情報保護學會論文誌
第18卷 第2號, 2008. 4

# 저비용 RFID 시스템에 적합한 효율적인 인증 방법*

김 진 호†, 서 재 우, 이 필 중‡
포항공과대학교

# Efficient Authentication Protocol for Low-Cost RFID System*

Jin Ho Kim†, Jae Woo Seo, Pil Joong Lee‡
POSTECH

## 요 약

RFID 시스템은 동시에 여러 개의 개체를 인식할 수 있는 장점을 바탕으로 기존의 광학 바코드 시스템을 대체할 새로운 기술로 주목받고 있다. 하지만 RFID 시스템을 이루는 태그와 리더 사이의 통신은 RF 신호를 이용하기 때문에 공격자에게 쉽게 노출될 수 있고, 그로 인해 여러 가지 보안위협이 존재한다. 이런 위협을 해결하기 위해서 많은 연구 활동이 있었다. 본 논문에서는 기존 연구를 기반으로 RFID 인증 프로토콜을 동기 필요의 유무에 따라 상태기반과 비 상태기반 인증 모델로 나누어 설명한다. 그리고 Bloom filter를 이용해서 백엔드 데이터베이스에서의 인증 시간을 단축하는 방법을 제안한다. 이는 상태기반과 비 상태기반 모두에 적용가능하며, RFID 인증에서 발생할 수 있는 태그 검색 문제를 해결하는 새로운 접근이다.

## ABSTRACT

Compared with the existing bar code system, RFID system has lots of advantages such as it identifies automatically massive objects. We might anticipate RFID technology will be a substitution for an optical bar code system in the near future. However, their feature that uses radio waves may cause various security problems. Many kinds of solutions have been researched to overcome these security problems. In this paper, we analyze the previous proposed protocols. And then, we categorize RFID authentication into two types according to the synchronization requirement between a Back-end Database and a Tag. In addition, we introduce the previous proposed approaches to tag search problem in RFID authentication. And we propose an efficient method which provides fast tag search by using membership test algorithm, a Bloom filter.

**Keywords** : RFID system, Hash Function, Authentication, Bloom Filter

## Ⅰ. 서 론

RFID 시스템은 직접적인 물리적 접촉 없이 자동으로 개체를 인식, 식별하는 시스템으로 기존에 사용되고 있는 광학 바코드 시스템과 달리 한 번에 여러 개의 개체를 인식할 수 있는 장점을 가진다. 이 때문에 RFID 시스템은 기존의 광학 바코드를 대체하면서 생산, 공급 망

접수일: 2007년 11월 26일; 채택일: 2008년 3월 11일
* 본 연구는 Brain Korea 21과 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구 결과로 수행되었음 (IITA -2008-C1090-0801-0026).
† 주저자, jhkim@oberon.postech.ac.kr
‡ 교신저자, pjl@postech.ac.kr

관리, 재고 관리 등의 분야에서 중요한 역할을 할 것으로 기대되고 있다. 뿐만 아니라, 일상생활에서도 RFID의 활용분야는 다양하여 가까운 미래에는 우리사회 전반에서 중요한 요소로 자리 잡을 것이다[1].

RFID의 비접촉 무선인식 기술은 기술적인 유용성 측면에서는 그 가치가 막대하나, 시스템의 특성상 태그와 리더간의 통신이 무선채널 상에서 이루어지고, 이런 무선채널은 공격자에게 완전히 노출되기 때문에 보안상의 위협요소가 많은 기술이라고도 볼 수 있다. 따라서 RFID 기술이 가지는 보안상의 취약점이 무엇인지 명확하게 분석하고 이를 해결하려는 노력이 필요하다. 본 논문에서는 기존 연구를 기반으로 안전한 RFID 인증 프로토콜을 동기화 필요유무에 따라 상태기반과 비 상태기반 인증 모델로 나누어 설명한다. 또한 RFID 인증에서 발생할 수 있는 태그의 검색 문제를 해결하기 위한 방법을 소개한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서 RFID 시스템의 구성요소와 RFID 인증에서 요구되는 보안 요소들을 살펴본다. 3장에서는 지금까지 연구된 RFID 인증 프로토콜들에 대해 분석하고 이를 바탕으로 4장에서는 RFID 인증을 상태기반 인증과 비 상태기반 인증이라는 두 가지 모델로 나누어 설명한다. 5장에서는 4장에서 제시한 두 가지 형태의 모델에 적용될 수 있는 태그의 빠른 검색에 대한 방법을 제안한다. 마지막으로 6장에서 본 논문의 결론을 내린다.

## Ⅱ. RFID 시스템

### 2.1. 구성요소

RFID 시스템은 태그, 리더, 백엔드 데이터베이스의 세 가지 핵심요소로 구성된다[2, 3]. 본 절에서는 RFID 시스템을 이루는 세 가지 핵심요소에 대해 간략히 알아본다.

#### 2.1.1. RFID 태그

RFID 태그는 제한된 연산과, 데이터 저장을 위한 마이크로칩, 무선 통신을 위한 안테나 코일로 구성된 소형의 장치이다. 전력이 공급되는 방식에 따라 능동형 태그와 수동형 태그로 분류 하는 것이 일반적이다. 수동형 태그는 보드에 배터리가 부착되어 있어 데이터를 멀리 전송할 수 있는 장점을 가지지만 가격이 비싸다는 단점을 지닌다. 수동형 태그는 내장된 배터리 없이 리더로부터 나오는 RF 신호를 통해 전력을 공급받는다. 수동형 태그의 전력은 리더의 전력보다 상대적으로 약하기 때문에 일반적으로 근거리 데이터 통신에 사용되며, 가격이 저렴한 장점을 가진다.

#### 2.1.2. 리더

리더는 일반적으로 RF 모듈, 컨트롤유닛, RF 신호를 이용해 태그를 활성화하기 위한 커플링 소자로 구성되며, 무선 인터페이스를 통해 태그로부터 인증을 위한 데이터를 수신하고, 이를 백엔드 데이터베이스에 전달, 백엔드 데이터베이스로부터 받은 결과를 다시 태그에 전달하는 중재자 역할을 한다. 이때 리더와 태그 사이에 형성되는 채널은 무선통신을 기반으로 하기 때문에 공격자의 도청이 가능한, 안전하지 않은 채널이다.

#### 2.1.3. 백엔드 데이터베이스

백엔드 데이터베이스는 리더로부터 받은 데이터를 저장하며, 계산능력이 제한된 태그나 리더를 대신하여 복잡한 계산을 수행한다. 또한 백엔드 데이터베이스는 태그를 인증하기 위한 정보를 저장하고 있기 때문에 리더로부터 받은 데이터가 맞는지 틀린지를 구별해 낼 수 있고, 이를 통해 태그를 인증할 수 있다. 보통 리더와 백엔드 데이터베이스의 저장, 처리 능력의 제약은 태그에 비해 심하지 않기 때문에, 이들 두 요소 간에는 암호학적 프리미티브를 사용할 수 있다. 따라서 리더와 백엔드 데이터베이스 사이에 형성되는 채널은 일반적으로 안전하다고 가정한다.

### 2.2. 보안 요구사항

RFID 인증에서 리더와 태그 사이의 통신은 무선구간으로, 공격자가 개입할 수 있는 오픈 채널이다. 따라서 인증을 방해하는 여러 위협요소들이 존재한다. 본 절에서는 RFID 시스템 내에 존재하는 위협요소들을 바탕으로 RFID 인증에서 요구되는 보안 요소들을 살펴본다.

#### 2.2.1. 도청

리더와 태그 사이의 통신이 무선채널에서 이루어지

기 때문에, 공격자는 리더와 태그 사이의 통신내용을 쉽게 엿들을 수 있다. 따라서 RFID 인증 프로토콜을 설계할 때, RFID 시스템 특성상 도청을 피하는 것은 불가능할 지라도 엿들은 통신 내용으로부터 보안을 위협하는 어떠한 비밀 정보의 유출도 없도록 인증 프로토콜을 설계해야 한다[4].

