*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.24, NO.5, Oct. 2014 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2014.24.5.987

# PUF 기반 RFID 인증 프로토콜의 효율적 설계에 관한 연구

변 진 욱†‡
평택대학교 정보통신학과

# A Study on Efficient Design of PUF-Based RFID Authentication Protocol

Jin Wook Byun†‡
Department of Information and communication, Pyeongtaek University

요 약

PUF(physically unclonable function)는 태그 혹은 디바이스 내에 삽입되어 구현되며, 해당 디바이스의 고유한 물리적인 성질을 이용해서, 입력 값 $x$에 대해 노이지 $y$값을 출력한다. 비록 $x$가 동일하게 입력되더라도 매번 다른 출력 값($y_1$,...$y_n$)을 출력하며, 탬퍼 방지 (tamper-resistance) 성질로 인해, 암호 프로토콜에 활용도가 매우 높다. 본 논문에서는 이러한 PUF를 이용하여 RFID 인증 프로토콜을 안전하고 효율적으로 설계하는 방법에 대해 연구한다. 본 논문에서 제안된 방법은, 기존의 방법과 비교했을 시, 공격자가 메모리 노출 공격을 통해 비휘발성 메모리에 존재하는 롱텀(long-term) 키 값을 알게 되더라도, 그 세션 전후에 사용된 태그(tag)의 안전성 및 프라이버시가 보장되도록 설계하였다. 또한, PUF에 사용된 키 값을 복원하는 알고리즘이 태그 측이 아닌 RFID 리더에서 수행하도록 설계함으로써, 태그 구현 비용 및 전체 프로토콜 실행시간을 최소화할 수 있도록 하였다.

ABSTRACT

A PUF is embedded and implemented into a tag or a device, and outputs a noise y with an input of x, based on its own unique physical characteristics. Although x is used multiple times as inputs of PUF, the PUF outputs slightly different noises, ($y_1$,...$y_n$), and also the PUF has tamper-resistance property, hence it has been widely used in cryptographic protocol. In this paper, we study how to design a PUF-based RFID authentication protocol in a secure and an efficient way. Compared with recent schemes, the proposed scheme guarantees both authentication and privacy of backword/forward under the compromise of long-term secrets stored in tag. And also, the most cost and time-consumming procedure, key recovery algorithm used with PUF, has been desgined in the side of RFID reader, not in the tag, and, consequently, gives possibility to minimize costs for implementation and running time.

**Keywords:** RFID authentication, PUF, authentication protocol

## I. 서 론

RFID(radio frequency identification) 기술은 리더(reader)가 무선 주파수를 이용하여 태그(tag)가 부착된 사물을 인식할 수 있는 기술이다. 처음에는 유통물류, 생산관리 분야의 바코드를 대신할 수 있는 기술로 소개되었으나, 현재는, 그 활용범위가 넓어져, 대중교통 및 전자 요금 징수 시스템, 사람 및 동물 추적 장치, 자동차 및 전자기구 안전장치, 개인 입·출입 시스템 등 모든 일상생활의 필수적인 기술로

접수일(2014년 8월 22일), 수정일(2014년 9월 15일), 게재확정일(2014년 9월 15일)
† 주저자, jwbyun@ptu.ac.kr
‡ 교신저자, jwbyun@ptu.ac.kr(Corresponding author)

자리 잡았다.

RFID 기술의 급격한 대중화 및 편리함의 이면에는, 각종 시스템 보안 사고에 대한 위험성이 존재하며, 무엇보다 사용자들의 위치추적으로 인한 프라이버시가 노출될 수 있는 가능성이 더욱 높아지게 되었다. 그리하여 많은 문헌에서 [3,5,8,10,11,12,14] RFID 시스템에 존재하는 공격자의 행위, 안전성, 프라이버시에 대해서 논하고, 모델들을 정의하였다. 하지만, 스마트 디바이스 환경이 새롭게 변화함에 따라 (예: 사물 인터넷) 여전히 그리고 항상, 새로운 RFID 안전성 모델 및 인증 프로토콜을 요구하고 있다. 이러한 변화에도 불구하고, 프로토콜 설계에서 변함없는 사실은, RFID 프로토콜은 해당 환경에 맞게 항상 안전하고 경량화 되어야 한다는 점이다.

본 논문에서는 현재 암호학 분야에서 주목받는 기술 중 하나인 PUF(physically unclonable function)를 이용하여 RFID 인증 프로토콜을 안전하고 효율적으로 설계하는 방법에 대해 연구한다.

PUF는 디바이스 안에 안전하게 구현되며, 물리적인 복제가 불가능한 성질로 인해, 디바이스 기반 안전한 저장 기술 및 경량화된 인증 프로토콜 설계에 많이 활용되고 있다. PUF는 해당 디바이스의 고유한 물리적인 성질을 이용해서, 입력 값 $x$에 대해 출력 $y$값을 출력한다. 비록 $x$가 동일하게 입력되더라도 매번 다른 출력 값($y_1$,...$y_n$)을 출력한다. 해당 디바이스의 물리적 특징을 모르기 때문에, 출력되는 값을 예측할 수 없다는 장점이 있다.

매번 달라지는 출력 값을 기반으로 하여 매번 새로운 키를 대칭키(암호용 키, 인증용 키)로 활용하기 위해서, PUF는 퍼지 익스트랙터(fuzzy extractor) [1]와 함께 활용된다. 퍼지 익스트랙터는 매번 달라지는 PUF의 출력 값을 수정(correction)하고, 랜덤성을 보장하여, 두 참여자가 매번 암호학적으로 우수한 동일한 키를 유도할 수 있는 알고리즘을 제공해준다. 다시 말하면, 태그의 롱텀(long-term) 비밀 키 값 $x$가 안전하게 저장된다고 가정한다면, $x$가 PUF 및 퍼지 익스트랙터를 적용하여 나온 값은 암호학적으로 우수한 키이고 매번 달라지므로 그 키가 노출되더라도 다른 키에 영향을 주지 않을뿐더러 롱텀 비밀 값 $x$에 대한 정보도 노출되지 않기 때문에 전체적인 안전성을 향상 시킬 수 있다. 사실, PUF를 RFID 인증 프로토콜에 적용하는 가장 큰 이유는 효율성에 있다. 공개키 기반 RFID 인증 기술이 상당히 많이 제안되었지만, 사실, 공개키 기반 기술은, 제약된 자원을 보유한 태그에게, 인증서 관리라는 번거로움과 상대적으로 긴 연산 시간 및 처리 능력을 요구한다. 하지만, PUF는 대칭키 기반의 인증 방식이므로, 태그 차원에서, 공개키 기반의 단점들을 고려할 필요가 없으므로, 상대적으로 보다 경량화된 프로토콜 설계가 가능하다. 이러한 이유로 최근 PUF기반 프로토콜에 대한 연구가 비단 RFID뿐 아니라 여러 암호 시스템에 활용되고 있다.

### 1.1 관련 연구 및 공헌도

2004년, Ranasinghe 등에 의해 처음으로 PUF를 RFID에 적용하여 설계한 프로토콜이 제안되었다 [11]. PUF의 입·출력 값들을 리더의 데이터베이스에 미리 저장하여 이후 해당 토큰의 시도/응답을 통해 인증하는 구조이다. 가장 큰 단점은 한번 사용한 시도/응답 값은 재사용할 수 없으며, 이로 인해 제한된 횟수 내에서 프로토콜을 사용할 수밖에 없는 단점이 있다. 이와 유사한 구조로, Holcomb 등은 RFID 태그의 SRAM-PUF 환경에서 인증 프로토콜을 설계하기도 하였다 [4].

Tuyls와 Batina는 PUF를 사용하여, 인증에 사용되는 비밀 값을 비휘발성 메모리에 저장하는 대신, 필요할 때 마다 인증 비밀 값을 생성하는 인증 방식을 제안하였다 [13]. 하지만, 공개키 암호에 기반을 하고 있어, 저 비용 RFID 태그에는 적합하지 않다.

Kulseng 등은 키 갱신 절차가 포함된 RFID 인증 프로토콜을 제안하였으나 [7], Kardas 등에 의해 프로토콜이 중간자 공격(man in the middle attack)에 취약함이 밝혀졌다 [6].

