RFID tag를 위한 강력한 Yoking Proof Protocols

정회원 조 정 식*, 여 상 수**, 김 성 권*

Strong Yoking Proof Protocols for RFID Tags

Jung-Sik Cho*, Sang-Soo Yeo**, Sung-Kwon Kim* ***Regular Members***

요 약

RFID 시스템은 작고 값싼 태그를 물품에 부착하여, 이를 이용해서 물품들을 식별하는 비접촉식 자동인식 시스템이다. 이 시스템은 현재 가장 많이 사용되고 있는 접촉식 판독 기법인 바코드를 대처할 것으로 예상된다. RFID 시스템은 다양한 응용분야가 존재한다. 그 중 Ari Juels는 2개의 태그가 동시에 인식된 것을 증명할 필요가 있는 환경을 설명하고, 이를 위한 프로토콜로 yoking proof 프로토콜을 제안하였다. 하지만 yoking proof 프로토콜은 재생 공격에 약하다. 이를 보완하기 위한 변형된 yoking proof 프로토콜들이 제안되었지만, 역시 재생 공격에 자유롭지 못했다. 본 논문에서는 기존 yoking proof 프로토콜들의 문제점을 분석하고 이를 바탕으로 재생 공격을 어렵게 하는 새로운 프로토콜을 제안한다. 또한 이 프로토콜을 n개의 태그에 대한 yoking proof를 제공할 수 있도록 프로토콜을 확장시켰다.

Key Words : RFID, Yoking proof, Timestamp, Random number, Replay attack

ABSTRACT

The RFID system is a non-contact automatic identification system that identifies tags through a reading device by attaching small, inexpensive tags on goods. This system is expected to supplant barcodes, the contactless reading technique that is most widely used at present. The RFID system can be applied in a variety of areas. Among those, Ari Juels proposed an environment to prove that a pair of tags has been scanned simultaneously. And he presented a yoking proof protocol for this. But the yoking-proof protocol is vulnerable to replay attack. Although modified yoking-proof protocols for alleviating this drawback have been proposed, they are not immune to replay attack, either. In this paper, we analyze problems of existing yoking-proof protocols and present a new protocol, which will make replay attack difficult, based on this analysis. We have also extend this protocol so that it can provide yoking proofs for n tags.

Ⅰ. 서 론

RFID 시스템은 작은 크기의 값싼 microchip으로 이루어진 RFID 태그와 RFID 리더, 그리고 back-end 서버로 이루어진 것을 말한다. RFID 리더는 짧은 거리의 무선 통신을 통해 RFID 태그에 저장된 고유한 식별 정보를 전달받고, 이를 back-end 서버로 전송하여 해당 RFID 태그가 부착된 물품의 정보를 인식하는 시스템이다.

RFID 시스템은 현재 물류에 대한 인식 시스템인 접촉식 판독 기법의 바코드를 대처할 것으로 예상되고 있다. RFID 시스템 사용으로 인하여 취할 수 있는 장점은 대량의 RFID 태그에 대한 정보를 한 번에 인식 할 수 있다는 것이다. 이런 RFID 기술을 활용한 응용분야는 다양하게 존재한다. 그 중 Ari Juels는 yoking proof라는 개념을 제안하였다. 이는

※ 이 논문은 2005년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No.R01-2005-000-10568-0).
* 중앙대학교 컴퓨터공학부 알고리즘 및 정보보호 연구실 (mfg@alg.cse.cau.ac.kr)
** 규슈대학교 정보공학부 (ssyeo@itslab.csece.kyushu-u.ac.jp)
논문번호 : KICS2006-11-496, 접수일자 : 2006년 11월 16일, 최종논문접수일자 : 2007년 3월 9일

두 개의 태그가 동시에 인식되었다는 것을 검증자에게 증명하는 프로토콜이다[2]. 하지만 yoking proof 프로토콜은 악의를 가진 공격자에게 쉽게 재생 공격을 받을 수 있다는 문제를 가지고 있다. 이는 다음과 같이 RFID 시스템이 내재하고 있는 몇 가지 취약점으로 인해 발생한다.

- RFID 태그는 값싼 chip 으로써 한정된 기계적 성능을 가지고 있다.
- RFID 시스템은 무선 주파수를 사용함으로써 쉽게 도청될 수 있다.
- RFID 시스템은 태그와 리더 간 상호 인증을 제공해 줄 수 없다.

이러한 취약점은 비단 Ari Juels가 제안 환경에서만 있는 것은 아니다. RFID 시스템을 사용하는 모든 응용 분야에서는 소비자의 privacy 문제와 local privacy 문제, forgery 문제 등이 문제화 되고 있다. 따라서 이를 막기 위한 많은 연구들이 진행되어 지고 있다.

