

일회성 난수를 이용한 안전한 RFID 상호인증 프로토콜 설계

(Design of RFID Mutual Authentication Protocol using One Time Random Number)

김대중[†] 전문석^{**}

(Kim-Dae Jung) (Jun Moon Seog)

요약 최근 연구된 RFID 인증 프로토콜들에는 위치추적, 재전송 공격, 스푸핑 공격 등에 취약점이 여전히 남아있다. 본 논문에서는 리더나 태그에서 난수를 생성하는 기존의 연구된 프로토콜들과는 달리 Back-End DB에서 일회성 난수를 생성하고, 이 난수를 상호인증에 사용함으로써 위치추적, 재전송 공격, 스푸핑 공격에 안전하게 프로토콜을 설계하였다.

키워드 : RFID, 일회성 난수, 인증, 상호인증, RFID 시스템

Abstract Recently, researched RFID authentication protocols still have vulnerability of attack, such as location tracking attack, replay attack, spoofing attack etc. This paper designed method of making one time random number in DB server side unlike previously researched protocols, and it protects RFID communication from location tracking, replay attack and spoofing attack.

Key words : RFID, one time random number, Authentication, Mutual Authentication, RFID system

1. 서론

RFID(Radio Frequency Identification) 시스템은 리더기와 태그의 접촉 없이 무선 통신을 하여 태그(Tag)안의 정보를 읽거나 쓸 수 있는 시스템이다. 이러한 기술은 칩의 저장 공간의 확장과 인식능력이 발전하면서 유비쿼터스 환경에서 필수적인 기술로 인식되고 있다. 또한 기존의 바코드나 자기 인식 장치의 단점을 보완하여, 물류관리, 재고관리, 항만관리, 동물관리 등 소비가 증가하고 있는 차세대 핵심기술로 주목받고 있다[1,2].

그러나, RFID태그의 사용으로 인해 개인의 프라이버시 침해의 위험요소가 존재한다. 단순한 전자태그의 경

우 근처의 리더기의 영향으로 EPC(Electric Product Code)인 ID 및 고유식별번호를 일정반경내의 모든 리더나 태그들에게 보내게 된다. 이 ID나 고유식별번호는 기술 표준 개발을 위한 연구단체인 AutoID Center가 부여하는 것으로 제품의 정보를 갖게 된다. 예를 들면 제조업자, 제품의 형태, 제품의 생산지, 제품의 유효기간 등의 정보를 갖게 된다. 제품을 구입한 사용자는 다른 리더기 근처를 지나치면서 자신이 산 제품의 정보를 흘리게 된다. 어떠한 옷을 샀는지, 사이즈는 얼마인지 등의 정보가 노출될 수 있다. 개인의 취향이 외부로 알려지게 되면 상업적 목적으로 불법적인 사용이 일어날 수도 있고, 또한 자신의 위치정보가 추적될 수 있다[3].

RFID의 편리함 뒤편에는 이를 악용하는 사례가 발생하면서 정보를 정당한 권리를 가진 개체가 소유해야 한다는 인식을 하지만, 모든 RFID 시스템 또한 이런 공격에 대해 반드시 안전하지는 않다. 이러한 침해 문제를 해결하기 위해 많은 연구가 진행되어 왔음에도 불구하고, 기존의 RFID 통신에 취약점이 존재한다[4,5]. 태그의 재사용이 불가능하거나, 위치추적이 쉬우며, 재전송이나 스푸핑 공격에 약하다[6]. 본고에서 기존의 방법들의 문제점을 지적하고 제안한 프로토콜을 통해 이러한 취약점을 보완하고 유비쿼터스 환경에 적용 가능한 RFID 상

[†] 정 회 원 : 상호저축은행중앙회 IT본부 시스템운영팀 팀장
djkim@fsb.or.kr

^{**} 종신회원 : 숭실대학교 컴퓨터학부 교수
mjun@ssu.ac.kr

논문접수 : 2007년 10월 11일

심사완료 : 2008년 2월 25일

Copyright © 2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 정보통신 제35권 제3호(2008.6)

호인증 프로토콜을 제안하였다. 2장에서는 기존의 연구된 RFID 프로토콜을 살펴보고 존재하는 문제점을 기술한다. 3장에서는 기존의 RFID 인증 프로토콜을 보완하여 RFID 상호 인증 프로토콜을 제안한다. 리더-태그간 보다 안전한 데이터 통신을 보장하고 상호인증이 가능하며, DB서버(Back-end DB)에서 발생한 일회성 난수를 이용하여 RFID 상호 인증하는 프로토콜을 제안하고 설계한다. 4장에서는 제안한 프로토콜의 보안성과 성능을 비교분석 한다. 기존의 RFID 인증 프로토콜과 제안 프로토콜의 구조와 태그의 각 기능 연산횟수를 비교하였고 후반부에는 프로토콜의 보안성을 비교한 결과를 기술한다. 마지막으로 5장으로 결론을 맺도록 하겠다.

2. 관련연구

RFID 시스템의 소프트웨어적 보안기법으로 다음과 같이 해쉬-락 기법, 확장된 해쉬-락, 해쉬기반 ID 변형 기법, 개선된 해쉬기반 ID 변형기법 등이 있다[4].

2.1 해쉬-락 기법

MIT에서 Weis가 연구한 기법으로 저렴한 태그로 리소스 제한문제를 해결하면서 인가 받은 리더에게만 태그 정보를 전송한다[4]. 저비용의 태그의 리소스 제한을 해결하기 위해 태그에 하드웨어적으로 해쉬함수만을 구현했다 가정한다.

해쉬-락 기법에서 연산도구로 해쉬 함수를 갖는 태그의 소유자에 의해 자신의 실질적인 데이터 ID를 비롯하여 인증과정에 필요한 일시적인 metaID 값을 저장한다.

