RFID 시스템에서 안전하고 효율적인 프라이버시 보호 기법*

이승구° 여상수 조정식 김성권
중앙대학교 컴퓨터공학과
{leevers°, ssyeo, mfg}@alg.cse.cau.ac.kr, skkim@cau.ac.kr

Secure and Efficient Privacy Protection Scheme in RFID System

Seung-Koo Lee° Sang-Soo Yeo Jung-Sik Cho Sung Kwon Kim
School of Computer Science Engineering, Chung-Ang University

요 약

최근 RFID 시장이 급성장함에 따라 여러 분야에서 주목을 받고 있다. RFID 태그는 기존에 쓰이고 있는 바코드를 대체하는 기술로서, 바코드처럼 정적인 데이터만을 주는 것이 아니라 제품에 관한 여러 가지 정보를 제공할 수 있다. 이는 공급망관리(Supply Chain Management)나 재고 관리 등에 적용하여 시스템을 보다 효율적이고 편리하게 이용할 수 있게 된 것이다. 반면 편리성이 주는 이익은 오히려 프라이버시나 안정성 등 여러 가지 문제를 야기하게 되었다. 이러한 위험성으로부터 사용자 프라이버시 보호와 태그 위조 문제에 대처하기 위한 프로토콜 개발이 필요하다. 이를 위해 본 논문에서는 RFID 프라이버시 보호와 안정성에 대해 논하고, 기존의 프로토콜을 분석한 후 위치추적에 대한 사용자 프라이버시를 보호하는 면과 서버의 효율성을 높이는 면에서 향상된 프로토콜을 제안하기로 한다.

1. 서 론

유비쿼터스(Ubiquitous)라 불리는 미래의 생활환경에서 언제 어디서나 누구라도 간단하고 편리하게 컴퓨터와 네트워크를 이용할 수 있는 사용자 중심의 IT환경이 조성될 것이다. 이러한 컴퓨팅 환경의 발전 방향에서 유비쿼터스 기술 중 가장 부각되고 있는 것이 바로 RFID 통신 기술이다.

RFID는 무선 주파수 인식기술로서 70년대 처음 미군이 탄도미사일 추적을 목적으로 개발된 후 칩 제조와 무선통신 기술 발달로 각 산업분야에 적용되기 시작했다. 초기 RFID 시스템은 비용문제 때문에 널리 사용되지 못했지만, 최근 들어 칩 기술 발달로 인해 태그 가격이 5센트 이하로 떨어지고 메모리의 양이 늘어남에 따라 여러 산업분야에서 각광을 받게 되었다. 바코드 시스템과 비교했을 때, 시장성은 더 많은 가능성을 가질 수 있게 된 것이다.

RFID는 무선 통신을 하는 방식이기 때문에 그에 대한 공격이 일반 네트워크 환경보다 쉽다. 바코드에 비하여 편리성은 향상되었지만 태그의 정보를 누구든지 항상 읽을 수 있다는 점 때문에 정당하지 않은 사용자로부터 불법적인 접근이 가능하다. 몇 가지 공격방법에 대해 살펴보면, 도청은 태그와 리더사이의 통신내용을 엿듣는 방식으로 도청한 내용은 여러 가지 기본정보로 활용될 수 있다. 따라서 공격자가 도청을 통해 얻은 내용을 다른 공격에 활용할 수 없어야 한다. 위치추적은 태그에서 전송되는 정보가 항상 동일하다는 점을 이용해서, 태그 소유자의 위치를 파악하는데 악용되는 것을 의미하며, 사용자 프라이버시를 침해할 가능성이 매우 크다[1]. 무선통신을 하는 태그는 도청자로부터 태그의 정보를 읽어내기가 더 쉽다. 최근 해킹방법은 흔한 도청에서부터 직접 시스템의 내용을 변경하거나 망가뜨리는 공격기술까지 다양한 형태를 띠고 있다. 따라서 태그와 리더 사이의 원활한 통신을 위해서는 태그에 장점을 살리면서 프라이버시 문제를 해결해야 한다.

본 논문에서는 RFID 무선 통신 환경에서의 사용자 프라이버시 보호를 위한 프로토콜을 제안한다.

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

2. RFID 구성요소

RFID는 크게 태그와 리더 그리고 백엔드 시스템 총 3가지로 구성되어 있다.

2.1 태그-리더 시스템

RFID 시스템에서 태그와 리더의 통신에서 능동형(active) 태그와 수동형(passive) 태그로 나누어질 수 있다. 능동형 태그는 태그 내 자체 배터리로 동작하는 형태이고, 수동형 태그는 리더에서 오는 반송파로 에너지를 얻어 동작하는 형태이다. 태그에서 전송된 데이터는 리더에게 전달되고 리더는 태그정보를 확인하기 위해 백엔드 서버 시스템에 태그 정보를 넘긴다. 일반적으로 배터리가 없는 수동형 태그는 능동형 태그에 비해 전송거리가 짧다는 단점이 있다.