Saiko와 Sakurai는 Ari Juels가 제안한 프로토콜의 취약점을 보완하기 위해 타임스탬프를 사용하는 프로토콜[3]을 제안하였다. 하지만 타임스탬프는 순차적으로 증가한다는 특징과 유효범위가 노출되면 쉽게 재생 공격 할 수 있다는 약점이 있다. 이어 Selwyn Piramuthu는 검증자로부터 전달 받은 임의의 수를 바탕으로 한 프로토콜[4]을 제안하였다. 하지만 공격에 필요한 복잡도가 크지 않다는 점에서 이 역시 재생 공격에 노출될 수 있는 가능성이 남아있다. 따라서 본 논문에서는 재생 공격에 대응할 수 있는 Strong yoking proof 프로토콜을 제안한다. 제안 프로토콜은 공격자가 재생 공격을 시도 할 때 높은 복잡도를 요하는 방법을 사용하였다. 이밖에도 컨테이너 혹은 박스 안에 있는 물류를 파악하는 응용분야가 존재할 수 있다. 이때 n개의 태그가 동시에 인식되었다는 것을 검증자에게 증명 해줄 필요가 있다. 따라서 본 논문은 향상된 yoking proof 프로토콜을 확장시켜 이를 달성할 수 있는 프로토콜에 대해서도 제안한다.

Ⅱ. 관련 연구

Yoking proof 프로토콜의 중요 관점은 두 개의 태그가 동시에 인식되어야하는 것이다. 이때 리더는 믿을 수 없는 존재다. 리더는 검증자에게 두 태그가 동시에 인식되었다는 것을 증명해주어야 한다. 다음 설명되는 프로토콜들은 이를 위한 Ari Jules의 프로토콜[2]과 그 프로토콜이 가지고 있는 약점인 재생 공격을 막기 위해 제안된 프로토콜[3][4]들의 설명과 분석들이다. 다음 표1은 사용되어지는 표기에 대한 설명이다.

표 1. 표기 설명.

표기법	설 명
V	증명자(verifier)
r, r_A, r_B	임의의 값(random number)
x_A, x_B	태그 T_A, T_B 각각에 대한 비밀 키
MAC	Message Authentication Code
$MAC_x[m]$	비밀 키 x를 통해 만들어진 매시지 m 의 MAC
TS	타임스탬프(time stamp)
P_{AB}	태그 A와 B 가 동시에 인식되었다는 것을 검증자에게 증명하는 증명들

2.1 Yoking proof protocol

Yoking proof 프로토콜의 구성 요소는 우선 프로토콜에 참가하는 두 개의 태그 T_A, T_B 가 있으며, 이를 동시에 인식하는 리더와 동시에 인식하였다는 것을 증명할 수 있는 검증자로 구성된다. 이때 태그 T_A, T_B 는 검증자와 각각의 비밀 키 x_A, x_B를 공유하고 있는 상태며, 리더는 이를 모른다. 프로토콜은 다음 그림 1 과 같이 이루어지며, 주어진 시간 t 안에 이루어져야한다.

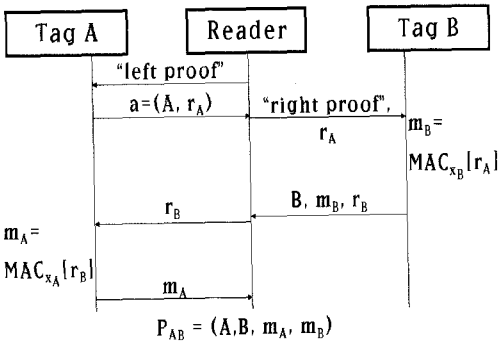


그림 1. minimalist yoking proof protocol[2]

자세한 프로토콜은 다음과 같이 이루어진다.

단계 1 : 리더는 태그 A에게 "left proof" 를 통해 request 한다.

단계 2 : 태그 A는 request 에 대한 response로

임의의 수 r_A를 생성하여 리더에게 보낸다.

단계 3 : 리더는 태그 A로부터 받은 r_A를 태그 B에게 보낸다.

단계 4 : 태그 B는 다음과 같이 비밀 키 x_B를 사용하여 r_A의 MAC을 생성한다. 이를 m_B라 한다.

$$m_B = MAC_{x_B}[r_A]$$

단계 5 : 태그 B는 단계 4에서 생성된 m_B와 임의의 수 r_B를 생성하여 리더에 전달한다.

단계 6 : 리더는 태그 B에게 받은 r_B를 태그 A에게 전달한다.

단계 7 : 태그 A는 다음과 같이 비밀 키 x_A를 사용하여 r_B의 MAC을 생성한다. 이를 m_A라 한다.

$$m_A = MAC_{x_A}[r_B]$$

단계 8 : 리더는 다음과 같은 P_{AB}를 생성하여 검증자에게 태그 A,B가 동시에 인식되었다는 것을 증명한다.

$$P_{AB} = (A, B, m_A, m_B)$$

위의 yoking proof 프로토콜에 대하여 다음 그림 2와 같은 방법으로 재생 공격이 가능하다.

