情報保護學會論文誌
第15卷 第4號, 2005. 8

위치 추적과 서비스 거부 공격에 강한 RFID 인증 프로토콜

강 전 일,† 양 대 헌‡
인하대학교 정보통신대학원

RFID Authentication Protocol with Strong Resistance against Traceability and Denial of Service attack

Jeonil Kang,† DaeHun Nyang‡
INHA University Graduate School of IT&T

요 약

RFID 시스템에는 많은 인증 프로토콜이 존재함에도 불구하고, 오로지 몇몇 프로토콜만이 위치 추적에 대한 안전성을 보장해준다. 태그의 하드웨어적 제한 사항으로 인하여 이러한 프로토콜은 서비스 거부 공격을 포함하는 많은 보안 위협으로부터 안전하지 못하다. 이 논문에서는 위치 추적에 대한 문제를 설명하고 RFID 인증 프로토콜들의 취약점을 보인다. 그리고 위치 추적과 위장 공격, 서비스 거부 공격 등에 대하여 강한 인증 프로토콜을 제시한다.

ABSTRACT

Though there are many authentication protocols for RFID system, only a few protocols support location privacy. Because of tag's hardware limitation, these protocols suffer from many security threats, especially from DoS (Denial of Service) attack. In this paper, we explain location privacy problem and show vulnerabilities of RFID authentication protocols. And then, we suggest an authentication protocol that is strong against location tracing, spoofing attack and DoS attack.

Keywords : *RFID system, Location Privacy,Authentication protocol, Denial of Service attack*

Ⅰ. 서 론

무선 인식, 즉 RFID(Radio Frequency Identification) 시스템은 바코드(Bar-Code) 시스템과 비교하여 빠른 인식 속도, 먼 인식 거리, 장애물과 오염으로부터 비교적 자유롭다는 등의, 불과 몇 초 안 수백, 수천 개의 RFID 태그를 동시에 읽어낼 수 있다는 의미하는 이점 때문에 유통과 물류를 중심으로 사회 전반에 걸쳐 그 사용 폭을 넓혀가고 있다. 또한 RFID 시스템은 바코드 시스템을 사용하기 힘든 동물 태깅이나 고속도로 요금 부과, 도난 방지 등에 사용할 수 있다는 장점이 있어 바코드 시스템을 빠르게 대체해가고 있는 중이다.

하지만 낮은 연산 능력과 작은 기억 용량 등 태그의 하드웨어적 제약사항 때문에 높은 연산 능력과 큰 기억 용량을 필요로 하는 기존의 보안 기술을 그대로 RFID 시스템에 적용할 수 없다. 물론, 태그의 연산 능력과 기억 용량을 기존의 보안 기술을 탑재할 수 있을 만큼 사용할 수도 있지만, 이는 태그의 가격을 상승시키는 요인이 된다. 하지만 RFID 시스템이 바코드 시스템을 대신하기 위해는 싼 가격의

접수일 : 2005년 4월 12일 ; 채택일 : 2005년 8월 9일
† 주저자, dreamx@seclab.inha.ac.kr
‡ 교신저자, nyang@inha.ac.kr

태그가 필수이기 때문에 이는 바람직하지 못하다. 따라서 RFID 시스템은 여전히 많은 보안 문제가 남아 있으며 RFID 시스템을 높은 보안을 요구하는 곳에서 사용하기란 힘들다. 이러한 이유 때문에, 싼 가격의 RFID 시스템을 위한 보안 기술이 많은 사람들에 의하여 연구되고 있다.[1-9,11]

RFID 시스템의 보안 문제는 보통 두 가지로 나눈다. 첫 번째는 정보 누출(Information Leakage)에 관한 문제이다. 수동적인 공격자는 RFID 시스템이 무선을 이용하기 때문에 리더와 태그 사이의 정보를 엿들을 수 있고, 능동적인 공격자는 데이터베이스(또는 리더)나 태그를 속이기 위한 거짓 메시지를 만들어 보냄으로서 이들의 정보를 얻어낼 수 있다. 이러한 공격을 막기 위해서 몇 가지 해결책이 제시되었는데, 블로커 태그,[6] AES를 태그에 탑재한 RFID 시스템,[9] XOR 원 타임(one-time) 암호화[4] 등이다. 특히, 블로커 태그는 정보 누출을 막는 매우 간단하고 강력한 방법 중에 하나이다.

두 번째는 위치 추적(Location Privacy)에 관한 문제인데, 이 개념은 RFID 시스템의 정보뿐만이 아니라 RFID 태그의 위치 자체가 하나의 정보가 될 수 있다는 것이다. 이 문제를 위해서도 몇 가지 제시된 해결책이 있는데, 해시 기반 인증 프로토콜,[1] 해시 체인,[2] 보편적인 재 암호화,[7] 상태 기반 인증 프로토콜[11] 등이 그것이다. 하지만, G. Avoine에 따르면 응용, 통신, 물리로 나누는 각각의 RFID 프로토콜 계층이 갖는 기술적이고 현실적인 결함으로 인하여 위치 추적 문제를 완전히 해결하는 것은 쉬운 것이 아니라고 한다.[10] 따라서 이에 관한 연구가 보다 더 필요하며 절실하다.

이 논문에서는, 2장에서 RFID 시스템의 위치 추적 문제와 RFID 시스템의 위협 모델에 대해서 이야기할 것이다. 3장에서는 서비스 거부(Denial of Serve) 공격을 포함하여 기존의 RFID 인증 프로토콜을 어떻게 공격할 수 있는지 살펴볼 것이고, 4장에서 이러한 공격과 보안 문제를 해결하기 위한 인증프로토콜을 제시할 것이다. 그리고 5장에서 논문을 정리하며 마무리한다.

Ⅱ. 위치 추적(Location Privacy)

2.1 응용 계층에서의 위치 추적 가능성

일반적으로 RFID 통신 프로토콜은 응용(Application), 통신(Communication), 물리(Physical) 계층으로 구성된다. 응용 계층에서의 정보는 사용자에 의해서 다루어진다. 다른 RFID 응용 프로그램들은 일반적으로 상품번호와 일련번호로 이루어진 정보를 이용하여 사물을 구별하고 인식한다. 또한 이 정보에는 관리 목적을 위한 비밀키나 패스워드를 포함할 수도 있다. 통신 계층에서는 어떻게 하면 리더와 태그 사이에서 발생하는 비트의 충돌을 회피할 수 있는가에 대해서 정의한다. 이러한 프로토콜을 "충돌 방지 프로토콜(Collision Avoidance Protocol 또는 Anti-collision Algorithm)"이라고 한다. 충돌 방지 프로토콜은 보통 이진트리탐색기법(Binary-Tree Search)과 같은 결정적 방법과 알로하(ALOHA)와 같은 확률적 방법으로 나눌 수 있다. 물리 계층에서는 주파수, 변조, 데이터 인코딩, 동기화 등 에어 인터페이스(Air Interface)를 사용하여 물리적으로 어떻게 리더와 태그 사이에 주고받을 수 있느냐 하는 것을 정의한다.