#### 2.2.2. 스푸핑 공격과 재생 공격

스푸핑 공격은 정당하지 않은 개체를 정당한 것처럼 속여 인증과정을 통과하거나 정상적인 통신을 방해하는 것을 말한다. 스푸핑 공격은 일반적으로 다음과 같이 이루어진다. 공격자는 정당한 리더로 가장하여 공격 대상이 되는 태그에게 악의적인 질의를 전송하여 응답 메시지를 수집한다. 즉, 정당한 태그로부터 수집된 정보들을 이용해 정당하지 않은 태그에 대해서 인증을 시도하는 순으로 공격이 진행된다[4].

재생 공격은 공격자가 정상적인 인증이 진행되는 과정에서 리더의 질의에 대한 태그의 응답 메시지를 저장해 두었다가 나중에 재전송함으로써, 정당하지 않은 태그를 정당한 것으로 가장하는 공격 방법으로 정당하지 않은 개체를 정당한 것처럼 속인다는 측면에서 넓게 보면 스푸핑 공격의 한 예라고 할 수 있을 것이다. 이런 공격이 성공할 경우, 태그의 impersonation 문제가 발생할 수 있고, 이는 RFID 시스템에 큰 위협이 될 수 있기 때문에, 인증 프로토콜을 설계할 때는 스푸핑 공격과 재생 공격에 안전하게 설계해야 한다.

#### 2.2.3. 위치 추적

위치추적은 공격자가 태그의 응답을 바탕으로 태그의 위치를 파악함으로써 태그 소유자의 이동경로를 알아내는 공격이다. 이는 태그 소유자의 프라이버시와 밀접한 관계가 있으므로 안전한 RFID 시스템을 위해서는 위치 추적이 불가능한 인증 프로토콜이 절실히 요구된다. 이를 위해서는 어떤 특정 태그가 전송하는 메시지 값과, 다른 태그들이 전송하는 메시지 값들을 공격자가 구분하기 힘들어야 한다[5].

#### 2.2.4. 전방향 안전성

전방향 안전성은 공격자가 태그에 저장된 비밀 데이터 값을 얻게 되더라도 이를 통해서 이전의 태그의 위치를 추적하거나 이전에 발생했던 거래에 대해 알아낼 수는 없어야 한다는 것을 의미한다. RFID 시스템에서 태그는 폐기되거나 포획되기 쉽고, 저장된 비밀 값에 대한 부 채널 공격에도 취약하기 때문에 비밀 값이 노출되더라도 과거에 이루어졌던 거래에 대한 프라이버시를 보장해주는 전방향 안전성의 개념이 중요한 보안요소로 떠오르고 있다[6].

## Ⅲ. 이전 연구들에 대한 분석

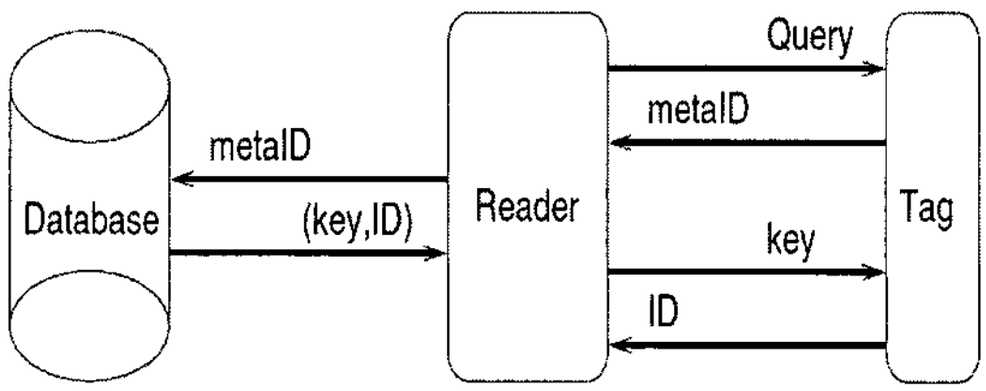
RFID 태그들은 계산능력, 저장 공간이 떨어지는 등 자원적으로 제한된 장치이기 때문에 기존 암호화 알고리즘을 그대로 적용하기에는 무리가 있다. 따라서 계산능력이 떨어지는 저비용 RFID 태그에서도 작동할 수 있는 새로운 인증 프로토콜이 요구된다. 이에 해쉬와 같은 가벼운 연산으로 인증을 하는 프로토콜에 대한 연구들이 많이 이루어졌다. 본 장에서는 이와 관련한 이전 연구들에 대한 분석을 할 것이다.

### 3.1 Hash Lock and Randomized Hash Lock

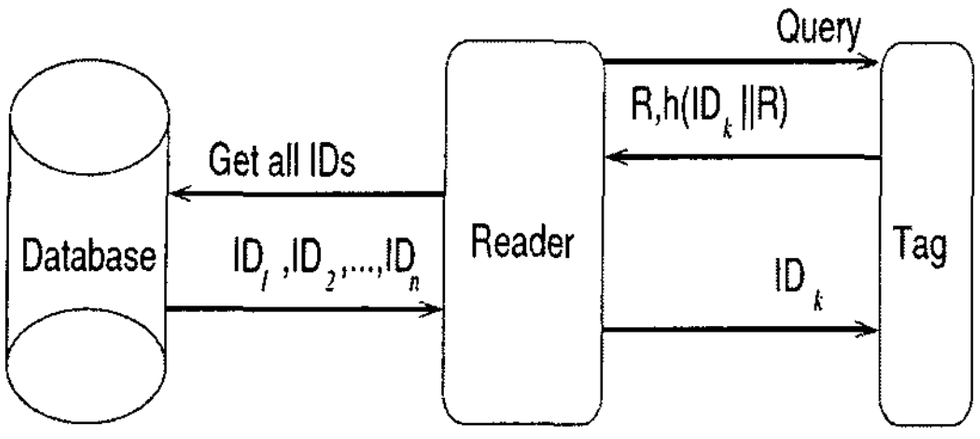
Stephen 등은 *ID* 대신 *metaID*를 사용하는 인증 프로토콜을 [그림 1].과 같이 제안하였다[5].

이 프로토콜에서는 태그의 진짜 *ID*를 숨기기 위해 *metaID*를 사용한다($metaID \rightarrow hash(key)$). 하지만 *metaID* 값이 고정되기 때문에 공격자는 태그의 위치를 추적할 수 있고, 재생 공격에도 취약하다. 또한 공격자가 진짜 리더인 척 접근해서 태그로부터 *metaID*를 받고 이를 이용해 진짜 태그인 척 가장할 수 있다. 따라서 스푸핑 공격에도 취약하다.

같은 논문에서 태그의 위치추적을 피하기 위해 Randomized Hash Lock 방법을 [그림 2].와 같이 제안했다. 이 방법에서는 난수 *R*을 이용해서 태그의 응답을




〔그림 1〕 Hash-Locking

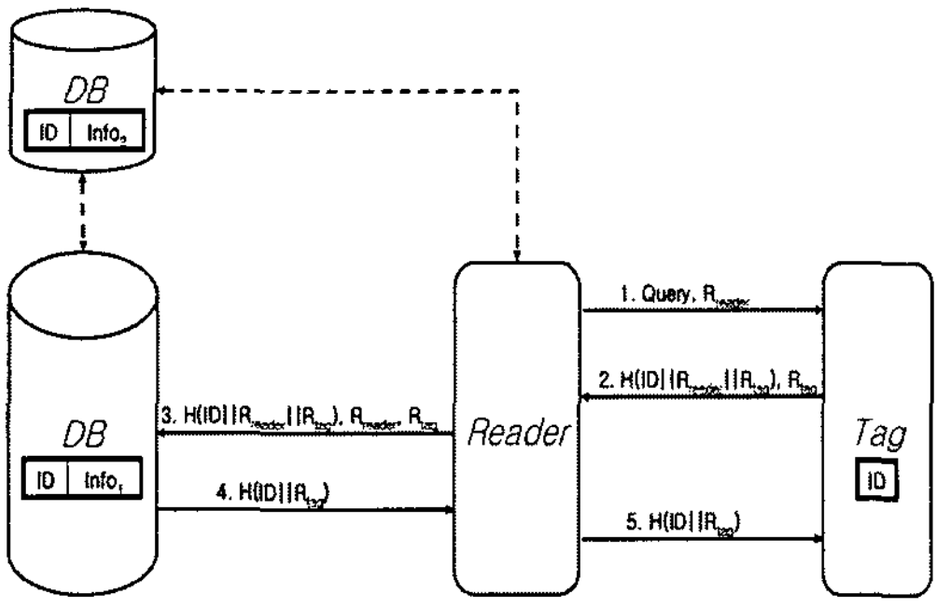



〔그림 2〕 Randomized Hash-Locking

랜덤화한다. 하지만 이 방법에서, 리더가 태그에게 안전하지 않은 채널을 통해 $ID_k$ 값을 전달하기 때문에 위치 추적문제를 여전히 해결하지 못하였다. 그리고 이 방법 역시 재생 공격과, 스푸핑 공격에 취약하다.