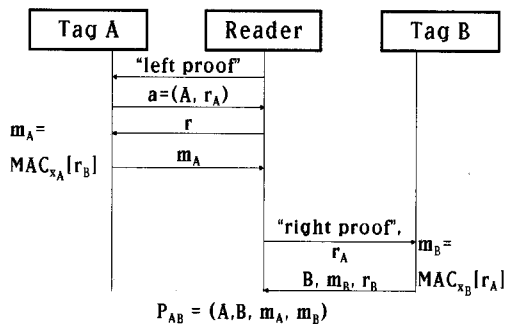


그림 2. yoking proof protocol 의 재생 공격 방법[3]

자세한 공격 방법은 다음과 같다.

공격 1 : 공격자는 태그 A에게 "left proof" 를 통해 r_A를 받는다.

공격 2 : 공격자는 자신이 생성한 임의의 수 r을 태그 A에게 보낸다.

공격 3 : 공격자는 r을 통해 태그 A가 생성한 m_A를 받는다.

이후 공격자는 태그 A가 없어도 공격 1~3 단계에서 수집해둔 정보를 바탕으로 다음과 같이 공격 4~5단계를 통해 태그 B와 정상적인 정보를 주고받을 수 있다.

공격 4 : 태그 A로부터 받아 두었던 r_A를 태그 B에게 보낸다.

공격 5 : 태그 B로부터 정상적인 m_B와 r_B를 받는다.

공격 6 : 공격자는 $P_{AB} = (A, B, m_A, m_B)$를 생성하여 검증자에게 전달해 준다.

이런 단계를 수행함으로써 공격자는 태그 A로부터 미리 정보를 받아둘 수 있다. 공격자는 정보를 바탕으로 이후 태그 A 없이도 태그 B만으로도 검증자에게 두 태그 A, B가 동시에 인식 되었다는 것을 증명할 수 있게 된다. 이와 같은 공격이 가능한 것은 각 태그가 리더에게서 받은 임의의 값으로부터 MAC을 생성할 때 그 임의의 수가 상대 태그로부터 전달된 정당한 값인지 확인하지 않고 있으며 검증자 또한 이를 확인하지 않고 있기 때문이다.

2.2 타임 스탬프를 사용한 Yoking proof protocol

Saiko 와 Sakurai 는 yoking proof의 재생 공격을 보안하기 위하여 다음 그림3과 같이 타임스탬프를 사용하는 프로토콜을 제안하였다.

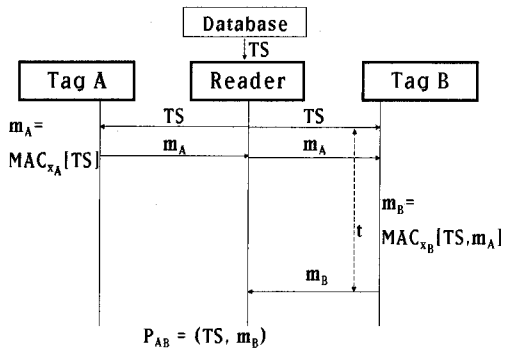


그림 3. 타임스탬프를 사용한 yoking proof protocol

Jules의 yoking proof와 달리 리더는 데이터베이스로부터 타임스탬프 TS를 받아 이를 두 태그에게 전달하여 MAC 생성에 사용하고 있다. 이때 태그

A는 TS만을 이용하고 있으며, 태그 B는 TS와 태그 A가 만든 MAC을 함께 이용하여 MAC을 생성하게 된다. 리더는 TS와 태그 B로부터 생성된 MAC을 받아 이를 검증자에게 제시함으로써 두 태그가 동시에 인식되었다는 것을 증명하게 된다.

이 프로토콜은 데이터베이스가 전달해준 TS를 기준으로 하여 주어진 시간 t 안에 프로토콜이 끝나는 전제 조건을 두고 있다. 이는 공격자로 하여금 시간적 제약을 주기 위함이다. 또한 m_A를 m_B의 생성 요소로 사용함으로써 두 태그의 생성값을 연관시키고 있다. 이렇게 함으로써 yoking proof 프로토콜이 가지고 있던 문제점인 MAC 생성에 사용되는 값에 대해 확인하지 않던 것을 보안하고 있다. 하지만 이 방법 역시 다음 그림 4 와 같은 방법으로 재생 공격이 가능하다.

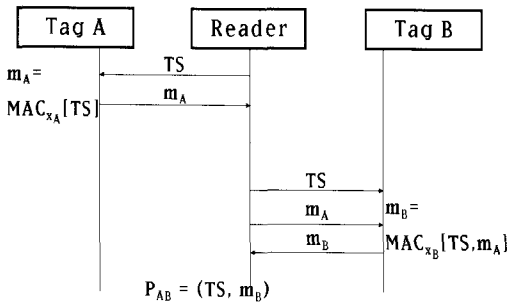


그림 4. 타임스탬프를 사용한 yoking proof protocol 에 대한 재생 공격[4]

자세한 공격 방법은 다음과 같다.

공격 1 : 공격자는 도청을 통해 TS를 알아 낼 수 있다. 이를 바탕으로 공격자는 유효한 TS의 범위를 구할 수 있다.