위치 추적 문제는 위의 모든 계층에서 발생할 수 있다. 만약 공격자가 리더와 태그 사이의 모든 메시지를 엿들을 수 있고, 메시지에 변하지 않는 데이터가 있다면 공격자는 이 데이터를 이용하여 태그의 움직임을 추적할 수 있다. 또한, 공격자는 태그가 충돌 방지 프로토콜의 종류와 부분적으로 통신 계층에서 사용하는 구분자를 이용하여 태그의 위치를 알아낼 수도 있다. 물리 계층 또한 이와 다르지 않다. 따라서 공격자로부터 태그의 위치가 추적당하는 것을 막기란 매우 힘든 일이다.

여기서, 우리는 추적의 정확도에 대해서 생각해 볼 필요가 있다. 보통의 경우, 통신과 물리 계층의 문제를 가지고 태그의 위치를 추적하는 것은 비교적 정확도가 떨어지게 되는데, 왜냐하면 동일한 통신과 물리 계층의 특징을 가진 태그는 비교적 많기 때문이다. 태그를 제조하는 회사가 어떠한 충돌 방지 프로토콜과 에어 인터페이스를 사용하느냐에 따라서 이러한 특징이 발생하게 된다는 사실은, 사용자들이 많이 쓰는 태그일수록 통신과 물리 계층의 특징을 이용하여 태그의 위치를 추적하는 것은 어렵게 된다는 것을 의미한다. 하지만 응용 계층의 메시지를 이용하는 방식을 이용하는 위치 추적은 정확도가 매우 높다. 데이터베이스나 리더는 태그가 부착된 객체를 명확하게 구별해야 하기 때문에, 이 계층에서 사용하는 구분자는 통일된 코드 체계를 사용한다고 가정하면 전 세계에서 오로지 하나 뿐이다. 따라서 응용

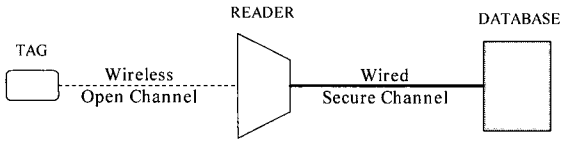

그림 1. RFID 시스템의 위협 모델

계층에서 위치 추적을 막는 것은 중요한 일이며, 이 논문에서도 응용 계층에서의 위치 추적 문제와 해결 방안에 대해서 이야기하고자 한다.

2.2 RFID 시스템의 위협 모델

제안하는 인증프로토콜을 설명하기에 앞서, 기존의 인증 모델이 갖는 취약점을 분석하기 위하여 위협 모델(Threat Model)을 정의하고자 한다. 이러한 위협 모델을 정의함으로써, RFID 시스템에서 공격자의 능력을 합당한 수준으로 제한할 수 있다.

다른 위협 모델과 마찬가지로,[13] 이 논문에서는 공격자가 메시지를 가로채거나 수정할 수 없다고 가정한다. 리더와 태그 사이의 무선 채널(Wireless Open Channel)에서 공격자는 모든 메시지를 엿듣거나 거짓 메시지들을 끼워 넣을 수 있지만, 그는 메시지를 가로채거나 수정할 수 없는데 데이터 전송을 위한 매체가 공기이기 때문이다. 리더와 데이터베이스 사이의 유선 채널(Wired Secure Channel)에서 공격자는 메시지를 엿듣거나 가로채거나 삽입하거나 수정할 수 없다. 또한 리더와 데이터베이스 서버 사이에서는 서로에 대해서 사전 인증이 이루어졌다고 가정한다.

이 위협 모델에서, 공격자는 태그의 위치를 추적하거나 시스템을 붕괴시키기 위해 노력할 수 있다. 여기서 시스템의 붕괴란 서비스 거부공격이나 위장공격, 특정 또는 불특정 태그나 데이터베이스를 상대로 하는 비동기 공격이 성공했을 때를 의미한다.

Ⅲ. 인증 프로토콜의 취약점들

이 장에서는, 기존의 프로토콜들이 가지고 있는 구조적 문제점을 이용하여 RFID 시스템을 공격하는 몇 가지 방법에 대해서 기술할 것이다.

3.1 정상 종료되지 못한 세션을 이용한 공격

많은 경우 공격자가 인증 세션을 올바르게 종료할

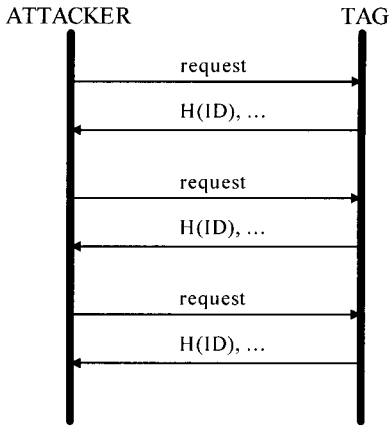

그림 2. 정상 종료되지 못한 세션을 이용한 위치 추적

것이라고 확실할 수 없다. 공격자는 세션에 필요한 모든 메시지를 전송할 의무가 없으며, 또한 이를 역으로 이용할 수도 있다. 만약 어떤 인증 프로토콜이 식별자(identifier)를 단지 해시해서 보내거나 리더로부터 받은 임시 변수(nonce)와 함께 해시하여 리더에게 보낸다고 하면 이러한 인증 프로토콜을 사용하는 태그는 보다 추적에 취약하다. 왜냐하면 태그는 공격자의 요청에 매 세션마다 동일한 해시된 식별자를 돌려줄 것이기 때문이다. 불행하게도 태그는 데이터베이스와의 동기화를 유지해야만 하므로 이러한 공격에 대해서도 자신의 식별자를 다른 것으로 바꿀 수가 없는데, 데이터베이스와 태그 사이의 비동기화는 그 태그를 더 이상 사용할 수 없다는 것을 의미하기 때문이다.

3.2 Preemptive Locking

태그가 이미 인증 세션에 있을 때 새로운 요청 메시지가 도착하였을 때를 가정해보라. 태그는 자신의 행동을 결정해야만 하는데, 새로운 요청 메시지를 단지 무시하거나 또는 새로운 요청 메시지에 대해서 즉시 새로운 인증 세션을 시작하는 것이다. 전자의 경우 공격자로 하여금 태그를 선점하여 잠글 수 있는 공격 기회를 부여하게 된다. 공격자는 태그가 합법적인 인증 세션을 시작하기 전에 단시 요청 메시지를 보내주는 것만으로 태그를 침묵하게 만들 수 있다. 이러한 Preemptive Locking을 이유로, 태그 제조사는 이러한 공격을 막을 수 있는 어떠한 메커니즘을 제공하거나 태그의 행동으로서 후자를 선택해야만 한다.