2.1.1 문제점

MIT 해쉬-락 기법은 ID대신 metaID를 사용하여 ID의 노출을 방지한다는 장점이 있기는 하지만 그림 1의 ⑥ 단계에서 ID를 리더에게 넘겨주는 과정이 metaID의 사용을 무의미하게 만든다. 리더와 태그 사이는 불안정한 채널이므로 암호화나 복잡성을 부여하는 과정 없이 data를 넘겨주게 되면 도청의 위험이 따르게 된다. MIT 해쉬-락 기법은 다음과 같은 공격에 취약하다.

- 위치 추적 공격 : 리더 요청에 의한 응답으로 매번 같은 metaID 값이 전송되므로 data 추적이 가능하며,

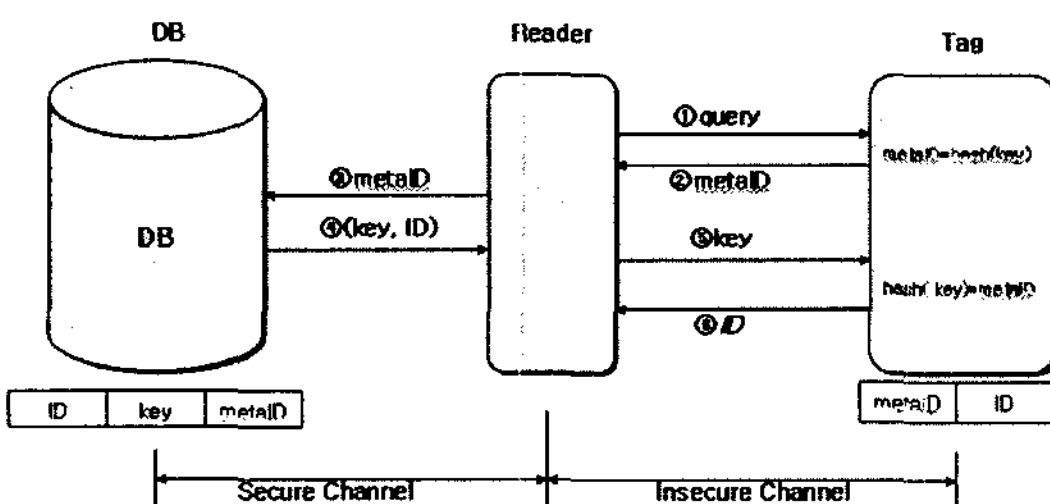


그림 1 MIT의 해쉬-락 기법

metaID가 ID대신 식별자의 역할을 수행하기 때문에 사용자가 추적되는 문제를 피할 수 없다.

- 재전송 공격 : 악의적인 리더가 정당한 리더로 가장하여 태그로부터 metaID를 얻을 수 있다. 이 정보를 악의적인 태그 안에 저장하며, 이 악의적인 태그가 정당한 태그로 가장하여 리더 요청의 대한 응답으로 탈취한 metaID값을 전송하게 되면 리더는 의심 없이 다음 단계를 수행하게 되고, 그 결과로 태그는 리더로부터 key까지 얻을 수 있다. 이 key를 가지고 다시 정당한 리더로 가장하여 태그에게 전송한다면 태그는 리더는 정당한 리더로 인증하고 ID를 넘겨주게 된다.

- 스푸핑 공격 : 앞에서 재전송 공격의 취약성으로 metaID, key, ID를 모두 획득할 수 있기 때문에 정당한 태그로 가장하여 공격이 가능하다. 악의적인 공격자가 key를 한번 획득하게 되면 그 태그 보안성이 사라졌기 때문에 태그의 보안을 필요로 하는 곳에서는 사용할 수 없게 된다.

2.2 확장된 해쉬-락 기법

이 방법은 해쉬-락 기법을 확장하였다. 그림 2에서 태그는 랜덤 넘버를 이용하여 리더 요청 쿼리에 매번 다른 응답을 하기 때문에 태그의 위치추적은 어렵게 된다[4]. 악의적인 공격자가 리더의 요청에 대한 응답값 $R, h(ID_k || R)$ 을 획득하였다 하더라도 다음 세션에는 또 다른 값이 생성되기 때문에 어느 태그에서 고정적인 data값이 나오는지 관찰하기는 어렵다. 따라서 위치추적은 어렵다.

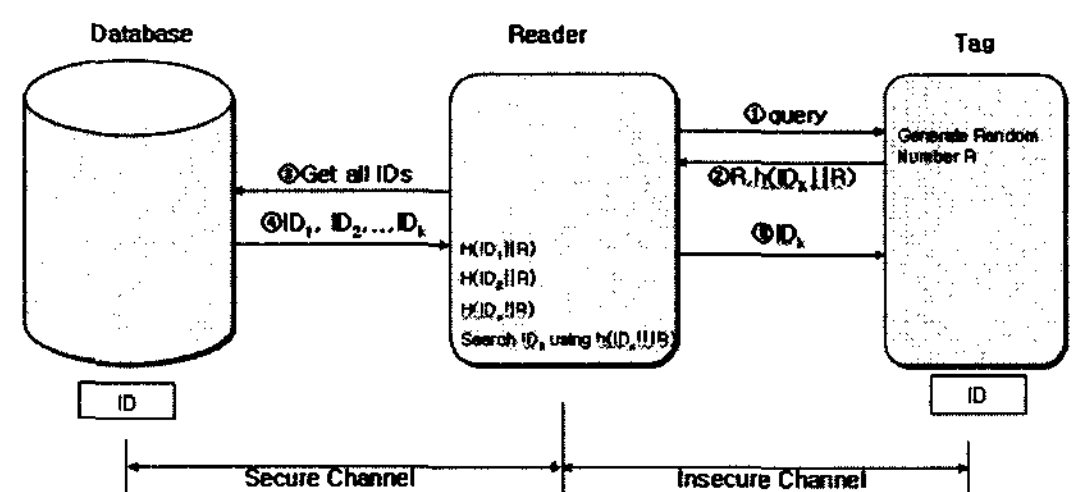


그림 2 확장된 해쉬-락 기법

2.2.1 문제점

- 재전송 공격 : 악의적인 리더가 정당한 리더로 가장하여 태그로부터 $R, h(ID_k || R)$ 을 획득할 수 있다. 다시 이 정보를 악의적인 태그 안에 저장한다. 이 정당한 태그로 가장한 태그에게 리더가 요청 쿼리를 전송하면 그에 대한 응답으로 저장되었던 정보를 보내고 다음 단계들을 의심 없이 수행하여 이 태그를 정당한 태그로 인증하게 된다.