아주 최근에, Moriyama 등은 비휘발성 메모리 노출 공격에 안전한 PUF 기반 인증 프로토콜을 제안하였다 [9]. 매 세션마다 비휘발성 메모리에 저장되어 있는 비밀 값을 안전하게 갱신시킴으로써, 비록 비밀 값이 노출되더라도 한 세션에만 영향을 미치고, 전후 세션에는 안전성 영향을 받지 않도록 설계하였다. 이 프로토콜은 Herrewege가 2012년도에 제안한 리버스 퍼지 익스트랙터(reverse fuzzy extractor) [2] 구조를 최대한 기반 하되, 키 갱신 절차를 추가하여, 비휘발성 메모리 노출 공격에 안전하도록 설계한 것이다. 하지만, 비용이 많이 소모되는 *FE.REC* 알고리즘이 태그 쪽에서 수행되도록 설계되었기 때문에, 여전히 효율성 개선의 여지가 남아 있다.

본 논문에서는 PUF를 기반으로 하여 메모리 노출

공격에도 안전한 RFID 인증 프로토콜을 보다 효율적으로 설계한다. 제안된 프로토콜은, 메모리 노출공격과 리버스 퍼지 익스트랙터를 기반으로 하므로, 이 두 기술을 적용한 프로토콜[2,9]들과 밀접한 관련이 있다. 기존 프로토콜들보다 우수한 측면을 정리하면 다음과 같다.

- 제안된 프로토콜은 공격자가 메모리 노출 공격을 통해 롱텀 키 값을 알게 되더라도, 그 세션 전후에 사용된 태그의 안전성 및 프라이버시가 보장된다. 즉, 전후에 사용된 프로토콜의 실행 메시지들이 어떤 태그에 해당하는 것인지 구분 불가능하여 프라이버시를 만족하게 된다. 하지만, 최근에 Herrewege 등에 의해 제안된 인증 프로토콜 [2]은 매번 동일한 롱텀 비밀 값을 PUF에 적용시키는 구조이므로, 메모리 노출 공격 시, 전/후 사용된 프로토콜 실행 메시지가 어느 태그 것인지 쉽게 구분할 수 있어, 결국 프라이버시를 보장하지 못하게 된다. 제안된 프로토콜에서는 매 세션 마다 비휘발성 메모리에 저장되는 롱텀 키가 갱신되도록 하는 절차를 설계하였다. 매 세션 마다 참여자들이 매번 롱텀 비밀 값을 갱신하기 때문에 프라이버시 보장이 가능하도록 설계하였다. 또한, 안전성 모델을 바탕으로 인증 및 프라이버시에 대한 공격자의 이점이 무시할 확률임을 보임으로, 증명 가능한 프로토콜 설계를 하였다.
- 제안된 프로토콜은 효율적인 측면에서 Moriyama 등에 의해 제안된 프로토콜 [9] 보다 우수하다. $|p|$를 의사랜덤함수의 출력크기, $|m|$을 주고받는 시도 값의 크기, $|PE|$를 퍼지 익스트랙터의 출력 크기라고 각각 정의했을 때, Moriyama 등이 제안한 프로토콜은 $6|p|+2|m|$을 요구하지만 본 프로토콜은 $4|p|+2|m|+|PE|(\leq 5|p|+2|m|)$를 요구한다. 효율성 측면에서, 가장 중요한 것은 PUF에 사용된 키 값을 복원하는 퍼지 익스트랙터 복원 알고리즘이 태그 측이 아닌 리더 측에서 수행하도록 설계함으로써, 구현 비용 및 실행시간을 최소화할 수 있도록 하였다. 즉, 1) 키 갱신 작업 설계, 2) 퍼지 익스트랙터 복원 알고리즘을 리더 측에서 수행, 3) 증명 가능한 RFID 안전성 및 프라이버시 만족, 이 세 가지를 동시에 고려하여 안전하고 효율적으로 설계했다는 점이 기존 방식과의 차이점이다. 아래 Table 1에 기존 프로토콜과의 차이점을 정리하여 나타내었다. 단, T는 전송량을 의미하고, MR은 메모리 노출 시 프라이버시 공격 가능 여부를 의미한다. R은 라운드 수를 나타내며, FE는 퍼지 익스트랙터의 복원 알고리즘의 수행 주체를 나타낸다. PR은 증명 가능성 여부이며, $|ID|$는 $ID$의 길이를 나타낸다. 항목 중 여부 판단은 ○, ×로 표현하고 나머지는 직접적인 수식과 내용으로 나타내었다.

### 1.2 논문의 구성

먼저, 다음 장에 본 논문에 사용되는 PRF, PUF, 퍼지 익스트랙터를 설명한다. 3장에서는 안전성 모델을 정의하였으며, 4장에서는 프로토콜의 제안 배경 및 패러다임을 설명하고, 5장에서 프로토콜을 제안한다. 6장에서는 안전성 증명을 수행하고, 7장에서 논문의 결론을 맺는다.

## II. 암호 프리미티브

제안된 프로토콜을 구성하는데 핵심적인 역할을 수행하는, 의사랜덤함수(PRF), PUF, 퍼지 익스트랙터(fuzzy extractor)에 대한 개념 및 정의는 다음과 같다.

### 2.1 PRF(pseudo random function)

**[정의 1]** 의사 랜덤 함수 패밀리는 함수 $F_K : K(F) \times D(F) \rightarrow R(F)$의 모음으로 정의한다. 단, $K(F)$는 $F_K$의 키들의 집합이고 D(F)는 $F_K$의 도메인에 대한 집합이다. 의사 랜덤 함수 $F_K$는 다항식 시간 알고리즘(polynomial time algorithm) $B$에 대해서 다음을 만족해야 한다.

- 주어진 $x \in D(F)$ 와 $K \in K(F)$에 대해서, $F_K(x)$를 계산하는 다항식 알고리즘이 존재해야 한다.
- 어떠한 다항식 알고리즘 $B$에 대해서 다음의 $\epsilon_{prf}$ 이점이 무시할 수 있을 만큼 작아야 한다.

$$\left|\Pr\left[B^{F_K}=1 : K \xleftarrow{R} K(F)\right] - \Pr\left[B^{F}=1 : F \xleftarrow{R} U_{F_K}\right]\right| \leq \epsilon_{prf}$$

단, $U_{F_K}$는 전체 가능한 $F_K : K(F) \times D(F) \rightarrow R(F)$ 함수의 총 집합이다.

Table 1. Analysis and comparison with recent results

| Protocol | T | MR | R | FE | PR |
|---|---|---|---|---|---|
| Herrewege et al's protocol [2] | $3\|m\|+2\|h\|+\|ID\|$ | × | 4 | Reader | ○ |
| Moriyama et al's protocol [9] | $6\|p\|+2\|m\|$ | ○ | 3 | Tag | ○ |
| Proposed Protocol | $4\|p\|+2\|m\|+\|PE\|$ | ○ | 3 | Reader | ○ |

### 2.2 PUF(physcially unclonable function)

일반적으로, PUF는 사전에 IC칩에 삽입되어 운영된다. PUF, $f$는 질의 $c$가 있을 시 디바이스 내 물리적인 특징 $x$를 이용해 반응하며, 이를 $r \leftarrow f(x,c)$로 표현한다. 하지만, PUF는 매번 같은 질의에 대해서도 틀린 값으로 반응한다.

문헌에서 언급되는 PUF는 강인함(robust), 예측 불가능(unpredictable), 물리적 복제 불가능(physically unclonable) 성질을 보유하고 있다. 먼저, 강인함은, 동일한 질의에 대해서도 출력의 변동 범위(variation)값은 HD(hamming distance) $d$ 값을 유지한다는 성질이다. 둘째, 예측 불가능성은 비록 PUF에 사용된 입력, 출력 값을 많이 알고 있더라도, 알려지지 않은 질의 값에 대한 PUF의 출력 값은 알기가 어렵다는 것이다. 끝으로, 복제 불가능 성질은 구분될 수 없는 두 개의 PUF를 생산해내는 것이 어렵다는 성질이다.