한편, 타이머를 사용하여 Preemptive Locking 문제를 쉽게 해결할 수 있을 것처럼 보인다. 태그는 세션이 시작하면 타이머를 시작하고, 타이머가 종료되면 세션을 그 즉시 닫으면 된다. 그러나 여전히 공격자가 태그를 선점하고 있는 동안에는 여전히 다른 어떠한 요청에도 응답할 수가 없다. 때문에 공격자는 이러한 공격을 시도함으로서 전체적인 시스템의 성능을 저하시킬 수 있다.

그렇기 때문에, 새롭게 도착한 요청 메시지에 대해서 새로운 인증 세션을 시작하는 것이 더 현명해 보인다. 하지만 이 경우는 다음 절에서 기술하는 공격에 여전히 약하다.

3.3 Stealth Bombing

만약 태그가 새롭게 도착한 요청 메시지에 대해서 새로운 인증 세션을 시작하도록 구현되었다면, Stealth Bombing이라고 불리는 서비스 거부 공격에 취약하게 될 것이다. 이미 Ⅱ장 2절에서 우리는 공격자가 거짓 메시지를 생성하고 끼워 넣을 수 있음을 가정했다. Stealth Bombing은 인증 세션 동안 공격자에 의해서 거짓 메시지가 삽입하여 실행되는 서비스 거부 공격을 의미한다. 만약 매 세션마다 무작위로 결정한 임시 변수를 사용하여 응답 메시지를 바꾸는 태그가 거짓 메시지에 대해서 새로운 세션을 열었다면, 기존의 진행 중이던 인증 세션은 이러한 거짓 인증 시도에 의하여 실패하게 될 것이다.

또한 다른 공격 방법도 생각해볼 수 있는데, 거짓 요청 메시지를 끼워 넣는 대신 공격자는 합법적 세션 요청 동안 올바르지 않는 확인 메시지를 보내는 것이다. 어떤 종류의 태그들은 이 확인 메시지를 받은 뒤 인증 세션을 중단할지도 모른다.

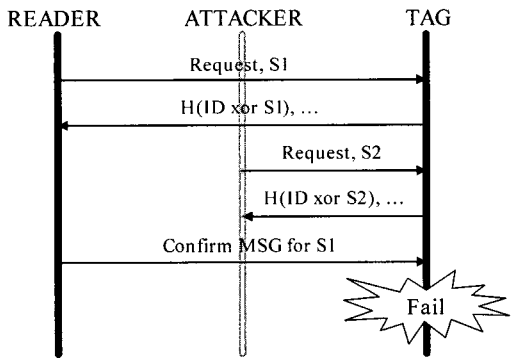

그림 3. Stealth Bombing

비록 이 Stealth Bombing 공격이 통신 계층에서 정의된 식별자가 올바르지 않다면 통신 계층은 이를 거부할 것이기 때문에 매우 힘들어 보이지만, 리더와 태그 사이의 일반적 통신에서 사용되는 식별자는 노출되어 있고 공격자는 이를 이용하여 올바른 통신 계층의 식별자를 포함하는 거짓 요청 메시지를 생성하고 태그에게 보낼 수 있기 때문에 이러한 공격은 성립할 수 있다.

3.4 데이터베이스에 대한 서비스 거부 공격

Ⅲ장 1절에서 거론되었던 문제를 막기 위하여 어떠한 인증 프로토콜[11]은 상태 기반 표식을 이용하기도 한다. 만약 앞의 세션이 성공적으로 종료되지 못했다면, 태그는 이를 어떠한 공격으로 가정하고, 보조키와 실제 식별자를 찾아내기 위한 힌트를 사용한 메시지를 생성함으로써 응답한다. 이러한 프로토콜은 올바른 상태를 복원하기 위해서 많은 연산 자원을 필요로 하게 된다. 다른 예로써, Okubo의 해시 체인 프로토콜[2]은 태그의 식별자를 찾기 위하여 많은 해시 함수를 연산해야만 하는데, 왜냐하면 데이터베이스 서버는 최초의 공유 키 s_1을 찾기 위하여 $s_i=h(h(\cdots h(s_1)\cdots))$를 계산해야만 하기 때문이다. RFID 태그는 아주 적은 연산 능력을 가지고 있기 때문에 이러한 방법들은 태그의 계산량을 데이터베이스 서버에게 이전시키는 것이므로 좋은 선택일 수 있다.

하지만, 이러한 계산의 비효율성 때문에 공격자는 이러한 프로토콜들에 대하여 아주 작은 노력만으로 서비스 거부 공격을 시도할 수 있다. 어떠한 쓰레기 메시지들을 데이터베이스 서버에게 보내기만 하는 것으로 충분하다는 것이다. 그러면, 서버는 식별자를 찾거나 모든 레코드를 검색할 때까지 식별자를 검색하기 시작할 것이다. 공격자는 거짓 메시지를 생성할 때 이런 식별자가 존재하는지에 대해서 신경쓸 필요가 없다.

결론적으로, 서비스 거부 공격에 강한 인증 프로토콜은 숨겨진 식별자를 찾아내기 위하여 많은 검색 공간을 뒤져서는 안 된다는 것이다. 이러한 지적은 Ⅲ장 1절에서 거론되었던 문제의 해결책과는 서로 호환이 안 되어 보인다. 왜냐하면 이는 데이터베이스 서버에게 태그의 계산량을 넘겨서는 안 된다고 말하는 것과 같기 때문이다.

Ⅳ. 제안 프로토콜

이 장에서는 서비스 거부 공격에 내성을 갖고 있으며, 위치 추적에 안전하고, 다른 취약점에 강한 인증 프로토콜을 제안하고자 한다.

4.1 태그와 데이터베이스 구조

이 프로토콜을 사용하는 태그는 다음과 같은 내부 메모리 공간을 갖는다.