- 스푸핑 공격 : 앞에서 태그의 랜덤 값 R과 $h(ID_k || R)$ 를 탈취 가능하고 ⑤ 단계에서의 ID_k도 탈취가 가능

하다. 거의 모든 정보를 탈취할 수 있기 때문에 정당한 태그로 가장하는 것이 가능하다.

2.3 해쉬기반 ID 변형 기법

Henrici와 Muller가 제안한 해쉬에 기반하여 ID를 갱신함으로써 위치트래킹 공격을 방지한다.

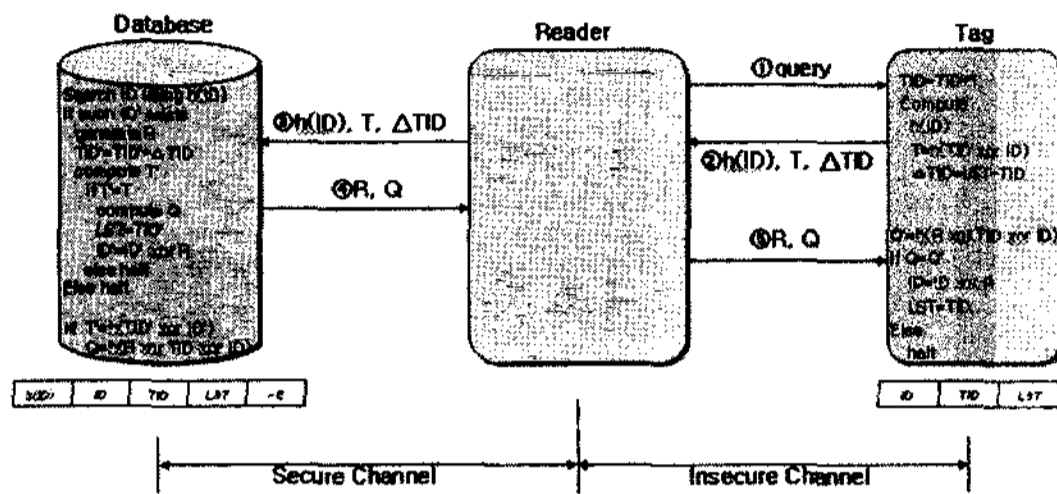


그림 3 해쉬기반 ID 변형 기법

2.3.1 문제점

이 기법은 ID갱신으로 위치추적을 막는다는 것이나 역시 다음과 같은 공격에 취약하다[4].

- 위치트래킹 공격 : 인증이 완료될 경우 ID가 주기적으로 바뀌어서 위치트래킹 공격에 안전해 보이지만 태그가 항상 동일한 $h(ID)$ 를 응답하므로 태그의 위치가 트래킹 될 수 있다. 그러나 재전송 공격에는 안전하다.
- 스푸핑 공격 : 공격자는 질의를 통해 그림 3의 ②단계를 얻어낼 수 있으며 태그가 정상적인 세션에서 ②단계를 전송하게 되면 데이터베이스는 공격자를 정당한 태그로 인증할 수밖에 없게 된다. 또한 공격자가 정당한 리더로 가장하여 $h(ID)$, $h(TID \text{ xor } ID)$, ΔTID 를 획득하고 ⑤단계에서 전송되는 정보를 전송하지 않는다면, 태그는 ID 갱신이 이루어지지 않는다. 이 경우 공격자가 여러 개의 리더를 곳곳에 설치해두었다면, 태그가 정당한 리더와 인증 세션을 수행하여 $h(ID)$ 가 갱신되기 전까지 $h(ID)$ 를 통해 태그의 위치를 추적할 수 있다. 그리고 해쉬기반 ID 변형 기법은 ID가 인증 세션마다 바뀌므로 변형되는 ID를 저장하고 있는 유일한 데이터베이스가 존재해야만 한다. 그러나 단일 데이터베이스인 경우 많은 태그를 인증하기 위해서 데이터베이스가 많은 연산을 해야 한다.

2.4 개선된 해쉬기반 ID 변형 기법

해쉬기반 ID 변형 기법의 문제점인 스푸핑에 대한 취약점을 보완하고, 태그의 해쉬 횟수를 줄였다[4,7].

2.4.1 문제점

향상된 해쉬기반 ID 변형 기법에서는 데이터베이스에서 해쉬함수와 리더에서 난수생성기 구현을 요구한다. 태그와 데이터베이스에서는 Exclusive-OR 연산과 연결 연산을 구현해야 한다. 또한 이 기법에서는 기존의 기법과 다르게 태그와 데이터베이스에서 문자열 나눴다