### 3.2. 도전-응답 방법에 기반 하는 인증 방법

Keunwoo 등은 해쉬 함수와 도전-응답 방법을 이용한 인증 프로토콜을 [그림 3].과 같이 제안하였다[2]. 이 프로토콜을 살펴보면 백엔드 데이터베이스와 태그가 공유하는 *ID*라는 비밀 값은 고정되어 있지만 리더와 태그에서 매 세션 마다 난수 $R_{reader}$와 $R_{tag}$를 각각 생성하여 이를 해쉬 함수의 입력에 포함시키고 있기 때문에, 태그에서 리더로 보내지는 응답 메시지가 고정되지 않고 항상 변화하게 된다. 이를 통해 위치추적문제를 피할 수 있게 된다. 이와 마찬가지로 정당한 리더가 생성한 값을 바탕으로 다음 응답이 이루어져야 하기 때문에, 이전의 응답을 저장했다가 재전송해서 공격하는 재생 공격이나 스푸핑 공격에 안전하다. 하지만, 비밀 정보인 *ID* 값이 고정 되기 때문에 태그의 비밀 정보인 *ID* 값이 노출되면 과거에 이루어 졌던 거래들에 대한 정보 역시 노출된다. 따라서 전방향 안전성을 만족시키지는 못한다. 하지만 이를 제외한 대부분의 보안 요구조건을 충족한다.




〔그림 3〕 도전-응답 방법에 기반한 인증 프로토콜

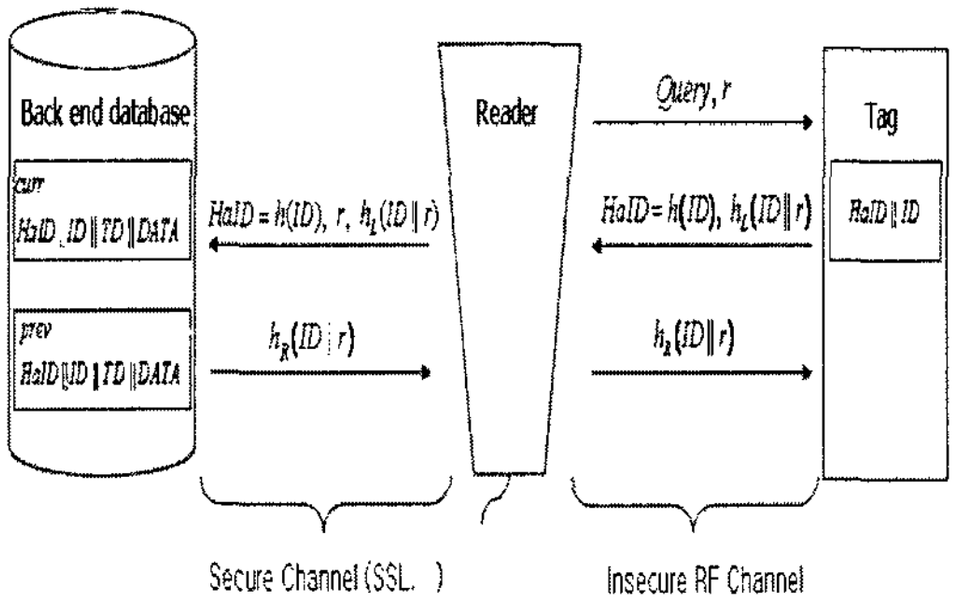
### 3.3. 동기화된 비밀 값에 기반 하는 인증 방법

앞서 살펴본 연구들은 비밀 값이 초기화 단계부터 계속 같은 값으로 고정되는 알고리즘들이었다. 이런 경우, 비밀 값이 노출 되면 과거 거래들에 대한 정보 역시 노출된다. 이를 방지하기 위해서 비밀 값을 갱신하면서 인증을 하는 프로토콜들이 제안되었다. Su Mi 등은 [3]에서 비밀 값을 갱신하면서 인증을 하는 방법을 [그림 4]. 와 같이 제안하였다.

이 프로토콜에서는 성공적인 인증이 이루어진 후, 태그의 *ID* 값을 갱신함으로써 전방향 안전성을 달성하려고 했다. 따라서 이 방법에서는 백엔드 데이터베이스와 태그 사이의 동기화가 요구된다. 프로토콜을 안전도 측면에서 살펴보면 리더가 매 세션 마다 난수 $r$을 생성하고 이를 입력으로 한 결과 값($h_L(ID\|r)$)을 태그의 응답으로 요구하기 때문에, 재생 공격이나 스푸핑 공격에 안전하다. 하지만 위치 추적 문제가 발생할 수 있다. 즉, 공격자가 리더에서 태그로 보내지는 메시지를 계속 차단해서 비동기 상태를 만든다면, 첫 번째 응답 값인 $h(ID)$가 항상 고정되는 결과가 나타나고 따라서 위치 추적문제가 야기될 수 있다. 또한 이 프로토콜에서는 *ID* 갱신을 통해서 전방향 안전성을 얻으려 했으나 *ID*가 갱신되는 방식이 단순한 exclusive OR 연산이기 때문에 현재 사용되는 *ID*가 노출됐다면, 이전 거래 역시 노출 될 수 있다. 예를 들어 현재의 *ID* 값이 $ID_2$이고 이 $ID_2$ 값이 노출되었다면 $r_1$ 역시 알 수 있는 값이기 때문에 $ID_1$ 역시 알 수 있다($ID_2 = ID_1 \oplus r_1 \rightarrow ID_1 = ID_2 \oplus r_1$).

## Ⅳ. 해쉬 함수 기반 인증 프로토콜의 분류

지금까지 우리는 저비용 RFID 시스템에 적용 가능한 해쉬 기반 인증 프로토콜에 대해 살펴보았다. 본 장에서는 기존 연구들을 기반으로 안전한 RFID 인증 프로토콜을 백엔드 데이터베이스와 태그 사이의 동기화 필요 유무에 따라 상태기반과 비 상태기반 인증 모델로 나누어 설명한다. 현재 제안되고 있는 해쉬 기반 인증 프로토콜들은 크게 이 두 가지 모델로 분류 가능하다. 이는 지금까지 진행되었던 RFID 인증프로토콜 연구에 대한 정리이며, 안전한 형태의 프로토콜을 설계하기위한 참조 모델로써의 의미를 부여할 수 있을 것이다. 본 장에서는 다음과 같은 용어들을 사용한다.




〔그림 4〕 Low-Cost RFID Authentication Protocol

- H : 일방향 해쉬 함수, $H:\{0,1\}^{*}\rightarrow\{0,1\}^{l}$
- $R_Q$ : 리더가 생성한 난수
- $R_T$ : 태그가 생성한 난수
- $R_D$ : 백엔드 데이터베이스가 생성한 난수
- $Sync$ : 동기 상태비트($Sync$= 1; 동기 상태)
- ‖ : 연접