- □ SFlag (session flag): 이 필드는 태그가 인증 세션 중인지 아닌지를 나타낸다. 세션이 시작하면, 이 필드는 true가 되며, 세션이 정상 또는 비정상으로 끝나게 되면 이는 false가 된다. SFlag는 태그가 전원이 들어온 즉시 false로 초기화 된다.
- □ ID (identifier): 이 필드는 인증 세션 동안 다른 태그들과 구별하기 위하여 사용한다. $\{0,1\}^n$
- □ CWD (confirm word): 이 필드는 데이터베이스에 의해 태그의 ID를 확인하는데 사용한다. $\{0,1\}^n$
- □ R1 (random nonce 1): 이 필드는 매 세션마다 무작위로 바뀐다. 만약 SFlag가 true라면 기존에 저장되었던 R1으로 대체되지만, false라면 태그에 의해서 새롭게 생성된다. $\{0,1\}^n$
- □ C (counter): 이 필드는 매 세션마다 일정하게 증가하거나 무작위로 바뀌게 된다. 만약 SFlag가 true라면 기존에 저장되었던 C로 대체되지만, false라면 태그에 의해서 바뀌게 된다. $\{0,1\}^m$
- □ THR_COUNT (threshold counter): 이 필드는 얼마나 많은 시도가 있었는지 나타낸다. 만약 THR_COUNT가 THR_MAX에 다다르게 되면, 태그는 현재의 세션을 종료하고 새로운 세션을 시작한다. 선택적으로 서비스 거부 공격을 보고할 수도 있다.

한편, 데이터베이스는 그림 4에 묘사된 것과 같은 구조를 갖는다. H가 암호학적 해시 함수를 의미하고, ‖가 연결(concatenate)을 의미한다고 했을 때, 데이터베이스는 모든 가능한 H(ID ‖ C)와 같은

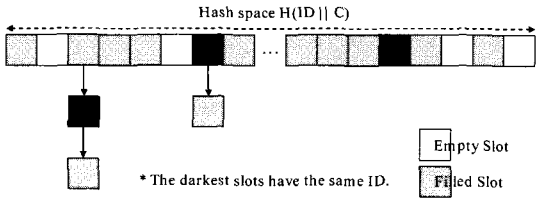

그림 4. 데이터베이스 구조의 예. 가장 검은 슬롯은 동일한 ID를 가지고 있다

수의 슬롯을 준비해야만 한다. ID가 일정할 때, 데이터베이스는 동일한 ID를 갖는 2^m개의 슬롯을 가지고 있다. 만약 다른 ID들의 H(ID ‖ C) 값이 서로 충돌이 발생하였을 때(즉, 같은 값을 가질 때), 이 슬롯들은 같은 값을 갖는 슬롯에 '링크(linked)' 되게 된다. 만약 m이 10이라면, 데이터베이스 서버는 1024번의 해시 값을 계산해야한다. 이러한 사실은 데이터베이스 서버에게 많은 계산량을 요구하는 것처럼 보이지만 해시 값의 계산은 인증이 끝난 뒤 휴지 시간(idle time)에 계산하면 되므로 이는 큰 문제가 아니다. 또한 해시 값의 계산을 위한 특수한 연산 유닛을 사용하는 것도 좋은 방법일 것이다.

4.2 기본 프로토콜

제안하는 프로토콜은 그림 5에 간략히 묘사 되어 있다. 이 프로토콜은 Ⅲ장 1절에서 4절까지 언급된 보안 문제들을 해결한다.

정상 종료되지 못한 세션을 이용한 공격을 막기 위해서는 세션 사이에서 태그가 보내는 응답 메시지가 매 세션마다 새롭다는 사실(freshness)을 보장해야만 한다. 그러기 위해서는 매 세션마다 모든 메시지들이 안전하고 무작위 임시 변수들과 함께 생성되어야만 한다. 하지만 만약 인증 세션에서 안전하지 못한 임시 변수가 쓰이게 되거나 공격자가 자신이 생성한 임시 변수를 사용할 수 있다면, 이러한 공격을 막을 수 없다. 왜냐하면 태그는 요청 메시지에 포함된 무작위 임시 변수가 어디에서 왔는지 알 수가 없기 때문에, 합법적인 리더로부터 온 무작위 수 또한 신뢰할 수 없다. 그래서 태그는 자신이 스스로 생성한 무작위 수에 대해서만 믿을 수 있는 것이다. 결론적으로, 태그는 새로운 세션이 시작할 때 무작위 임시 변수를 생성해야만 한다.

Preemptive Locking과 Stealth Bombing 공격을 동시에 막기 위해서는, 세션을 조심스럽게 다루어야할 필요가 있다. Preemptive Locking은

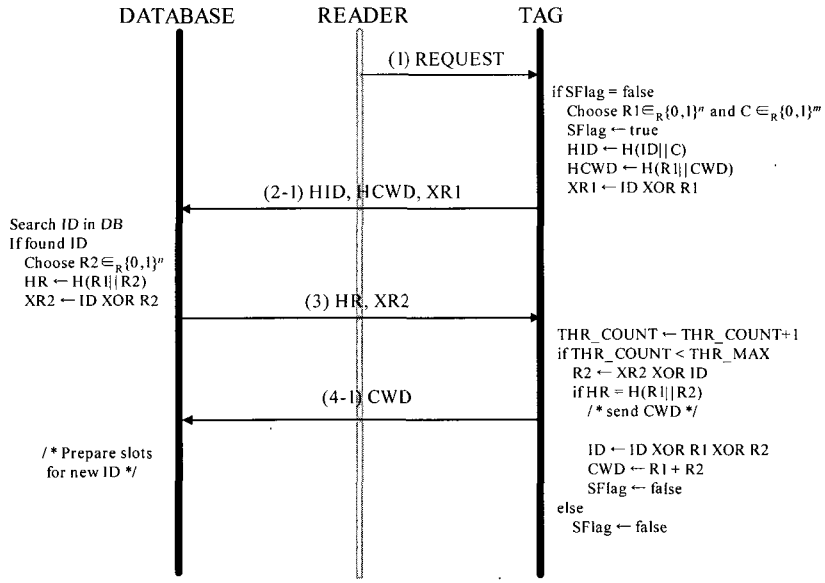

그림 5. 제안 프로토콜

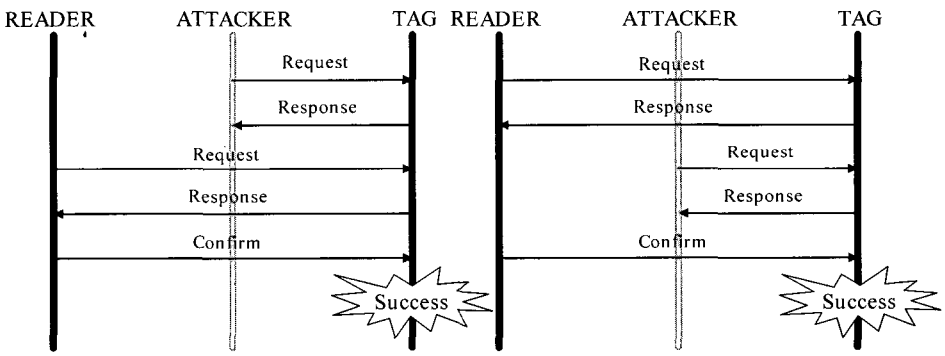

그림 6. Preemptive Locking과 Stealth Bombing을 막는 방법. 모든 응답(response) 메시지는 동일한 값을 갖는다.