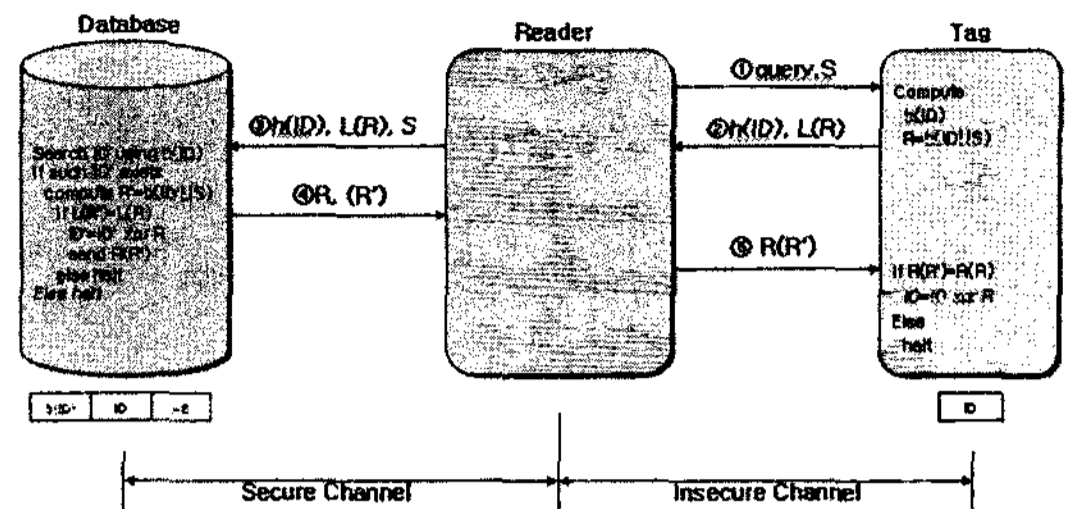


그림 4 개선된 해쉬기반 ID 변형기법

이 필요하다. 그리고 다음과 같은 공격에 취약하다.

- 위치추적 공격 : 앞의 해쉬기반 ID변형 기법과 마찬가지로 그림 4의 ②단계에서 태그는 매 세션마다 동일한 $h(ID)$ 를 전송하기 때문에 태그의 위치가 노출된다.
- 스푸핑 공격 : 태그가 데이터를 가지는 시스템에 적용될 경우, 공격자는 리더인 척 가장하여 태그를 속이고 태그 내의 데이터를 얻을 수 있다. 공격자는 리더와 태그사이의 데이터를 탈취하여 그림 4의 ①, ②, ⑤단계를 얻고 ⑤단계를 태그에게 주지 않고 가로채면 태그의 ID는 갱신되지 않는다. (예를 들어 동물용 RFID의 경우 125KHz~134KH 저주파대역의 RF 신호를 사용하기 때문에 리더와 태그간의 인식거리가 60cm 미만으로 짧다. 리더와 태그사이에서 매우 빠른 시간 내 통신이 이루어지기 때문에 세션 가로채기가 거의 어렵지만 가능하다고 가정한다.) 태그는 아직 갱신되지 않았고 ①단계와 동일한 S를 요청메시지와 함께 전송하고 ②단계 응답에 대해 도청한 ⑤단계의 정보를 태그에게 주면 태그는 리더를 인증하고 데이터를 주게 된다.

3. 제안 프로토콜

3.1 일회성 난수를 이용한 안전한 RFID 상호인증 프로토콜

본 논문에서는 기존 리더에서 태그에게만 보내던 Query를 태그의 ID를 유지하는 DB에게도 동시에 보냄으로써 DB도 난수 값을 생성하게 된다. DB와 리더 사이의 채널은 Secure channel이라 가정한다. Insecure Channel인 리더와 태그사의 모든 정보를 탈취하여 재전송 공격에 이용하더라도 DB가 가지고 있는 난수 값은 매 인증 시도마다 갱신되기 때문에 과거의 정보는 쓸모없는 데이터가 된다.

제안하는 RFID 인증 프로토콜은 쿼리가 태그와 DB로 동시에 전송되고, DB는 쿼리를 받음과 동시에 난수를 생성한다. 매 인증시마다 난수 값이 갱신되기 때문에 매번 다른 데이터 값이 생성된다는 장점이 있다.