공격 2 : 공격자는 미리 구해둔 TS값들을 태그 A에게 request 하여 m_A값들을 수집해 둘 수 있다.

공격 3 : 차후 공격자는 태그 A가 없는 상태에서 데이터베이스로부터 정상적인 TS를 받게 되면 수집해둔 m_A중 해당 m_A를 찾아 TS와 함께 태그 B에게 보냄으로써 정상적인 m_B를 받을 수 있게 된다. 이렇게 함으로써 검증자에게 두 태그 A, B가 동시에 있다는 것을 증명해 줄 수 있다.

이때 TS의 범위는 검증자의 필요에 따라 정할 수 있으며 그 크기가 클 수록 유연한 공격이 가능하지만 복잡도는 높아 질것이다.

이는 타임스탬프가 순차적으로 증가하는 성질로 인하여 발생하는 문제점이다. 공격자는 도청과 같은 경로를 통해서 최근의 TS를 알 수 있고 따라서 공격자는 유효한 범위의 TS를 미리 구해 태그 A에게 보냄으로써 m_A를 수집해 둘 수 있다는 것이 문제다. 또한 태그가 TS를 확인하지 않는 것도 문제가 될 수 있다.

2.3 Modified yoking proof protocol

Selwyn Piramuthu는 위의 두 프로토콜에서 나타나는 재생 공격을 막기 위해 그림5와 같이 타임스탬프 대신 검증자로부터 임의의 수 r을 전달 받는 방법을 사용하였다.

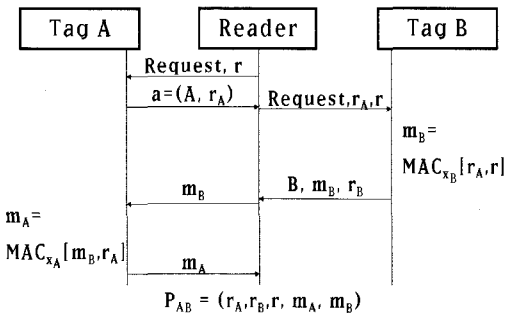


그림 5. Modified yoking proof protocol

자세한 프로토콜은 다음과 같은 단계를 거친다.

단계 1 : 리더는 검증자로부터 r을 전달 받는다.

단계 2 : 리더는 태그 A에게 r을 보냄으로써 request 를 수행한다.

단계 3 : 태그 A는 r을 시드로 하여 자신의 임의의 수 인 r_A를 생성하여 리더에게 보낸다.

단계 4 : 리더는 r_A와 r을 함께 태그 B에게 보냄으로써 request 를 수행한다.

단계 5 : 태그 B는 비밀 키 x_B를 이용하여 r_A와 r의 MAC, m_B를 생성한다.

단계 6 : 태그 B는 r을 시드로 하여 생성한 임의의 수, r_B를 생성하여 m_B와 함께 리더에게 보낸다.

단계 7 : 리더는 m_B를 태그 A에게 보낸다.

단계 8 : 태그 A는 자신의 비밀 키 x_A를 이용하

여 r_A와 m_B의 MAC, m_A를 생성한다.

단계 9 : 태그 A는 m_A를 리더에게 보낸다.

단계 10 : 리더는 $P_{AB}=(r_A,r_B,r,m_A,m_B)$ 를 생성하여 검증자에게 전달하여 태그 A, B 가 동시에 있다는 것을 증명한다.

이렇게 함으로써 타임스탬프를 사용한 프로토콜에 비해 얻어지는 이점으로는 r값이 random 하기 때문에 공격자가 그 값을 미리 추정하기 힘들어진다는 것이다. 하지만 r값의 범위가 크지 않다면 공격자는 브루트 포스 공격방법을 통해 공격이 가능하다.

우선 공격자는 검증자에게 정상적인 $P_{AB}=(r_A,r_Br,m_A,m_B)$를 전달한다면 공격은 성공한 것이다. 따라서 다음과 같은 공격 방법을 통해 공격자는 검증자에게 $P_{AB}=(r_A,r_B,r,m_A,m_B)$를 전달함으로써 공격을 성공시킬 수 있다.

공격 1 : 공격자는 구할 수 있는 r의 값을 모두 구해둔다.

공격 2 : 태그 A 에게 가능한 r에 대한 request 을 보내서 해당하는 r_A를 수집한다.

공격 3 : r과 수집된 r_A를 태그 B 에게 보내어 m_B와 r_B를 수집해 둘 수 있다.

공격 4 : 공격자는 차후 데이터베이스로부터 유효한 r를 받았을때 해당하는 값들을 바탕으로 재생 공격이 가능하다.

추가적으로 m_A까지 수집해둔다면 두 태그 없이도 검증자에게 두 태그가 동시에 인식되었다고 속일 수 있을 것이다. r이 n-bit 라 하고 MAC 을 d-bit 라 할 때 각 공격단계에 필요한 복잡도를 분석해 보면 다음과 같다.