## 4.1. 비 상태기반 인증모델

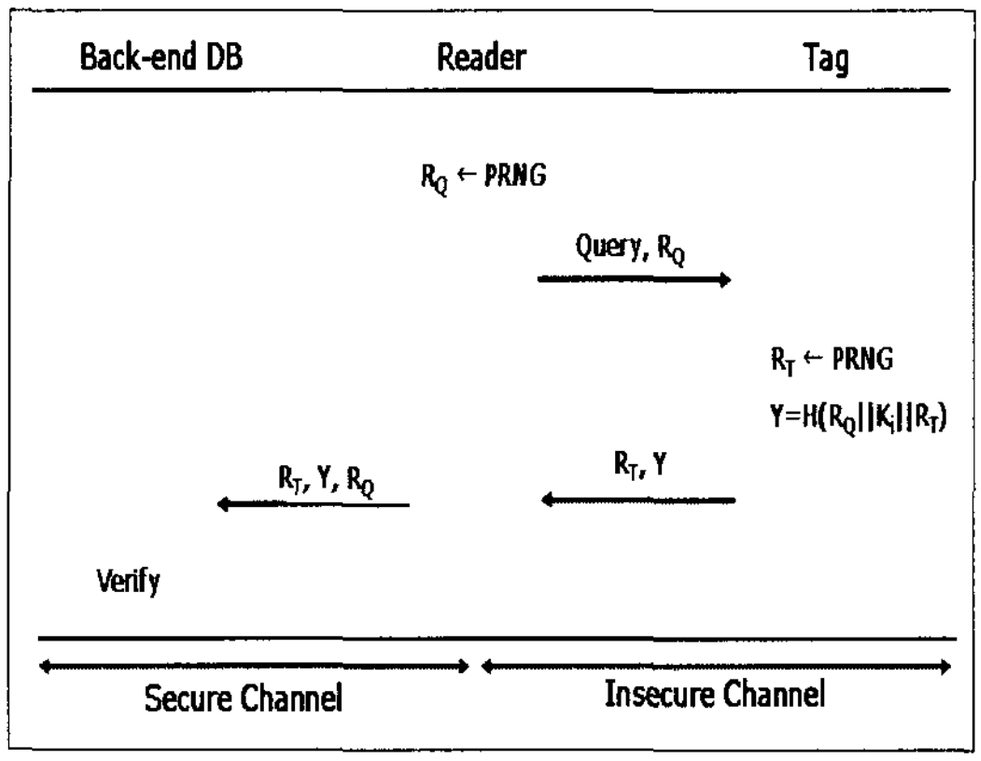
### 4.1.1. 비 상태기반 인증모델에 대한 기술

첫 번째로 태그와 백엔드 데이터베이스 사이에 동기가 필요 없는 비 상태기반 인증모델이다. 비 상태기반 모델로 분류 가능한 프로토콜에는 [5, 6, 7, 8, 9]등이 있으며, 이들은 큰 변형이 거의 없고 정규화 되어있기 때문에 안전도와 효율성을 고려하여 다음의 네 가지 알고리즘들로(Setup, Query, Respond, Verify) 구성된 일반적인 모델을 생각할 수 있다.

- **Setup** : 백엔드 데이터베이스에 등록된 태그의 수를 $N$ 이라고 하자. 백엔드 데이터베이스는 난수 키 $K_i$를 선택하여 각 태그에게 배분한다. 이 키 $K_i$는 백엔드 데이터베이스와 각 $Tag_i$가 공유하는 비밀정보이다.

- **Query** : 리더가 태그를 인식하면 난수 $R_Q$를 생성하여 $Query$ 신호와 함께 태그에 전달한다.

- **Respond** : $R_Q$와 $Query$를 받은 $Tag_i$는 난수 $R_T$를 생성하여 $Y= H(R_Q\|K_i\|R_T)$를 계산한다. 그리고 $Tag_i$는 $R_T$와 $Y$를 리더에게 보낸다. 리더는 $Tag_i$에게 받은 응답 $R_T$, $Y$와 자신이 생성했던 난수 $R_Q$를 백엔드 데이터베이스에게 보낸다.




〔그림 5〕 비 상태기반 인증모델

- **Verify** : $R_T$, $Y$, $R_Q$를 받은 백엔드 데이터베이스는 모든 태그들에 대해서 $Y_i'= H(R_Q\|K_i\|R_T)$를 계산하고 받은 $Y$ 값과 $Y_i'$ 값을 비교한다. 만약 $Y_i'= Y$인 $Y_i'$를 찾으면 $Tag_i$에 대한 정보를 전송한다. 그렇지 않은 경우에는 $Invalid$ 신호를 전송한다.

### 4.1.2. 안전도 분석

비 상태기반 모델에서 리더의 $Query$에 대한 태그의 응답은 해쉬 함수의 결과 값으로 만들어 지기 때문에, 공격자가 메시지의 내용을 엿듣는다 해도 태그의 비밀정보를 알아 낼 수는 없다. 즉, 도청을 피할 수는 없더라도 이를 통해 비밀정보가 유출되지는 않으므로 넓은 의미의 도청공격에 안전하다고 할 수 있다. 그리고 인증과정에서 리더와 태그는 매 세션마다 난수 $R_Q$ 와 $R_T$ 를 각각 생성하여 이를 해쉬 함수의 입력에 포함시키고 있다. 따라서 태그에서 리더로 보내지는 응답 메시지가 고정되지 않고 항상 변화하게 되며, 이를 통해 위치 추적 문제를 피할 수 있게 된다. 마찬가지로 태그의 응답은 정당한 리더가 매 세션마다 생성하는 난수 $R_Q$ 값을 바탕으로 이루어져야 하기 때문에, 이전의 응답을 저장했다가 이를 재전송해서 공격하는 재생 공격이나 스푸핑 공격에 안전하다.

하지만, 비 상태기반 인증 모델의 경우 백엔드 데이터베이스와 태그 사이에 비밀정보를 초기에 공유하고,

그 후에 갱신하지 않기 때문에 만약 태그가 공격자에 의해 포획되어 비밀 값이 노출될 경우, 과거에 이루어졌던 거래들에 대한 정보들 역시 노출된다. 다시 말해 전방향 안전성을 만족시키지는 못한다고 볼 수 있다. 하지만 이런 종류의 프로토콜은 몇 가지 장점을 가진다. 태그 쪽에서 비밀 정보를 갱신할 필요가 없기 때문에, 태그 쪽에서는 리더가 보내는 *Query*에 대한 응답만을 할뿐 리더 쪽의 응답을 굳이 받을 필요가 없다. 따라서 인증을 위해 요구되는 리더와 태그 사이의 pass의 수를 줄일 수 있고, 자연히 비동기 문제도 없다. 또한 미리 정해진 비밀정보만을 이용해 인증을 하기 때문에 비밀 정보의 갱신을 위해 필요한, 값 비싼 R/W 메모리가 필요 없다. 이는 태그의 가격을 낮추는 역할을 할 수 있다.

## 4.2. 상태기반 인증모델

### 4.2.1. 상태기반 인증모델에 대한 기술

두 번째로 태그와 백엔드 데이터베이스 사이에 동기가 필요한 상태기반 인증모델이다. 상태기반 모델로 분류 가능한 프로토콜에는 [3, 10, 11, 12, 13, 14, 15, 16, 17, 18]등이 있으며, 이들은 비밀 값을 갱신함으로써 프로토콜의 안전도를 높이고자 하는 공통점을 가지나 비밀 값을 갱신하는 방법이나 변화하는 비밀 값을 인증에 어떻게 이용하는지는 프로토콜마다 조금씩 다르다. 일반적으로 상태기반 인증모델의 경우 Setup, Query, Respond, Verify, Update의 다섯 가지 알고리즘들로 구성되며, 비 상태기반 모델과는 달리 비밀 정보를 갱신하는 Update 과정이 추가됨을 알 수 있다. 상태기반의 경우 앞서 언급한 것처럼, 비밀 값을 갱신하는 방법이나 변화하는 비밀 값을 인증에 어떻게 이용하는지에 따라 다양한 프로토콜이 설계될 수 있으므로 우리는 상태기반 인증모델에 속하는 하나의 형태를 제시하고 이를 통해 상태기반 인증모델을 설명한다. 프로토콜은 다음의 다섯 가지 알고리즘들로(Setup, Query, Respond, Verify, Update) 구성된다.

- **Setup** : 백엔드 데이터베이스에 등록된 태그의 수를 $N$ 이라고 하자. 백엔드 데이터베이스는 난수 키 $K_i$와 $TID_i$를 선택하여 각 태그에게 배분한다. 이 값들은 백엔드 데이터베이스와 각 $Tag_i$가 공유하는 비밀정보이며, 태그 쪽에서는 $K_i$와 $TID_i$를 각각 저장하고, 백엔드 데이터베이스는 $TID_i$ 값과 비동기가 발생했을 때 다시 동기를 회복하기 위해 $K_i$를 $K_{i0}$와 $K_{i1}$ 두 가지 형태로 저장한다. 이때, $K_{i0}$는 이전 인증에 사용된 비밀 키 값이고, $K_{i1}$는 현재 인증에 사용되는 비밀 키 값이다. 동기를 잃은 상태에서는 $Tag_i$를 찾기 위해 $K_{i0}$를 이용하고, 동기가 맞는 상태에서는 $Tag_i$를 찾기 위해 $TID_i$ 값이 사용된다. 프로토콜이 시작될 때 백엔드 데이터베이스는 $K_{i0}$와 $K_{i1}$에 같은 값 $K_i$를 설정한다.

