하게 구현하고 활용할 수 있는 가이드라인을 제시하며, 6절에서 끝을 맺는다.

II. PUF 구현 기술 연구 동향

2.1 트랜지스터 V_T기반 PUF

IC를 식별하기 위한 기술로 처음 등장한 기술은 ICID라고 불리는 트랜지스터를 이용한 기법이었다[3]. 이 기법은 동일하게 만들어진 트랜지스터라 하더라도 공정상에서 발생하는 변동에 의해 각각 다른 임계 전압 V_T를 가지게 되는 특성을 이용하였다. Fig. 1.은 V_T 기반 PUF의 구조를 나타내었다. 본 기법에서는 저항에 연결된 동일하게 만들어진 두 개의 트랜지스터의 드레인 전류를 비교하기 위해서 auto-zeroing 비교기를 사용하였다. 첫 번째 트랜지스터의 스위치가 닫혔다가 열리고 다음 트랜지스터의 스위치가 닫히면, 트렌지스터 사이의 불일치로 인해 발생하는 드레인 전류의 차이로 인해 저항에서의 전압 강하가 변하게 되고, 이에 따라 비교기의 출력이 바뀌게 된다.

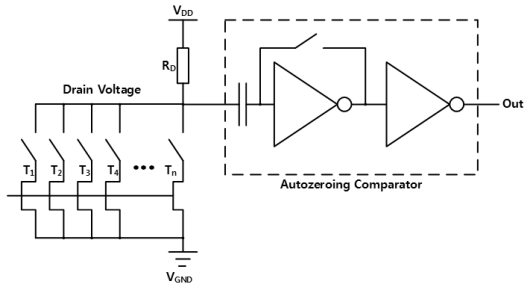


Fig.1. PUF circuit based on drain voltage variations

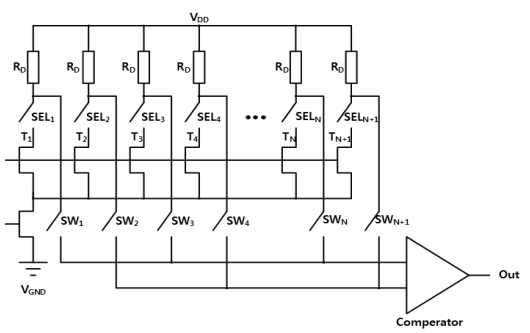


Fig.2. PUF circuit using a differential amplifier.

[5]에서는 [3]의 기법이 실제 IC로 구현하였을 때, 전력 공급에 있어서의 노이즈와 측정 노이즈로 인해 출력이 불안정함을 지적하고, 이를 해결하기 위해 Fig. 2.의 차동 증폭기를 이용한 회로를 제안하기도 하였다.

2.2 Arbiter 기반 PUF

2.2.1 Arbiter PUF

Arbiter PUF는 가장 많이 알려져 있는 PUF중

하나이다. Arbiter PUF는 설계상으로 동일한 거리를 가지는 두 경로에 동일한 신호를 보내 어떤 신호가 먼저 Arbiter에 도착하는지에 따라 출력이 결정되는 PUF이다. 비록 설계상으로는 동일한 거리일지라도 칩 생산 과정에서의 미묘한 물리적 차이로 인해 동일하게 만들어진 다른 칩들 간에 서로 다른 지연 시간을 가지게 된다. Arbiter PUF는 멀티플렉서 등으로 이루어진 연속된 k개의 단계로 이루어진다[6]. 두 경로에 동일한 신호가 주어지면, 동일한 각 단계를 거쳐 경로를 따라 경쟁하는데, 이 경로는 외부에서 각 단계로 들어오는 k개의 challenge 입력 비트에 의해 정해진다. i번째 비트는 i번째 단계의 경로 선택 영향을 미친다. 마지막 단계까지 통과한 전기 신호는 래치로 이루어진 arbiter로 들어가고 둘 중 어떤 신호가 먼저 도달하였는지를 판별하여 0이나 1의 결과를 출력한다.

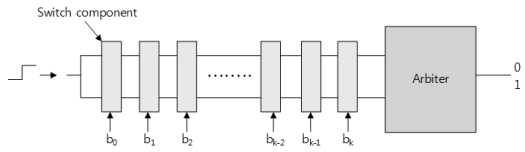


Fig.3. Structure of Arbiter PUF

2.2.2 Feed-Forward PUF

기존의 Arbiter PUF가 기계 학습으로 인한 모델링 공격에 취약한 단점을 보완하기 위해서 제시된 구조이다. Arbiter PUF와 유사한 구조에서 외부의 bits만 단계에 영향을 미치는 것이 아니라 앞부분의 단계에서 나온 결과가 뒷부분의 단계에 영향을 미치는 것이다. 이를 위해 단계의 중간 지점에 또 다른 arbiter가 필요하다[6]. Fig.4.에서 보는 것과 같이 다른 스위치들에는 challenge 비트가 들어가지만 스위치 a는 challenge 대신 중간에 놓여있는 arbiter의 출력이 들어가서 경로 선택에 영향을 준다.

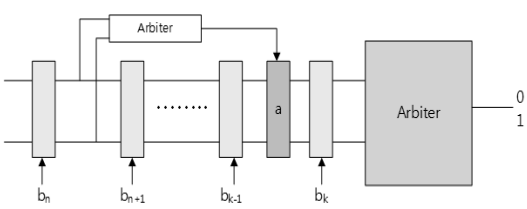


Fig.4. Structure of Feed-Forward PUF

2.2.3 XOR PUF

모델링 공격 취약점을 개선하기 위한 또 다른 방법으로써, XOR PUF가 제시된 바 있다[7]. XOR PUF는 병렬로 l개의 독립적인 arbiter PUF를 두는 것이다. 동일한 challenge가 모든 arbiter PUF에 적용되고 i번째 PUF에서는 개별 출력 t_i가 나온다. 이 값들이 XOR 되어 XOR PUF의 response 값을 생성한다.

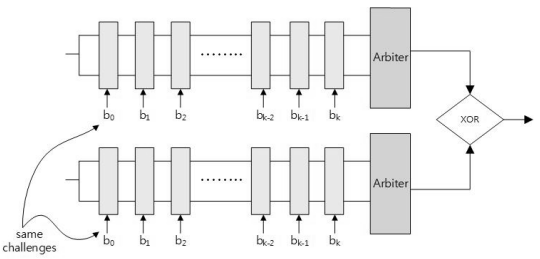


Fig.5. Structure of XOR Arbiter PUF

Fig.5.는 2개의 arbiter PUF를 병렬로 놓아 만든 XOR PUF이다. 각 challenge bits는 두 arbiter PUF에 동일하게 들어가며 그 출력 결과를 XOR하여 response 1 bit을 생성한다.

2.2.4 Lightweight PUF

Lightweight PUF[8] 역시 모델링 공격을 회피하기 위해 제시된 구조로서, Input network, output network, interconnect network, parallel PUF의 4개의 블록 구조로 이루어져 있다.

Input network는 모델링 공격을 어렵게 만들기 위해 parallel PUF에 들어가는 challenge 값을 변환한다. Input값이 d라고 할 때, 이로 인해 생성되는 challenge c는 아래와 같다. 이때, N은 짝수이다.

$$c_{\frac{N+i+1}{2}} = d_i, \text{ for } i=1 \quad (1)$$

$$c_{\frac{i+1}{2}} = d_i \oplus d_{i+1}, \text{ for } i=1,3,5,\ldots N-1 \quad (2)$$

$$c_{\frac{N+i+2}{2}} = d_i \oplus d_{i+1}, \text{ for } i=2,4,\ldots,N-2 \quad (3)$$

Output network는 XOR PUF와 유사하다. Output network는 arbiter PUF에서 나온 결과 R을 O로 Mapping한다. O는 $H(R)$로 정의되며

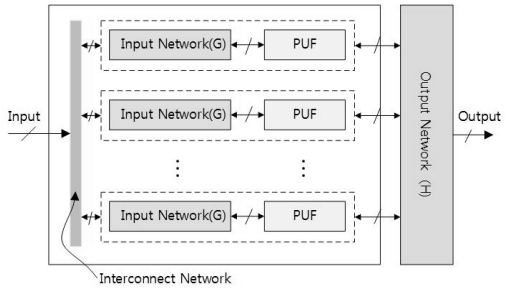


Fig.6. Structure of Lightweight PUF,

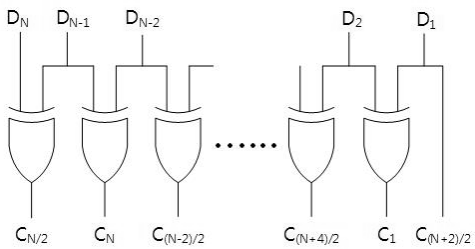


Fig.7. Detailed structure of Input Network

$H: B^{Q} \rightarrow B^{Q'}$, $B = \{0,1\}$, $Q' < Q$ 이다.

$$o_j = \bigoplus_{i=1,\ldots,x} r_{(j+s+i)} \bmod Q, \text{ for } j = 1, \ldots, Q' \qquad (4)$$

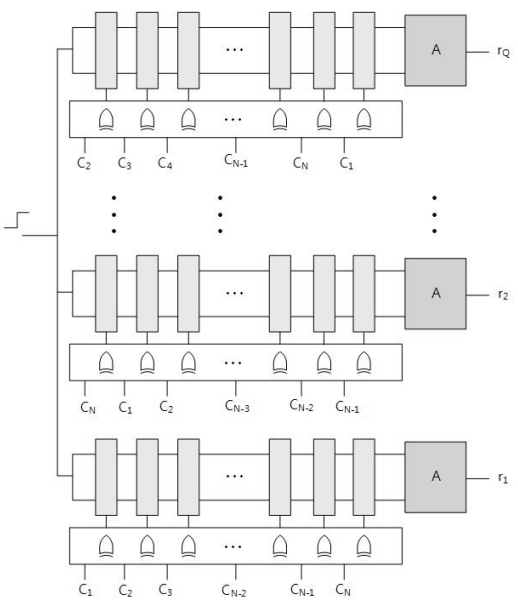


Fig.8. Detailed structure of Interconnect Network

식 (4)에서 ⊕는 parity 생성 함수이며 s는 shifting 단계를 의미하며 x는 parity input 크기이다.