태그는 오로지 ID값만 저장한다. 리더는 리더의 감응

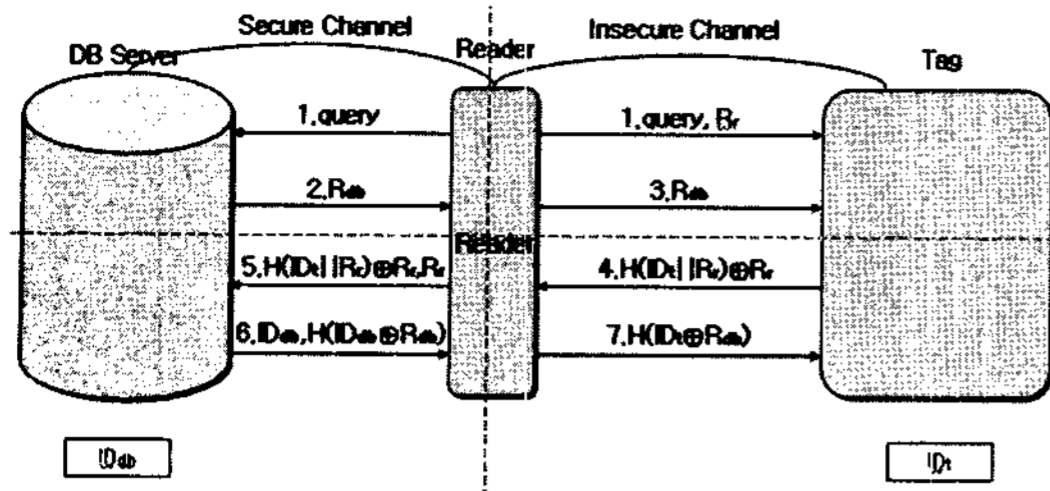


그림 5 일회성 난수를 이용한 안전한 RFID 상호인증 Protocol

인지거리 내에 태그가 존재하게 되면 태그와 DB에 쿼리를 동시에 전송한다. 쿼리를 받은 태그는 쿼리와 함께 전송된 리더 난수를 태그 안에 임의의 저장소에 저장한다. 리더로부터 쿼리를 받은 DB는 자신의 난수생성기(Random Number Generator)를 이용하여 난수를 생성하고 그 난수를 리더에게 전송한다. DB로부터 난수를 받은 리더는 바로 태그에 DB난수를 전송한다. 이전 과정을 통해 태그는 DB로부터의 난수, 리더로부터의 난수 이렇게 두 개의 난수를 갖게 된다. 태그는 순식간에 $H(ID_t||R_r) \oplus R_r$ 을 계산해 내어 다시 리더에게로 전송한다. 리더는 태그로부터 받은 데이터 중 $H(ID_t||R_r) \oplus R_r$ 와 R_r 을 DB로 전송한다. DB에서는 태그로부터 온 데이터 중 ID 매칭 여부를 판단하게 된다. $H(ID_{db}||R_r) \oplus R_r$ 과 $H(ID_t||R_r) \oplus R_r$ 을 비교하여 서로 같다면 태그로부터 온 데이터가 정확한 것이라 판단을 하고 인증을 통과시킨다. 반대로 같지 않다면 신뢰할 수 없는 데이터 값일 수 있고 이는 인증을 통과시키지 않고 또한 다음 과정을 진행 할 수 없게 된다. DB로부터 인증이 통과된다면, DB는 자신이 갖고 있는 ID 값과 $H(ID_{db} \oplus R_{db})$ 을 리더에게 보낸다. $H(ID_{db} \oplus R_{db})$ 값의 용도는 인증한 ID값과 DB 자신이 처음에 만들어낸 난수 값을 XOR 후 해쉬하여 태그가 리더를 인증할 때 사용된다. 리더는 받은 $H(ID_{db} \oplus R_{db})$ 값을 태그에게 보내고 태그는 이 값이 정당한 리더로부터 온 것인지 자신이 가지고 있던 R_{db} 값과 ID이용하여 비교 후 리더 인증을 끝내고 인증이 통과 되었다면 다음 과정을 진행하게 된다.

3.2 제안 프로토콜의 세부 실행과정

인증은 그림 6과 같이 총 7단계로 이루어진다.

- ① 감응 인식 범위 내에 태그가 존재하게 되면 리더는 난수 값 R_r 을 생성하고 동시에 DB서버 쪽에도 쿼리를 전송한다.

Reader(query, R_r) → Tag, Reader(query) → DB 서버

쿼리를 받은 Tag는 자신의 임시저장장소에 R_r 을 저장하게 된다. 또한 DB서버에서는 쿼리를 받음과 동시에 난수 값 R_{db} 를 생성한다.

- ② 이미 생성된 R_{db} 를 리더에게 전송한다.
DB서버(R_{db}) → Reader
- ③ DB서버로부터 받은 R_{db} 를 태그에게 전송한다.
Reader(R_{db}) → Tag
Rdb를 받은 태그는 임시저장소에 Rdb값을 저장하게 된다. 태그는 R_r (Reader에서 생성된 난수 값)과 R_{db} (DB서버에서 생성된 난수 값) 이 두 값을 갖게 된다.
- ④ 태그에서는 자신이 갖고 있는 ID_t 값과 기 저장된 R_r 값을 연결하여(concatenate) 해쉬 한다. 이 값을 이용하여 $H(ID_t||R_r) \oplus R_r$ 을 생성하여 리더에게 전송한다.
Tag($H(ID_t||R_r) \oplus R_r$) → Reader
- ⑤ 리더는 태그로부터 받은 $H(ID_t||R_r) \oplus R_r$ 과 R_r 을 DB서버로 전송한다.
Reader($H(ID_t||R_r) \oplus R_r, R_r$) → DB서버
- ⑥ DB서버에서는 자신의 ID_{db} 값으로 $H(ID_{db}||R_r) \oplus R_r$ 을 생성하고 태그로부터 받은 $H(ID_t||R_r) \oplus R_r$ 과 일치 여부를 비교하여 일치한다면 ID_{db} 와 $H(ID_{db} \oplus R_{db})$ 를 리더에게 전송한다. 그렇지 않다면 시스템은 중지한다.
IF $H(ID_{db}||R_r) \oplus R_r = H(ID_t||R_r) \oplus R_r$
Send $ID_{db}, H(ID_{db} \oplus R_{db})$ to Reader
Else
Halt
- ⑦ DB서버로부터 받은 $H(ID_{db} \oplus R_{db})$ 를 태그에게 전송한다. 태그는 자신의 ID_t 값으로 $H(ID_t \oplus R_{db})$ 을 생성하고 리더로부터 받은 $H(ID_{db} \oplus R_{db})$ 와 비교하여 값이 일치하면 다음 과정을 수행하고 그렇지 않다면 시스템은 중지 한다.
IF $H(ID_t \oplus R_{db}) = H(ID_{db} \oplus R_{db})$
Do next process
Else
Halt

3.3 제안한 프로토콜의 특징과 보안성

- DB에서도 일회성 난수를 생성하여 인증에 이용한다. 리더에서 태그로 최초 쿼리시 동시에 DB로도 쿼리를 전송하게 된다. 이 쿼리를 받은 DB는 자신이 갖고 있는 난수 생성기로 난수 값을 생성하고, 이 난수 값은 향후 태그 인증시 사용된다. 앞선 많은 연구에서는 일반적으로 DB에서 난수 값을 생성하지 않는데, DB서버는 태그의 하드웨어적 구조보다 제약사항이 심하지 않기 때문에 DB서버에서 생성한 난수를 이용함으로써 보안성을 증가시켰다.
- 위치추적이 어렵다.
순차적으로 리더로부터 리더난수를 받고, DB서버에서 DB난수를 받은 태그는 ④단계에서 $ID_t||R_r$ 를 해쉬하기 때문에 매번 다른 data 값이 발생한다. 따라서 매

