향상된 Yoking Proofs 프로토콜†

조정식*, 여상수**, 김성권*

*중앙대학교 컴퓨터공학부, **단국대학교 정보컴퓨터학부

Enhanced Yoking Proofs Protocol

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

*School of Computer Science & Engineering, Chung-Ang University

**School of Information & Computer Science, Dankook University

요 약

RFID 시스템은 전자태그를 이용한 자동 무선 식별 시스템으로써 RFID 전자 태그를 물체나 사람 또는 동물에게 부착하여 무선 주파수를 통해 태그의 정보를 인식할 수 있도록 해주는 시스템이다. 이는 동시에 다량의 정보를 인식할 수 있다는 장점을 무기로 현재 접촉식 판독 기법의 바코드 시스템을 대처할 수 있을 것이다. 반면 이러한 장점에도 불고하고 RFID 시스템이 사용되는데 걸림돌이 되는 가장 큰 단점은 RFID 태그 정보에 대한 접근이 자유롭다는 점에서 프라이버시 문제를 야기하기 때문이다. 현재 이러한 문제를 해결하기 위해 많은 연구가 진행되고 있으며, 그 중 Ari Juels 는 두 개의 RFID 태그가 동시에 있다는 것을 증명하기 위한 프로토콜인 yoking proof 프로토콜을 제안하였다. 하지만 이는 재생(replay) 공격이 가능하다는 취약점을 가지고 있으며, 이를 보안하기 위해 제안된 여러 프로토콜 들에서도 역시 재생 공격에 대한 취약점이 발견되고 있다. 따라서 본 논문에서는 이러한 yoking proof 프로토콜의 취약점을 보안하기 위하여 공격에 대한 복잡도를 높여 공격자로 하여금 재생 공격이 어렵게 하는 프로토콜을 제안한다.

I. 서론

RFID 태그는 작은 크기의 값싼 마이크로 칩에 저장된 고유한 식별 정보를 짧은 거리의 무선 통신을 통해 전달하는 방식으로써, 현재 물류 대한 인식 시스템인 접촉식 판독 기법의 바코드를 대처할 것으로 예상되고 있다. RFID 시스템 사용으로 인하여 취할 수 있는 장점으로써는 대량의 RFID 태그에 대한 정보를 한번에 인식 할 수 있다는 것과 RFID 기술을 활용하여 제품이 생산단계에서 최종 소비자에게 이르는 모든 과정의 데이터를 실시간으로 파악할 수 있어, 물류량 예측을 통한 물류비용의 절감이 가능하다는 것이다.

RIFD 시스템이 사용되는 응용 분야는 다양하게 존재할 수 있으며, 그중 Ari Juels 는 두 개의 태그가 하나의 리더를 통해 동시에 인식되어야 하는 응용분야를 전제로 이를 증명자에게 증명하는 yoking proof 프로토콜을 제안하였다.[1] 하지만 Ari Juels 가 제안한 프로토콜은 재생 공격이 가능하다는 취약점이 있으면, Saiko 와 Sakurai 는 이를 보안하기 위해 타임스탬프를 사용하는 프로토콜[2]을 제안하였지만 역시 재생 공격에 자유롭지 못했다. 이어 Selwyn Piramuthu 는 증명자로부터 전달받은 임의의 값을 바탕으로한 프로토콜[3]을 제안하였으나 이 역시 공격자의 재생 공격에 노출될 수 있는 가능성이 남아있다. 따라서 본 논문에서는 이러한 재생 공격에 대응할 수 있도록 공격자로 하여금 재생 공격시 높은 복잡도를 요하는 향상된 yoking proof 프로토콜을 제안하고자 한다.