PUF에 대한 정의가 다양하지만, 그 성질은 위에서 언급한 세 가지 성질로 축약된다. 본 논문에서는 참고문헌 [9]에 정의한 $(d,n,l,h,\epsilon_{puf})$-안전한 PUF의 정의를 따른다.

**[정의 2]** 다음 조건들을 만족하면, $(d,n,l,h,\epsilon_{puf})$-안전한 PUF, $f(x,\bullet)$라 정의한다.

- 임의의 입력 $y \in \{0,1\}^k$에 대해 출력 값의 변화가 최소 $d$이다.

$$\Pr[HD(z_1,z_2) \le d | z_1 \xleftarrow{R} f(x,y), z_2 \xleftarrow{R} f(x,y)] = 1$$

  $HD$는 해밍 거리이며, $x$는 디바이스 내 물리적인 고유 성질을 의미한다.
- $n$개의 서로 다른 PUF에 대해 다른 입력 값 $y_1,...,y_l$을 고려하자. 이에 대한 출력 값을 $Z$라고 정의했을 때 다음과 같이 정의된다.

$$Z = \left\{ z_{i,j} \xleftarrow{R} f(x_i,y_j) \right\} \quad \text{for } 1 \le i \le n, 1 \le j \le l$$

  임의의 $z_{i^*,j^*}$에 대한 최소 엔트로피(min-entropy)가 비록, $Z \backslash z_{i^*,j^*}$에 대한 출력 값을 알고 있더라도, $h$로 귀결되어야 한다. 즉, 다음의 식을 만족해야한다.

$$H(z_{i^*,j^*} | Z \backslash z_{i^*,j^*}) = h$$
$$\text{for} 1 \le i^* \le n, 1 \le j^* \le l$$

- 비록 악의적인 공격자가 $f$에 대해 물리적인 공격을 수행하더라도, $f$의 입력, 출력 값에 대한 정보를 제외하고 어떠한 $f$의 정보를 얻을 수 없다. $A$를 다항식 시간 내에 $f$를 물리적으로 공격하는 공격자라 하자. 구분자(distinguisher) $D$ 및 $f$와 상호작용하는 알고리즘 $S$를 고려했을 때 다음 식을 만족해야 한다.

$$|\Pr[D(1^k,st) \to 1 | st \xleftarrow{R} A(1^k, f(x,\cdot))]$$
$$- \Pr[D(1^k,st) \to 1 | st \xleftarrow{R} S^{f(x,\cdot)}(1^k)]| \le \epsilon_{puf}$$

### 2.3 퍼지 익스트랙터 (fuzzy extractor)

일반적으로 $f(x,\cdot)$의 출력 값을 암호학적인 롱텀 키로 사용하기 위해서는 퍼지 익스트랙터가 활용된다 [1]. 즉, PUF 출력 값은 일정한 노이지(noisy)를 띄는데, 그러한 노이지를 모두 수렴하여 랜덤성이 보장된 암호학적 키를 만들기 위해, PUF는 퍼지 익스트랙터와 결합하여 사용한다.

**[정의 3]** $(d,h,\epsilon_{pe})$-퍼지 익스트랙터는 키 생성 알고리즘 *FE.GEN*과 복원 알고리즘 *FE.REC*으로 구성된다. 먼저 *FE.GEN* 알고리즘은 입력 $z$에 대해 $r$과 보조데이터 $hd$를 출력한다. *FE.REC* 알고리즘은 입력으로 $z'$($z$와 HD가 $d$인 $z'$)와 $hd$를 입력으로 받으면 $r$을 출력한다. $(d,h,\epsilon_e)$-퍼지 익스트랙터는 다음의 조건을 만족한다.

- *FE.GEN* 알고리즘에 입력되는 $z$의 최소 엔트로피가 $h$를 만족한다면, 비록 $hd$값이 공격자에게 노출되더라도, $r$값은 $\{0,1\}^k$에서 유니폼하게 선택된 랜덤 값과 구분할 가능성이 무시할 확률, $\epsilon_{pe}$로 낮다.

$f(x,\cdot)$의 출력 값은 정의에 의해 최소 엔트로피 $h$를 만족한다. 그러므로, $z(=f(x,y))$값을 *FE.GEN* 알고리즘의 입력 값으로 주어 $(r,hd)=FE.GEN(z)$ 값을 얻은 후, $r$값을 암호학적인 키로 활용한다.

## III. 안전성 모델

본 논문에서는, 참고문헌 [9]에 정의된 안전성 및 프라이버시 정의를 따른다. 이 정의는 기존에 RFID에서 연구되어온 안전성 정의와 크게 다르지 않다. 가장 큰 차이점은 공격자가 태그의 비휘발성 메모리 안에 있는 비밀 값을 알 수 있다는 것이고, 그럼에도 불구하고, 태그 가장 공격을 하지 못하도록 하는데 있다. 먼저, 인증 프로토콜에 대한 안전성 정의에 공격자의 가장 공격에 대한 안전성을 고려하였고, 프라이버시 정의에 태그의 추적 및 구분에 대한 공격을 고려하였다.

### 3.1 인증 프로토콜에 대한 안전성 정의

RFID 프로토콜 $P$에는 한 개의 리더 $R$과 $n$개의 RFID 태그 $T=\{t_1,...,t_n\}$가 존재하며, 리더 $R$은 각각의 태그들과 함께 설정 단계와 인증 단계를 수행한다. RFID 프로토콜 $P$에서 공격자가 취할 수 있는 오라클 질의들은 다음과 같다.

- **Launch**$(1^k)$ : 리더에게 세션을 시작하도록 한다.
- **SendReader**$(m)$ : 임의의 메시지 $m$을 리더에게 전달한다.
- **SendTag**$(t,m)$ : 임의의 메시지 $m$을 태그 $t$에게 전달한다.
- **Result**$(sid)$ : 리더가 세션 $sid$를 수락했다면 1을 출력하고, 아니면 0을 출력한다. (단, $sid$는 통신되는 메시지들을 이용해 고유한(unique) 값으로 정의된다.)
- **Reveal**$(t)$ : 태그 $t$의 비휘발성 메모리에 존재하는 비밀 값을 반환한다.

RFID 인증 프로토콜 $P$에 반드시 만족해야 할 안전성 성질은 태그 가장 공격이다. 만약, 공격자가 위에서 정의한 질의들을 활용하여, 최종적으로 리더 $R$이 수락할 수 있는 $sid^*$를 만들 수 있다면, 태그 가장 공격이 성공한 것이다. 이에 대해, Table 2에 잘 정의하였다.

RFID 인증 프로토콜 P에 대해서 공격자 $A$에 대한 이점 $Adv_{P,A}^{Scc}(k)$는 $\mathrm{Exp}_{P,A}^{Scc}(k)$이 1을 출력하는 확률로 정의한다. $sid^*$를 구성하는 통신 메시지는 공격자에 의해 수정, 변경되는 메시지를 의미한다.

Table 2. A definition of security for RFID authentication protocol

$\mathrm{Exp}_{P,A}^{Scc}(k)$
$(pk,sk)\xleftarrow{R}Setup(1^k);$
$sid^*\xleftarrow{R}A_1^{Launch,SendReader,SendTag,Result,Reveal}(pk,R,T);$
$b=Result(sid^*);$
$Output\ b$

**〔정의 4〕** 안전성 파라미터 $k$에 대해 $Adv_{P,A}^{Scc}(k)$가 무시할 정도의 확률일 때, RFID 인증 프로토콜 $P$는 다항식 시간 공격자 $A$에 대해 가장 공격 및 메모리 노출 공격에 안전하다고 정의한다.