Interconnect network는 Fig.6.에서 가장 좌측에 있는 부분이다. Interconnect network는 각 행의 한 challenge 비트가 다른 행의 다른 challenge 비트와 연결되는 것이다. Interconnection 룰은 다음과 같이 표현된다.

$$c_i^m = c_j^{m+1} \text{ for } i,j \in \Omega, m = 1,2,\ldots,Q-1 \qquad (5)$$

c_i^m은 m번째 행의 i번째 challenge 비트이다. $\Omega = \{1,2,\ldots,N\}$이고 $j = g_m(i)$, $g: \Omega \rightarrow \Omega$의 일대일 치환함수이다. Fig.8.은 interconnect network의 구조를 쉽게 알려준다.

2.3 Ring Oscillator 기반 PUF

Ring Oscillator PUF(RO-PUF)는 인버터를 이용한 회로상의 루프를 이용하여 신호가 계속하여 반전되는 현상을 이용한다[4,7]. RO의 진동주파수는 루프를 구성하는 루프 및 인버터 회로의 딜레이, 그리고 기타 반전 루프를 구성하는 지연시간에 의해 결정되는데, 미세한 공정 차이 혹은 온도 등 여러 가지 요인에 의해 이 지연시간이 달라짐으로써, 동일하게 만들어진 RO라 할지라도 다른 지연 시간 특성을 가지게 된다. 이러한 특성을 이용하여 일정시간동안 RO에 의한 진동수를 측정하여 RO-PUF에 활용하게 된다.

RO-PUF는 대체적으로 Fig.9.와 같은 형태를 가지고 있다. 출력의 안전성을 높이기 위해서, [7]에서는 k개의 RO쌍을 비교하여 그 중 가장 차이가 현저하게 나는 1쌍을 선택하여 출력하는 1-out-k 마스킹 기법이 소개되기도 하였다. 혹은 인접한 두 개의 RO

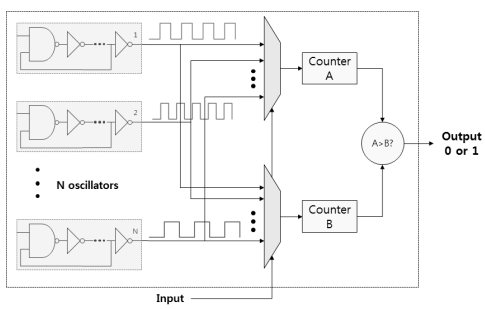


Fig.9. General structure of RO-PUF

는 온도와 같은 주변 환경의 영향도 비슷하게 받을 것이라는 가정 하에 인접한 두 개의 RO를 비교하는 $n-1$ 기법이 제안되기도 하였다[9]. 또한 RO의 루프 내에 멀티플렉서를 삽입하여, 재구성 가능한 루프를 만드는 방법을 제안하기도 하였으며[9], RO 루프에 공급되는 전압을 달리 함으로써 다른 특성을 가지게 하는 방법이 제안되기도 하였다[10].

Maiti 등[11]은 Identity Mapping 이라는 기법을 통해 RO-PUF의 CRP(Challenge Response Pair)를 늘릴 수 있는 방법을 제안하기도 하였다. 일반적으로 서로 다른 두 개의 RO-PUF를 구별하기 위해서는 하나의 RO-PUF가 가지는 주파수들의 순위를 정하여, 순위 정보(각 RO의 주파수 간의 크기에 관한 상관관계)에 따라 칩을 구별하게 되는데, 이 경우 전혀 다른 주파수 집합을 가지는 두 개의 RO-PUF라 할지라도, 주파수의 순위 정보가 같을 경우 동일한 PUF로 인식될 수 있다. 이러한 문제를 방지하기 위해 Identity Mapping이라는 방법을 사용하는데, 이는 둘 이상의 주파수의 조합에 유클리디안 거리를 적용하여 Q-value라는 집합을 만들어 이를 이용해 순위를 적용하는 방법이다. 서로 다른 RO-PUF의 Q-value 집합은 이전의 순위 정보와는 달리 주파수들의 순위 정보가 같더라도 전혀 다른 양상을 보이게 된다. m개의 RO를 이용한 기존의 기법의 경우 $log_2(m!)$의 CRP를 만들 수 있는 반면 Identity Mapping을 이용하면 2^m-m-1개의 Q-value를 만들 수 있다.

2.4 메모리 기반 PUF

2.4.1 SRAM PUF

SRAM(Static Random Access Memory) PUF는 Guajardo[12] 및 Holcomb[13] 등에 의해서 제안되었다. SRAM은 본래 하나의 비트를 저장할 수 있는 셀로 구성된 임시 저장 장치이지만, SRAM의 본질적인 특성을 활용하면 PUF를 구성할 수도 있다. SRAM PUF는 2개의 동일한 인버터로 구성되어있는 SRAM 구조를 이용하여, 전원을 인가할 때의 초기값 변이에 기반을 둔 것이다. Cell의 출력은 전원이 인가되면서 0과 1중 하나의 상태로 바뀌나 어느 상태로 변화하는지는 명확하지 않고, 교차 결합(cross-coupled)된 인버터 회로의 MOSFET중 어느 것이 더 강한가에 따라 달려있다.

2.4.2 Latch PUF

Fig.10.은 교차 결합된 NOR 게이트로 구성된 Latch를 나타낸다. 불안정 상태에서 리셋 신호가 인가되면 두 개의 NOR 게이트 로직의 불일치로 인해 최종적으로 0 또는 1 중 하나로 수렴하게 된다. Latch PUF[14]는 이러한 특성을 활용한다. 또한 NOR 게이트 대신 NAND 게이트로 대치한 Latch PUF 구현도 생각할 수 있다. 전원이 인가되었을 때의 초기 값을 이용하는 SRAM PUF와 달리 전원을 통해 PUF를 제어하는 것이 아니기 때문에, 전원이 공급된 상태에서 리셋 신호를 제어함으로써 재수행 가능하다는 장점을 가진다.

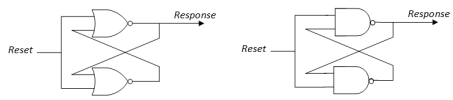


Fig.10. NOR Latch and NAND Latch

또한 [15]에서는 Latch PUF의 유일성(uniqueness)을 강화하기 위한 방법을 제안하였다. 리셋 비인가에 따른 Latch 셀의 값이 0/1로 고정되거나 0/1 이 랜덤하게 발생하는 3가지 상태를 가진다는 사실을 확인하고 이를 각각의 셀의 출력 값으로 사용한다. 만약 n개의 Latch 셀이 주어진다면 3^n개의 출력을 만들어 낼 수 있게 된다.

또한 버스 구조에서의 유사한 특성을 활용한 Buskeeper PUF가 제안되었다[16]. Buskeeper 또는 Bus-holder라고 알려져 있는 이 소자는 latch이지만 제어신호가 없는 구조이다. 드라이버가 여러 개인 버스 구조에서 주로 사용되며 버스에 마지막으로 로드된 값을 저장한다. 이러한 특성을 사용하는 Buskeeper PUF는 패스가 짧고 컨트롤 신호가 따로 없어 1/4 latch 정도의 면적만을 요구한다.

2.4.3 플래쉬(Flash) 메모리 PUF

[17]에서는 일반적인 플래쉬 메모리를 아무런 조작 없이 PUF로 사용할 수 있는 기법이 제안되었다. 플래쉬 메모리 공정상 발생하는 변화(variation)에 따라서 플래쉬 메모리 셀마다 값이 프로그래밍 되는 시간이 달라지며 이러한 특성을 활용하여 플래쉬 메모리 PUF를 구현한다. 메모리 페이지에 대한 프로그래밍 되는 시간에 따라 순서를 배정하고 순서가 특정 임계

값 t보다 크면 1, 작으면 0을 출력하도록 한다.

2.4.4 Memristor PUF

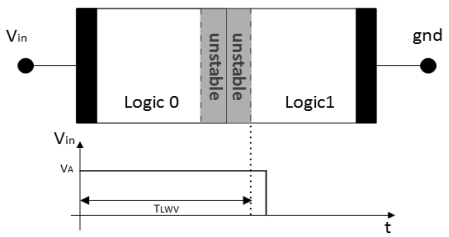


Fig.11. T_{LWV}, minimum time for logic state change, and unstable interval.

Memristor는 저항, 커패시터, 인덕터에 이어 4번째 기본 소자로서 역할을 수행하며 이전 상태를 기억하는 성질을 가지기 때문에 메모리로 사용가능하다. Fig.11.에서 Memristor 메모리는 인가되는 전압에 따라 로직의 상태가 결정된다. 로직 0이나 로직 1을 결정하는 최소한의 시간 T_{LWV}과 최소한의 전압 V_{LWV}이 존재하며 두 조건을 만족하지 못할 경우 메모리는 0과 1중 어느 로직을 가질지 알 수 없는 상태가 된다. Memristor PUF는 이러한 특성을 사용하며 이는 SRAM 이외의 새로운 메모리 구조를 이용한 PUF 구현 사례이다[18].

2.5 재구성 가능한(Reconfigurable) PUF

[19]에서는 2가지 Reconfigurable PUF를 소개하였다. 변화 이전의 상태로 돌아갈 수 없으며 완전히 새로운 임의성을 가지는 변화를 재구성(Reconfiguration)으로 정의하며 이를 통해 PUF의 CRP 쌍을 재구성하는 것이 주된 목표이다.

첫 번째 소개된 방법은 optical PUF에 재구성 기법을 적용하였으며 레이저 빔을 통해 CRP 획득을 위한 광학매체에 변화를 일으킨다. 이러한 광학 매체의 변화는 challenge에 대응하는 response의 변화를 가져온다.