2.2 백엔드 서버 시스템

리더로부터 받은 태그 정보와 백엔드 서버 시스템의 데이터베이스에 저장되어 있는 태그 ID 와 비교하여 올바른 태그인지 검사하는 역할을 한다. 일반적으로 백엔드 서버는 위험이 없는 상태라 가정한다.

3. 관련 연구

Saito Junichiro는 리더와 태그 통신 시 전송되는 데이터를 ElGamal 암호화 방식을 이용하여 프라이버시를 보호하는 기법을 제안하였다. 이 기법은 공개키 기반의 ElGamal 암호화 방식을 이용하여 ID 를 재 암호화(Re-Encryption) 함으로써 공격자가 태그의 데이터 내용을 도청하더라도 ID 를 예측 할 수 없도록 하였다. 그러나 이 방식은 공격자가 임의의 값을 태그에 쓸 경우 재 암호화 하더라도 공격자가 쓴 똑같은 값이 그대로 나오기 때문에 태그의 위치를 추적 할 수 있다는 단점이 있다.[2]

Weis, Sarma, Rivest는 태그에서 전송하는 키를 해시하여 생성된 값 $metaID$ 를 전송함으로써 중간에 공격자가 데이터를 가로채더라도 태그 데이터의 내용을 알 수 없도록 하는 프로토콜을 제안하였다. 이 기법은 리더에서 정보 요청 시 태그는 똑같은 데이터만 내보

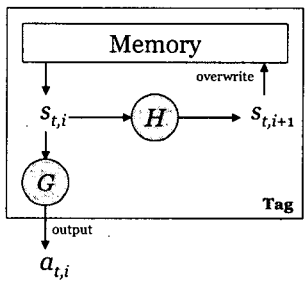
내게 되므로 태그의 위치추적이 가능하다는 단점이 있다. 이러한 단점을 보완하기 위해 Randomized Hash Lock 기법을 제안 하였는데, 이는 임의의 값을 ID 정보와 함께 보내 태그에서 나오는 정보를 공격자가 예측할 수 없게 하여 태그 위치추적을 불가능하게 하였다[3].

ID 를 예측할 수 없는 기법 중 이와 비슷한 기법을 Henrici와 Muller가 제안하였는데, 이는 ID 를 갱신함으로써 위치트래킹 공격을 방지하는 프로토콜이다[4]. 이것의 단점은 데이터베이스에 있는 모든 ID 들이 태그 ID 와 일치하는가를 조사해야 하기 때문에 태그 판별을 위한 계산량이 늘어날 수 있다.

해시 함수를 이용한 또 다른 프라이버시 보호 기법은 해시체인 기법을 이용한 기법이 있다[5]. 우리는 이 논문에서 제안된 프로토콜을 살펴보고 안정성과 효율성에 대해 논하면서 보다 향상된 프로토콜을 제안한다.

3.1 Ohkubo의 프로토콜

기존에 Ohkubo가 제안한 프로토콜에 대해 간략하게 설명한다[2].



[그림 1] 태그 시스템

백엔드 서버에는 비밀 값 $s_{t,1}$ 와 ID 값이 미리 저장되어 있으며 태그에도 동일한 값이 저장되어 있다. 태그는 리더의 요청을 받아 $s_{t,i}$ 에 대하여 G 해시함수 연산을 하여 $a_{t,i}$ 를 계산하고 그 값을 리더에게 보낸다. 그리고 태그는 다시 $s_{t,i}$ 값을 H 해시함수를 통해 $s_{t,i+1}$ 값으로 갱신한 후, 내부 메모리에 바뀐 값으로 다시 쓰게 된다(그림 1).

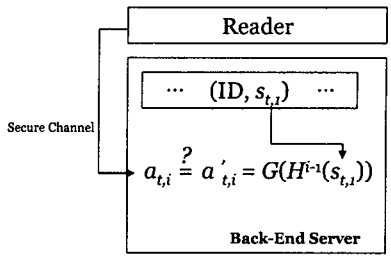
위 과정을 통해 생성된 $a_{t,i}$ 값은 무선 통신을 통해 리더에게 전달되고, 리더는 태그로부터 받은 $a_{t,i}$ 값을 안전한 채널로 백엔드 서버 시스템에게 보낸다. 백엔드 서버는 모든 $s_{t,1}$와 모든 $i\,(1 \le i \le n)$에 대하여 값을 가지고 다음과 같은 식의 (1)번 과정을 수행한다.

$$a'_{t,i} = G(H^{i-1}(s_{t,i})) \quad (1)$$
$$a_{t,i} ? = a'_{t,i} \quad (2)$$

이러한 과정 중에서, 태그에서 온 $a_{t,i}$ 와 백엔드 시스템에서 계산하여 얻어진 결과 $a'_{t,i}$ 값을 비교하여 일치하는 $a_{t,i}$를 찾아낸다. $a'_{t,i}$ 를 찾으면 이를 계산해 낸 $s_{t,1}$ 이 밝혀지고, 데이터베이스로부터 ID 를 알 수 있게 된다(그림 2).