### 3.2 프라이버시에 대한 정의

프라이버시를 세 단계로 정의하였다. 각 단계의 공격자는 $A_1,A_2,A_3$로 가정한다. 첫 번째 단계는 시도(challenge) 단계로, 공격자가 구분하고 싶은 태그를 직접 선택하는 단계이다. 이 후, 공격자는 선택된 두 태그가 리더와 수행한 프로토콜 실행 값 (주고받는 메시지 값)들, $\pi_0,\pi_1$을 알 수 있다. 두 번째 단계는, 익명 접근 단계로, 공격자 $A_2$는 선택된 두 태그에 대해 익명을 유지하며 접근할 수 있다. 이를 가능하게 해주는 것이 $I$ 알고리즘이다. $I$는 $A_2$와 $t_b^*$ 중간에서 공격자의 질의 및 응답을 익명을 유지하면서 전달해 준다. 예를 들어, $A_2$가 **SendTag**$(I,m)$를 질의하면, $I$는 $t_b^*$에게 $m$을 전달하고 그 응답을 다시 공격자에게 익명으로 전달한다. 두 번째 단계에서는 공격자에게 선택된 태그를 제외한 태그들에 대해 모든 질의를 허용한다. (그렇지 않을 경우, 즉, 선택된 태그들의 비밀

값을 통해 쉽게 선택된 태그들이 수행했던 통신이 어떤 것인지 구분할 수 있기 때문이다.) 이후 또 다른 프로토콜 실행 값 $\pi_0'$,$\pi_1'$에 대한 정보가 공격자에게 주어지고, 세 번째 단계에서는, 응답 단계로, 두 번째 단계의 정보를 이용하여, 최종적으로 $b'$값을 출력한다. 두 번째를 제외한 첫 번째, 세 번째 단계의 공격자에게는 선택된 태그에 대한 **Reveal** 질의를 수행할 수 있다. Table 3에, 참고문헌 [9]에 정의한 표기를 동일하게 이용하여 정의하였다.

Table 3. A definition of privacy of RFID authentication protocol

$\mathrm{Exp}_{P,A}^{Prv}(k)$
$(pk,sk) \xleftarrow{R} Setup(1^k)$;
$(t_0^*, t_1^*, st_1) \xleftarrow{R} A_1^O(pk, R, T)$;
$b \xleftarrow{R} \{0,1\}$, $T' = T \setminus \{t_0^*, t_1^*\}$;
$\pi_0 \xleftarrow{R} Execute(R, t_0^*)$, $\pi_1 \xleftarrow{R} Execute(R, t_1^*)$;
$st_2 \xleftarrow{R} A_2^O(R, T', I(t_b^*), \pi_0, \pi_1, st_1)$ :
$\pi_0' \xleftarrow{R} Execute(R, t_0^*)$, $\pi_1' \xleftarrow{R} Execute(R, t_1^*)$;
$b' \xleftarrow{R} A_3^O(R, T, \pi_0', \pi_1', st_2)$
*Output* $b'$

RFID 인증 프로토콜 P에 대해서 공격자 $A$에 대한 이점 $Adv_{P,A}^{Prv}(k)$는 $\mathrm{Exp}_{P,A}^{Prv}(k)$이 정확히 $b$값을 추측하는 확률로 정의한다.

**[정의 5]** 안전성 파라미터 $k$에 대해 $Adv_{P,A}^{Prv}(k)$가 무시할 정도의 확률일 때, RFID 인증 프로토콜 $P$는 다항식 시간 공격자 $A$에 대해 프라이버시에 안전하다고 정의한다.

### 3.3 비밀 키 값 갱신, 안전성, 프라이버시 논의

본 논문에서 제안하는 RFID 인증 프로토콜은 키 갱신 과정이 포함되어 있다. 즉, 매 세션마다 리더와 태그는 퍼지 익스트랙터 생성 및 복구 작업을 통해 $r_i$ 값을 상호 공유한다. $r_i$값은 매 세션 마다 랜덤하게 생성되고, PRF함수인, $G, G'$의 시드(seed)값이 되어, 주고받은 랜덤 값, $a_1$,$a_2$의 인증 값을 생성해내는 역할을 수행한다. 그러므로 재생공격 및 가장공격에 안전하다. 안전성 및 프라이버시 정의를 만족하기 위해 가장 중요한 것은 비휘발성 메모리에 존재하는 비밀 값 $y_i$가 매 세션마다 새롭게 갱신되어야 한다는 사실이다. 만약, 매 세션마다 비휘발성 메모리에 있는 비밀 키 값이 갱신되지 않으면, 세션이 진행되어도 같은 키가 비휘발성 메모리에 남아있게 되고, 이 경우는 여기서 고려하는 프라이버시 정의를 만족하지 못하게 된다. 예를 들어, 첫 번째 단계에서 공격자는 선택된 태그 $t_0$,$t_1$의 비휘발성 메모리에 대한 **Reveal**질의를 하여, 비밀 값 $y_0$,$y_1$을 얻었다고 하자. 그리고, 세 번째 단계에서, 어느 태그의 세션인지 구분해야 할 $\pi_0'$,$\pi_1'$ 프로토콜이 주어진다. 만약, 키 갱신이 이루어지지 않으면, $\pi_0'$,$\pi_1'$ 프로토콜의 세션에 **Reveal**질의를 했을 때 $y_0$값을 얻는 경우는 $t_0$이고 $y_1$값을 얻는 경우는 $t_1$ 태그인지 쉽게 구분할 수 있다. 하지만, 매 세션 마다 비밀 키 값이 갱신된다면, 세 번째 단계에서 비록 선택된 태그에게 **Reveal**질의를 하더라도 공격자가 얻을 수 있는 비밀 값은 이미 갱신된 비밀 값이므로, 예전의 비밀 키 값으로 생성된 $\pi_0'$,$\pi_1'$값을 구분할 수 없다. 그러므로, 강화된 프라이버시 모델을 만족시키기 위해서는, 매 세션 마다 비휘발성 메모리에 저장되어 있는 비밀 키 값들에 대한 갱신 작업이 필요하다.

## IV. 퍼지 익스트랙터와 PUF를 RFID 인증 프로토콜에 효율적으로 적용하는 방안

일반적으로, 프로토콜의 참여자는 $z(=f(x,y))$값을 계산하고, $(r,hd) = FE.GEN(z)$를 통해 유도된 $r$값을 인증 프로토콜에 활용한다. 프로토콜의 다른 참여자는 자신의 PUF를 이용하여, $z'(=f(x,y))$ 값을 직접 계산하고, $z'$값과 다른 참여자로부터 제공되는 $hd$값을 입력으로하여 $r = FE.REC(z', hd)$을 유도한다. 이러한 방식으로 $r$값을 상호 공유하여 인증 프로토콜의 비밀 값으로 활용한다.

RFID 인증 프로토콜에서는, 계산 및 저장 자원이 풍부한 리더와 자원들의 사용이 제약되어 있는 태그로 구성된다. 그래서 퍼지 익스트랙터를 PUF와 함께 인증 프로토콜에 적용할 때, 가장 큰 이슈는, PUF를 태그에 구현시킬 때 소요되는 비용을 최소화하는 것이다. PUF는 퍼지 익스트랙터와 함께 사용되며, 특히, 생성할 때($r = FE.GEN(z)$) 보다 상대적으로 복원할 때 ($r = FE.REC(z', hd)$) 매우 많은 수의 게이트 및

and/or 연산을 요구한다 [2]. 즉, 복원 알고리즘을 태그에 그대로 구현 시, 태그의 연산 시간 및 태그의 구현 비용이 급격히 높아지는 단점이 있다. 그러므로 퍼지 익스트랙터와 PUF를 RFID 인증 프로토콜에서 효율적으로 설계하기 위해서는, 복원 알고리즘, *FE.REC*를 태그 쪽에 설계하는 것이 아니라 연산 및 저장 공간이 풍부한 리더 쪽에서 수행되도록 설계해야만 한다. Herrewege 등은 참고문헌 [2]에서, 이러한 접근법을 이용하여, 리버스(reverse) 퍼지 익스트랙터라는 개념을 처음 제안하였다.

Fig. 1에 전형적인 PUF 기반 RFID인증 프로토콜의 구조에 대해 나타내었고, Fig. 2에서는 리버스 퍼지 익스트랙터를 이용한 RFID 인증 구조를 나타내었다. 그림에서 보듯이, 두 그림의 차이점은 Fig. 2에서는 *FE.REC* 알고리즘이 태그가 아닌 리더에서 수행된다는 점이다.