공격 1 분석 : 가능한 r을 모두 구하기 위해서는 n * 2^n bit 저장 공간이 필요하다.

공격 2 분석 : 공격자는 r_A를 수집하기 위해 (공격 1)에서 소집 해둔 r값들을 모두 보내 본다. 이때 필요한 통신 횟수 2^n번 필요하며, n * 2^n bit 저장 공간이 필요하다.

공격 3 분석 : 공격자는 수집된 r_A를 바탕으로 태그 B 로부터 m_B,r_B 를 수집하기 위해서는 2^n의 통신이 필요하며, 추가적으로 n * 2^n bit + d * 2^n bit 저장 공간이 필요하다. 이후 m_A까지 수집한다면 2^n의 통신 횟수 와 d * 2^n bit 저장 공간이 추가 될 것이다.

이를 전체적으로 보면 공격에 필요한 저장공간은 2^n(3n+2d)bit 가 필요하고 통신 횟수는 3 * 2^n 번 필요하다.

Ⅲ. 제안 기법 - Strong Yoking Proof Protocol

2장의 관련연구를 통해 yoking proof와 관련된 프로토콜들을 살펴보았다. 하지만 제안되었던 프로토콜들은 RFID 시스템의 특성상 재생 공격에 취약하다는 것이 증명되었다. 따라서 이를 보완하기 위한 프로토콜이 필요하며, 이때 다음과 같은 사항을 만족해야 한다.

- 전체 시스템에는 부담을 주지 않으면서 보안을 강화 시킬 수 있는 프로토콜 필요, 특히 태그에게는 특별히 부담을 주지 않아야 할 것이다.
- 공격자에게는 공격이 어렵거나 불가능한 프로토콜이 필요하다.

본 논문은 위와 같은 사항을 만족 할 수 있도록 하기 위해 다음과 같이 각 사항에 대하여 고려하였다.

- RFID 시스템에 추가적인 부담을 주지 않는 프로토콜
  - ▸ 기존 프로토콜에서 태그들이 수행한 작업에 최대한 변형을 가하지 않는다.
  - ▸ 기존 프로토콜의 전체적인 형태는 유지한다.

- 공격자에게 공격에 있어 부담을 주는 프로토콜
  - ▸ 기존 프로토콜의 보안상 장단점을 파악한다.
  - ▸ 기존 프로토콜의 장점을 유지하면서 취약점을 보완할 수 있는 방안을 강구한다.
  - ▸ 가장 적절한 방안으로 공격자가 공격에 부담을 주기 위해 복잡도를 높이는 방법을 강구한다.

본 논문에서 위와 같은 고찰을 통해 "Strong yoking proof protocols"을 제안한다. 제안 프로토콜은 기존에 제안되었던 프로토콜인 "Modified yoking proof protocol"을 기반으로 하여 변형을 가하는 방법을 사용하였다. 이렇게 함으로써 얻을 수 있는 장점으로는 다음과 같다.

- 기존 프로토콜의 시스템 상의 추가적인 부담을 주지 않고 yoking proof를 쉽게 제공해준다.
- 태그에게 추가적인 부담을 주지 않고 공격자가 공격을 하기위한 복잡도를 쉽게 증가 시킬 수 있다.

우선 제안하는 프로토콜은 다음 그림 6과 같이 데이터베이스로부터 전달 받는 임의의 수를 태그 A를 위한 값 r_1과 태그 B를 위한 값 r_2로 각각 나누어 전달해 줌으로써 시작된다.

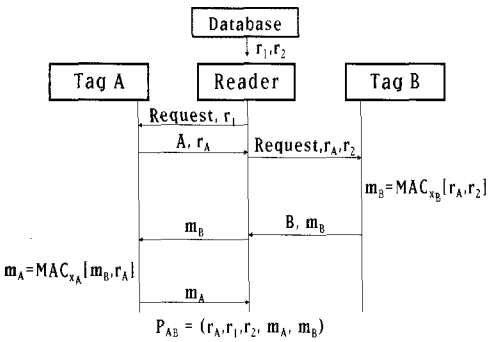


그림 6. Strong yoking proof protocol

자세한 프로토콜은 다음과 같다.

단계 1 : 리더는 r_1을 태그 A에게 전달해준다.

단계 2 : 태그 A 는 r_1을 시드로 하여 r_A를 생성하고 이를 리더에게 response 해준다.

단계 3 : 리더는 r_A와 r_2를 태그 B 에게 전달해 준다.

단계 4 : 태그 B 는 비밀 키 x_B를 사용하여 r_A, r_2의 MAC, m_B을 생성한다.

단계 5 : 태그 B는 m_B를 리더에게 전달한다.

단계 6 : 리더는 m_B를 태그 A에게 전달한다.

단계 7 : 태그 A는 비밀 키 x_A를 사용하여 자신의 임의의 수 r_A와 리더로부터 전달받은 m_B의 MAC, m_A를 생성하여 리더에게 전달한다.