두 번째 방법은 열 변화에 대한 상변화 기록(phase change memory : PCM) 기술을 사용한다. PCM은 미래의 비휘발성 메모리로써 주목 받고 있다. 결정 상태(crystalline state)와 비정질 상태(amorphous state)를 가지는 저항 재료를 셀로 활용한다. 주사된 열에 따라 결정의 상태가 변화하며 결정 상태에서는 저항 값이 논리적으로 1이 되며 비경질 상태에서는 논리적으로 0을 이루게 된다. 이러한 결정의 변화를 측정하여 데이터의 상태를 판단한다.

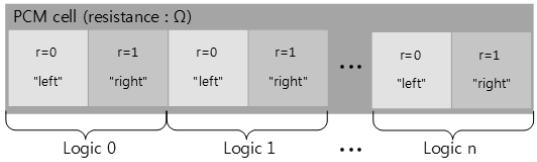


Fig.12. PCM-based PUF for 2 logic states

Fig.12.에서와 같이 결정 상태에 대한 로직 값은 다시 left, right로 나누어지며 보통 r 값을 측정하여 상태를 판단한다. 메모리 읽기 작업 시에는 r 값을 쉽게 얻을 수 있지만 메모리 쓰기 작업이 이루어질 때 r 값을 제어하기 힘든 점을 이용하여 PUF로 사용한다.

삭제 가능한(Erasable) PUF는 한번 사용한 CRP를 재사용하지 못하도록 특정 CRP만 지울 수 있는 특징을 가진다. 재구성 가능한 PUF의 경우 전체 CRP 리스트의 변경을 가하는 것에 반해 삭제 가능한 PUF는 하나의 CRP 쌍만을 변경하는 것이 특징이다. [20]에서는 크로스바(Crossbar) 구조의 삭제 가능한 PUF를 제안하였다. Fig.13.과 같이 크로스바를 구성하고 ALLIE(Aluminum-induced layer exchange) 결정으로 이루어진 다이오드를 스위치로 사용한다. ALLIE 결정의 경우 실리콘과 알루미늄의 비율이 매우 랜덤하기 때문에 이러한 특징에 따라 측정되는 전류 $I(V)$도 임의성을 가진다. 이러한 특성은 PUF의 엔트로피를 높이는데 일조한다.

삭제 동작을 수행하기 위해서는 Fig.13.에서Word Line과 Bit Line을 선택하고 다이오드의 역방향으로의 전압을 4~5 V까지 순간적으로 높여 준다. 이를 통해 선택한 ALLIE 다이오드만을 파괴하는 것이다.

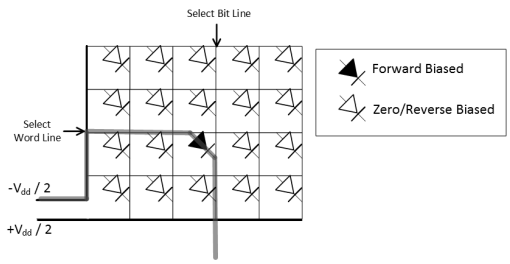


Fig.13. Crossbar structure of diodes.

III. 3. PUF 활용 사례

3.1 식별(Identification)

PUF를 활용하는 첫 번째 사례로 식별이 있다[21]. PUF를 식별에 활용하면, 각 개체에 유일한 ID를 할당하기 위해 외부에서 ID를 만들어서 주입하는 처리과정을 생략할 수 있으며, 개체 내부에서 PUF를 사용하여 ID를 생성할 수 있다. 더불어, ID를 저장하기 위해 내부에 비휘발성 메모리를 두지 않아도 되므로, 비용절감도 기대할 수 있다. 식별을 위해서는 PUF의 CRP 리스트를 데이터베이스에 저장해야 한다. 이 후, PUF를 통해 생성된 CRP와 데이터베이스에 저장된 CRP 리스트 값을 비교함으로써 식별이 가능하다.

3.2 Key Generation

암호화를 위해 키를 생성하는데 PUF를 사용할 수 있다[22]. 이는 하나의 PUF에 동일한 challenge를 보내면 항상 같은 response를 얻을 수 있다는 사실에 기인한다. 그러나 PUF는 항상 어느 정도의 비트 에러를 만들어내기 때문에 오류 수정이 필요하다. 불안정한 PUF의 출력으로부터 안정적인 키값을 출력하기 위해 에러 정정과 Fuzzy Extractor(FE)와 같은 방법이 주로 사용된다[23].

3.3 Random Number Generation

[24]에서는 PUF를 이용한 난수 생성기를 제안하였다. 해당 연구에서는 PUF에서 활용하고 있는 공정에 따른 미묘한 변화(variance)를 랜덤니스의 원천으로 사용될 수 있음을 제시하였다. Fig.14.는 PUF를 이용한 난수 발생기의 예를 보여주고 있다. 본 난수 발생기에서는 PUF가 0과 1의 비율을 골고루 출력해 주게 하는 challenge를 "unpredictable challenge"라고 하며, PUF의 출력에서 0과 1의 비율을 확인하여, 출력에 편향(bias)이 없도록 하는 challenge를 PUF의 입력으로 계속하여 사용할 수 있도록 하는 기법이다. 이 unpredictable challenge는 Last Challenge라는 스토리지에 저장되어 PUF의 challenge로 사용되는데, N bit의 출력을 뽑아서 response의 바이어스를 확인하여 e보다 클 경우에는 N bit를 다시 $M-1$번 뽑아 보고, 총 M번

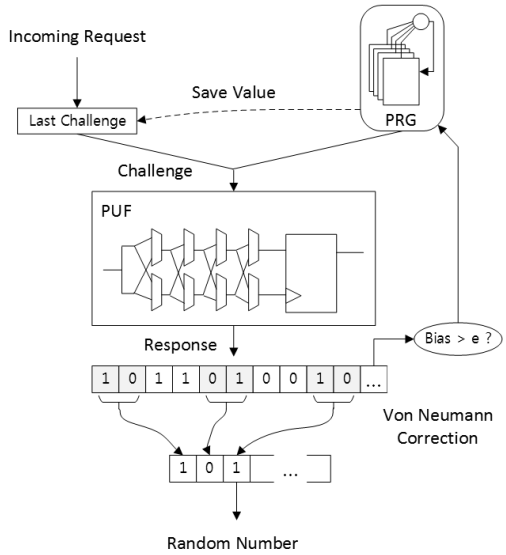


Fig.14. PUF-based random number generator

의 모든 수행에서 편향이 생길 경우에 challenge를 PRNG로부터 새로 뽑아 Last Challenge에 저장하여 다음 challenge로 활용하게 된다. 만약에 편향성 테스트에 통과할 경우 N 비트에 Von Neumann Correction을 적용하여, 두 비트씩 읽어 두 비트가 다른 경우에만 첫 번째 비트를 최종 난수 출력으로 활용하게 된다.

3.4 Public Key Cryptography

PUF를 Public-Key Cryptography처럼 쓰는 방법을 Beckmann과 Potkonjak [25]이 제안하였다. 그들은 XOR게이트의 행렬을 이용하여 Fig.15. 와 같은 Public PUF(PPUF) 라는 구조를 제안하였고, 이를 이용한 PKC를 제안하였다. 그림의 PPUF는 설명을 돕기 위한 예시로써, 실제로는 더 넓은 행과 열을 가지는 행렬로 구성이 된다.

Fig.15. Example of PPUF

PPUF는 회로의 초기 값에 대해 안정 상태(steady-state)에 도달한 이후에 입력을 바꾸어 그 다음 안정 상태에 도달하기 이전의 특정 시점에서의 출력이 각 게이트의 변화 시간(transition time)에 따라 결정이 된다. 이러한 특성을 이용해 공정이나 환경의 영향을 받는 PPUF를 만들 수 있고 이를 통해 PKC를 구성한다. Table 1.은 각 게이트의 입력에 따른 출력 변화에 필요한 시간에 대한 예시를 나타내고 있다. Alice와 Bob 사이에 비밀 키를 교환하는 예시를 살펴보자. 먼저 Alice는 개인키에 해당하는 그림 상의 PPUF를 가지고 있다고 가정하고, Table 1.과 같은 게이트 레벨의 회로 특성 표를 공개키로 사용한다고 가정한다. Bob은 이 표를 이용해서 초기의 안정 상태를 만드는 값 $x_0 = 01$와 이후 변경해주는 값 $x_1 = 10$을 정하고, 시간 $t = 2.7ps$를 정한다. 표에 있는 이 값들을 이용해 시뮬레이션을 수행하고, 결과 값이 $t = 2.7ps$에서 $y = 10$이라는 결론을 내린다. Bob은 x_0, t, y를 Alice에게 보내고, Alice는 x_1을 찾으려고 한다. Alice는 초기 값 x_0에서 $t = 2.7ps$에서 $y = 10$을 출력하도록 하는 값을 찾기 위해 x_1값을 바꾸어가며 PPUF를 반복해서 실행하게 되고, 해당 값이 $x_1 = 10$이란 것을 찾아내게 된다. 이러한 관점에서 개인키는 PPUF 회로 그 자체가 되는 것이다. 공격자의 입장에서는 Alice가 PPUF 회로를 가지고 동작시키는 것보다 불리하며, Bob이 한 번의 시뮬레이션을 통해 결과 값을 얻는 것에 비해서도 불리하다. 이러한 방법을 통해 PUF를 PKC로 사용이 가능하다는 것을 보여준다.

Table 1. Gate delay from input to output(ps).

	Input1	Input2		Input1	Input2
E	0.86	0.95	F	1.24	0.96
C	1.11	0.90	D	0.78	0.71
A	0.93	1.01	B	1.12	0.88

3.5 Oblivious Transfer

Oblivious Transfer(OT) 프로토콜은 여러 개의 정보를 가지고 있는 송신자와 그 중 하나를 전송 받고 싶어 하는 수신자 사이의 프로토콜이며, 수신자는 송신자로 하여금 어떤 자료를 요구 했는지를 모르게 전송 받고자 하며, 송신자는 수신자가 요구한 자료 외에 다른 자료는 수신자가 모르게끔 하는 프로토콜이다. [26]에서는 PUF를 사용한 OT-프로토콜을 제안하였으며, 이는 초기화 단계와 서브세션 단계로 구성된다. 초기화 단계에서는 수신자가 PUF를 가지고 CRP 리스트 $\mathcal{L}$을 생성하며 생성에 사용한 PUF를 송신자에게 전달한다. 이 후 서브세션 단계의 경우 Fig.16.과 같이 진행된다.