2010년, Sadeghi 등에 의해, 첫 번째 PUF 기반 RFID 인증 프로토콜이 제안되었다 [12]. 이 프로토콜은 Fig. 1의 구조를 기반으로 하고 있다. Herrewege 등은 Fig. 2의 리버스 퍼지 익스트랙터를 처음 소개하고 효율적인 RFID 프로토콜을 설계하였다 [2]. 2013년, Moriyama 등은 [2][12]의 연구 결과를 바탕으로 비 휘발성 메모리에 존재하는 키의 업데이트 과정을 추가하여 설계하였다 [9]. 하지만, Moriyama 등이 제시한 프로토콜에서는 여전히 태그에서 *FE.REC* 알고리즘을 수행하도록 설계되어 효율성이 좋지 못하다. 이는 안전한 키 갱신, RFID 안전성 및 프라이버시 만족, *FE.REC* 알고리즘의 태그 측 수행, 이 세 가지를 동시에 만족하도록 설계하는 것이 결코 쉬운 작업이 아니기 때문이다.

| Reader [$y$] | | Tag [$y, f, hd_i$] |
|---|---|---|
| $z_i = f(x_i, y)$<br>$(k_i, hd_i) = FE.GEN(z_i)$ | | |
| $a_1 \in \{0,1\}^k$ | $\xrightarrow{a_1}$ | |
| | | $z_i' = f(x_i, y)$<br>$k_i = FE.REC(z_i', hd_i)$<br>$a_2 \in \{0,1\}^k$<br>$g = PRF(k_i, a_1 \| a_2)$ |
| $g = PRF(k_i, a_1 \| a_2)$ | $\xleftarrow{a_2, g}$ | |

Fig. 1. A paradigm of PUF-based RFID authentication

제안하는 구조는 Fig. 2의 구조를 기반으로 하여, 키 갱신 작업을 효율적으로 추가하였고, RFID 리더 측에서 *FE.REC*가 수행되며, 동시에 인증 및 프라이버시에 대해 증명 가능하도록 설계하였다. 전체적인 패러다임은 Fig. 3에 나타내었으며, Fig. 2와의 차이점은 키 갱신 과정이 프로토콜에 있다는 것이다.

키 갱신 작업은 최대한 효율적으로 설계하였다. 태그가 갱신될 키를 선택하고, 그 키를 일회용 패드형식으로 리더에게 안전하게 전달하는 구조를 취하였다. 인증 작업이 모두 끝나면, 태그와 리더는 새로운 키로 갱신해서 비휘발성 메모리에 저장한다. Fig. 2에서는, 기존 패러다임과의 비교를 위해, 태그에 대한 인증만 고려하였지만, 프로토콜 설계에서는 리더도 응답을 하여 태그가 리더를 인증할 수 있도록 설계하였다.

| Reader [$y$] | | Tag [$y, f$] |
|---|---|---|
| $z_i = f(x_i, y)$ | | |
| $a_1 \in \{0,1\}^k$ | $\xrightarrow{a_1}$ | |
| | | $z_i' = f(x_i, y)$<br>$(k_i, hd_i) = FE.GEN(z')$<br>$a_2 \in \{0,1\}^k$<br>$g = PRF(k_i, a_1 \| a_2)$ |
| $k_i = FE.REC(z_i, hd_i)$<br>$g = PRF(k_i, a_1 \| a_2)$ | $\xleftarrow{a_2, g, hd_i}$ | |

Fig. 2. An efficient design with reverse fuzzy extractor

| Reader [$y$] | | Tag [$y, f$] |
|---|---|---|
| $z_i = f(x_i, y)$ | | |
| $a_1 \in \{0,1\}^k$ | $\xrightarrow{a_1}$ | |
| | | $z_i' = f(x_i, y)$<br>$(k_i, hd_i) = FE.GEN(z')$<br>$a_2 \in \{0,1\}^k$<br>$g = PRF(k_i, a_1 \| a_2)$ |
| $k_i = FE.REC(z_i, hd_i)$<br>$g = PRF(k_i, a_1 \| a_2)$ | $\xleftarrow{a_2, g, hd_i}$ | Key Update |
| Key Update | | |

Fig. 3. An efficient and secure design with reverse fuzzy extractor, key update procedure

## V. 프로토콜 설계

### 5.1 프로토콜 설정 (Setup) 단계

설정 단계는 안전한 채널로 구성된다. 먼저, 리더 $R$은 $y_1$을 $\{0,1\}^k$으로부터 랜덤하게 선택하여, 태그 $t_i$에게 전달한다. 각각의 PUF는 모든 태그 $t_i$에 고유한 태그의 물리적 정보를 바탕으로 이미 각각 설치되어있음을 가정한다. 리더 $R$은 $(y_1, y_{old}=y_1, t_i)$를 데이터베이스에 보관하고, 태그는 $(f, y_1)$값을 보관한다.

### 5.2 인증 단계

프로토콜에 사용되는, PRF 함수, $G$, $G'$를 다음과 같이 정의한다.

$$G: \{0,1\}^k \times \{0,1\}^{2k} \rightarrow \{0,1\}^{4k}$$
$$G': \{0,1\}^k \times \{0,1\}^{2k} \rightarrow \{0,1\}^k$$

- 리더 $R$은 $a_1$을 $\{0,1\}^k$으로부터 랜덤하게 선택하여, 태그 $t_i$에게 전달한다.
- 태그 $t_i$는 비휘발성 메모리에 있는 비밀 값 $y_1$과 $f(x, \cdot)$를 이용해서 $z_1$을 계산한다. 그 후 $z_1$은 퍼지 익스트랙터 생성 함수를 이용하여 $r_1$과 $hd_1$를 다음과 같이 계산한다.

$$z_1 \leftarrow f(x, y_1)$$
$$(r_1, hd_1) = FE.GEN(z_1)$$
$$(g_{1,\ldots,}g_4) = G(r_{1,}a_1 \| a_2)$$

이후, 태그는 $r_1$을 PRF 함수, $G$의 seed값으로 활용하여, $a_1 \| a_2$에 대한 PRF값, $g_1,\ldots,g_4$값을 구한다. 물론, $a_2$는 태그 $t_i$에 의해 랜덤하게 선택된 값이다. 태그 $t_i$는 다음 세션에 사용할 비밀 값, $y_2$를 $\{0,1\}^k$으로부터 랜덤하게 선택하여, $G$ 의 최종적인 출력 값 $g_2$를 키로 이용하여, 배타적연산을 수행하고, 수행된 값 $E_1$에 대해 메시지 인증 값을 $G'$을 통해 다음과 같이 계산한다.

$$y_2 \leftarrow \{0,1\}^k$$
$$E_1 = g_2 \oplus y_2$$
$$M_1 = G'(g_3, a_2 \| E_1)$$

- 최종적으로, $a_2, g_1, E_1, M_1, hd_1$을 리더 $R$에게 전달한다. 리더 $R$은 먼저 PUF 함수, $f$에 $y_1$값을 입력으로 하여 $z_1'(=f(x,y_1))$을 계산한다. 계산된 $z_1'$ 값은 주어진 $hd_1$값과 함께 퍼지 익스트랙터 복원 함수의 입력으로 사용되어 $r_1$값을 태그와 함께 공유하게 된다. 이후, $r_1$값은 리더와 태그가 각각 선택한 $a_1, a_2$값에 대한 $G$ 값을 생성하는데 사용된다. 그 과정은 다음과 같다.

$$z_1' \leftarrow f(x, y_1)$$
$$r_1 = FE.REC(z_1', hd_1)$$
$$(g_1', \ldots, g_4') = G(r_{1,}a_1 \| a_2)$$

리더 $R$은 전달 받은 $g_1$ 값을 확인하여 태그 $t_i$를 인증한다. 즉, $g_1', \ldots, g_4'$값을 계산하여 전달 받은 $g_1$값과 동일한지 검증한다. 또한 전달받은 $M_1$값이 올바른 값인지를, $g_3'$값을 이용하여 검증한다.$(M_1 = G'(g_3', a_2 \| E_1))$에 적용하여 검증한다. 만약, 두 개의 검증이 동일하지 않으면, 인증을 실패하게 된다. 동일하면, 전달 받은 $E_1$값에 $g_2'$ 값을 배타적 연산하여 $y_2$값을 얻는다. 최종적으로 리더 $R$은 $[y_2, y_{old}=y_1, t_i]$로 갱신하여 데이터베이스에 보관한다. 리더 $R$은 $g_4'$값을 $t_i$에게 전달한다.