단계 8 : 리더는 다음과 같은 P_{AB}를 통해 두 태그가 동시에 인식되었다는 것을 검증자에게 증명하게 된다.

$$P_{AB} = (r_A, r_B, r_1, r_2, m_A, m_B)$$

제안 프로토콜의 보안상 특징은 modified yoking proof 프로토콜과 동일하다. 하지만 각 태그에게 전달되는 임의의 수를 다르게 하고 있다는 차이에 의해 공격자 입장에서는 복잡도가 증가 할 수 밖에 없다. 이에 대한 자세한 분석은 다음 장에서 논하도록 하겠다.

제안 프로토콜에서 태그에 필요한 기능은 MAC 과 random number generator 이다. 앞에서 언급한 것과 같이 태그는 하드웨어적으로 한계를 가지고 있기 때문에 이러한 기능들은 태그의 특성에 적합한 것들이여 한다. MAC은 Ari Juels의 "yoking proofs"에서 제안한 Lamport digital signature scheme 의 truncated version을 사용하면 될 것이다. random number generator는 시드를 입력 받아 임의의 수를 생성하는 함수이여야 한다.

Ⅳ. 보안 분석

제안 프로토콜은 modified yoking proof 프로토콜을 변형한 것으로써 이 역시 재생 공격 이 가능하다. 하지만 공격자는 공격을 하면서 증가하는 복잡도에 대하여 고민해야할 것이다. 제안 프로토콜에 대한 재생 공격은 다음과 같이 modified yoking proof 프로토콜의 공격 방법과 비슷하게 수행된다. 공격자는 검증자에게 $P_{AB} = (r_A, r_B, r_1, r_2, m_A, m_B)$ 를 전달함으로써 공격을 성공시키는 것이 목적이다.

공격 1 : 공격자는 구할 수 있는 r_1 값과 r_2 값을 모두 구해둔다.

공격 2 : 태그 A에게 미리 계산해둔 r_1에 대한 request 를 보내서 해당하는 r_A를 수집한다.

공격 3 : 태그 B에게 수집된 r_A와 미리 계산해둔 r_2간의 나올 수 있는 모든 조합을 보내어 m_B를 수집한다.

공격 4 : 차후 데이터베이스로부터 유효한 r_1, r_2를 받았을 때 해당하는 값들을 바탕으로 재생 공격이 가능하다.

그렇다면 위 와 같은 공격에 대한 복잡도를 다음과 같이 각 단계별로 분석 해볼 수 있다.

공격 1 분석 : 미리 r_1, r_2를 모두 구한다. 이때 r_1과 r_2값의 범위는 동일하기 때문에 중복하여 구할 필요는 없다. 따라서 n * 2^n bit 저장 공간이 필요하다.

공격 2 분석 : 공격자는 r_A를 수집하기 위해 (공격 1)에서 미리 계산해둔 r_1값들을 모두 태그 A 에게 request 한다. 이때 필요한 통신 횟수 2^n번 필요하면, n * 2^n bit 저장 공간이 필요하다.

공격 3 분석 : 공격자는 태그 A로부터 수집된 r_A와 미리 계산해둔 r_2의 조합을 바탕으로 태그 B 로부터 m_B를 수집하기 위해서는 r_A와 r_2의 조합의 수인 2^{2n}의 통신이 필요하며, 저장 공간은 d * 2^n bit 필요하다.

이후 추가적으로 m_A까지 수집한다면 2^{2n}의 통신 횟수 와 d * 2^n bit 저장 공간이 추가 될 것이다. 이를 전체적으로 보면 공격에 필요한 저장공간은 2^n(2n+2d)bit 가 필요하고 통신 횟수는 $2^{2n+1}+2^n$번 필요하다. 이를 기존에 제안되었던 프로토콜들의 복잡도와 비교한다면 표 2와 같이 나타낼 수 있다.

표 2 에서 보는 거와 같이 yoking proofs와 타임스탬프를 이용한 프로토콜에서는 재생 공격을 수행하는데 그리 많은 복잡도가 요구 되지 않는다. 반면 modified yoking proof 프로토콜에서는 재생 공격에 비교적 안전하지만 r 값의 범위에 따라 브루트 포스 공격에 약할 수 있다. 따라서 제안 프로토콜은 이를 방지하기 위해 복잡도를 높이는 방안을 사용하여 modified yoking proof 프로토콜을 강화시키고 있다. 이때 저장공간에 대한 복잡도는 비슷하나 공격자가 재생 공격을 위해 수행하는 통신에 대한 복잡도가 공격 3단계, 즉 r_B와 m_B를 수집하는 단계부터 급격히 증가하는 것을 볼 수 있다. 따라서 제안하는 Strong yoking proof 프로토콜은 yoking proofs의 목적을 만족하면서 브루트 포스 공격 통한 재생 공격에도 안전한 프로토콜이라 할 수 있겠다.