- **Query** : 리더가 태그를 인식하면 난수 $R_Q$ 를 생성하여 *Query* 신호와 함께 태그에 전달한다.

- **Respond** : $R_Q$와 *Query*를 받은 $Tag_i$는 $R_T$를 선택한다. 이 때, *Sync* 값이 1인 경우에는 $R_T$를 $TID_i$로 설정한다. 만약 비동기 상태인 경우라면 ($Sync = 0$), $R_T$값을 난수 발생기로 생성한 값으로 설정한다. 이렇게 $R_T$를 선택한 다음 $Tag_i$는 $Y_T = H(R_Q \| K_i \| R_T)$를 계산하고 *Sync* 값을 0으로 한 후, 리더에게 $R_T$와 $Y_T$를 전송한다. 리더는 $Tag_i$에게 받은 응답 $R_T$, $Y_T$와 자신이 생성했던 난수 $R_Q$ 를 백엔드 데이터베이스에게 보낸다.

- **Verify** : $R_T$, $Y_T$, $R_Q$ 를 받은 백엔드 데이터베이스는, 저장하고 있는 $TID_i$ 필드에서 받은 $R_T$와 일치하는 값이 있는지를 찾는다. 만약 발견이 된다면, $Y_T' = H(R_Q \| K_{i1} \| R_T)$를 계산한다. 이때 $Y_T$ 값과 $Y_T'$ 값이 같다면 $b$값을 1로 하고 $Y_D = H(R_D \| K_{ib})$와 *Valid* 신호를 return한다. 그렇지 않은 경우에는 $b$를 0으로 하고 모든 태그들에 대해서 $Y_T$ 값과 $Y_T'$ 값을 비교한다. 백엔드 데이터베이스가 $Y_T = Y_T'$를 만족하는 $Y_T'$를 찾으면 $Y_D = H(R_D \| K_{ib})$와 *Valid* 신호를 전송한다. 만약 조건을 만족하는 $Tag_i$를 찾지 못하면 *Invalid* 신호를 전송한다.

- **Update** : Update 알고리즘은 각 개체에서 다음과 같이 서로 다르게 이루어진다.
  백엔드 데이터베이스 : $TID_i = H(R_T \| K_{ib})$,
  $K_{i0} = K_{ib}$, $K_{i1} = H(TID_i \| K_{ib})$ 로 갱신
  태그 : 리더로부터 받은 $R_D$와 $Y_D$를 이용해서 $Y_D' = H(R_D \| K_i)$를 계산한 다음 $Y_D$ 값과 $Y_D'$ 값

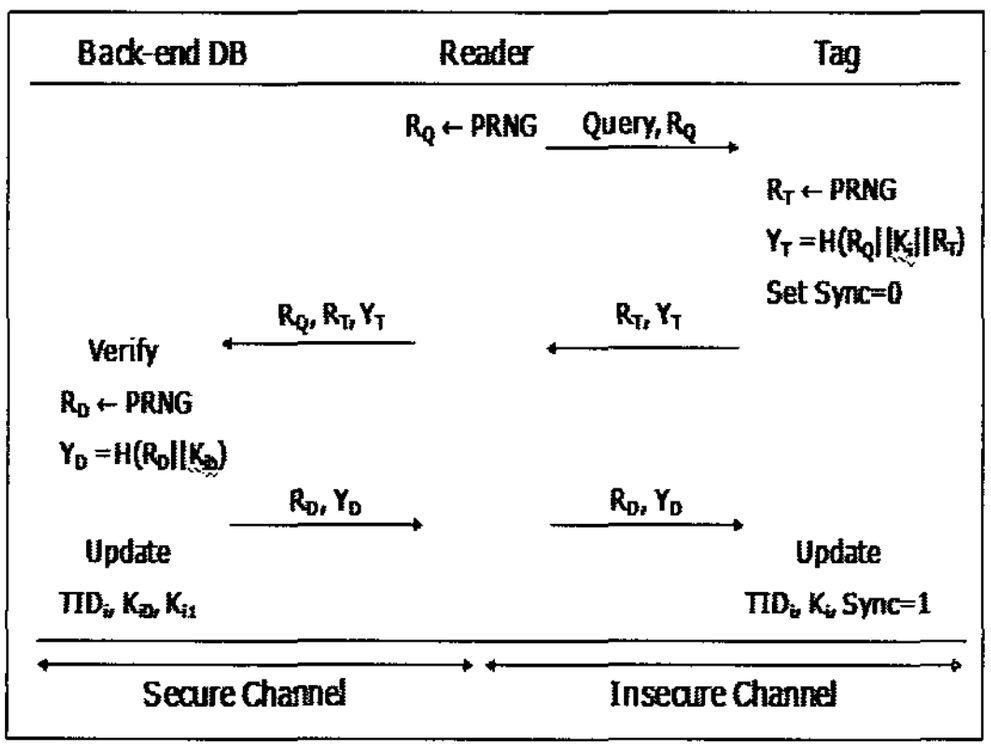
이 같은지를 비교해서 같은 값을 가지면 $K_i = H(TID_i \| K_i)$, $TID_i = H(R_T \| K_i), Sync = 1$로 갱신

#### 4.2.2. 안전도 분석

상태기반 인증모델은 태그와 백엔드 데이터베이스 사이에 동기가 필요한 모델로, 태그와 백엔드 데이터베이스는 비밀 정보를 바탕으로 인증을 하고, 성공적인 인증 후에는 이 비밀 정보를 갱신한다. Update 알고리즘이 추가되고, 이로 인해 백엔드 데이터베이스에서 기록하고 있어야 하는 비밀 값이 늘어나는 것을 제외하고는 비 상태기반 인증모델과 유사하다. 따라서 보안측면에서도 비 상태기반 인증모델과 거의 유사한 특징을 가진다.

상태기반 인증모델 역시 앞서 설명한 비 상태기반 인증모델에서처럼 리더의 *Query*에 대한 태그의 응답은 해쉬 함수의 결과 값으로 만들어지기 때문에, 공격자가 메시지의 내용을 엿듣는다 해도 태그의 비밀 정보를 알아 낼 수는 없다. 즉, 도청을 피할 수는 없더라도 이를 통해 비밀정보가 유출되지는 않으므로 넓은 의미의 도청공격에 안전하다고 할 수 있다. 그리고 인증과정에서 리더와 태그는 매 세션마다 난수 $R_Q$와 $R_T$를 각각 생성하여 이를 해쉬 함수의 입력에 포함시키고 있다. 따라서 태그에서 리더로 보내지는 응답메시지가 고정되지 않고 항상 변화하게 되며, 이를 통해 위치 추적 문제를 피할 수 있게 된다. 마찬가지로 태그의 응답은 정당한 리더가 매 세션마다 생성하는 난수 $R_Q$ 값을 바탕으로 이루어져야 하기 때문에, 이전의 응답을 저장했다가 재전송해서 공격하는 재생 공격이나 스푸핑 공격에 안전하다. 그리고 상태기반 인증모델의 경우 비밀정보가 고정된 것이 아니라 변화하기 때문에, 태그가 공격자에 의해 포획되어 비밀 값이 노출될 경우라도 과거에 이루어졌던 거래들에 대한 정보들이 노출되지 않는다. 즉, 전방향 안전성 역시 만족한다고 볼 수 있다.

하지만 이런 종류의 프로토콜 역시 몇 가지 단점을 가진다. 태그는 리더로부터 응답을 받고 그를 바탕으로 비밀정보를 갱신할지의 여부를 결정하기 때문에 리더와 태그 사이의 pass가 최소한 3번은 형성되어야한다. 또한 매 세션마다 비밀정보를 갱신해야하기 때문에, 값 비싼 R/W 메모리가 필요하게 되고 이는 태그의 가격을 높이는 역할을 할 수 있다.