† 본 연구는 한국과학재단 특정기초연구 (R01-2005-000-10568-0) 지원으로 수행되었음

II.관련 연구

Yoking proof 프로토콜의 모티브는 두 개의 태그가 동시에 동일한 리더에게 인식되어야 하는 것에서 출발한다. 이때 리더는 믿을 수 없는 존재며, 리더는 백엔드 서버(Back-end server)에게 두 태그가 동시에 인식되었다는 것을 증명해주어야 한다. 다음 설명되는 프로토콜들은 이를 증명하기 위한 Ari Jules의 프로토콜[1]과 Ari Jules의 프로토콜이 가지고 있는 약점인 재생 공격을 막기 위해 제안된 프로토콜[2][3] 들의 설명과 분석들이다. 다음 표1 은 사용되어지는 표기에 대한 설명이다.

표 1. 표기 설명

표기법	설명
V	증명자(verifier)
r, r_A, r_B	임의의 값(random number)
x_A, x_B	태그 T_A, T_B 각각에 대한 비밀키
MAC	Message Authentication Code
$MAC_x[m]$	비밀키 x 를 통해 만들어진 메시지 m 의 MAC
TS	타임 스탬프(time stamp)
P_{AB}	태그 A와 B 가 동시에 인식되었다는 것을 증명자에게 증명하는 증명내용

1. Yoking proof

Yoking proof 의 구성 요소는 우선 프로토콜에 참가하는 두 개의 태그 T_A, T_B 가 있으며, 이를 동시에 인식하는 리더(reader) 와 리더가 동시에 인식하였다는 것을 증명할 수 있는 증명자(verifier)로 구성된다. 이때 태그 T_A, T_B 는 증명자와 각각의 비밀 값, 즉 비밀키 x_A, x_B를 공유하고 있는 상태며, 리더는 이를 모른다. 프로토콜은 다음 그림1 과 같이 이루어지며, 주어진 시간 t 안에 이루어져야 한다. 자세한 내용은 다음과 같이 이루어진다.

단계 1 : 리더는 태그 A 에게 "left proof" 를 통해 요청한다.

단계 2 : 태그 A 는 요청에 대한 응답으로 임의의 값 r_A 를 리더에게 보낸다.

단계 3 : 리더는 태그 A로부터 받은 r_A를 태그 B 에게 보낸다.

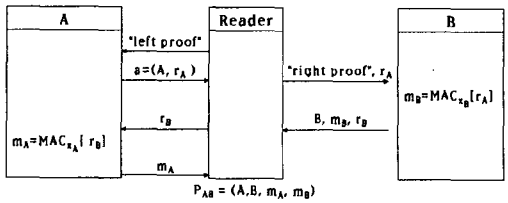

그림 1 yoking proof 프로토콜

단계 4 : 태그 B 는 다음과 같이 비밀 키 x_B를 사용하여 r_A 의 MAC 을 생성한다. 이를 m_B 라 한다.

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

단계 5 : 태그 B 는 p4에서 생성된 m_B 와 r_B를 생성하여 리더에 전달한다.

단계 6 : 리더는 태그 B에게 받은 r_B 를 받아 이를 태그 A에게 전달한다.

단계 7 : 태그 A는 다음과 같이 비밀 키 x_A 를 사용하여 r_B 의 MAC 을 생성한다. 이를 m_A 라 한다.

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

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

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

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

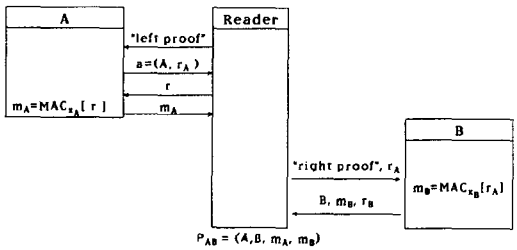

그림 2 yoking proof 의 재생 공격 방법

이와 같은 공격이 가능한 것은 각 태그가 리더에게서 받은 임의의 값으로부터 MAC 을 생성할 때 그 임의의 값이 상대 태그로부터 전달된 정당한 값인지 확인하지 않고 있으며 증명자 또한 이를 확인하지 않고 있다는 것이다.

2. 타임스탬프를 사용한 Yoking proof

Saiko 와 Sakurai 는 이를 보안하기 위하여 다음 그림3 과 같이 타임스탬프를 사용하는 프로토콜을 제안하였다.