인증 세션의 선점 특징 때문에 가능해지고, Stealth Bombing 공격은 인증 세션의 비 선점 특징 때문에 가능하다. 하지만 보통의 인증 프로토콜은 동시에 선점적이거나 비 선점적일 수 없다.

태그가 인증 세션 동안에 요청 메시지를 받더라도, 만약 태그가 정상적으로 종료될 때까지 동일한 응답 메시지를 보낸다면 공격자는 세션을 선점할 수 없게 된다. 또한, 동일한 응답 메시지의 반복적인 전송을 통해서 Stealth Bombing을 막을 수도 있다. 이러한 방법은 태그가 무작위 임시 변수를 생성하기 때문에 가능하다. 하나의 세션 동안 태그로부터의 응답 메시지의 동일성(identicalness)을 보장함으로써, 제안하는 프로토콜은 Preemptive Locking과 Stealth Bombing 공격에 대해서 매우 견고할 수 있다. 즉, 이 프로토콜은 공격을 어떻게 막느냐하는 방법을 제시하는 것이 아니라 어떻게 무시하느냐 하는 방법을 제시해준다.

하지만, 올바르지 않은 확인 메시지를 끼워 넣는 서비스 거부 공격을 막기 위해서는 다른 전략이 필요하다. 이 공격을 막기 위해서 태그는 약속된 시간만큼 올바른 확인 메시지를 기다려야만 한다. RFID 태그에 타이머를 직접 사용하는 것은 비싸기 때문에 타이머 대신 한계 카운터 변수(Threshold Counter)를 사용할 수 있을 것이다.

그림 5에서 묘사되었던 프로토콜은 다음과 같은 순서로 동작한다.

단계 1. 태그의 데이터를 리더가 필요로 할 때, 리더는 (1) {REQUEST}메시지를 보낸다.

단계 2. 태그가 REQUEST메시지를 받았을 때, 태그는 SFlag를 확인한다. 만약 SFlag가 true라면, 태그는 메모리에 있는 R1과 C를 사용하고, 그렇지 않다면 $R1 \in_R \{0,1\}^n$과 $C \in_R \{0,1\}^m$를 생성하고 SFlag를 true로 만든다. 마지막으로 태그는 (2-1) {HID←H(ID ‖ C), HCWD←H(R1 ‖ CWD), XR1←ID⊕R1} 메시지를 계산한 뒤 리더를 통해 데이터베이스 서버로 전송한다.

단계 3. (2-1) 메시지의 HID를 사용하여, 서버는 ID와 CWD의 후보들을 얻을 수 있고, XR1⊕(후보ID)를 계산하여 R1의 후보들을 얻어낼 수 있다. 그런 뒤 HCWD가 H(후보R1 ‖ 후보CWD)와 같은 지를 확인하여 CWD와 짝을 이루고 있는 실제 ID를 찾아낼 수 있다.

단계 4. 만약 서버가 이전 단계에서 만족하는 ID를 찾아낼 수 없었다면, 서버는 이 메시지를 공격으로 간주하고 무시한다. 만약 서버가 ID를 찾았다면 서버는 새로운 무작위 임시 변수 $R2 \in_R \{0,1\}^n$를 생성하고 (3) {HR←H(R1 ‖ R2), XR2←ID⊕R2} 메시지를 계산하여 태그에게 전송한다.

단계 5. 만약 태그가 (3) 메시지를 받았다면, THR_COUNT를 증가시키고, H(R1 ‖ (ID⊕XR2))이 HR과 일치하는 지 확인한다. 만약 일치한다면 태그는 (4-1) {CWD}를 데이터베이스 서버에게 전송하고 단계 7을 실행한다.

단계 6. 만약 태그가 거절할 어떠한 이유가 있거나 (3) 메시지가 올바르지 않다면, 태그는 THR_COUNT가 만료되기까지 다른 메시지들을 기다려야 한다. THR_COUNT가 THR_MAX에 도달하면, 태그는 SFlag를 false로 만들고, 다른 리더에 의해서 세션이 열리기 전까지 모든 다음 메시지를 무시한다.

단계 7. 태그는 (4-1) 메시지를 전송한 후, 그리고 데이터베이스 서버는 CWD를 올바르게 받은 후 해당 ID와 CWD를 새롭게 갱신한다. 만약 데이터베이스 서버가 CWD를 받지 못했다면, 이전 메시지를 재전송(replay)공격으로 판단하고 인증을 중단한다. 태그는 SFlag를 false로 만들고, 서버는 모든 가능한 H(ID ‖ C)를 위하여 해시 공간을 준비해야만 한다. 여기서 CWD는 이전 세션의 R1과 R2로 구성된다. 그것은 i가 현재 세션을, i-1이 이전 세션을 의미할 때

$$CWD_i = (R1_{i-1} + R2_{i-2}) \bmod 2^n$$

와 같고 또한

$$ID_i = ID_{i-1} \oplus R1_{i-1} \oplus R2_{i-1}$$

이다.

4.3 보안 안전성 분석

(위치 추적에 대하여) 공격자가 목표로 한 태그를 추적하고자 할 때, 그는 (2-1)과 (3) 메시지를 사용할 수밖에 없는데, 왜냐하면 (1)과 (4-1) 메시지에서는 어떠한 정보도 담고 있지 않기 때문이다. 그러나 만약 그가 해시된 메시지들로부터 충돌 쌍(collision pair)을 찾아낼 수 없다면 공격자는 ID⊕R1과 ID⊕R2를 이용하여 태그를 추적할 단서를 얻어낼 수밖에 없다. 공격자는 이러한 필드들에서 ⊕연산의 결과만 얻을 수 있기 때문에, 그는 (ID⊕R1)⊕(ID⊕R2)=R1⊕R2 만을 알 수 있다. R1과 R2는 태그와 서버에 의해서 매 세션 새롭게 생성되기 때문에 공격자가 모든 인증 메시지를 관찰할 수 있다고 하더라도 R1⊕R2는 추적을 위한 단서가 될 수 없다.

혹, 어떤 공격자는 두 번의 세션을 연속 관찰함으로써, R1⊕R2(이전 세션에서)와 R1+R2(=CWD, 현재 세션에서)를 얻은 뒤 이를 이용하여 이전 세션의 R1과 R2를 구하려고 할지도 모른다. R1과 R2를 구할 수 있다면 이전 세션에서 ID⊕R1또는 ID⊕R2를 이용하여 ID를 찾아내고 다음 세션부터 이를 이용하여 위치추적이 가능하다. 물론, R1⊕R2와 R1+R2로부터 구할 수 있는 R1과 R2 쌍은 비트 열에서 0과 1이 교차하는 횟수를 k라고 하였을 때 2^k개(k의 기댓값은 $n/2$)보다 많으므로[1] n이 충분히 클 때 이러한 방법도 여의치 않아 보이지만,

근본적으로 이러한 공격은 있을 수 없다. 두 번 연속으로 세션을 관찰하였다는 것은 이미 이 태그가 위치 추적 되었다는 의미이기 때문이다. 많은 다른 태그들 속에서 동작함으로써 자신을 숨기는 것을 기본으로 하는 RFID 태그의 위치 추적 방지 프로토콜의 특성상 이러한 가정은 매우 '이상하다.'