〔그림 6〕 상태기반 인증 모델

## V. 빠른 태그 검색 방법

지금까지 우리는 여러 가지 위협에 안전한 RFID 인증 프로토콜을 백엔드 데이터베이스와 태그 사이의 동기화 필요유무에 따라 두 가지 모델로 나누어 살펴보았다. 본 장에서는 RFID 시스템의 인증과정에서 발생 할 수 있는 태그 검색 문제에 대해 정의하고 이를 해결하기 위해 *Bloom filter*를 이용하는 방법을 소개한다.

### 5.1. 태그 검색 문제

RFID 시스템에서 리더는 태그에 *Query*를 하고 태그는 이에 대한 응답을 한다. 리더를 통해 태그의 응답을 전해 받은 백엔드 데이터베이스는 사전에 등록된 태그들의 정보 중에서 응답한 태그와 일치하는 정보를 찾아내어 태그를 인증한다. 하지만 백엔드 데이터베이스에 등록되어 있는 태그의 수는 많고, 이렇듯 많은 태그 정보들 중에서 특정 태그에 대한 정보를 찾아내어 인증을 해야하기 때문에, 인증에 필요한 시간이 길어질 수 있다. 즉, 이전에 제안되었던 많은 인증프로토콜에서 백엔드 데이터베이스는 태그를 찾기 위해 등록되어 있는 모든 태그들에 대해서 하나씩 맞는지를 확인함으로써 태그를 찾는다. 이런 경우, 백엔드 데이터베이스에서는(등록된 태그의 수가 N개라고 했을 때) 평균 N/2번의 확인을 거쳐야 태그를 찾을 수 있다. 본 논문에서는 이러한 문제를 태그 검색 문제라 하고, 본 장에서는 태그 검색 문제를 해결하기 위한 접근을 소개한다.

Tassos 등은 [7]에서 태그들을 트리 형태로 구조화함으로써 백엔드 데이터베이스에서의 검색을 대수적으로

감소 시켰다[7]. 하지만, 이를 위해서 각 태그는 트리의 깊이에 해당하는 수만큼의 키를 가져야 하고, 마찬가지로 이 수만큼 의사 난수 함수를 계산하여야 하기 때문에 계산 능력이 떨어지는 태그에게 부담이 되는 단점이 있다. 그리고 N.W. Lo등은 [18]에서 fast matching key와 search index value를 사용해서 인증시간을 단축시키는 Efficient Identity Match Scheme을 제안하였으나 스푸핑 공격에 취약하다[18]. 본 장에서는 *Bloom filter*를 사용한 빠른 태그 검색 방법을 제안한다.

### 5.2. *Bloom filter*를 이용한 빠른 태그 검색 방법

우리는 본 절에서 미리 계산된 *Bloom filter*를 이용하여 백엔드 데이터베이스에서의 인증 시간을 단축시키는 방법을 제안한다.

#### 5.2.1. *Bloom filter*

프로토콜을 살펴보기 이전에, 멤버쉽 테스트 함수로 사용되는 *Bloom filter*에 대해 간단히 살펴본다. *Bloom filter*는 주어진 원소가 어떤 집합에 속하는지 여부를 검사하는데 사용할 수 있는 기법이다[19]. 이 기법은 공간효율성이 좋고 간단하다는 장점을 가지지만, 긍정오류율(False Positive Rate, 집합에 포함되어 있지 않은 원소를 포함되어 있다고 잘못 판단하는 확률)이 존재한다. 긍정오류율은 (1)과 같이 나타나며 $k, s, m$의 값을 통해 조정할 수 있다. 이 때, $k$는 *Bloom filter*가 사용하는 해쉬 함수의 수를 의미하고, $s$는 집합에 속해 있는 원소의 수를, $m$은 해쉬 함수의 범위($\{1, \ldots, m\}$)를 나타낸다.

$$f = (1-(1-\frac{1}{m})^{ks})^k \quad (1)$$

$k = \ln 2 \cdot (m/s)$일 때 긍정오류율은 최소값을 가지며, 이 경우 긍정오류율은 식 (2)와 같이 나타난다.

$$f \approx (\frac{1}{2})^k \approx (0.6185)^{\frac{m}{s}} \quad (2)$$

#### 5.2.2. 알고리즘에대한 기술

빠른 태그 검색 방법은 다음의 네 가지 알고리즘으로 (Setup, Query, Respond, Verify) 구성되며, 이는 일반적인 인증프로토콜에서의 전수 조사를 대체할 수 있다.

- **Setup** : 백엔드 데이터베이스는 모든 가능한 $R_{BF}$ (where $R_{BF} \in \{0, 1, \ldots, 2^t - 1\}$)에 대해서 $H(R_{BF} \| K_i)$를 계산하고, 이러한 해쉬 값들로부터 각 $Tag_i$의 $BF_i$를 계산한다. 이 $BF_i$ 값은 $Tag_i$의 *Bloom filter*이며 이는 빠른 태그 검색을 위한 identifier 역할을 한다. 백엔드 데이터베이스는 모든 $Tag_i$에 대한 $BF_i$를 순차적으로 계산한다.

- **Query** : 리더가 태그를 인식하면 난수 $R_Q$를 생성하여 *Query* 신호와 함께 태그에 전달한다.

- **Respond** : $R_Q$와 *Query*를 받은 $Tag_i$는 난수 $R_{BF}$를 생성하여 $R_T = H(R_{BF} \| K_i)$와 $Y = H(R_Q \| K_i \| R_T)$를 계산한다. 그리고 $Tag_i$는 $R_T$와 $Y$를 리더에게 보낸다. 리더는 $Tag_i$에게 받은 응답 $R_T$, $Y$ 와 자신이 생성했던 난수 $R_Q$를 백엔드 데이터베이스에게 보낸다.

- **Verify** : $R_T$, $Y$, $R_Q$를 받은 백엔드 데이터베이스는 $R_T$에 대한 멤버쉽 테스트를 수행하여 테스트를 통과하는 $BF_i$를 찾는다. 만약 백엔드 데이터베이스가 테스트를 통과하는 $BF_x$를 찾았다면 $Tag_x$에 대한 $Y_x' = H(R_Q \| K_x \| R_T)$를 계산하고 $Y_x'$ 값이 $Y$ 값과 같은지를 체크한다. 만약 $Y_x'$ 값이 $Y$ 값과 같은 값이면 백엔드 데이터베이스는 $Tag_x$에 대한 정보를 전송한다. 그렇지 않은 경우에는 멤버쉽 테스트를 통과하는 $BF_i$를 찾고 $Y_i'$ 값과 $Y$ 값이 같은지 체크하는 과정을 계속 수행한다. 만약 백엔드 데이터베이스가 $Y_i'$ 값과 $Y$ 값이 같은 $Y_i'$를 찾지 못하면 *Invalid* 신호를 전송한다. 이때 멤버쉽 테스트를 통과하는 $BF_i$의 수는 $p$라고 가정한다.

위에서 언급하였듯이, 우리가 제안한 빠른 태그 검색 방법은 *Bloom filter*를 이용한다. *Bloom filter*는 유한개의 원소로 이루어진 어떤 집합에 멤버쉽 질의를 제공하는 공간 효율적인 데이터 구조로, 가장 효율적인 멤버쉽 테스트 알고리즘으로 알려져 있다. 실제로 멤버쉽 테스트는 어떤 특정 위치의 비트 값이 1인지 0인지 체크하는 방식으로 이루어지기 때문에 해쉬 함수를 계산하는

것보다 훨씬 효율적이다. 이점을 이용하여, 우리의 방법에서는 *Bloom filter*를 태그의 빠른 검색을 위한 identifier로 사용한다.