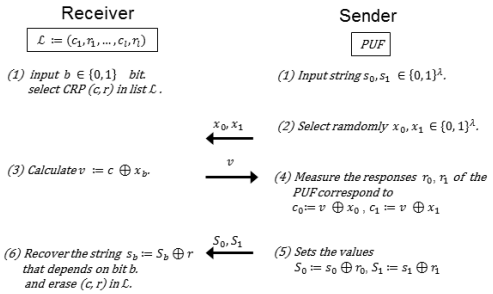


Fig.16. Subsession phase of OT protocol.

s_0, s_1은 송신자가 전달하기를 원하는 값이며 b 값은 수신자로 하여금 송신자가 가지고 있는 s_0, s_1 중 하나를 선택하기 위해 필요하다. 이 때 b 값은 단지 전달 받은 x_0, x_1을 선택하는데 사용되며 송신자에게 전달되지 않기 때문에 송신자는 수신자가 원하는 값을 알지 못하며 단지 v 통해 유추한 두 개의 CRP (c_0, r_0), (c_1, r_1)을 가지고 S_0, S_1을 생성하여 전달할 뿐이다. 이후 수신자는 전달 받은 S_0, S_1 중 원하는 값만 response를 사용하여 획득가능하다.

3.6 Bit Commitment

Bit Commitment(BC) 프로토콜은 OT 프로토콜을 사용한 또 다른 프로토콜로써, 여러 가지의 선택지 중 한 가지를 골랐을 때, 어떤 것을 골랐는지를 특정 시점에서는 알 수 없다가 나중에 이를 확인할 수 있는 기법이다. [26]에서는 PUF를 이용한 BC 프로토콜에 대해서도 언급하고 있다. BC 프로토콜의 경우 b 비트를 기밀성을 유지한 상태로 상대에게 전달하기 위한 방법이며 b 비트를 확인하고 싶은 시점에 확인 가능하도록 하는 특성(commitment)을 가진다. BC 프로토콜의 경우 OT 프로토콜과 동일하지만 단지 송신자와 수신자가 정반대로 수행되며 b 비트를 확인하고 싶은 시점에 BC 프로토콜의 송신자는 b와 함께 v 값을 전달한다. 이 때 v 값은 CRP와 XOR 연산을

통해서 b 값을 생성하기 때문에 송신자가 이전에 선택한 값이 b가 맞는지를 확인하는 용도로 사용된다.

하지만 OT 프로토콜과 더불어 BC 프로토콜은 XOR 연산을 통해 계산된 값을 전달하기 때문에 중간자 공격에 취약하며 이를 보안하기 위하여 Interactive Hash 기법이 제안되기도 하였다 [27].

3.7 Key Exchange

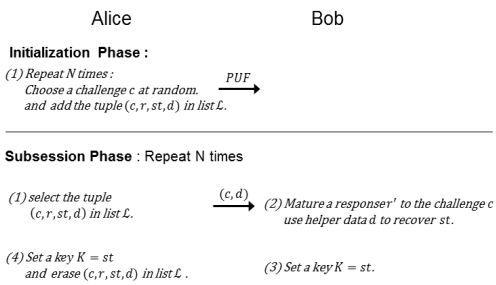


Fig.17. KE Protocol.

Fig.17.에서는 Key Exchange(KE) 프로토콜에 대한 절차를 보여 준다[26]. c와 r은 각각 challenge와 response를 나타내며, d는 FE에 사용되는 helper data, st는 FE의 출력을 나타낸다. 초기화 단계에서는 PUF를 사용하여 CRP 리스트 및 키로 사용될 값을 튜플로 저장하며, 서브세션 단계에서는 Alice는 c, d 값을 전달하며 Bob은 c를 사용하여 response r'를 획득하고 이후 r'과 d를 통해 키로 사용할 st를 생성하는 것이 가능해진다. 마지막으로 Alice는 사용한 튜플을 리스트 $\mathcal{L}$에서 제거한다.

IV. PUF 공격 기술 연구 동향

4.1 기계 학습 기반 모델링

PUF의 일부 CRP를 이용해서 나머지 CRP도 추측하기 위해서, 기계 학습 알고리즘을 수행하여 CRP를 추측해내는 공격이 시도되었다. [28,29]에서는 다양한 PUF에 대하여, 잘 알려져 있는 기계 학습 알고리즘을 이용하여 시도한 모델링 공격의 결과를 제시하고 있다. Table 2.는 대표적인 공격 결과가 정리되어 있다. 해당 연구는 기계 학습이 PUF를 모델링함에 있어서 얼마나 효율적인 도구인지를 보여주는 대표적인 사례라고 할 수 있다. [28]에서는 Arbiter PUF를 비롯해 XOR, Lightweight, Feedforward-PUF, 그리고 RO-PUF를 대상으로 비트 출력 모델을 제시하고, 시뮬레이션된 CRP를 대상으로 모델링 공격을 수행하였다. Arbiter PUF의 경우 지지 벡터 머신(SVM, Support Vector Machine), 로지스틱 회귀(Logistic Regression), 진화 전략(Evolution Strategy) 등의 기계 학습 알고리즘을 이용하여 모델링을 시도하였고, 그 중 로지스틱 회귀를 이용할 경우 가장 좋은 결과를 보였다고 한다. 64단계의 Arbiter-PUF는 Intel Quadcore Q9300을 이용한 실험에서, 단 640개의 CRP를 이용해 10 ms 만에 학습을 완료하였으며, 95%의 확률로 예측을 할 수 있었다고 한다. 2,555개의 CRP로 학습을 하면, 130ms의 학습 시간으로 99% 예측률을 달성할 수 있었다. Arbiter PUF의 단계를 늘리면, 동일한 확률로 예측하기 위해 더 많은 CRP를 필요로 하고, 학습 시간도 많이 필요하였다. 64단계의 XOR Arbiter PUF의 경우에는 4개의 XOR을 사용하였을 때, 12,000개의 CRP로 약 4분 만에 학습을 완료하여 99%의 예측 율로 예측할 수 있었다. 단계를 128로 늘릴 경우 2배의 CRP와 약 3시간의 학습 시간을 필요로 하였으며, XOR수를 5개로 늘리자 필요한 CRP의 수가 80,000개로 늘었으며, 학습시간도 2시간 8분에 이르렀다. 64단계 Lightweight PUF의 경우 3개의 XOR을 사용할 경우 6,000개의 CRP를 이용해 8.9초의 학습시간으로 99%의 확률로 예측할 수 있었다. 64단계 Feed-Forward PUF의 경우 6개의 FF-Loop를 사용할 경우 50,000개의 CRP를 이용해 약 8분 정도에 학습을 마치며, 97.7%의 확률로 예측을 할 수 있었다. 다른 Arbiter PUF의 경우 로지스틱 회귀가 가장 좋은 공격 결과를 보였지만, Feed-Forward PUF의 경우 진화 전략 알고리즘이 가장 좋은 공격 결과를 보였다.

RO-PUF를 공격할 수 있는 가장 직관적이고 쉬운 방법은 모든 CRP를 읽어 들이는 것인데, 이것은 k개의 ring oscillator를 이용해서 얻을 수 있는 CRP의 개수가 $k(k-1)/2$개 밖에 되지 않기 때문이다. 혹은 공격자가 적절한 방법으로 CRP를 선택할 수 있다면 정렬 알고리즘을 이용해 RO의 주파수를 오름차순으로 정렬하는 것이 가능해진다. 이러한 방법을 이용하면 각 주파수들 간의 우위 관계를 알 수 있으며, 남은 CRP의 모든 출력을 100%로 예측할 수 있다. 실험에서는 256개의 RO를 사용하는 RO-PUF에 대해서 약 14,000개의 CRP를 이용해서 99%로 예측하

는 것이 가능함을 보여주었다.

[29]에서는 [28]에서의 연구를 더욱 확장하였는데, 실제 FPGA(Field Programmable Gate Array)와 ASIC(Application Specific Integrated Circuit) 칩을 이용하여 구현한 PUF에 대하여 모델링 공격을 수행하였다. 실제 칩을 이용한 공격에는 Arbiter PUF와 XOR PUF에 대해서만 수행하였으며, 64단계의 Arbiter PUF의 경우 6,500개의 CRP로 FPGA의 경우 $830ms$, ASIC의 경우 $760ms$에 학습이 가능하였으며, 99%의 확률로 예측이 가능하였다. 64단계의 5-XOR Arbiter의 경우 FPGA와 ASIC 각각 78,000개의 CRP로 약 39분과 18분의 학습 시간을 통해 99%의 확률로 예측할 수 있었다.

Table 2. The main result of machine learning-based modeling attack. LR is Logistic Regression, ES is Evolution Strategy, QS is Quick Sort, S means simulation, F means FPGA, and A means ASIC.

PUF Type	Mach. Learn. Algo.	No. of Stage /RO	CRP Src.	No. of CRPs ($\times 10^3$)	Training Time	Pred. Rate
Arbiter	LR	128	S	39.2	2.1s	99.9%
		64	S	0.64	10ms	95%
		64	S	2.555	130ms	99%
		64	F	6.5	830ms	99%
		64	A	6.5	760ms	99%
4-XOR Arbiter	LR	128	S	24	2:52hr	99%
		64	S	12	3:42min	99%
5-XOR Arbiter	LR	64	S	80	2:08hr	99%
		64	F	78	39min	99%
		64	A	78	18:09min	99%
3-Light weight	LR	128	S	15	40sec	99%
		64	S	6	8.9sec	99%
5-Light weight	LR	128	S	1000	267day	99%
		64	S	300	13:06hr	99%
6-FF Arbiter	ES	128	S	50	3:15hr	99.1%
		64	S	50	7:51min	97.7%
Ring Osc.	QS	256	S	14.06	-	99%
		512	S	36.06	-	99%

4.2 부채널 공격

부채널 공격은 디바이스에 대한 물리적 공격의 일종으로써, 디바이스가 동작하는 동안 누설되는 정보를 수집하여 공격에 활용하는 것이다. 예를 들어, 디바이스가 동작하는 동안 소모되는 전력 소비량이나 전자기파를 이용하면, 디바이스가 어떤 동작을 수행중인지를 추측할 수 있다. 일반적으로 디지털 회로를 구성할 때 사용하는 방법인 CMOS(Complementary Metal Oxide Semiconductor)의 경우 회로를 구성하는 게이트가 상태를 바꿀 때 마다 상당한 량의 전력 소모가 일어나는데, 이러한 전력 소모는 회로 상에서 처리되고 있는 데이터와 동작에 따라 다른 패턴을 보이게 된다. 이러한 특성을 이용하면, 칩에서 발생하는 소모 전력 정보를 이용해 칩 상에서 처리되고 있는 비밀정보를 알아낼 수 있다. 많이 알려진 공격으로는 SPA(Simple Power Analysis), DPA(Differential Power Analysis), SEMA(Simple Electromagnetic Analysis), DEMA(Differential Electromagnetic Analysis) 등이 있다.