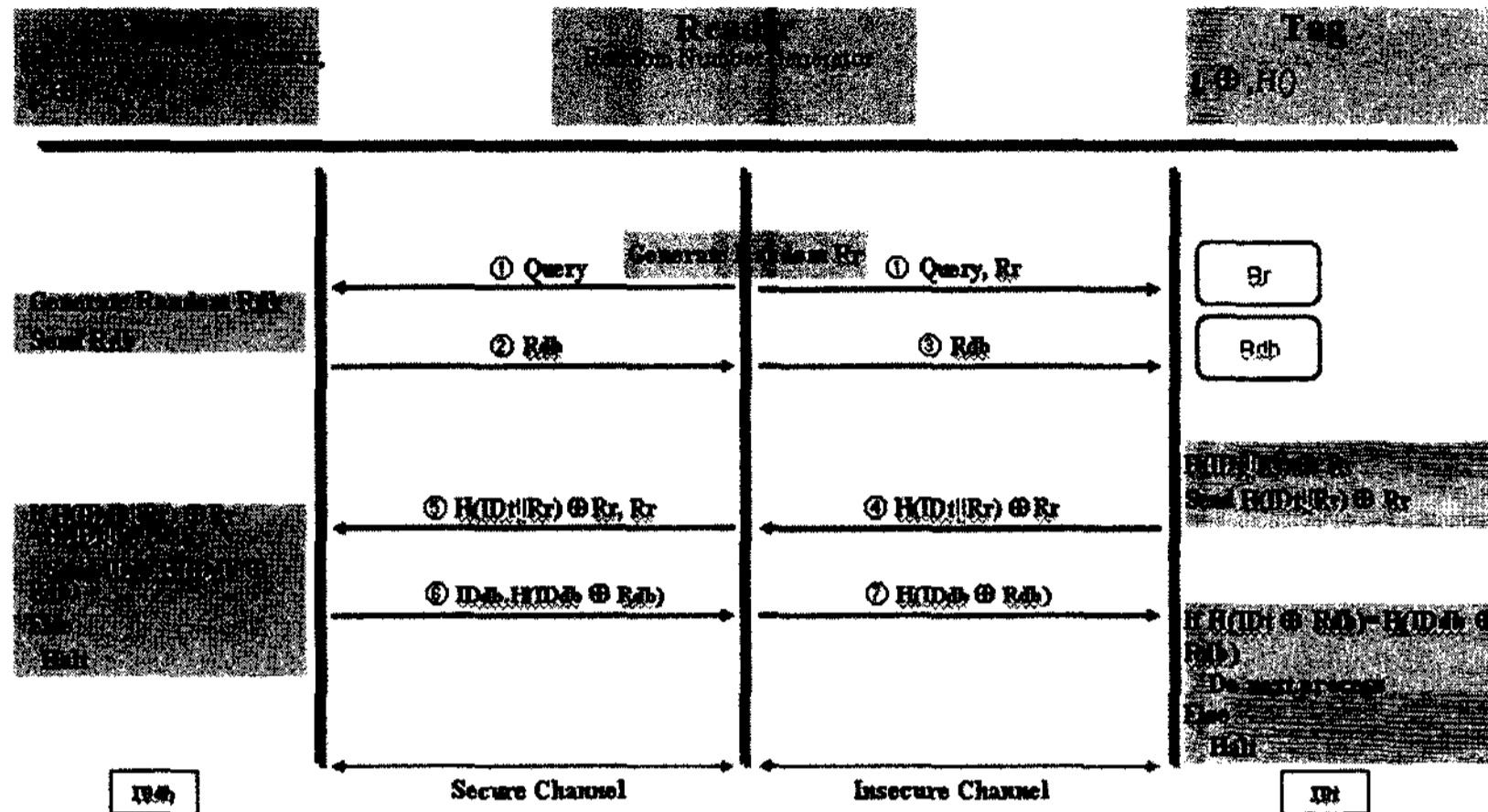


그림 6 제안 프로토콜의 세부 동작과정

번 다른 값을 정보로 현재 태그가 어느 지점의 리더기에서 읽혀지고 있는지를 추적할 수가 없다.

- 재전송 공격에 안전하다.

위치추적이 어려운 것은 리더와 태그 사이에 특정 data 값이 오가는 것이 아니라, 매번 다른 data 값이 오가기 때문이다. 제안한 프로토콜은 악의적인 공격자가 리더와 태그 사이의 공간에서 ①,③,④단계의 모든 값을 탈취하여 이 값을 다음 번 인증에 사용할 수는 없는 구조이다. 다음 번 인증에 사용함과 동시에 리더나 DB는 자신이 생성한 다른 난수 값을 가지고 인증하려 하기 때문에 이전에 사용된 ④단계의 값은 유효하지 않은 값이 되어버린다.

- 스푸핑 공격에 안전하다.

악의적인 공격자가 ①,③,④단계의 모든 값을 탈취하여도 매 단계에서 난수 값 혹은 난수 값을 포함한 데이터가 전송되기 때문에 정당한 리더로 가장한 태그가 이 데이터를 한 세션(한 인증)이 끝난 후 다음 번 인증에서 사용할 수 없다. 이유는 재전송 공격에 안전한 이유와 비슷하다. ④단계의 난수 값을 포함한 데이터 값은 이미 난수 값과 ID 값이 연접연산이 되거나 XOR 연산 후 해쉬 되었기 때문에 값에 대한 ID 값 유추가 불가능하다.

- 상호인증

그림 6의 ④~⑦단계 과정이 태그와 리더가 상호인증하는 과정이다. 태그는 자신이 갖고 있던 ID와 리더로부터 받은 난수 값을 이용하여 $H(ID||Rr) \oplus Rr$ 값을 리더를 통해 DB까지 전송한다. 이 값을 이용하여 DB는 태그가 정당한 태그인지 인증을 하게 된다. 또한 태그가 인증을 통과했다면 DB는 자신이 갖고 있던 ID와 자신이 생성했던 난수 값을 이용하여 $H(IDdb \oplus$

$Rdb)$ 를 리더에게 전송하게 된다. 태그에서는 이 값이 자신이 갖고 있던 값과 비교하여 같다면 정당한 리더로 받아들이고, 그렇지 않다면 프로세스를 중지하는 과정으로 상호인증을 실시하게 된다.