*Bloom filter*는 각 태그로부터 생성된 해쉬 값으로부터($H(R_{BF}\|K_i)$) 생성되며, 주어진 값 $R_T$가 특정 태그가 만들 수 있는 집합의 원소인지 아닌지를 테스트한다. *Bloom filter*는 긍정오류(False Positive)를 가지며, 이 긍정오류율에 의해 백엔드 데이터베이스에서 해쉬 함수 값 비교가 최대 몇 번까지 수행되는지가 결정 된다($p$ : 멤버쉽 테스트를 통과하는 $BF_i$의 수, 해쉬 함수 값을 비교하는 작업을 최대 $p$번까지 해야 함을 의미). 만약 긍정오류율이 낮다면 $p$값은 작을 것이고 반대로 긍정오류율이 높다면, $p$값은 커질 것이다. 따라서 긍정오류율을 조정한다면 $p$값을 조절할 수 있게 된다. 표 1.에서는 이전 연구 결과들과 제안한 방법의 효율성을 분석한 결과를 제시한다. 등록된 태그의 수가 1,000,000개라고 가정하면 [2, 14]에서는 태그를 찾기 위해 전수 조사를 하기 때문에 평균 500,000번, 최대로는 1,000,000번 해쉬 연산이 필요하다. 태그의 효율적인 접근을 시도한 [7]에서는 태그를 찾기 위해 20번의 해쉬 연산만을 필요로 한다. 하지만 태그에서도 20번의 해쉬 연산이 필요하므로 저비용 RFID 시스템에 부적합하다. [18]의 경우에는 시스템 파라미터 n과 태그가 선택하는 난수 m에 따라 효율성이 결정되고, 보안상 취약하므로 완벽한 해결책이 될 수는 없다. 제안한 방법은 [7]논문에서처럼 해쉬 연산수를 20회 정도로 가정하더라도, 태그에서 필요한 해쉬 연산수는 2회로 저비용 RFID 시스템에서도 효율적으로 사용될 수 있다.

또한, 제안하는 방법은 상태기반모델과 비 상태기반 모두에 적용될 수 있으며, 적용되는 모델에 기반하는 안전도를 가진다. 즉, 비 상태기반 모델의 경우에는 전방향 안전성을 제외한 도청, 스푸핑, 재생 공격, 위치 추적 등에 안전하다. 그리고 상태기반에 적용할 경우에는 2장에서 제시한 모든 보안요소를 만족한다.

### 5.2.3. 파라미터 설정

실제로, 파라미터들을 알고리즘에 적용시켜보면 다음과 같다. 우선 *Bloom filter*의 긍정오류율을 $f$라고 가정하면 $BF_i$를 통과하는 $R_T$의 개수는 다음과 같다.

$$N_{pass} = f \cdot (2^l - 2^t) + 2^t$$
$$= f \cdot 2^l + 2^t(1-f) \approx f \cdot 2^l \quad (3)$$

(3)의 수식은 다음을 통해 얻어진다. $R_T$가 될 수 있는 값의 총 개수는 해쉬 함수 결과 값의 비트길이가 $l$이므로 총 $2^l$개가 된다. 그리고 특정태그 하나에 대해서 가능한 $H(R_{BF}\|K_i)$ 값은 $R_{BF}$ 값이 $t$ 비트길이이기 때문에 총 $2^t$개가 된다. 따라서 $BF_i$를 통과하는 $R_T$의 개수는 정상적으로 멤버쉽 테스트를 통과하는 $2^t$개에, 집합의 원소가 아님에도 긍정오류율 때문에 통과할 수 있는 원소의 수 $f \cdot (2^l - 2^t)$개를 합한 $f \cdot (2^l - 2^t) + 2^t$개가 된다.

우리는 앞에서 멤버쉽 테스트를 통과하는 $BF_i$의 개수를 최대 $p$개라고 가정하였기 때문에 백엔드 데이터베이스에 등록된 태그의 개수가 $2^n$개라고 할 때의 $p$ 값을 다음과 같이 얻을 수 있다.

$$p = \frac{N_{pass} \cdot 2^n}{2^l} \approx \frac{f \cdot 2^l \cdot 2^n}{2^l} \approx f \cdot 2^n \quad (4)$$

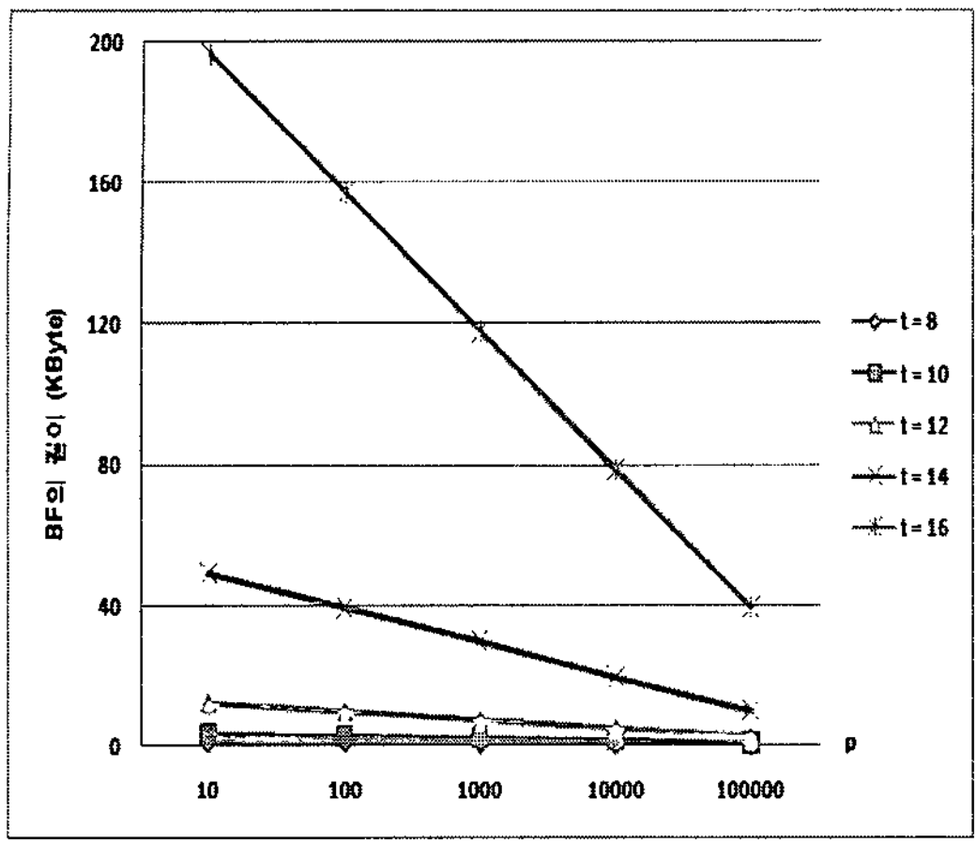
(4)의 수식을 통해서 $p$ 값이 긍정오류율 $f$와 등록된 총 태그의 개수에 의해 정해짐을 알 수 있다. 한편, 긍정오류율 $f$는 $BF_i$의 비트길이 $m$과 $R_{BF}$의 비트길이 $t$에 의해 결정된다. 따라서 $p$ 값은 $m$과 $t$값을 조정함으로써 다음과 같이 정할 수 있다.

$$p = (0.6185)^{\frac{m}{2^t}} \cdot 2^n \quad (5)$$

〔표 1〕 제안한 방법의 효율성 분석

| | [2] | [14] | [7] | [18] | 제안한 방법 |
|---|---|---|---|---|---|
| 태그에서 필요한 해쉬연산의 수 | 2 | 3 | 20 | m | 2 |
| 태그 검색을 위해 백엔드 데이터베이스가 수행하는 최대 해쉬연산의 수 | 1,000,000 (전수 조사) | 1,000,000 (전수 조사) | 20 | n-m | 20 |

* n : 시스템에서 미리 정해놓은 양수, m : 태그가 선택한 난수
* m 값이 작아지면 백엔드 데이터베이스의 부담이 늘어나고, 커지면 태그의 부담이 늘어나므로 trade off




〔그림 7〕 $R_{BF}$의 비트길이와 p값에 따른 $BF_i$의 길이

이제 $p$ 값을 어떻게 정할지 생각해보자. $p$ 값은 백엔드 데이터베이스에서 최대로 수행해야 하는 해쉬 연산 수이기 때문에 $p$ 값이 작을수록 효율성이 커진다. 따라서 $p$ 값을 줄일 수 있다면 가능한 가장 작게 해야 할 것이다. 하지만, $p$ 값이 줄어들면, $m$값이 커지거나 $t$ 값이 줄어야하기 때문에 $p$ 값을 무한정 줄일 수는 없다. $m$ 값은 $BF_i$의 비트길이를 나타내는 값이기 때문에 이는 백엔드 데이터베이스에서 저장해야 하는 *Bloom filter*의 크기와 연결된다. 즉, 백엔드 데이터베이스는 $m \cdot 2^n$ bits를 저장해야 한다. 백엔드 데이터베이스를 계산능력이 충분한 장치라고 가정하기는 하지만, 백엔드 데이터베이스가 제공하는 메모리 범위를 초과해서 사용 할 수는 없기 때문에, $m$ 값을 무한정 키울 수는 없다. 또한 $t$ 값은 $R_{BF}$의 비트 길이를 결정한다. 따라서 $t$ 값이 작아지면 $Tag_i$가 만들 수 있는 $R_T$의 개수가 줄어들게 된다. 따라서 $p$ 값을 정할 때는 위의 두 가지 요소를 생각해서 결정해야 한다.