V. Strong Yoking Proof for Tag Group

RFID 시스템을 이용하는 응용 중 n 개의 태그, 즉 여러개의 태그를 동시에 인식되었다는 것을 증명 해줄 응용이 존재 한다. 예를 들면 컨테이너, 혹은 박스에 물품이 정확이 들어 있는가를 증명해야 할 때가 있다. 이때 각 물품에는 태그가 부착되어 있고 컨테이너와 박스에도 태그가 부착 되어 있는 상태이다. 이때 컨테이너 혹은 박스에 부착된 태그는 물품에 부착된 태그 보다는 좀 더 연산 능력이 높은 태그로써 symmetric encryption 연산이 가능한 것이 사용되어야 할 것이다.

본 논문에서는 위에서 제안한 strong yoking proof 프로토콜을 바탕으로 하여 n개의 태그가 동시에 인식되었다는 것을 증명해주는 프로토콜을 다음 그림 7같이 제안한다. 우선 선행 작업으로써 박스 태그 A 는 검증자와 symmetric encryption 을 위한 비밀 키를 공유한다. 나머지 물품 태그들도 MAC 생성을 위한 비밀 키를 검증자와 공유한다. 이때 이 키는 태그 마다 다른 키여야 한다.

자세한 프로토콜은 다음과 같다.

단계 1 : 리더의 request 에 의해 검증자의 DB 에서는 r_1, r_2값이 리더에게 전달된다.

표 2. 각 프로토콜들의 공격에 대한 비용 비교

공격 단계	Yoking proofs		Timestamp		Modified		Enhanced yoking	
	저장 공간 (bit)	통신 횟수	저장 공간 (bit)	통신 횟수	저장 공간 (bit)	통신 횟수	저장 공간 (bit)	통신 횟수
r 수집					$2^n \times n$		$2^n \times n$	
r_A 수집	n bit	1			$2^n \times n$	2^n	$2^n \times n$	2^n
r_B, m_B 수집					$2^n(n+d)$	2^n	$2^n \times d$	2^{2n}
m_A 수집	d bit	1	$x \times d$ bit	x	$2^n \times d$	2^n	$2^n \times d$	2^{2n}
총 비용	$n+d$ bit	2	$x \times d$ bit	x	$2^n(3n+2d)$	3×2^n	$2^n(2n+2d)$	$2^{2n+1}+2^n$

n : random number bit, d : MAC 의 출력 bit, x: 공격 가능한 timestamp 의 유효범위 값

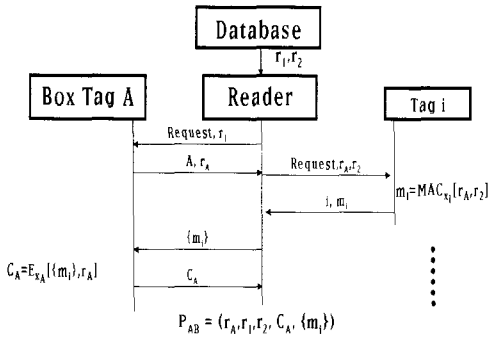


그림 7. n 개의 태그를 통시에 인식할 수 있는 프로토콜

단계 2 : 리더 는 r_1을 박스 태그 A 에게 request 와 함께 보낸다.

단계 3 : 박스 태그 A 는 r_1를 시드로 하여 r_A 를 생성하여 리더에게 전달해준다.

단계 4 : 리더는 r_A 와 r_2를 각 물품 태그들에게 각각 전달해준다.

단계 5 : 각 물품 태그들은 자신의 비밀 키 x_i를 사용하여 r_A, r_2의 MAC, m_i을 생성한다.

단계 6 : 각 물품 태그들은 m_i를 리더에게 전달한다.

단계 7 : 리더는 각 물품 태그들로부터 받은 m_i 를 수집하여 박스 태그 A에게 전달한다.

단계 8 : 박스 태그 A는 비밀 키 x_A를 사용하여 자신의 임의의 값 r_A와 리더로부터 전달 받은 $\{m_i\}$를 암호화한 C_A를 생성하여 리더에게 전달한다.

단계 9 : 리더는 다음과 같은 P_{AB}를 통해 박스 태그를 기반으로 하여 n 개의 태그가 동시에 인식되었다는 것을 검증자에게 증명하게 된다.

$$P_{AB} = (r_A, r_1, r_2, C_A, \{m_i\})$$

본 장에서 제안 하는 프로토콜은 3장에서 제안한 "Strong yoking proof protocol"을 기반으로 하고 있기 때문에 그 보안상의 복잡도를 그대로 상속 받고 있다. 즉 공격자는 하나 이상의 물품 태그에 대한 재생 공격이 성공하기 위해서는 하나의 태그 당 $2^{2n}+2^n$의 통신이 필요하게 된다. 또한 "Strong yoking proof protocol"을 사용하는 태그와 시스템에 추가적인 변형없이 그대로 사용할 수 있다는 것이 장점이다.