PUF 역시 회로로 구성되는 하드웨어의 일종이기 때문에, 부채널 공격의 가능성을 가지고 있다. 실제 PUF를 대상으로 하는 여러 가지 공격 사례가 제시되었다. [30]에서는 PUF를 이용해 암호화에 사용되는 비밀 키를 생성해 내는 FE를 대상으로 하는 부채널 공격을 시도하였다. FE 내부에는 불안정한 PUF의 출력의 에러를 정정해 주는 에러 정정 과정이 포함되는데, 연구에서는 많이 사용되는 BCH 코드 내부에서 PUF 출력 값에 따라 오증(Syndrome) 계산 과정에서 연산이 다르게 수행되는 점을 이용하여 Differential Template Attack을 수행하였다. 해당 연구에서는 TI MSP430 마이크로 컨트롤러에 구현한 RS Decoder에 대한 공격 실험에서 50개 정도의 전력 측정 트레이스를 이용해 공격이 가능함을 보였다.

[31]에서는 FE내에서 사용되는 해쉬 알고리즘을 대상으로 부채널 공격을 수행하였다. FE 알고리즘에서는 출력되는 키의 분포를 균일화하기 위해서 해쉬 알고리즘을 사용하게 되는데, 해쉬 알고리즘에서 부채널 정보를 흘리게 되면 PUF의 출력을 알아낼 수 있게 되고, 해당 출력을 통해 FE를 통해 추출되는 키를 추측할 수 있게 된다. 해당 연구에서는 특정 FE 구현에서 사용된 Toeplitz 해쉬 함수가 단순한 LFSR 구조로 구현 시 매우 효율적이지만, PUF response에 따라 동작이 확연히 달라져 부채널 공격에 취약한 점을 이용하였다. 해당 공격에서는 Xilinx Spartan 3E FPGA 칩에 511개의 RO를 사용하는 RO-PUF와 64비트의 키를 생성하는 Toeplitz 해쉬 함수를 구현한 뒤 동작 중에 발생하는 EM을 측정하여 부채널

공격에 활용하였다. 공격에서는 단 200개의 트레이스만을 가지고도 10개 내외의 비트만 틀렸을 뿐 나머지 비트들을 성공적으로 추출하였으며, 8,000개의 트레이스를 사용하였을 때, 오류 없이 비트를 복원할 수 있었다.

또 사례로는 RO를 기반으로 하는 PUF를 대상으로 하는 부채널 공격 기반의 모델링 공격이 있었다[32]. PUF가 가져야 할 특성에는 PUF 칩에 대한 변조(tamper) 공격을 시도할 때에 PUF의 특성이 완전히 바뀌어져야 한다는 특성이 있지만, 해당 연구에서는 FPGA 칩의 패키지를 개봉할 경우에 실제로는 그 특성이 완전히 바뀌지 않는 것을 실험을 통해서 밝혀내었다. 실험에서는 패키지를 제거하기 전과 제거한 후에 FPGA 상에 구현된 RO의 주파수의 차이를 비교하였으나 개봉전과 개봉 후에 거의 차이가 없음을 보였다. 이를 이용해 FPGA 칩의 바닥면 패키지를 제거한 뒤에 EM Probe를 좀 더 칩에 가까운 곳에 접촉 시켜, 칩 다이에 X, Y축으로 움직일 수 있도록 하여, 다이 상에서 EM 지도(cartography)를 작성하는 것을 가능하게 하였다. 부채널 공격을 위해서 이들은 9개의 RO를 FPGA에 구현하고, 각 challenge에 대해 n, $n+1$ 번째 RO 두 개를 선택하여 response를 출력되도록 하였다. RO-PUF가 수행되는 동안 발생하는 EM 신호를 Fourier Transform을 통해 주파수에 따른 진폭으로 변환하고, 각각의 challenge에 대해 어떤 주파수가 발생하는 지를 스펙트럼을 통해서 확인할 수 있다. challenge를 n, $n+1$번째 RO를 선택하도록 하였기 때문에, 인접한 challenge 사이에는 중첩을 확인할 수 있으며, 전체 RO에 대한 스펙트럼 리스트를 생성함으로써 전체 CRP에 대한 모델을 만드는 공격이다.

4.3 복제 공격

PUF가 가져야 할 특성 중 가장 중요한 특성은 바로 복제 불가능(Unclonable)에 대한 특성일 것이다. 하지만 최근 SRAM을 이용한 PUF에 대한 복제 공격이 제시되었다[33]. SRAM PUF의 CRP를 알아내는 방법은 비교적 단순한데, 단순히 메모리 주소를 주고 데이터를 읽어 오는 방법이지만, 이러한 동작에 대비한 방어책에 의해 메모리 인터페이스를 통해 CRP를 획득이 어려운 상황을 가정해 공격을 시도하였다. 해당 연구에서는 SRAM의 주소에 따른 초기 값을 알아내기 위해 SRAM에서 발생하는 광자 방출(photon emission) 현상을 관찰하여 메모리 값을 얻어내는 방법을 활용하였다. SRAM의 경우 MOS(Metal Oxide Semiconductor)를 통해 구현되는 경우가 많은데, MOS의 경우 off, linear, saturation의 세 가지 동작 모드를 가지게 된다. 이때 saturation 영역에서 디바이스의 공핍 영역에 캐리어가 지나갈 만큼 충분한 채널이 생성되지 않았을 때 잉여 캐리어가 빛의 형태로 방출되며 광자 방출 현상이 발생하게 되는데, CMOS 칩의 경우 이 순간이 매우 짧기 때문에, 이 순간을 포착하기가 어렵다. 이를 포착하기 위해 칩의 초기 전원 인가 동작을 반복한 후 Near Infrared CCD를 이용해 빛의 방출을 포착하여 SRAM PUF의 CRP를 획득하였다.

이후 동일한 SRAM PUF로 복제하기 위해 집속이온빔(Focused Ion Beam)을 활용하였다. 이는 칩의 뒷면을 통해 회로를 수정하는 기법으로써, 칩의 전원 인가 후 초기 동작을 수정하기 위해 이온빔을 이용해 정밀하게 n-well을 깎아 내거나, 혹은 컨택(contact)을 제거하여 고정된 값이 출력되게 하는 방법이다. n-well을 깎아 낼 경우 초기 특성만 바꿀 뿐 메모리로서의 기능 (읽기, 쓰기)에 문제가 없지만, 컨택을 제거할 경우 메모리로서의 기능은 상실한다. 이러한 방법으로 ATmega328P 마이크로 컨트롤러 내의 SRAM을 이용한 SRAM PUF를 복제하는 것에 성공하였다. 비록 이러한 공격을 수행하기 위해서는 고가의 장비들이 필요하지만, PUF 에 대한 복제 가능성을 보여 주었다는 것에서 의미가 있다고 할 수 있다.

4.4 프로토콜 공격

[34]에서는 앞서 살펴본 PUF를 사용한 OT, BC, 및 KE 프로토콜에 대한 공격 방법을 제안하였다. [26]에서는 PUF를 이용한 OT, BC 및 KE프로토콜을 제안하였지만, [34]에서는 제안되는 프로토콜에 있어 PUF에 대한 다른 공격 모델을 가정하고 있어 서로 비교하는 것이 어려움을 언급하였다. 따라서 [34]에서는 PUF 기반 프로토콜에 대한 공격 모델은 분석 하였고, 이를 기반으로 PUF를 이용한 OT, BC 및 KE 프로토콜의 안전성을 분석하였다.

[34]에서는 PUF 기반 프로토콜의 공격 모델을 Stand-Alone, Good PUF Model, PUF Re-Use Model, 그리고 Bad PUF Model 로 분류하였다. Stand-Alone, Good PUF Model 은 안

전한 모델을 대표하고 있으며, 프로토콜 수행 중에 단 한번만 수행되고, 이후에는 접근이 불가능한 PUF 모델이다. PUF 분배 시에는 사전에 명시된 사용자에게만 전달되며, PUF 하드웨어에 대한 임의의 조작이나 변경은 불가능하다고 가정한다. 따라서 공격자는 단지 프로토콜을 임의로 수정하거나 프로토콜 임의로 개입하는 것만이 허용된다. 따라서 이 모델은 실제 응용에서 거의 적용 불가능 하지만, PUF 프로토콜 공격을 위해 기본적으로 가정할 수 있는 환경으로 사용된다. PUF Re-Use Model은 프로토콜 수행 후 다른 세션에 대한 추가적 접근을 허용하는 PUF 공격 모델이다. Bad PUF Model은 PUF 하드웨어의 특징을 변경하거나 변조하는 것을 허용하는 모델이다. 외관상으로 PUF처럼 보이지만, 내부에는 시뮬레이션 가능한(복제 가능한) 형태로 구성되어 있어 입출력의 형태만으로는 Bad PUF인지 아닌지를 따지는 것이 불가능 하다.