어떠한 공격자는 물리적으로 태그의 옆에서 연속으로 REQUEST 메시지를 보냄으로서 모든 H(ID ‖ C)를 알아내려고 할지도 모른다. 그리고 이러한 정보를 다음에 태그가 인증 받을 때 그 위치를 추적하는 데 사용할 수 있을 것이다. 하지만, 이러한 공격은 태그가 매 REQUEST 메시지마다 다른 세션으로 인식하게 만드는 것이 중요하다. 만약 태그에 충분한 용량의 콘덴서와 약간의 고의적 처리지연 기능이 내장되어 있다면 공격자가 이러한 일을 수행하는데 걸리는 시간은 그에 비례하여 증가하게 된다. 그러한 콘덴서를 제외하더라도, 대략 5초를 한 번의 세션으로 인식하게 하는데 걸리는 시간이라고 한다면, 공격자는 m이 10인 상황에서, 공격자는 5120초 동안을 위치추적하려는 태그와 물리적으로 근접한 상태에 있어야 한다는 이야기가 된다. 또한, 다음번 인증이 일어나면 그 위치를 알 수는 있겠지만, 다음 번 추적을 위해서는 또다시 5120초를 물리적으로 근접해 있어야 한다. 그래도, 만약 이러한 비현실적인 공격이 걱정된다면 간단히 태그를 인증할 때는 제외하면 Faraday Cage에 넣어두면 된다.

(위조 공격에 대하여) 만약 공격자가 데이터베이스와 태그를 비 동기화시키기 위하여 데이터베이스 서버나 태그를 위조하려 할 때, 공격자는 (2-1)이나 (3) 메시지를 올바르게 생성해야만 한다. (비 동기화는 서비스 거부 공격의 일종이 될 수 있다.) 공격자는 그가 만약 이전 세션을 관찰하여 올바른 CWD의 값을 알아낸다고 하더라도 올바른 H(R1 ‖ CWD)를 2^{-n} 확률로만 생성해낼 수 있는데, 이는 공격자가 올바른 R1을 얻기 위해서 올바른 ID를 추측해야만 하기 때문이다. 한편, 공격자는 불특정 태그를 공격하기 위하여 ID와 R1을 고를 수 있는데, 이 경우 올바른 CWD를 2^{-n} 확률로 생성해낼 수 있을 뿐이다. 데이터베이스에서 공격자가 선택한 ID가 존재할 확률은 $|ID|\times 2^{-n}$이므로 공격자가 이러한 공격을 성공할 확률은 $|ID|\times 2^{-2n}$과 같다.

공격자는 불법적인 (3) 메시지를 보냄으로서 태그를 속일 수 있는데, 만약 공격자가 올바른 ID를 추측할 수 있다면 (2-1) 메시지의 ID⊕R1에서 R1 추출할 수 있으므로, ID⊕R2에 맞는 올바른 H(R1 ‖ R2)를 생성해낼 확률은 2^{-n}과 같다.

(replay 공격에 대하여) 공격자는 태그로부터 미리 정당한 (2-1) 메시지를 얻어낸 뒤, 이를 리더에게 전송함으로써 태그와 데이터베이스 간에 동기화되지 못하도록 시도할지 모른다. 그러나 인증 세션은 CWD가 서버에게 안전하게 전송된 다음에 끝나게 되므로 공격자는 이러한 공격이 성공하기 위해서는 CWD를 알아야 한다. 이전 세션을 도청한다면, R1⊕R2를 얻어낼 수 있지만 이로부터 구할 수 있는 R1과 R2 쌍은 2^n개만큼 존재하므로 R1+R2를 알아내는 것은 사실상 불가능하다.

어떠한 공격자는 정당한 인증 세션을 관찰하다 CWD가 전송되는 것을 막아 인증 세션을 끝마치고 다음에 자신이 모아두었던 유효한 (2-1)과 (4-1) 메시지를 재전송하여 자신을 정당한 태그로 속이려 할지도 모른다. 기본적으로 이 논문에서는 이러한 유형의 공격은 불가능하다고 가정하지만, (CWD를 데이터베이스 서버는 모르고 공격자 자신만 알 수 있도록 해야 한다.) 메시지 차단이 이루어져서 CWD가 막혔다고 하더라도, 시스템 및 사용자는 시작한 인증 세션을 처음부터 인증 세션을 다시 시작하더라도 완벽히 끝마치려고 노력할 것이기 때문에 CWD는 끝내 전송될 것이고, 공격자는 이 짧은 순간에 replay 공격을 실행하지 않으면 이를 위해 수집해 놓은 메시지들은 더 이상 유효하지 않을 것이다.

(서비스 거부 공격에 대하여) 태그에 대한 서비스 거부 공격은 Ⅲ장 2절과 3절에서 이미 기술 되었다. 또한 이 문제에 대해서 어떻게 동시에 해결할 것인가 하는 해결책을 Ⅳ장 2절에서 설명하였고, 제안한 프로토콜은 이러한 원칙을 따라 작동하게 된다. 태그가 생성하는 무작위 임시 변수를 사용하고 공격 시도에 대한 한계 카운터 변수를 사용함으로써, 이 프로토콜은 서비스 거부 공격에 대해서 내성을 가지고 있다. 공격자가 메시지를 수정할 수 없다

1) 이는 R1과 R2가 동일한 포지션에서 서로 다른 비트, 즉, (0,1) 또는 (1,0)을 가질 때 이를 서로 교체하여도 R1⊕R2와 R1+R2의 연산 결과에 영향을 미치지 않는다는 사실로부터 쉽게 알 수 있다. 따라서 R1⊕R2, R1+R2로부터 R1과 R2를 하나로 특정할 수 있는 경우는 오로지 R1=R2인 경우뿐이다.

고 가정한다면 서비스 거부 공격을 시도하기 위해서는 메시지를 보내거나 중간에 끼워 넣어야만 한다. 공격자가 (1-1) 메시지를 보냄으로써 Preemptive Locking이나 Stealth Bombing 공격을 시도할 수 있지만 태그는 한 세션 동안 동일한 (2-1) 메시지로 응답하기 때문에 공격은 성공할 수 없다. 또한 (2-1) 메시지로부터 ID나 R1 또는 C를 올바르게 추측할 수 없다.