Ⅵ. 결 론

본 논문에서는 먼저 여러 yoking proof 프로토콜들에 대하여 분석하였다. 하지만 기존의 yoking proof 프로토콜들은 재생 공격이 가능하다는 단점이 있었다. 기존에 제안된 프로토콜들이 재생 공격에 쉽게 노출되는 가장 큰 이유를 분석해보면, RFID 시스템에서 사용되어지는 태그의 한계에서 비롯된다. 태그는 작고, 값싸다는 특징으로 인해 하드웨어적 한계를 가지고 있기 때문이다. 따라서 공격자에 의한 재생 공격을 완전히 봉쇄한다는 것은 아주 힘든 일이다.

따라서 본 논문에서는 이를 인정하고 시각을 달리하여 공격자로 하여금 재생 공격이 어렵도록 하는 방향을 선택하였다. 이때 중요한 점은 기존 시스템에 큰 변화 없이 몇 가지 요소의 추가로 이를 달성해야 하며, 특히 태그에 추가되는 연산을 최소화해야 한다는 것이다. 이 취지를 바탕으로 제안된 "Strong yoking proof protocol"은 기존에 제안된 "modified yoking proof protocol" 을 바탕으로, 태그에 추가적인 요구 사항 없이 시스템 의 작은 변화를 줌으로써, 공격자가 재생 공격하기 위한 복잡도를 높인 프로토콜이다. 또한 이를 확장시켜 n개의 태그를 동시에 인식되었다는 것의 증명을 만들어내는 프로토콜을 제안하였다.

참 고 문 헌

[1] K. Finkenzeller, RFID Handbook, *John Wiley& Sons*, 2002.

[2] A. Juels, "Yoking Proof" for RFID Tags", *IEEE Computer Society*, Proceedings of the First International Workshop on Pervasive Computing and Communication Security - PerSec 2004, pp.138-143, 2004

[3] J. Saito and K. Sakurai, "Grouping Proof for RFID Tags", *IEEE Computer Society*, Proceeding of the 19th International Conference on Advanced Information Networking and Applications(AINA'05), vol 2, pp. 621-624, 2005

[4] Selwyn and Priamuhtu, "On Existence Proofs for Multiple RFID tags", *IEEE Computer Society*, IEEE International Conference on

Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing - SecPerU 2006, pp.317-320, 2006

[5] G. Avoine and P. Oechslin, "RFID Traceability: A Multilayer Problem", *Springer-Verlag*, Financial Cryptography - 0FC'05, LNCS, vol 3570, pp.125-140, 2005

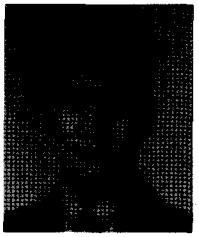
[6] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", *Springer-Verlag*, International Conference on Security in Pervasive Computing - SPC 2003, LNCS, Vol 2802, pp. 454-469, 2004

[7] Zhaoyn Liu and Dichao Peng, "True random number generator in RFID systems against traceability", *IEEE Computer Society*, Consumer Communication and Networking Conference, vol 1, pp.620-624, January 2006

[8] T. Dimitriou, "A Lightweight RFID Protocol to Protect Against Traceability and Cloning Attacks", *IEEE Computer Society*, Proceeding of IEEE International Conference on Security and Privacy for Emerging Areas in Communication Networks - SecureComm, pp.59-66, 2005.

[9] A. Lenstra and E. Verheul, "Selecting Cryptographic Key Sizes", *Journal of Cryptography*, vol 14, number 4, pp.255-293, 2001

조 정 식 (Jung-Sik Cho) 정회원



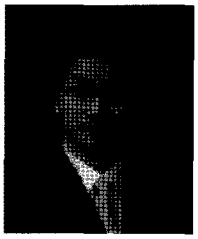
2003년 2월 : 강남대학교 전자계산학과 학사 졸업

2005년 2월 : 중앙대학교 컴퓨터공학과 석사 졸업

2005년 3월~현재 : 중앙대학교 컴퓨터공학과 박사과정

<관심분야> RFID 보안, Sensor network 보안, 암호 응용 및 정보보호

여 상 수 (Sang-Soo Yeo) 정회원



1997년 2월 : 중앙대학교 컴퓨터공학과 공학사

1999년 2월 : 중앙대학교 컴퓨터공학과 공학석사

2005년 8월 : 중앙대학교 컴퓨터공학과 공학박사

2006년 3월~2007년 2월 : 단국대학교 강의전임강사

2007년 3월~현재 : 류슈대학교 정보공학부 방문연구원

<관심분야> RFID 보안, 암호 응용 및 정보보호, 컴퓨터 알고리즘

김 성 권 (Sung-Kwon Kim) 정회원

1981년 2월 : 서울대학교 계산통계학과 학사 졸업

1983년 2월 : 한국과학기술원 전산학과 석사 졸업

1990년 8월 : University of Washington 전산학 박사 졸업

1991년 3월~1996년 2월 : 경성대학교 전산통계학과 조교수.

1996년 3월~현재 : 중앙대학교 컴퓨터공학과 교수

<관심분야> 생물정보학, 계산기하학, 암호응용 및 정보보호