〔34〕에서는 Re-Use 모델을 사용할 경우 OT 및 BC 프로토콜의 초기와 단계 이후 서브세션 단계가 끝난 뒤 PUF의 CRP를 획득 가능 하다고 가정할 때, 도청한 CRP를 이용해 송신자가 송신한 모든 값을 알아 낼 수 있음을 제시하였다. 다만 공격자는 수신자가 선택한 값은 알아내지 못한다. 또한 Bad PUF Model을 이용할 경우 프로토콜 과정에서 도청한 값과 시뮬레이션으로 얻은 response 값으로 송신자가 전달한 값을 모두 알아낼 수 있음을 제시하였다. KE 과정에서는 Re-Use 모델을 사용하면, 단순히 프로토콜 상에서 전달된 challenge 값을 다시 PUF에 접근하여 query를 함으로써 response를 통해 세션 키를 취득할 수 있음을 지적하였다.

〔35〕에서는 PUF가 ATM(Automated Teller Machine)과 PUF가 장착된 카드를 이용해 은행 사이에서 세션 키를 나누어 가지는 프로토콜을 제안하였다. 하지만 〔20〕에서는 해당 프로토콜에서 ATM/PUF 사이의 통신을 도청할 수 있고, PUF에 대한 접근이 최소 2회 정도 가능하여 CRP 측정이 가능하면, 프로토콜에서 생성되는 임시키를 알아낼 수 있고, 이를 통해 생성되는 세션 키를 알아낼 수 있음을 지적하기도 하였다.

V. 안전한 PUF 구현을 위한 가이드라인

이 절에서는 안전하게 PUF를 구현하기 위한 가이드라인을 제시하고자 한다. 제시되는 가이드라인은 앞서 소개했던 PUF의 구현 사례와 응용 사례, 그리고 PUF를 대상으로 하는 다음의 가이드라인을 통해서 완벽하게 공격을 방어할 수는 없지만, 공격자로 하여금 공격의 비용을 늘이거나, 공격을 어렵게 할 수 있을 것으로 기대된다.

5.1 모델링 공격에 대처하기 위한 가이드라인

5.1.1 PUF challenge 길이 증가

〔28,29〕에서는 기계 학습으로부터 PUF를 보호하기 위해서는 PUF challenge 길이를 크게 하는 것만으로도 공격에 필요한 시간을 늘릴 수 있음을 시사 하였다. 실제 Arbiter PUF의 경우 모델링 공격에서 단계수를 64에서 128로 늘리는 것만으로 학습에 필요한 CRP를 두 배 이상 필요로 하도록 하였으며, 학습에 필요한 시간도 늘어나게 하였다. 하지만, challenge의 길이가 늘어나는 만큼 하드웨어 비용도 증가하기 때문에, 제공하고자 하는 안전성을 고려하여 크기를 결정하는 것이 중요하다.

5.1.2 비선형성 추가

〔28,29〕에서 가장 기본적인 형태의 Arbiter PUF의 경우 기계 학습을 이용한 모델링 공격에 매우 취약한 것으로 나타났다. 64단계의 Arbiter PUF의 경우 단 640개의 CRP만으로 $10ms$ 내에 학습하여 95% 이상의 예측 율로 공격 가능한 것으로 알려져 있다. 이러한 취약점을 보완하기 위해 XOR PUF, Lightweight PUF, 그리고 Feed-Forward PUF 등이 제안 되었으며, 이들 기법은 모두 Arbiter PUF에 비선형성을 추가하기 위한 방법이다. 이렇게 비선형성을 추가하면 기계 학습에 필요한 시간을 늘려 공격을 어렵게 할 수 있다. 비선형성을 추가하는 방법에는 AND, OR, XOR 연산을 활용하거나, 최소값(Min), 최대값(Max), 다수(Majority)에 의한 방법이 활용될 수 있다.

5.1.3 랭킹 기반의 출력 지양

RO-PUF의 경우 단순히 CRP가 노출되는 것만으로 공격자에게 많은 정보를 줄 수 있다. 예를 들어 $f_1 > f_2$에 대한 응답으로 yes를 얻고, $f_2 < f_3$에 대한 응답으로 no를 얻었다면, 자연스럽게 $f_1 > f_3$에 대한

응답은 yes 라는 것을 추측할 수 있다. 혹은 새로운 challenge가 주어졌을 때, 이전의 CRP로 알 수 있는 RO 간의 크기 관계에서 나타날 수 있는 나머지 크기 관계의 경우의 수를 추측하여 확률적으로 어떤 response가 나올 확률이 높은지를 예상할 수 있다. 이는 일정 수 이상의 CRP를 가지게 되면, RO-PUF를 구성하는 RO 주파수 간의 순위를 추측할 수 있으며, 이와 연관 있는 새로운 질의를 할 경우에 공격자가 기존의 정보를 이용해 쉽게 response를 추측할 수 있음을 의미한다. 따라서 RO-PUF를 사용할 때에는 랭킹 기반의 출력을 지양하거나, 연관 관계에 있는 CRP는 사용하지 않아야 한다.

앞서 2.2절에서 소개한 것처럼 Identity Mapping[11]이라는 기법을 통해 랭킹 기반의 출력을 회피할 수 있다. 이는 둘 이상의 RO 주파수를 조합하여 유클리디안 거리를 측정하여 Q-value라는 집합을 만들어 출력에 활용함으로써, 동일한 RO 주파수 크기 관계를 가지는 다른 RO-PUF라 할지라도, 전혀 다른 CRP를 얻을 수 있도록 하는 기법이다.

[32]에서는 CRP 간의 연관 관계를 추측할 수 없도록 한 번 challenge에 사용된 RO는 다른 challenge에서 사용하지 않는 기법을 제안하였다. 이 기법의 단점은 n개의 RO를 이용해서 만들 수 있는 CRP의 수를 $n/2$로 줄어들게 한다는 점이다. 이러한 단점을 보완하기 위해 [36]에서는 사용가능한 CRP 데이터베이스를 만드는 두 가지 알고리즘을 제시하였다. 첫 번째 방법은 challenge pool C에서 challenge를 랜덤하게 선택한 후 해당 challenge를 데이터베이스에 추가하고, challenge pool에서 해당 challenge와 연관이 있는 모든 challenge를 C에서 제거하는 작업을 challenge pool C가 완전히 소진될 때 까지 반복하는 것이다. 두 번째 알고리즘은 RO들의 pool P에서 랜덤하게 RO를 하나 선택한 뒤, 해당 RO보다 작은 RO에 대해서만 비교를 하는 CRP를 데이터베이스에 추가하고, 선택한 RO를 기준으로 더 큰 그룹 P_f와 작은 그룹 P_s로 나눈 다음 재귀적으로 다시 알고리즘을 수행하는 방법이다. 랜덤하게 선택된 RO보다 작은 RO에 대해서만 비교하는 CRP의 경우 challenge들 간에 연관 관계를 전혀 찾을 수 없으며, P_f와 P_s로 나누어 재귀적으로 수행할 경우 P_f와 P_s에 속해 있는 challenge 사이에는 아무런 연관 관계가 없기 때문에, 결과적으로 서로 연관 관계가 없는 challenge만으로 구성된 데이터베이스를 만들 수 있다.

5.1.4 프로토콜 차원의 CRP 노출 억제

모델링 공격의 근본적인 원인은 공격자가 CRP를 수집할 수 있음으로부터 기인한다. 따라서 challenge와 response가 안전한 채널을 통해 전달되거나 기밀성이 유지되는 상황에서 전달된다면, 모델링 공격을 회피할 수 있다. 하지만, 이와 같은 경우 PUF가 인증이나 키 교환 등에도 활용될 수 있다는 점에서 PUF의 사용 자체가 무의미해질 가능성이 있으므로, 위와 같은 방법이 정답이 될 수는 없다. 따라서 프로토콜 차원에서 CRP가 노출되지 않으면서도 정상적인 프로토콜을 수행할 수 있는 새로운 방법이 연구될 필요가 있으며, 이미 사용한 특정 CRP를 사용 못하게 하거나(Erasable), 전체 PUF의 특성을 완전히 바꿀 수 있는(Reconfigurable) 기능이 요구된다.

5.2 부채널 공격에 대처하기 위한 가이드라인

5.2.1 기존의 부채널 공격 방지 기법 적용

부채널 공격은 기존에 많은 암호화 알고리즘을 구동하는 하드웨어를 대상으로 내부의 비밀정보를 알아내기 위해 행해져왔다. 이와 함께 부채널 공격에 대한 방어책도 끊임없이 연구되어 왔다. PUF를 대상으로 하는 부채널 공격도 기존의 암호 알고리즘을 대상으로 하는 부채널 공격과 기본적인 원리에 있어서 크게 다르지 않기 때문에, 기존의 암호 알고리즘을 대상으로 하는 부채널 공격 방지 기법을 그대로 적용할 수 있다. 대표적인 부채널 공격 방지기법으로는 마스킹(masking)과 하이딩(hiding)이 있다. 마스킹은 내부의 누설되어는 안 되는 중요한 데이터에 대해 동작할 때마다 랜덤하게 바뀌도록 바꾸어주는 기법이다. 이를 위해 기존의 중요한 값에 랜덤한 마스크 값을 적용하여 연산을 수행하고, 연산이 끝난 뒤에 마스크를 제거하여(unmask) 최종 결과 값을 얻는 방식이다. 따라서 동일한 입력이라도 어떤 값이 마스크로 선택되어졌느냐에 따라 중간 값이 다른 값으로 바뀌며, 매번 다른 전력 소모 특성을 보이게 된다. 하이딩은 중요한 정보와 연관이 있는 전력 소모 패턴을 숨기기 위한 기법으로, 칩이 구동 되는 동안 노이즈 전력 소모 패턴을 발생시키는 더미(dummy) 모듈을 구동하거나, 칩에 공급되는 클록을 랜덤하게 변경 하는 등의 방법을 이용하여, 연산이 실행되는 시간 축 상의 위치를 매 실행시마다 변경하는 방법 등을 통해 공격자로 하여금

공격을 어렵게 하는 방법이다. 하이딩을 하는 다른 방법으로는 Dual-Rail Logic[37]을 사용하는 방법이 있다. Dual Rail Logic은 칩 내부의 모든 신호를 2비트로 인코딩한다. 즉 0은 (0,1), 1은 (1,0)과 같은 방법으로 인코딩하여 시그널이 0인지 1인지에 관계없이 항상 동일한 HW(hamming weight)를 가지게 하여, 0인지 1인지에 따라 전력 소모 차이가 나지 않게 하는 방법이다. Dual-Rail Logic을 사용할 경우 칩의 면적이 약 2배 정도로 늘어난다는 단점이 있다.