- 태그 $t_i$는 $G(r_1, a_1 \| a_2)$의 출력 값 $(g_1,\ldots,g_6)$ 중 $g_4$값과 전달 받은 $g_4'$값을 비교 확인하여 리더 $R$을 인증하고, 비휘발성 메모리에, 비밀 키 값을 $[f, y_2]$값으로 갱신한다.

## VI. 안전성 분석 및 증명

### 6.1 인증에 대한 안전성 분석 및 증명

제안된 프로토콜에서 정당한 태그는 $f$, 초기 비밀 값 $y_1$, $FE.REC$ 알고리즘을 이용해서 $r_1$을 올바르게 유도한 후 $a_1, a_2$에 대해 $(g_1,\ldots g_4) = G(r_1, a_1 \| a_2)$ 값을 계산하여 인증을 성공적으로 통과한다. 만약, 공격자가 태그의 비휘발성 메모리에 대해 물리적 공격을 수행한다면, $f$는 탬퍼링 방지 성질(tamper resistance)로 인해 공격자의 접근이 불가능하고, 오직 $y_1$값만 얻을 수 있다. 하지만, 이 값으로는 태

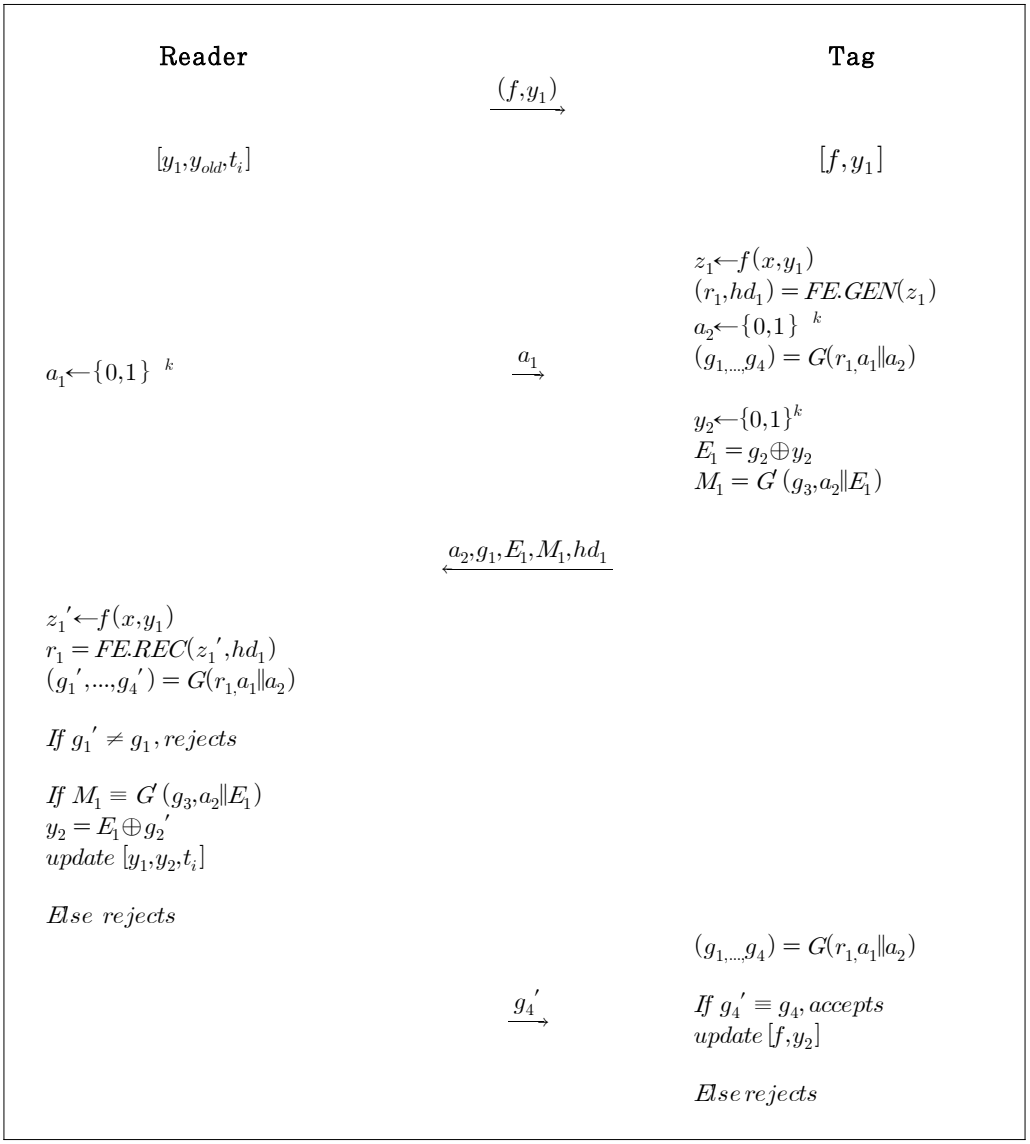


Fig. 4. An efficient PUF-based RFID authentication protocol

그 인증 시 중요하게 사용되는 $r_1$값을 유도할 가능성은 무시할 정도로 낮다. 또한, 매 세션마다 갱신되는 키(예: $y_2$)는 이러한 $r_1$값을 기반으로 만들어진 값을 키로 사용하여 배타적 연산을 수행하기 때문에 $r_1$을 알지 못하는 한 다음에 갱신되는 키 $y_2$값을 알 수 없으며, 궁극적으로 다음 세션에 사용될 $r_2$값을 유도 하지 못하게 된다. 그러므로 임의의 세션에서 태그의 비휘발성 메모리의 비밀 값이 노출되더라도 그 세션 전방향 후방향으로 인증에 필요한 $r_i$ 값을 유도하지 못해서 인증을 통과할 수 없다. 안전성 모델을 바탕으로 제안된 프로토콜이 가장 공격에 안전함을 다음의 정리를 통해 보인다. 다음의 정리는 태그 가장 공격만을 고려하였으며, 리더 가장 공격은 태그 가장 공격과 동일한 방법으로 증명 가능하므로, 지면 관계상 생략하였다.

**〔정리 1〕** $n$개의 태그와 한 개의 리더가 존재하는 RFID 시스템에서, $(d, h, \epsilon_p)$-퍼지 익스트랙터, $FE$와 $(d, n, l, h, \varepsilon_{puf})$-안전한 PUF 함수 $f$를 가정했을

때, 제안된 프로토콜 P는 다항식 시간 공격자 $A$에 대해 가장 공격 및 메모리 노출 공격에 안전하다. 즉, 공격자의 이점이 다음과 같이 무시할 확률로 표현된다.

$$Adv_{P,A}^{Sec}(k) \le (\frac{1}{n} \cdot \frac{1}{l})(5 \cdot \epsilon_{prf} + \epsilon_{pe} + \epsilon_{puf})$$

**〈증명〉** 공격자 $A$의 목적은 태그 $R$이 수락(accept) 할 수 있는 세션 아이디 $sid^*$를 만드는 것이다. 각기 다른 환경의 안전성 게임이 정의되고 $i$번째 게임에서 공격자가 선택한 $sid^*$에 대해 **Result**$(sid^*)=1$이 될 사건을 $S_i$라 정의한다.