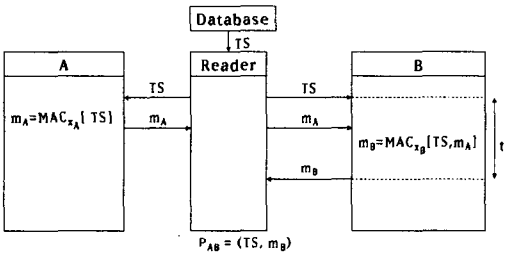

그림 3 타임스탬프를 이용한 yoking proof

Jules 의 yoking proof 와 달리 리더는 데이터베이스로부터 타임스탬프 TS 를 받아 이를 두 태그에게 전달하여 MAC 생성에 사용하고 있다. 이때 태그 A는 TS만을 이용하고 있으며, 태그 B는 TS 와 태그 A가 만든 MAC 을 함께 이용하여 MAC 을 생성하게 된다. 리더는 TS와 태그 B로부터 생성된 MAC 을 받아 이를 증명자에게 제시함으로써 두 태그가 동시에 인식되었다는 것을 증명하게 된다.

역시 이 프로토콜도 주어진 시간 t 안에 이루어져야 하는 제한 조건을 가지고 있다. 하지만 이 방법 역시 다음 그림4와 같은 방법으로 재생 공격이 가능하다.

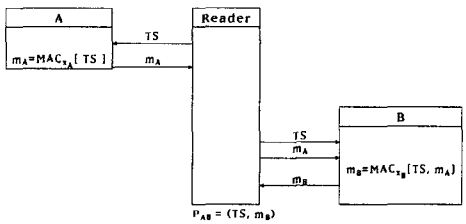

그림 4 타임스탬프를 사용한 yoking proof 에 대한 재생 공격

이는 타임스탬프가 순차적으로 증가하는 성질로 인하여 발생하는 문제점이다. 공격자는 도청과 같은 경로를 통해서 최근의 TS 를 알 수 있고 따라서 공격자는 유효한 범위의 TS 를 미리 구해 태그 A 에게 보냄으로써 m_A 를 수집해 둘 수 있다는 것이 문제다.

3. Modified proof

Selwyn Piramuthu 는 위의 두 프로토콜에서 나타나는 재생 공격을 막기 위해 그림5 와 같이 타임스탬프 대신 증명자로부터 임의의 값 r 을 전달 받는 방법을 사용하였다. r을 전달 받은 태그들은 이를 시드(seed)로 하여 각각 자신의 임의의 값 r_A, r_B 를 생성하는데 사용한다. 각 태그는 이를 바탕으로 MAC 을 생성하게 된다.

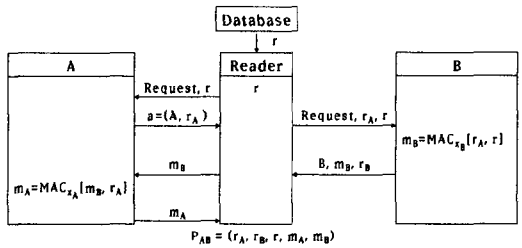

그림 5 Modified proof

이에 대한 공격 방법으로는 다음과 같은 브루트 포스 공격(brute force attack) 방법을 통해 가능하다.

단계 1: 공격자는 구할 수 있는 r 의 값을 모두 구해둔다.

단계 2: 태그 A 에게 가능한 r 에 대한 요청을 보내서 해당하는 r_A 를 수집한다.

단계 3: r 과 주집된 r_A를 태그 B 에게 보내어 m_B와 r_B를 수집해 둘 수 있다.

단계 4: 공격자는 차후 데이터베이스로부터 유효한 r 를 받았을때 해당하는 값들을 바탕으로 재생 공격이 가능하다.

추가적으로 m_A까지 수집해둔다면 두 태그 없이도 증명자에게 두 태그가 동시에 인식되었다고 속일 수 있을 것이다.

III. 향상된 Yoking proof 프로토콜

1. 프로토콜

본 논문이 제안하는 "향상된 yoking proof 프로토콜"은 "modified proof" 에 변형을 가하여 공격을 하기위한 복잡도를 증가시켜 공격에 어려움을 주고 yoking proof의 궁극적인 목적인 두 태그가 동식에 인식되었다는 것에 대한 증명을 제공해 주기 위함에 있다.