5.2.2 물리적 보호 장치 적용

PUF는 변조 공격 수행 시에 그 특성이 현저히 변화되어 복제가 불가능하거나, 내부의 구조를 파악하기 어려워야 하지만, 앞서 소개한 부채널 공격사례에서는 FPGA 칩의 뒷면을 개봉하였음에도 그 특성이 현저히 변화하지 않았다. 부채널 공격에서는 이러한 특성을 이용해 EM Probe를 다이에 더 가까이 접근 시켜 정밀한 EM 센싱이 가능하도록 하였으며, SRAM PUF에 대한 복제 공격에서는 PEA를 통해 칩의 동작 상태를 파악하거나, 진속이온빔을 이용해 n-well이나 contact을 절삭하기도 하였다. 칩을 개봉하게 되면, 여러 가지 변조 공격을 수행할 수 있는 교두보가 될 수 있기 때문에, 칩의 개봉을 감지할 수 있는 물리적 보호 장치를 PUF를 제작할 때에 고려해야 할 필요가 있다. 따라서 칩의 바닥면을 제거할 경우 칩의 오동작을 유발하거나, PUF의 특성이 현저하게 바뀔 수 있도록 하는 기법에 대한 연구가 필요하다.

5.2.3 주기를 가지는 설계 지양

RO-PUF의 경우 RO의 주파수에 따라 계속해서 신호가 진동하기 때문에, Fourier Transform을 수행할 경우 각 RO가 발생시키는 주파수를 추정할 수 있다. 따라서 주기를 가지는 설계의 경우에는 주파수에 대한 정보를 쉽게 노출함으로써, PUF의 특성을 누설할 수가 있다. 따라서 PUF를 구현할 때에는 주기에 따라 response가 생성되는 방법은 지양되어야 한다.

5.3 복제 공격에 대처하기 위한 가이드라인

5.3.1 PUF 회로의 은닉

SRAM PUF에 대한 복제 공격은 고가의 장비를 요구하기 때문에, 흔히 일어날 수 있는 공격은 아니지만, 복제 공격의 가능성을 보여주었다. 이는 칩을 개봉하였을 때, 칩 내부의 레이아웃을 살펴봄으로써, SRAM의 위치를 파악할 수 있으며, 주소에 따른 셀의 위치를 파악하는 것이 어렵지 않기 때문에 공격이 가능하게 한다. 따라서 앞서 부 채널 공격에 대비한 가이드라인에서 소개했던 칩의 개봉에 따른 탐지 기법뿐만 아니라, 칩을 개봉하더라도 내부의 PUF 회로를 찾지 못하도록 하는 은닉 기술이 요구된다. 기존에 칩 제작 시 사용되는 메모리 블록보다는 SRAM PUF와 유사한 원리를 활용하는 Butterfly PUF 형태를 사용하고, 이를 기존 디지털 셀 라이브러리로 합성하는 것도 한 가지의 방법이 될 수 있다. 또한 SRAM의 메모리 주소를 스크램블링하거나, 주소를 암호화 방법도 있다. 혹은 여분의 SRAM 셀을 활용하여, 어느 SRAM 블록이 response와 연관관계에 있는지를 숨기는 방법도 활용할 수 있다.

5.3.2 광자 방출(Photon Emission) 억제

PEA(Photon Emission Analysis)는 6T SRAM 셀에서 방출되는 광자 방출 현상을 CCD를 통해 관찰하는 방법으로 이루어졌다. 주로 주소를 가진 SRAM cell의 경우 긴 bitline의 커패시턴스와 읽기 성능을 좋게 하도록 드라이브하게 되는 증폭기의 전류에 의해 필요 이상의 캐리어가 채널을 통해 이동하면서 광자 방출 현상이 발생할 수가 있다. 이는 SRAM 셀 트랜지스터의 드레인에 강한 전압 강하를 만들게 되며, saturation 영역에 오래 머물도록 하게 된다. 또한, bitline 에 연결된 NMOS 트랜지스터는 낮아진 bitline 신호 때문에, saturation 영역에서 동작하게 되며, 이러한 현상으로 인해 SRAM 셀에 많은 전류를 흐르게 하며, 광자가 방출되는 원인이 된다. 따라서 이러한 현상을 최소화하기 위해서는 트랜지스터에 공급되는 전류와 커패시턴스를 적절히 분배할 수 있는 회로 설계 기술이 요구된다.

5.4 프로토콜 공격에 대처하기 위한 가이드라인

5.4.1 공격 모델을 고려한 PUF 프로토콜 설계

PUF를 이용한 프로토콜을 설계할 때에는 반드시 PUF 공격 모델을 고려하여 설계해야 한다. [34]에서 본 것과 같이 프로토콜을 설계할 때에는 공격 모델이

Stand-Alone, Good PUF Model인지, Re-Use Model인지, Bad PUF인지에 따라 프로토콜의 안전성이 달라질 수 있다. 일반적으로 PUF 프로토콜을 제안할 때에는 이상적인 PUF를 대상으로 제시되어서는 안 되며, 현실적으로 PUF가 갖는 취약성을 고려하여 프로토콜을 설계해야 한다.

5.4.2 하드웨어 변조 탐지 및 방지 기법 고려

PUF의 배포 과정에서 Bad PUF와 같이 하드웨어 변조가 발생할 수 있으며, 이러할 경우 각종 프로토콜에서 여러 가지 공격에 무방비 상태로 노출될 수가 있다. 따라서 PUF를 실질적인 프로토콜에 적용되기 위해서는 하드웨어적인 변조가 있었는지를 탐지하거나 혹은 방지할 수 있는 기술이 요구된다. 현재까지는 연구된 바가 없지만, 프로토콜 상에서 Bad PUF임을 판단할 수 있는 수단이나 프로토콜이 고려되어야 할 필요가 있다.

5.4.3 Certifiable PUF 설계

앞서 살펴본 프로토콜 차원에서의 Bad PUF를 판단할 수 있는 기법뿐만 아니라, PUF 자체적으로 신뢰성을 제공할 수 있는 기술이 요구된다. Certifiable PUF[34]는 PUF가 신뢰할 수 있는 환경에서 생산되었으며, 배포 이전에 조작이 없었음을 오프라인 상에서 확인하여 신뢰성을 인증 받은 PUF임을 의미한다. 현재까지 알려져 있는 인증 기법은 없지만, 앞으로 이러한 부분의 연구가 필요할 것으로 고려된다.

5.4.4 Erasable, Reconfigurable PUF 설계

[20]에서는 PUF에 대한 재접근을 통해 CRP를 획득하여 공격에 시도하였다. 이렇듯 PUF에 재접근하여 PUF에 같은 CRP를 얻는 것을 허용하지 않기 위해서는 다른 CRP 특성을 변화하지 않고, 특정 CRP만 제거할 수 있는 삭제 가능한 PUF의 특성이 요구된다. [20]에서는 임의의 전류 특성을 가지는 PN 접합 구조와 cross-bar 형태의 접근 방법을 이용해 삭제 가능한 PUF를 구현한 사례가 있으며, 앞으로도 여러 가지 다양한 방법으로 삭제 가능한 PUF에 관해 연구할 필요가 있다. 또한, CRP가 노출되었거나, 모델링 공격을 받더라도, 내부적으로 PUF의 구조를 완전히 바꿀 수 있는 재구성 가능한 PUF를 적용한다면, 공격자의 공격을 한층 힘들어지게 할 수 있다.

VI. 결 론

PUF는 새로운 디바이스 식별 기술에 대한 가능성을 보여주었고, 그 가능성으로 인해 많은 연구자들이 PUF라는 분야를 연구하게 하였다. PUF는 복제 불가능한 고유한 식별 정보를 생성해 낼 수 있게 함으로써, 기존에 해결되지 못했던 많은 응용에 적용되기도 하였다. 하지만 PUF를 대상으로 하는 많은 공격들이 제시되면서 PUF의 안전성이 심각하게 훼손되고 있으며, 상용화의 저해 요소가 되고 있다.

본 논문에서 우리는 지난 10여 년 동안 연구되었던, PUF를 구현하는 방법과 PUF를 활용하는 방법, 그리고 PUF 및 프로토콜을 대상으로 하는 공격사례들을 살펴보고, 그동안의 연구 동향을 바탕으로 PUF를 안전하게 구현하고 활용하기 위한 가이드라인을 제시하였다. 우리가 제시한 가이드라인은 PUF를 대상으로 하는 공격을 완벽하게 방어할 수 없지만, PUF를 대상으로 하는 공격을 어렵게 할 수 있으며, 공격의 비용을 증가시킬 수 있을 것이라 기대된다. PUF가 안전하게 사용되기 위해서는 아직 해결되어야 할 부분이 많이 있다. 효과적으로 PUF를 재구성 할 수 있는 기술이나, 삭제 가능한 구조의 경우 앞으로도 연구되어야 할 필요가 있으며, certifiable PUF에 관한 연구도 꼭 필요할 것으로 생각된다. 향후 PUF의 출력을 안정화할 수 있는 기술과 다양한 공격으로부터 방어할 수 있는 기술에 대한 끊임없는 연구를 통해 PUF가 이른 시일 안에 실생활에 적용되기를 기대하며 글을 맺고자 한다.

References

[1] R. S. Pappu, "Physical One-Way Functions," Ph.D. Thesis, Massachusetts Institute of Technology, Mar. 2001.

[2] R. S. Pappu, B. Recht, J. Taylor and N. Gershenfeld, "Physical one-way functions," Science, vol. 297, no. 5589, pp. 2026-2030. Sep. 2002.