4. 프로토콜의 보안성 및 성능

RFID시스템에서 태그는 하드웨어적 설계에 제약사항이 많다. 반면에 리더와 DB서버는 하드웨어적 제약사항은 많지 않다. 리더와 DB서버에서의 Hash Function, 난수생성기, $\oplus, ||$ 등과 같은 기능 구현은 그리 어려운 일이 아니다. 제안한 프로토콜에서는 하드웨어적 제약을 두지 않는 리더와 DB 서버에서 보안기능 구현상 용이함을 최대한 활용하였다.

표 1은 이미 연구된 프로토콜들의 기능비교를 통하여 각 프로토콜의 기능 구현과 안전성의 상관관계를 다음과 같이 분석하였다. 특히 리더기와 DB서버에서 기능구현상의 하드웨어적 제약은 없기 때문에 태그에 설계되어야 하는 기능에 초점을 맞추어 분석을 하였다.

4.1 해쉬락 기법

태그에서 해쉬함수만을 구현한다. 인증과정은 총 6단계로 이루어지나, 하드웨어적 암호화나 보안 기능이 거의 없기 때문에 이 기법은 위치추적, 재전송 공격, 스푸핑 공격에 약하다. 구현면에서는 오로지 해쉬함수만 작성만 하면 된다는 장점이 있지만 보안성은 매우 약하다.

4.2 확장된 해쉬-락 기법

태그에서 해쉬함수, 난수생성기, 연접연산을 구현해야 한다. 인증과정은 총 5단계로 이루어지나 여러기능이 구현되어 있음에도 불구하고 재전송 공격, 스푸핑 공격에 약하다. 난수생성기로 매번 다른 값이 생성 되기 때문에 위치추적에는 강하다.

표 1 각 프로토콜 기능비교

기능	구분	개수				
		MIT 해쉬-락 기법	확장된 해쉬-락 기법	해쉬기반 ID변형기법	개선된 해쉬기반 ID변형기법	제안한 프로토콜
해쉬함수	Tag	1	1	1	1	1
	Reader	-	1	-	-	-
	Database	-	-	1	1	1
난수생성기	Tag	-	1	-	-	-
	Reader	-	-	-	1	1
	Database	-	-	1	-	1
⊕	Tag	-	-	1	1	1
	Reader	-	-	-	-	-
	Database	-	-	1	1	1
	Tag	-	1	-	1	1
	Reader	-	-	-	-	-
	Database	-	1	-	1	1
추가기능	Tag	-	-	-	1(문자열 나눔기능)	-
	Reader	-	-	-	-	-
	Database	-	-	-	1(문자열 나눔기능)	-
인증과정의 Step 수		6	5	5	5	7

4.3 해쉬기반 ID 변형 기법

태그에서 해쉬함수, XOR를 구현해야 한다. 인증과정이 총 5단계로 이루어지나 위치추적, 스푸핑 공격에 약하다.

4.4 개선된 해쉬기반 ID 변형 기법

태그에서 해쉬함수, XOR, 연접연산을 구현해야 한다. 또한 추가적으로 문자열 나눔 연산이 구현이 되어야 한다. 데이터 전송시 정보를 반으로 쪼개어 전송하는 방법은 복잡성을 증가시켜 보안성을 강화했다. 인증과정이 총 5단계로 이루어지나 스푸핑 공격, 위치추적에 약하다.

4.5 제안한 프로토콜

개선된 해쉬기반 ID 변형 기법과 마찬가지로 태그에서 해쉬함수, XOR, 연접연산을 구현해야 한다. 인증과정이 총 7단계로 이루어진다. 개선된 해쉬기반 ID 변형 기법에 비하여 인증 단계가 2단계 증가하였지만 보안성은 증가하였다.

4.5.1 프로토콜의 효율성

제안한 프로토콜은 표 1과 같이 개선된 해쉬기반 ID 변형기법에 비하여 인증단계가 증가하였지만 리더나 데이터베이스는 태그에 비해 상대적으로 연산능력이 충분

하기 때문에 연산으로 인한 오버헤드는 크지 않다.

제안 프로토콜의 효율성을 기존 RFID 인증 기법들과 비교하여 정의하였다. 객관적으로 정의하기 위해 태그 인증시 사용되는 정보를 다음과 같이 가정한다.

metaID → 128bit(metaID는 해쉬함수 h()를 거쳐 만들어진다), ID → 64bit(일반적으로 64bit의 ID를 사용한다), key → 64bit, Random number → 16bit, hash() → 128bit, 나눔연산L(),R() → 피연산자/2 bit, S → 16bit(Random number)

데이터베이스나 리더는 태그에 비해 고가이기 때문에 간단한 기능(hash function, random number generator, ⊕,||)의 구현은 큰 비용을 들이지 않고 구현이 가능하다. 하지만 태그는 하드웨어 설계적면에서 제약사항이 많기 때문에 각 프로토콜에서 태그는 중요한 위치를 차지한다. 제안 프로토콜에서 태그는 해쉬함수 구현만 필요로 한다. 표 3에서와 같이 개선된 해쉬기반 ID 변형기법과 차이 없이 태그에서 연산횟수가 거의 동일하다. 태그는 인증 과정동안 2번의 해쉬만 수행하면 된다. 단 제안 프로토콜은 개선된 해쉬기반 ID 변형기법보다 XOR연산이 한번 더 일어난다. 리더-태그 사이의 송/수

표 2 각 프로토콜의 보안성 비교표

공격유형 보안프로토콜	스푸핑 공격	재전송 공격	위치추적공격
해쉬-락 기법	약함	약함	약함
확장된 해쉬-락 기법	약함	약함	강함
해쉬기반 ID 변형 기법	약함	강함	약함
개선된 해쉬기반 ID 변형기법	약함	강함	강함
제안 기법	강함	강함	강함