우선 제안하는 프로토콜은 다음 그림 6과 같이 데이터베이스로부터 전달 받는 임의의 값을 태그 A를 위한 값 r_1 과 태그 B를 위한 값 r_2 로 각각 나누어 전달해 줌으로써 시작된다.

단계 1 : 리더는 r_1 을 태그 A에게 전달해준다.

단계 2 : 태그 A 는 r_1를 시드로 하여 r_A를 생성하여 리더에게 전달해준다.

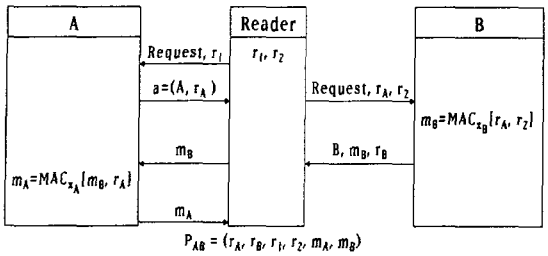

그림 6 향상된 yoking proof 프로토콜

단계 3 : 리더는 r_A 와 r_2를 태그 B 에게 전달해준다.

단계 4 : 태그 B 는 비밀 키 x_B를 사용하여 r_A, r_2 의 MAC, m_B 을 생성한다.

단계 5 : 태그 B는 r_2를 시드로 하여 생성된 r_B와 m_B 를 리더에게 전달한다.

단계 6 : 리더는 m_B를 태그 A 에게 전달한다.

단계 7 : 태그 A 는 비밀 키 x_A를 사용하여 자신의 임의의 값 r_A 와 리더로부터 전달 받은m_B의 MAC, m_A 를 생성하여 리더에게 전달한다.

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

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

2. 안전성 분석

제안 프로토콜 역시 공격자는 브루트 포스 공격을 통해 공격을 할 수 있다. 하지만 다음 표 2를 통해 "modified proof"와 비교해 봤을

표 2. Modified proof 와 Enhanced yoking proof 간 공격에 대한 비용 비교

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

n : random number bit, d : MAC 의 출력 bit

때 통신 횟수 대한 비용이 증가하는 것을 볼 수 있다. r을 수집하는 것과 r_1, r_2 를 수집한 것과의 저장 공간의 차이가 없는 것은 r_1, r_2 둘다 모두 같은 비트를 가지므로 따로 만들 필요가 없기 때문이다. 마찬가지로 각 단계별 저장 공간의 차이는 없다. 단지 해당 값들 간의 링크만이 증가할 뿐이다. 통신 횟수는 r_B, m_B 를 수집하는 단계부터 증가를 나타내고 있다. 이렇게 함으로써 공격자에게 공격에 대한 비용을 증가 시킬 수 있다.

IV. 결론

제안한 프로토콜은 Selwyn Piramuthu의 modified proof 에 간단한 변형을 가하여 공격자로 하여금 공격을 어렵게 하는 방법을 제안하고 있다. 하지만 제안 프로토콜이나 modified proof 모두 공격에 노출된 그 근본적인 이유는 같은 입력에 대하여 같은 결과 값을 출력하는 특성에서 야기된 것이라고 할 수 있다. 향후 과제로는 이러한 근본적인 문제점을 보안할 수 있는 방법에 대하여 연구가 필요하며, 두 개 이상의 태그를 동시에 인식할 수 있는 프로토콜의 대한 연구가 필요하다.

[참고 문헌]

[1]A. Juels, "Yoking Proofs" for RFID Tags, Proceedings of the First International Workshop on Pervasive Computing and Communication Security. IEEE Press. 2004.

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

[3]Piramuthu, Selwyn, On Existence Proofs for Multiple RFID Tags, IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPerU 2006.

[4] G. Avoine and P. Oechslin, RFID Traceability: A Multilayer Problem, Financial Cryptography - FC'05, : LNCS, Springer, 2005