[3] K. Lofstrom, W. R. Daasch and D. Taylor, "IC identification circuit using device

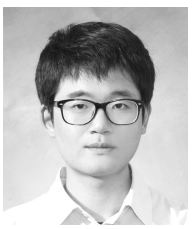
mismatch," Solid-State Circuits Conference, pp. 372-373, Feb. 2000.
[4] B. Gassend, D. Clarke, M. Van Dijk and S. Devadas, "Silicon Physical Random Functions," ACM Conference on Computer and Communications Security, pp. 148-160, Nov. 2002.
[5] B.-D. Choi, T.-W. Kim, M.-K. Lee, K.-S. Chung, D. K. Kim, "Integrated circuit design for physical unclonable function using differential amplifiers," Analog Integrated Circuits and Signal Processing, vol. 66, no. 3, Mar. 2011.
[6] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "Extracting secret keys from integrated circuits," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol 13, no. 10, pp. 1200-1205, Oct. 2005.
[7] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," Proceedings of the 44th Annual Design Automation Conference, pp. 9-14, Jun. 2007.
[8] M. Majzoobi, F. Koushanfar and M. Potkonjak, "Lightweight Secure PUFs," Proceedings of the 2008 IEEE/ACM International Conference on Computer-Aided Design, pp. 670-673, Nov. 2008.
[9] A. Maiti, P. Schaumont, "Improved Ring Oscillator PUF: An FPGA-friendly Secure Primitive," Journal of Cryptology, vol. 24, no. 2, pp. 375-397, Apr. 2011.
[10] S. S. Mansouri, E. Dubrova, "Ring Oscillator Physical Unclonable Function with Multi Level Supply Voltages," 2012 IEEE 30th International Conference on In Computer Design (ICCD), pp. 520-521, Sep. 2012.
[11] A. Maiti, I. Kim and P. Schaumont, "A Robust Physical Unclonable Function With Enhanced Challenge-Response Set," IEEE Transactions on Information Forensics and Security, vol. 7, no. 1, pp. 333-345, Feb. 2012.
[12] J. Guajardo, S. S. Kumar, G. J. Schrijen and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection," Cryptographic Hardware and Embedded Systems, CHES 2007, LNCS 4727, pp. 63-80, Sep. 2007.
[13] D. E. Holcomb, W. P. Burleson and K. Fu, "Initial SRAM State as a Fingerprint and Source of True Random Numbers for RFID Tags," Proceedings of the Conference on RFID Security, Jul. 2007.
[14] Y. Su, J. Holleman, and B. Otis. "A 1.6 pJ/bit 96% Stable Chip-ID Generating Circuit Using Process Variations," Proceedings of the Solid-State Circuits Conference, pp. 406-611, Feb. 2007.
[15] D. Yamamoto, K. Sakiyama, M. Iwamoto, K. Ohta, T. Ochiai, m. Takenaka and K. Itoh, "Uniqueness Enhancement of PUF Responses Based on The Locations of Random Outputting RS latches," Cryptographic Hardware and Embedded Systems, CHES 2011, LNCS 6917, pp. 390-406, Oct. 2011.
[16] P. Simons, E. van der Sluis, and V. van der Leest. "Buskeeper PUFs, a Promising Alternative to D Flip-Flop PUFs," 2012 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp. 7-12, Jun. 2012.
[17] Y. Wang, W. K. Yu, S. Wu, G. Malysa, G. E. Suh and E. C. Kan, "Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints," 2012 IEEE Symposium on Security and Privacy (SP), pp. 33-47, May. 2012.
[18] P. Koeberl, U. Kocabas, and A. R. Sadeghi. "Memristor PUFs: A New Generation of Memory-based Physically Unclonable Functions," Proceedings of the Design, Automation & Test in Europe

Conference & Exhibition (DATE), pp. 428-431, Mar. 2013.
[19] K. Kursawe, A. R. Sadeghi, D. Schellekens, B. Škoric, and P. Tuyls, "Reconfigurable Physical Unclonable Functions - Enabling Technology for Tamper-Resistant Storage," IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). pp. 53-54, Jul. 2009.
[20] U. Rührmair, C. Jaeger and M. Algasinger, "An Attack on PUF-Based Session Key Exchange and a Hardware-Based Countermeasure: Erasable PUFs," Financial Cryptography and Data Security, FC 2011, LNCS 7035, pp. 190-204, Mar. 2011.
[21] B. Gassend, D. Lim, D. Clarke, M. Van Dijk and S. Devadas, "Identification and Authentication of Integrated Circuits," Concurrency and Computation: Practice & Experience, no. 16, vol. 11, pp. 1077-1098, Sep. 2004.
[22] B. Škorić, P. Tulys and W. Ophey, "Robust Key Extraction from Physical Unclonable Functions," Applied Cryptography and Network Security, ACNS 2005, LNCS 3531, pp. 407-422, Jun. 2005.
[23] Y. Dodis, L. Reyzin, A. Smith, "Fuzzy Extractors: How to generate strong secret keys from biometrics and other noisy data," Advances in Cryptology - Eurocrypt 2004, LNCS 3027, pp. 523 - 540, May. 2004.
[24] C. W. O'Donnell, G. E. Suh and S. Davadas, "PUF-Based Random Number Generation," In MIT CSAIL CSG Technical Memo 481, 2004.
[25] N. Beckmann and M. Potkonjak, "Hardware-Based Public-Key Cryptography with Public Physically Unclonable Functions," Information Hiding, LNCS 5806, pp. 206-220, Jun. 2009.
[26] C. Brzuska, M. Fischlin, H. Schröder and S. katzenbeisser, "Physically Uncloneable Functions in the Universal Composition Framework," Advances in Cryptology, CRYPTO 2011, LNCS 6841, pp. 51-70, Aug. 2011.
[27] U. Rührmair, and M. van Dijk, "Practical Security Analysis of PUF-based Two-player Protocols," Cryptographic Hardware and Embedded System, CHES 2012, LNCS 7428, pp. 251-267, Sep. 2012.
[28] U. Rührmair, F. Sehnke, J. Sölter, G. Dror, S. Devadas and J. Schmidhuber, "Modeling attacks on physical unclonable functions," In: Proceedings of the 17th ACM conference on Computer and Communications Security. ACM, pp. 237-249, Oct. 2010.
[29] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "PUF Modeling Attacks on Simulated and Silicon Data," IEEE Transactions on Information Forensics and Security, vol. 8, no. 11, pp. 1876-1891, Nov. 2013.
[30] D. Karakoyunlu and B. Sunar, "Differential template attacks on PUF enabled cryptographic devices," 2010 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1-6, Dec. 2010.
[31] D. Merli, D. Schuster, F. Stumpf and G. Sigl, "Side-Channel Analysis of PUFs and Fuzzy Extractors," Trust and Trustworthy Computing, LNCS 6740, pp. 33-47, Jun. 2011.
[32] D. Merli, D. Schuster, F. Stumpf and G. Sigl, "Semi-invasive EM Attack on FPGA RO PUFs and Countermeasures," In Proceedings of the Workshop on Embedded Systems Security, pp. 2, Oct. 2011.
[33] C. Helfmeier, C. Boit, D. Nedospasov and J. P. Seifert, "Cloning Physically Unclon-

able Functions," In 2013 IEEE Hardware-Oriented Security and Trust (HOST), pp. 1-6, Jun. 2013.
[34] U. Rührmair and M. van Dijk, "PUFs in Security Protocols: Attack Models and Security Evaluations," 2013 IEEE Symposium on Security and Privacy, pp. 286-300, May. 2013.
[35] P. Tulys and B. Škorić, "Strong Authentication with Physical Unclonable Functions," Security, Privacy, and Trust in Modern Data Management, Springer Berlin Heidelberg, pp. 133-148, 2007.
[36] S. S. Mansouri and E. Dubrova, "Protecting Ring Oscillator Physical Unclonable Functions Against Modeling Attacks," In Proceedings of Information Security and Cryptology(ICISC 2013), pp. 27-29, Nov. 2013.
[37] D. Sokolov, J. Murphy, A. Bystrov and A. Yakovlev, "Design and Analysis of Dual-rail Circuits for Security Applications," IEEE Transactions on Computers, vol.54, no.4, pp. 449-460, Apr. 2005.

〈저 자 소 개〉

이 동 건 (Dong-geon Lee) 종신회원
2009년 2월: 부산대학교 정보컴퓨터공학부 졸업
2011년 2월: 부산대학교 컴퓨터공학과 석사
2011년 3월~현재: 부산대학교 컴퓨터공학과 박사과정
〈관심분야〉 정보보호, VLSI 설계, 물리적 복제 방지 함수, 부채널공격, IoT



이 연 철 (Yeon-cheol Lee) 학생회원
2013년 2월: 부산대학교 정보컴퓨터 공학부 졸업
2013년 3월~현재
〈관심분야〉 IoT, PUF, 정보보호, 전자공학

김 경 훈 (Kyoung-hoon Kim) 학생회원
2014년 2월: 부산대학교 정보컴퓨터공학부 졸업예정
〈관심분야〉 정보보호, VLSI 설계

박 종 규 (Jong-gyu Park) 학생회원
2014년 2월: 부산대학교 정보컴퓨터공학과 졸업
〈관심분야〉 정보보호, 데이터 마이닝

최 용 제 (Yong-je Choi) 정회원
1996년 8월: 전남대학교 전자공학과 졸업
1999년 2월: 전남대학교 전자공학과 석사
1999년 2월~8월: 전남대학교 전자통신연구소 인턴연구원
1999년 8월~현재: 한국전자통신연구원 사이버융합보안연구단 선임연구원
〈관심분야〉 보안프로세서 설계, 부채널 분석시스템, RFID/USN 보안

김 호 원 (Howon Kim) 종신회원
1993년 2월 : 경북대학교 전자공학과졸업(학사)
1995년 2월 : 포항공과대학교 전자전기공학과 석사(공학석사)
1999년 2월 : 포항공과대학교 전자전기공학과박사(공학박사)
1998년 12월~2008년 2월 : 한국전자통신연구원(ETRI) 정보보호연구단 선임연구원/팀장
2008년 3월~현재 : 부산대학교 정보컴퓨터공학부 부교수
〈관심분야〉 스마트그리드 보안, RFID/USN 보안, PKC 암호, VLSI,
Embedded system 보안, IoT