[그림 7].은 $R_{BF}$의 비트 길이와 $p$ 값에 따른 $BF_i$의 비트길이를 보여 준다. 예를 들어, 백엔드 데이터베이스에 등록된 태그의 수를 1,000,000으로, $p$ 값을 20으로, 빠른 태그 검색을 위한 $R_{BF}$를 8bits 라고 하면, 백엔드 데이터베이스는 $BF_i$의 길이를 723 Bytes로 하면서 최대 20번의 해쉬로 태그를 검색할 수 있다.

## Ⅵ. 결 론

본 논문에서는 RFID 시스템의 특성에 기인한 여러 보안 측면에서의 취약점을 분석하고, 이들 취약점을 해결하기 위한 방안을 기존의 연구를 바탕으로 제시하였다. 이를 위해서 RFID 인증 프로토콜의 초기 모델부터 시작해서 현재까지 제안된 여러 인증 프로토콜을 분석했고, 이를 통해 보안 측면에서 알려진 위협에 대응하기 위한 인증 프로토콜의 일반적인 형태를 두 가지로 분류하였다. 그 첫째가 비 상태기반 인증모델이고 둘째가 상태기반 인증모델이다. 이는 인증에 참여하는 두 개체, 태그와 백엔드 데이터베이스 사이에 동기가 요구되느냐의 여부에 따라 분류한 것이다. RFID 인증에서의 일반 모델은 기존의 RFID 인증연구에 대한 총정리의 의미로 볼 수 있고, 안전한 형태의 프로토콜을 설계하기 위한 참조모델의 의미를 부여할 수 있을 것이다. 또한 본 논문에서는 멤버쉽 테스트 알고리즘을 이용해 태그 검색을 빨리하는 방법을 제안했다. 우리의 방법은 인증프로토콜의 일반적인 두 가지 모델에 모두 적용 가능하며, 백엔드 데이터베이스 쪽에서 미리 계산된 *Bloom filter*를 이용하여 검색공간을 줄일 수 있기 때문에 태그 검색에 필요한 시간을 단축시킬 수 있는 새로운 방법이다.

## 참고문헌

[1] Stephen August Weis, ``Security and Privacy in Radio-Frequency Identification Devices," *Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY, May 2003.*

[2] Keunwoo Rhee, Jin Kwak, Seungjoo Kim, and Dongho Won, ``Challenge-Response Based RFID Authentication Protocol for Distributed Database Environment," *SPC 2005*, LNCS 3450, pp. 70-84, 2005.

[3] Su Mi Lee, Young Ju Hwang, Dong Hoon Lee, and Jong In Lim, ``Efficient Authentication for Low-Cost RFID Systems," *ICCSA 2005*, LNCS 3480, pp. 619-627, 2005.

[4] Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan M. Estevez-Tapiador, and Arturo Ribagora, "RFID Systems: A Survey on Security Threats and Proposed Solutions," *PWC 2006*, LNCS 4217, pp. 159-170, 2006.

[5] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels, ``Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," *Security in Pervasice Computing 2003*, LNCS 2802, pp. 201-212, 2004.

[6] Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshita, "Cryptographic Approach to Privacy-Friendly Tags," RFID Privacy Workshop, Nov 2003

[7] Tassos Dimitriou, "A Second and Efficient RFID Protocol that could make Big Brother (partially) Obsolete," *PERCOM'06*, IEEE, 2006.

[8] Gene Tsudik, "YA-TRAP: Yet Another Trivial RFID Authentication Protocol," *PERCOMW'06*, IEEE, 2006.

[9] Christy Chatmon, Tri van Le and Mike Burmester, "Secure Anonymous RFID Authentication Protocols" *Technical Report TR-060112,* Florida State University, Department of Computer Science, 2006.

[10] Dirk Henrici and Paul Muller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers," *PERCOMW'04*, IEEE, 2004.

[11] Tassos Dimitriou, "A Lightweight RFID Protocol to protect against Traceability and Cloning attacks," *SECURECOMM'05*, IEEE, 2005.

[12] Jang-Su Park and Im-Yeong Lee, “RFID Authentication Protocol Using ID Synchronization in Insecure Communication," *ICHIT'06*, IEEE, 2006.

[13] 하재철, 하정훈, 박제훈, 김환구, 문상재, ``분산 환경에 적합한 저비용 RFID 인증 프로토콜," *한국정보보호학회 하계학술대회 2007*, Vol. 17, pp. 78-83, 2007.

[14] Sangshin Lee, Tomoyuki Asano, Kwangjo Kim, "RFID Mutual Authentication Scheme based on Synchronized Sectet Information," *SCIS 2006*, IEICE, 2006.

[15] JaeCheol Ha, JungHoon Ha, SangJae Moon, and Colin Boyd, ``LRMAP: Lightweight and Resynchronous Mutual Authentication Protocol for RFID System," *ICUCT 2006*, LNCS 4412, pp. 80-89, 2007.

[16] 하재철, 박제훈, 윤신숙, 김환구, 문상재, ``경량화된 RFID 상호 인증 프로토콜 제안," *한국정보보호학회 동계학술대회 2007*, Vol. 17, No. 2, pp. 331-336, 2007.

[17] Song Han, Vidyasagar Potdar, and Elizabeth Chang, "Mutual Authentication Protocol for RFID Tags Based on Synchronized Secret Information with Monitor," *ICCSA 2007*, LNCS 4707, pp. 227-238, 2007.

[18] N.W. Lo and Kuo-Hui Yeh, ``Novel RFID Authentication Schemes for Security Enhancement and System Efficiency," *SDM 2007*, LNCS 4721, pp. 203-212, 2007.

[19] Andrei Broder and Michael Mitzenmacher, "Network Applications of Bloom Filters: A Survey," *Internet mathematics*, Vol. 1, pp. 485-509, 2004.

## 〈著者紹介〉


**김 진 호 (Jin Ho Kim) 학생회원**

2006년 2월 : 이화여자대학교 정보통신학과 졸업
2008년 2월 : 포항공과대학교 전자전기공학과 석사
2008년 3월~현재 : 삼성전자 정보통신총괄
<관심분야> 정보보호, 암호이론

**서 재 우 (Jae Woo Seo) 학생회원**

2005년 2월 : 경북대학교 전자전기공학부 졸업
2005년 3월~현재 : 포항공과대학교 전자/전기공학과 박사과정
<관심분야> 정보보호, 암호이론, 암호 프로토콜

**이 필 중 (Pil Joong Lee) 종신회원**

1974년 2월 : 서울대학교 전자공학과 학사
1977년 2월 : 한국대학교 전자공학과 석사
1982년 6월 : U.C.L.A System Science. Engineer
1985년 6월 : U.C.L.A Electrical Engineering. Ph.D.
1980년 6월~1985년 8월 : Jet Propulsion Laboratory. Senior Engineer
1985년 8월~1990년 2월 : Bell Communications Research. M.T.S.
1990년 2월~현재 : 포항공과대학교 전자전기공학과 교수
1996년 2월~1997년 2월 : NEC Research Institute 방문 연구원
2000년 9월~2003년 8월 : 포항공과대학교 정보통신 연구소장 (정보통신 대학원장 겸임)
2004년 1월~2004년 12월 : 한국정보보호학회 회장
2004년 1월~2004년 12월 : KT 정보보호 자문위원
2007년 1월~현재 : 한국공학한림원 정회원
<관심분야> 정보보호전반