데이터베이스 서버가 태그의 ID를 찾으려고 할 때, 모든 H(ID ‖ C)의 값은 이미 계산되어 있기 때문에 서버가 해시를 할 필요는 없다. 이러한 전략을 사용함으로써, 공격자는 데이터베이스 서버에게 어떠한 경우라도 해시 연산을 수행하게 할 수 없다. 공격자가 데이터베이스 서버에게 보낼 수 있는 메시지는 오로지 (2-1) 메시지뿐이라는 사실을 유의하라. 하나의 공격 메시지에 대해서 데이터베이스가 경험하게 되는 작업량은 오로지 ID 공간을 한 번 뒤져보는 일뿐이다. 그렇기 때문에, 공격자가 세션 도중에 자신이 만든 (2-1) 메시지를 끼워 넣는다 할지라도 데이터베이스 서버는 그 메시지를 무시할 것이다.

태그가 (3) 메시지를 기다릴 때, 공격자는 인증을 실패하도록 하기 위해서 올바르지 않은 메시지를 태그에게 보낼 수 있다. 그러나 이러한 메시지에 대해서 태그 내부의 THR_COUNT가 증가하게 되고, 만약 올바른 메시지가 THR_COUNT가 종료되기 전에 도착한다면 인증은 성공적으로 마칠 수 있을 것이다. 서버가 인증에 성공하기 위해서는 서버는 (3) 메시지를 반드시 제 시간에 보내주어야만 한다.

4.4 효율성 분석

이 프로토콜은 다른 프로토콜들에서 공격자가 태그를 가장하여 데이터베이스에 대한 서비스 거부 공격을 수행하였을 때 의무적으로 수행해야하는 ID 검색을 위한 해시 연산을 수행하지 않는다. 왜냐하면 데이터베이스는 H(ID ‖ C)에 대한 모든 가능한 슬롯을 미리 준비해두었기 때문이다. [13]과 같이 ID가 리더와 태그에서 받은 임시 변수와 함께 해시된 경우 이러한 슬롯을 미리 준비해둔다는 것은 그 크기가 매우 크기 때문에 불가능한 일이다. |ID|가 시스템 안에서 사용 중인 태그의 수를 나타낸다고 하면, 슬롯의 총 개수는 $2^m \times |ID|$이다. 만약, H(ID ‖ C)의 모든 가능한 후보가 균등하게 분포되고 2^n이 $2^m \times |ID|$보다 크다면 동일한 슬롯에서 충돌이 발생하지 않는다. 시스템이 몇몇 충돌을 가지고 있다면, 약간의 링크된 슬롯이 있을 수 있다. 그래서 사용 중인 태그의 수가 $2^{(n-m)}$ 보다 작을 때 시스템에 적합할 수 있다. 만약 태그의 수가 약 4,294,967,296 ($=2^{32}$)개라면 n은 42, m은 10이면 데이터베이스 구조의 효율성 측면에서 충분하다. 그러나 n이 128 정도 된다고 하더라도, 이러한 구조일 때 예측 가능한 데이터베이스의 용량은 대략 1.1 Tbyte정도로 큰 문제가 되지 않아 보인다. n과 m은 크면 클수록 보안에 좋지만 m의 크기는 데이터베이스의 해시 연산 횟수와 태그의 위치추적 가능성에 크게 연관되어 있기 때문에 적합한 응용 분야에 따라 이 값을 조절할 수 있을 것이다.

또한 데이터베이스는 인증프로토콜이 끝난 뒤 휴지시간에 다음 인증 세션에서 사용할 슬롯을 미리 준비해야 한다. 만약 m이 10이라고 했을 때, 이 프로토콜은 모든 태그의 인증 마다 1,024번의 해시 연산을 수행해야한다는 것과 같다. 그렇기 때문에 이러한 부분에서 비효율적으로 보이지만, 내부 실험 결과 실제 MD5 해시 함수의 1,024번의 연속 연산을 수행하는데 걸리는 시간은 대략 0.017초 정도이다. 데이터베이스에 접근하여 내용을 기록하는 것까지 포함한 시간은 대략 0.283초 정도가 소요된다는 점으로 미루어 보았을 때 이는 '실시간'으로 처리한다고 해도 데이터베이스 서버에 크게 부담되는 일이 아니다.

또한, 이 프로토콜은 RNG를 태그가 내장해야하는데, 일반적인 RNG의 경우 이를 구현하기 위해서는 약 300,000개의 게이트가 있어야 한다고 하며, 이런 이유 때문에 이를 태그에 내장하기란 불가능해 보인다. 그런 이유에서 [12]에서는 해시 함수를 사용하여 이를 대체하는 방법을 소개하였지만, [14]에 따르면 불과 몇 백 개의 게이트를 사용하여 완전한 RNG를 구현하는 방법이 서술되어 있다.

표 1은 제안한 프로토콜과 다른 유사한 프로토콜 간의 성능 및 기능 비교를 나타낸다.

추가적으로, 메시지 복구에 대한 내용은 이 논문에서 다루고자 하는 이야기가 아니다. 데이터베이스 또는 태그의 동기화가 메시지 차단에 의해서 이루어지지 않았다는 것은 이 논문에서 처음에 가정한 위협모델에 적합하지 않다. 실제로, 응용 계층의 메시지 차단을 위해서는 물리 계층과 통신 계층에서 발생하는 재전송 요구 또는 ACK 또한 차단해야한다는 의미이며 이를 위해서 전파 방해(Jamming)

표 1. 다른 프로토콜과의 성능 및 기능 비교 (*H : hash function, R : random number generation, L : 해시 연산 결과 길이, 괄호는 구현 방법에 따라 달라지는 수치)

	해시기반 ID변형[1]	개선된 해시기반 ID 변형	상태기반[11]	시도-응답 기반[12]	제안프로토콜	대안프로토콜
SB / PL	둘 중 하나	둘 중 하나	PL에 안전	둘 중 하나	모두	모두
DB DoS	O	O	×	×	O	O
위치추적	×	×	O	O	O	O
스푸핑	O	O	O	O	O	O
Replay	×	O	O	O	O	O
태그 계산량	H:3	H:2	H:3	H:2 R:1	H:3 R:1(2)	H:2 R:1(2)
DB 해시 계산량 (인증시)	2	1	- or $\sqrt{\|ID\|}$	O(|ID|)	2m+1	2m+1
DB 해시 계산량 (공격시)	-	-	$\sqrt{\|ID\|}$	|ID|	-	-
태그 발생 메시지	< 3L	1.5L	2.5L	< 2L	< 3L	< 2L
DB 발생 메시지	< 2L	0.5L	0.5L	L	< 2L	< 2L

같은 물리적인 방법이 동원되어야 한다는 것과 같다.