표 3 프로토콜 연산 횟수를 통한 효율성 비교표

기능	해쉬-락 기법	확장된 해쉬-락 기법	해쉬기반 ID변형기법	개선된 해쉬기반 ID변형기법	제안 프로토콜
태그 난수발생횟수	-	1회	-	-	-
리더 난수발생횟수	-	-	-	1회	1회
태그 해쉬연산횟수	2회	1회	3회	2회	2회
리더 해쉬연산횟수	-	DB에 n개의 ID가 존재한다면 최대 n번	-	-	-
태그에서 ⊕	-	-	2~3회	0~1회	2회
태그에서	-	1회	-	1회	1회
리더-태그간 data 총 송/수신량	metaID, ID, key 128+64+64=256bit	R, h(ID _k R) 16+128=144bit	R, Q=h(.....) 16+128=144bit	S, h(ID), 문자열 나눴음L(R), 나눴음L 16+128+64+64=272bit	R, R ₀ , h(ID R)⊕R _{0}, h(ID_k R_{0}) 16+16+128+128=288bit}}

신된 데이터 량은 제안 프로토콜이 가장 크게 평가되었지만 개선된 해쉬기반 ID 변형기법 또는 해쉬-락 기법의 데이터 송/수신량과 작은 차이를 보이긴 하나, 보안성을 증가시켜 보상하였다.

4.5.2 프로토콜의 장점

프로토콜 특성상 난수 값이 일회성을 갖기 때문에 매 인증 시도마다 갱신되어 과거의 정보를 이용하여 재전송 공격에 이용할 수 없다. 하드웨어적 제약사항이 많은 태그에서는 해쉬함수 구현만이 필요로 하기 때문에 태그 생산 비용이 비싸지 않다. 기 연구된 다른 프로토콜에 비해 태그와 데이터베이스에서 필요로 하는 메모리 공간이 많이 필요로 하지 않는다. 예를 들어 해쉬기반 ID 변형 기법은 데이터베이스에서 h(ID)', ID, TID, LST, AE 태그에서 ID, TID, LST를 저장할 수 있는 공간을 필요로 하지만, 제안 프로토콜에서는 데이터베이스와 태그에서 오로지 ID를 저장할 수 있는 공간만을 필요로 한다. 또한 가장 중요한 부분인 기존에 연구에 남아있던 보안 취약성을 보완하였다.

5. 결론

RFID 시스템은 마이크로 칩을 내장한 태그, 라벨, 카드 등에 저장된 데이터를 무선 주파수를 이용하여 인증하는 편리한 기술이다. 그러나, 태그의 사용으로 인해 개인의 프라이버시 침해의 위험요소가 존재한다. 리더기의 영향으로 EPC(Electric Product Code) 인 ID 및 고유 식별번호를 일정 반경 내에 모든 리더나 태그들에게 자신의 정보를 보내게 된다. 의도하지 않은 정보의 누출로 악의적인 공격을 조장할 수도 있고, 범죄에 악용 될

수도 있다. 이러한 문제점들을 해결하고자 개선시킨 RFID 상호인증 프로토콜을 제안하였다.

제안한 프로토콜이 기존의 연구된 기법들과 인증을 위한 총 단계 횟수에 차이를 보이기는 하나 기존 연구에 남아있던 스푸핑 공격의 위험성이 확실하게 차단된다는 장점이 있다. DB서버 난수를 이용한 RFID 상호인증 프로토콜은 기존 연구된 RFID 인증 기법들의 가장 큰 문제점인 재전송 공격, 스푸핑 공격, 위치 트래킹 공격에 안전하게 설계되었음을 확인하였다.

참고 문헌

[1] S. A. Weis, S. e. Sarma, R. L. Rivest, and D. W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," Security in Pervasive Computing 2003, LNCS 2802, pp. 201-212, Springer-Verlag Heidelberg, 2004.

[2] S. A. Weis, "Security an Privacy in Radio-Frequency Identification Devices," MS Thesis. MIT. May, 2003.

[3] Sanjay Sarma, Stephen Weis, and Daniel Engels. "Radio systems, security and privacy implications," Technical Report MIT-AUTOID-WH-014, AutoID Center, 2002. Available from <http://www.autoidcenter.org>.

[4] 이근우, 오동규, 광진, 오수현, 김승주, 원동호, "분산 데이터베이스 환경에 적합한 Challenge-Response 기반의 안전한 RFID 인증 프로토콜", 한국정보처리학회 논문지C, 제12-C권, 제03호, pp. 309-316, 2005.

[5] 이영진, 정윤수, 서동일, 이상호, "부분ID를 이용한 읽기전용 RFID태그 인증프로토콜", 한국정보처리학회 논문지 C, 제13-C권, 제05호, pp. 595-600, 2006.10.

[6] Ari Juels, Ronald Rivest, and Michael Szydlo.

"The blocker tag : Selective blocking of RFID tags for consumer privacy 2002.

- [7] Dirk Henrici. and Paul Muller. "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifications," PerSec'04, pp. 149-153, March 2004.



김 대 중

2000년 8월 한국방송통신대학교 컴퓨터 과학과 졸업(이학사). 2004년 2월 숭실대학교 산업기술정보대학원 정보통신공학과 졸업(공학석사). 2006년 2월 숭실대학교 일반대학원 컴퓨터학과 박사과정. 관심분야는 보안, 정보통신



전 문 석

1981년 숭실대학교 전자계산학과(공학사). 1986년 University of Maryland Computer Science(공학석사). 1989년 University of Maryland Computer Science(공학박사). 1989년 3월~7월 Morgan State University 조교수. 1989년~1991년 New Mexico State University Physical Science Lab 책임 연구원. 1991년~현재 숭실대학교 부교수. 관심분야는 전자상거래 보안, 인터넷 보안, 멀티미디어 보안, 인증 시스템, PKI, RFID