- **게임 0** : 실제 게임으로, 〔정의 1〕에 묘사된 게임과 동일하다.

$$Adv_{P,A}^{Sec}(k) = S_0$$

- **게임 1** : 공격자 $A$가 가장에 성공할 수 있는 태그 및 세션을 선택하는 단계이다. 공격자가 $l$의 세션을 재개할 수 있다고 했을 때, 챌린저는 랜덤하게 $t^*$태그를 $\{t_1,...,t_n\}$으로부터 선택하고, $l^*$를 $\{1,...,l\}$으로부터 선택한다. 만약, 랜덤하게 선택된 $t^*$가 공격자가 가장할 수 없는 태그라면, 또한 공격자가 가장할 수 있는 세션이 $l^*$가 아니라면 안전성 게임이 중단(aobrt)된다. 그러므로 다음의 관계식을 얻을 수 있다.

$$S_0 \le (\frac{1}{n} \cdot \frac{1}{l}) \cdot S_1$$

- **게임 2** : 공격자의 다음의 질의들에 대해 다음의 방법으로 응답한다.
  - **Launch**$(1^k)$ : 이 질의에 대해 챌린저는 임의의 메시지 $a_1$을 태그에게 보냄으로써, 프로토콜을 시작한다.
  - **SendReader**$(m)$ : 이 질의에 대해 챌린저는 임의의 메시지 $m$을 리더에게 전달하고, 그 후 프로토콜 동작과정에 따라 리더가 보내게 되는 메시지를 반환한다.
  - **SendTag**$(t,m)$ : 이 질의에 대해 챌린저는 임의의 메시지 $m$을 태그에게 전달하고, 그 후 프로토콜 동작과정에 따라 태그가 보내게 되는 메시지를 반환한다.
  - **Result**$(sid)$ : 챌린저는 공격자가 질의한 세션 아이디 값 $sid$에 대해 리더가 수락할 수 있는지를 진행 중인 프로토콜 동작 결과를 통해 알 수 있고, 수락할 수 있다면 1을 출력하고, 아니면 0을 출력한다.
  - **Reveal**$(t)$ : 챌린저가 태그 $t$의 비휘발성 메모리에 존재하는 비밀 값을 반환한다.

질의들에 대한 응답과정을 통해 공격자는 게임 1과 게임 2를 구분할 수 없다. 그러므로 다음의 관계식을 얻는다.

$$S_1 = S_2$$

- **게임 3** : 이 게임에서는, 임의의 입력에 대한 PUF의 출력 값 $(r_1,hd_1)$이 충분한 엔트로피 값을 지니지 않고, 다른 출력 값과 연관이 있을 경우, 챌린저는 안전성 게임을 중단한다. 즉, 이 게임을 통해, PUF 출력 값은 충분한 엔트로피 값을 지니고, 다른 출력 값과 독립적이게 만들어 준다. 공격자가 게임 2와 게임 3을 구분할 차이점은, PUF 정의에 의해, 최대 $\epsilon_{puf}$가 된다.

$$S_2 - S_3 \le \epsilon_{puf}$$

- **게임 4** : 이 게임에서는 퍼지 익스트랙터의 출력 값을 유니폼하고 랜덤한 값으로 대체한다. 즉, 게임 3에 의해, PUF의 출력 값이 충분한 엔트로피 값을 지닌 채로 퍼지 익스트랙터에 입력으로 활용되며, 게임 4에서, 챌린저는 프로토콜의 $r_1$ 값에 대해 유니폼하고 랜덤한 값 $r_1'$을 생성하여 **SendTag**, **SendReader**에 활용한다. 만약, 게임 3과 게임 4를 구분하는 다항식 알고리즘이 존재한다면, 〔정의 3〕에 의해, 퍼지 익스트랙터의 안전성을 깨는 알고리즘을 쉽게 구축할 수 있다.

$$S_3 - S_4 \le \epsilon_{pe}$$

- **게임 5** : 이 게임에서는 프로토콜에 사용되는 $G(r_1', \cdot)$ 함수를 전체 $U_{F_K}$ 집합에서 랜덤하게 뽑은 함수, $\overline{G}(r_1', \cdot)$로 대체하여, 그 출력 값,

$\overline{g_1},\cdots,\overline{g_4}$, 을 공격자가 수행하는 질의에 대한 답변으로 시뮬레이션 할 때 사용한다. 게임 4와 게임 5를 공격자 $A$가 높을 확률로 구분한다면, 그 확률로 PRF의 안전성을 깨는 다항식 알고리즘 $B$를, 〔정의 1〕에 의해, 쉽게 구축할 수 있다. 그러므로, 다음의 관계식을 얻는다.

$$S_4 - S_5 \leq 4 \cdot \epsilon_{prf}$$

- **게임 6** : 이 게임에서는 배타적 연산의 결과 값, $g_2 \oplus y_2$값을 랜덤한 값 $E_1'$을 선택하여 대체한다. 또한, 마지막 $G(\overline{g_3}, \cdot)$값을 $\{0,1\}^k$에서 선택한 랜덤한 값 $M_1'$으로 대체한다. 게임 6에서는 비로소 모든 프로토콜 메시지가 유니폼하고 랜덤하게 변경되는 형태이므로, 공격자는 안전성 게임에서 얻을 수 있는 이점은 0이다.

$$S_6 = 0$$

공격자가 게임 5와 게임 6을 구분할 이점을 고려해 보자. 원래, $y_2$ 값이 랜덤하게 선택되고, 원타임 패드 역할을 수행하므로, $g_2 \oplus y_2$결과 값을 랜덤한 $E_1'$으로 대체하는 것은 공격자가 구분할 수 없다. 하지만 $G(\overline{g_3}, \cdot)$ 값의 대체는 PRF의 이점이 있으므로, 〔정의 1〕에 의해 $\epsilon_{prf}$의 이점을 지닌다.

$$S_5 - S_6 \leq \epsilon_{prf}$$

모든 게임에서 발생하는 이점들을 종합하여 계산하면, $S_0$에 대해 다음의 무시할 확률 값을 얻는다.

$$Adv_{P,A}^{Scc}(k) \leq (\frac{1}{n} \cdot \frac{1}{l})(5 \cdot \epsilon_{prf} + \epsilon_{pe} + \epsilon_{puf})$$

### 6.2 프라이버시에 대한 안전성 분석 및 증명

프라이버시에 대한 안전성 분석은 인증에 대한 안전성 분석과 유사하다. 다만, 공격자가 **SendTag** 질의들을 통해 임의로 프로토콜이 실패하도록 만들어 태그에 포함된 비밀 값이 갱신되지 않도록 할 수 있다. 이후, 공격자는 **Reveal** 질의를 통해, 만일, 동일한 비밀 값이 나온다면, 해당 세션이 어느 태그가 수행하는 것인지 추적할 수 있게 된다. 프라이버시 정의 〔정의 5〕에 의해, 선택된 두 태그에 대해서는 익명으로 접근하도록 하였고, 전후로 **Execute** 질의를 하게 함으로써, 비밀 값이 재 갱신되도록 하였으므로, **Reveal**질의를 통한 추적이 불가능하다.

**〔정리 2〕** $n$개의 태그와 한 개의 리더가 존재하는 RFID 시스템에서, $(d,h,\epsilon_p)$-퍼지 익스트랙터, $FE$와 $(d,n,l,h,\varepsilon_{puf})$-안전한 PUF 함수 $f$를 가정했을 때, 제안된 프로토콜 P는 다항식 시간 공격자 $A$에 대해 프라이버시를 만족한다. 즉, 공격자의 이점이 다음과 같이 무시할 확률로 표현된다.

$$Adv_{P,A}^{Prv}(k) \leq Adv_{P,A}^{Scc}(k) + (\frac{1}{n^2} \cdot \frac{1}{l})(5 \cdot \epsilon_{prf} + \epsilon_{pe} + \epsilon_{puf})$$

**〈증명〉** 먼저, 〔정의 4〕에 정의된 메모리 노출 및 가장 공격자 $A$가 존재하면, 본 논문에서 정의된 〔정의 5〕의 프라이버시 안전성을 깰 수 있다. 즉, 두 개의 선택된 $t_0^*, t_1^*$에 대해 태그 $t_1^*$를 가장할 수 있는 공격자를 가정하자. 그러면, 공격자가 프로토콜 메시지를 잘 조작하여 가장에 성공했을 때, 그 메시지에 포함된 갱신하려는 키 값과 공격자에게 가장을 당한 원래 태그 $t_1^*$가 갱신하려 했던 키 값은 다르다. 그러므로 공격 이후에, 가장된 태그 $t_1^*$는 리더로부터 인증을 받지 못하기 때문에 공격자는 항상 인증을 받지 못하는 태그가 $t_1^*$ 태그임을 구분할 수 있다. 그래서 프라이버시를 깨는 이점은 가장 공격 및 메모리 노출 공격을 깨는 이점을 기반으로 한다.

$$Adv_{P,A}^{Prv}(k) \leq Adv_{P,A}^{Scc}(k)$$

**게임 0 ~ 게임 5** : 〔정리 1〕과 동일하게 안전성 게임들이 정의되고 $i$번째 게임에서 공격자가 정확히 $b$를 출력할 확률을 $S_i$라 정의하자. 그러면, 선택된 두 개의 태그 $t_0^*, t_1^*$에 대해, 여섯 번의 게임을 통해 프로토콜에서 발생되는 메시지를 모두 랜덤하게 대체한다. 이 과정은 〔정리 1〕과 동일하므로, 그 이점은 크게 차이가 없다. 다만, 두 개의 태그를 $n$개 중에서

정확히 선택해야 하므로 파라미터의 일부가 $\frac{1}{n^2}$로 수정되었다. 게임 0부터 게임 5까지의 증명과정을 〔정리 1〕과 동일하므로 생략하였다.