4.5 대안 프로토콜

이 절에서는 그림 7에 묘사되어 있는, 원래의 프로토콜을 약간 수정한 다른 프로토콜을 제시하고자 한다. 원래의 프로토콜에서 데이터베이스 서버는 태그를 우선 인증하지만(재전송 공격의 가능성은 남아 있다.), 이 대안 프로토콜에서는 태그가 데이터베이스 서버를 우선 인증하게 된다. (2-2) 메시지가 보내진 뒤에 서버는 ID의 후보들을 얻어낼 수 있고, 이를 기초로 올바른 ID를 찾을 때까지 서로 다른 (3) 메시지를 여러 번 보내어 인증을 시도할 수 있다. 올바른 ID와 R2가 포함되어 있는 올바른 메시지를 받은 뒤에 태그는 자신의 ID와 CWD를 바꾸고 평문 형태로 CWD를 보낸다. 만약 올바른 CWD가 도착하면, 데이터베이스 서버는 ID와 CWD를 제안 프로토콜과 같이 바꾼다.

R1⊕R2는 이전 세션을 관찰한다면 쉽게 얻어질 수 있는 것이지만, R1⊕R2에서 R1+R2를 추출한다는 것은 어려운 일이다. 그렇기 때문에 공격자는 목표로 정한 태그를 추적하기 위한 어떠한 단서로 얻어낼 수 없다. 또한 태그가 (4-2) 이후 인증을 받을 수 있기 때문에 재생 공격에 대해서도 안전하다.

대안 프로토콜은 평균적으로 원래 프로토콜에 비하여 미세하게나마 더 많은 메시지를 요구하지만, 원래 프로토콜과 달리 오로지 단 두 번만의 해시 연산을 필요로 한다.

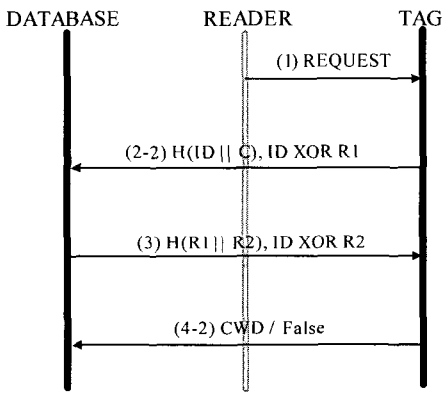

그림 7. 대안프로토콜

V. 결 론

RFID 시스템에는 많은 인증 프로토콜이 존재함에도 불구하고, 오로지 몇몇 프로토콜만이 위치 추적에 대한 안전성을 보장해준다. 태그의 하드웨어적 제한 사항으로 인하여 이러한 프로토콜은 서비스 거부 공격을 포함하는 많은 보안 위협으로부터 안전하지 못하다.

이 논문에서는 RFID 시스템의 위협 모델을 가정하였고, 어떠한 특수한 공격이 일반적인 인증 프로토콜에 시도될 수 있는지 설명하였다. 이러한 문제들을 풀기 위해서, 우리는 세션 동안 동일한 임시 변수를 유지하고 한계 카운터 변수를 사용하는 두 가지 전략을 사용하였다. 이러한 기법을 사용하여, 우리는 위치 추적에 대한 안전성을 보장하고 서비스

거부 공격에 강한 인증 프로토콜을 제시하였다.

마지막으로 제안한 프로토콜에 대한 공격을 세 가지 유형으로 나누어 확인하였으며, 이 프로토콜이 적당한 보안 강도를 가지고 있음을 보여주었다. 추가적으로, 우리는 한 번의 해시 연산을 줄인 대안 프로토콜 또한 제시하였다.

참 고 문 헌

[1] Dirk Henrici and Paul Müller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Device using Verying Identifiers," University of Kaiserslautern, Germany, Workshop on Pervasive Computing and Communications Security - PerSec 2004, pp. 149-153, IEEE, 2004

[2] Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshita, "Cryptographic Approach to 'Privacy-Friendly' Tags," NTT Laboratories, Japan, RFID Privacy Workshop MIT, 2003

[3] István Vajda and Levente Buttyán, "Lightweight Authentication Protocols for Low-Cost RFID tags," Budapest University of Technology and Economics, Hungary, 2003

[4] Ari Juels, "Minimalist Cryptography for Low-Cost RFID Tags," RSA Laboratories, USA

[5] Ari Juels, "Yoking-Proofs for RFID Tags," RSA Laboratories, USA

[6] Ari Juels, Ronal L. Rivest and Michael Szydlo, "The Blocker Tag: Selective Blocking of RFID Tag for Consumer Privacy," RSA Laboratories, USA

[7] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson, "Universal Re-encryption for Mixnets,"

[8] Stephen J. Engberg, Morten B. Harning and Christian Damsgaard Jensen, "Zero-knowledge Device Authentication: Privacy & Security Enhanced RFID preserving Business Value and Consumer Convenience," Privacy, Security and Trust 2004 - PST2004, EU Smarttag Workshop, 2004

[9] Martin Feldhofer, "A Propsal for an Authentication Protocol in a Security Layer for RFID Smart Tags," Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Austria

[10] Gildas Avoine and Philippe Oechslim, "RFID Traceability: A Multilayer Problem," École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, Financial Cryptography - FC'05, LNCS, Springer, 2005

[11] 유성호, 김기현, 황용호, 이필중, "상태기반 RFID 인증 프로토콜," 포항공과대학교, 대한민국, 정보보호학회 논문지 제14권 6호, 2004년 12월

[12] Keunwoo Rhee, Jin Kwak, Seungjoo Kim, and Dongho Won, "Challenge-Response based RFID Authentication Protocol for Distributed Database Environment," Proc. of SPC 2005, 2nd International Conference on Security in Pervasive Computing, Springer-Verlag GmbH, LNCS 3450, pp. 70-84, April 2005

[13] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," In First International Conference on Security in Pervasive Computing, 2003.

[14] Epstein, M., Hars, L., Krasinski, R., Rosner, M., Zheng, H.: Design and implementation of a true random number generator based on digital circuit artifacts. In Walter, C.D., C etin K. Koc, Paar, C., eds.: CHES 2003. Volume 2779 of LNCS., Berlin, Springer-Verlag pp.152-165, 2003

〈著者紹介〉

강 전 일 (Jeonil Kang) 학생회원
2003년 2월: 인하대학교 컴퓨터 공학과 졸업
2004년 3월~현재: 인하대학교 정보통신대학원 석사과정
〈관심분야〉 RFID 보안

양 대 헌 (DaeHun Nyang) 정회원
1994년 2월: 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월~2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월~현재: 인하대학교 정보통신대학원 조교수
〈관심분야〉 암호이론, 암호프로토콜, 인증프로토콜, 무선 인터넷 보안