**게임 6** : 〔정리 1〕의 증명과 동일하게, 이 게임에서는 배타적 연산의 결과 값, $g_2 \oplus y_2$값을 랜덤한 값 $E_1'$을 선택하여 대체하고, 마지막 $G'(\overline{g_3}, \cdot)$값을 $\{0,1\}^k$에서 선택한 랜덤한 값 $M_1'$으로 대체한다. 그러므로, 다음의 관계식을 얻는다.

$$S_5 - S_6 \leq \epsilon_{prf}$$

이제, 게임 6에서 공격자가 프라이버시 공격에 깨는 이점이 0임을 보이려 한다. 아래는 〔정의 5〕에서, $A_2, A_3$ 핵심 단계를 간추린 것이다. 현재까지, 게임들을 통해, 공격자에게, 모두 랜덤하게 변경된 $\pi_0, \pi_1, \pi_0', \pi_1'$ 값이 주어진다.

$$\pi_0 \xleftarrow{R} Execute(R, t_0^*), \pi_1 \xleftarrow{R} Execute(R, t_1^*);$$
$$st_2 \xleftarrow{R} A_2^O(R, T', I(t_b^*), \pi_0, \pi_1, st_1):$$
$$\pi_0' \xleftarrow{R} Execute(R, t_0^*), \pi_1' \xleftarrow{R} Execute(R, t_1^*);$$
$$b' \xleftarrow{R} A_3^O(R, T, \pi_0', \pi_1', st_2)$$

비록, 매번 임의의 메시지를 보내 고의로 프로토콜을 리젝시켜, 태그의 키 갱신이 안되게 하고, Reveal질의를 통해, 태그 추적을 시도하려 하더라도, 공격자 $A_2, A_3$ 전후에는 매번 랜덤하게 변경된 비밀 값이 생성되고, $A_2$ 공격자에게는 해당 태그에 대한 Reveal질의가 불허된다. 또한, 게임 6에서 항상 유니폼하고 랜덤한 $\pi_0, \pi_1, \pi_0', \pi_1'$정보가 공격자에게 주어지므로, 공격자는 $\pi_0, \pi_1, \pi_0', \pi_1'$에 대해 둘 중 어느 태그 것인지 구분할 이점은 0이다. 그러므로 이를 모두 정리하면 다음의 관계식을 얻는다.

$$Adv_{P,A}^{Prv}(k) \leq Adv_{P,A}^{Scr}(k) + (\frac{1}{n^2} \cdot \frac{1}{l})(5 \cdot \epsilon_{prf} + \epsilon_{pe} + \epsilon_{puf})$$

## VII. 결 론

본 논문에서는 PUF를 기반으로 RFID 인증 프로토콜을 최대한 효율적으로 설계하는 방안에 대해 연구하였다. 그 방법은, 첫째, 키 갱신 작업을 두어, 매번 롱텀 키 값이 변경되도록 설계하는 것이다. 이는 프라이버시 만족을 위해 반드시 필요했다. 둘째, 퍼지 익스트랙터의 복원 알고리즘을 태그 측이 아닌 리더측에 수행하도록 설계하였다. 이 두 방안을 적용하여 프로토콜을 제안하였다. 제안된 프로토콜은, 최근에 제안된 두 개의 PUF 기반 RFID 인증 프로토콜〔2,9〕 설계를 근간으로 하되, 보다 더 안전하고, 간단하며, 효율적인 증명 가능한 설계 방법이 존재할 수 있음을 보여준다.

## References

〔1〕 Yevgeniy Dodis, Leonid Reyzin, and Adam Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy dta," In Proceedings of Eurocyrpt, vol. 3027, LNCS, pp. 523-540, Springer-Verlag, May, 2004.

〔2〕 A. Herrewege, S. Katzenbeisser, R. Maes, R. Peeters A. R. Sadeghi, I. Verbauwhede, C. Wachsmann, "Reverse fuzzy extractors: enabling lightwewight mutual authentication for PUF-enabled RFIDs," In Proceedings of FC 2012, vol. 7397, LNCS, pp.374-389, Springer-Verlag, Feb. 2012.

〔3〕 J. Hermans, A. Pashalidis, F. Vercauteren, B. Preneel, "A new RFID privacy model," In Proceedings of ESORICS 2011, vol. 6879, LNCS, pp. 346-365, Springer-Verlag, Sep. 2011.

〔4〕 D.E. Holcomb, W.P. Burleson, K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags," In Proceedings of RFIDSec 2007, Jul. 2007.

〔5〕 A. Juels, S. Weis, "Defining strong privacy for RFID," ACM transactions on Information and System Security, vol. 13,

Issue 1, article no. 7, Oct. 2009.
〔6〕 S. Kardas, M. Akgun, M.S. Kiraz, H. Demirci, "Cryptanalysis of lightweight mutual authentication and ownership transfer for RFID systems," In Proceedings of Lightweight 2011, pp. 20-25, IEEE, Mar. 2011.
〔7〕 L. Kulseng, Z. Yu, Y. Wei, Y. Guan, "Lightweight mutual authentication and ownership transfer for RFID systems," In Proceedings of INFOCOM 2010, pp. 1-5, IEEE, Mar. 2010.
〔8〕 C. Ma, Y. Li, R.H. Deng, T. Li, "RFID privacy: Relation between two notions, minimal condition, and efficient construction," In Proceedings of ACMCCS 2009, pp. 54-65, Nov. 2009.
〔9〕 Daisuke Moriyama, Shin'ichiro Matsuo, and Moti Yung, "PUF-Based RFID Authentication Secure and Private under Complete Memory Leakage," IACR Cryptology ePrint Archive, Sep. 2014.
〔10〕 C.Y. Ng, W. Susilo, Y. Mu, R. Safavi-Naini, "New privacy results on synchronized RFID authentication protocols against tag tracing," In ESORICS 2009, vol. 5789, LNCS, pp.321-336, Springer-Verlag, Sep. 2009
〔11〕 D.C. Ranasinghe, D.W. Engels, P.H. Cole, "Security and privacy: Modest proposals for low-cost RFID systems," Auto-ID Labs Research Workshop, Sep. 2004.
〔12〕 S. Kardaş, S. Çelik, M. Yıldız, A. Levi "PUF-enhanced offline RFID security and privacy," Journal of Network and Computer Applications, vol. 35, Issue 6, pp. 2059-2067, Nov. 2012.
〔13〕 P. Tuyls and L. Batina, "RFID-tags for anti-counterfeiting," In Proceedings of CT-RSA 2006 vol. 3860, LNCS, pp. 115-131, Springer-Verlag, Feb. 2006.
〔14〕 S. Vaudenay, "On privacy models for RFID," In Proceedings of ASIACRYPT 2007 vol. 4833, LNCS, Springer-Verlag, Dec. 2007.

## 〈저자소개〉

변 진 욱 (Jin Wook Byun) 정회원
2001년 2월: 고려대학교 전산학과 이학사
2003년 2월: 고려대학교 정보보호대학원 정보보호 전공, 공학 석사
2006년 8월: 고려대학교 정보보호대학원 정보보호 전공, 공학 박사
2006년 11월~2007년 12월: 영국 런던대학교, ISG 박사후 연수
2008년 03월~현재: 평택대학교 정보통신학과 부교수
〈관심분야〉 사용자 인증, 프라이버시 보호 기술, 데이터베이스 보안, 암호 프로토콜