3.2 안전성

태그는 내부에서 비밀 값 $s_{t,i}$ 를 한번 통신할 때 마다 변경하므로 공격자가 태그를 구분하는 것이 쉽지 않고, 태그에서 항상 임의의 변경된 값이 전송되므로 태그의 위치추적에 대해 안전하다. 그리



[그림 2] 리더-백엔드 시스템

고 태그는 $s_{t,i}$ 값에 대해 일방향 해시함수의 특성을 제공하기 때문에 공격자

에 의해 $a_{t,i}$ 를 도청하거나 물리적인 공격에 의해서 $s_{t,i}$가 드러난다 하더라도 $s_{t,i-1}$을 알아내는 것은 불가능하다. 이렇게 현재의 비밀값을 획득하였어도 그전 비밀값을 계산하지 못하게 하여야지만 사용자의 이전 위치에 대한 보안도 유지될 수 있으며, 이러한 성질을 전방위 보안성(forward security)이라고 한다. 또한, 리더에 의해 태그 정보가 변경되지 않으므로 외부 환경에 대해 태그는 안전하다고 할 수 있다.

3.3 효율성

하나의 태그를 판별하기 위해 백엔스 서버 시스템에서는 초기 $s_{t,1}$ 값을 기초로 개개의 태그마다 n 번의 해시연산을 계산을 해야 한다. 그렇게 되면 백엔드 서버가 태그를 판별하기 위한 계산량이 많아지게 된다. 따라서, 태그의 수가 많아지고 해시연산이 많아짐에 따라 백엔드 서버의 태그 판별 시간이 급격하게 길어질 수 있다. Ohkubo 프로토콜의 계산 복잡도는 $O(mn)$이 된다.

4. 제안 프로토콜

우리가 제안하는 프로토콜은 Rabin 공개키 암호화 알고리즘을 이용한 프라이버시 보호 프로토콜이다. Rabin 암호화 방식은 특정 2차 잉여 값의 제곱근을 찾는 문제의 어려움에 기초를 둔 알고리즘이다. 이 암호 방식에서 개인키는 p 와 q 가 되며 이를 모르는 제 3자가 암호문으로부터 평문을 도출하는 것은 결국 법수 r 에 대한 2차 잉여류의 제곱근을 계산하는 문제가 발생하게 되고, 이는 r 을 소인수 분해하는 것과 같은 복잡도를 갖게 되는 형태를 띠게 된다.

Rabin 암호화 알고리즘은 현재 태그에 적용할 수 없지만, 우리는 수 년 내에 Rabin 암호화 알고리즘이 RFID 태그에 적용될 수 있다는 가정 하에 프로토콜을 제시하고 있다.

4.1 기본 구조

t : 태그
m : 태그의 개수
n : 해시체인의 최대 길이
$s_{t,1}$: 태그 t 의 해시함수 초기 값
$s_{t,n}$: 태그 t 의 마지막 해시 값

먼저 백엔드 서버에서는 큰 소수 p 와 q 를 만든다. 이 때 두 수의 곱을 r이라 하면, r은 공개키가 되고 p q는 비밀키가 된다.

$r = pq$ (백엔드 서버의 공개키)

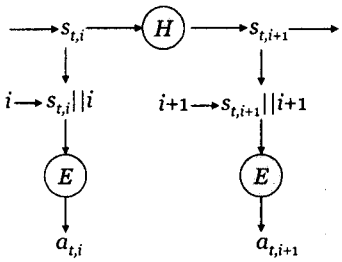
백엔드 서버는 p q 를 비밀 장소에 보관하고, r 은 모든 태그들 내부에 기억시켜 둔다.

4.2 태그에서의 연산

태그의 $s_{t,i}$ 값 갱신은 기존의 Ohkubo 프로토콜처럼 일방향 해시 함수를 이용하고, 리더로 보낼 $a_{t,i}$ 값은 기존의 해시함수가 아닌 Rabin 암호화 작업을 통해 얻는다. 태그는 두 값 $s_{t,i}$ 와 i 를 연결(concatenate)한 후 Rabin 암호화 작업을 수행한다.

$$a_{t,i}=(s_{t,i}||i)^2 \bmod r$$

이렇게 얻어진 태그 정보는 $a_{t,i}$ 가 되고, 이 값은 리더로 보내지게 된다(그림 3).



[그림 3] 제안된 프로토콜의 태그 연산

4.3 백엔드 서버에서의 연산

백엔드 서버에는 모든 태그에 대한 ID 와 $s_{t,1}$ 그리고 $s_{t,n}$ 을 저장한 데이터베이스가 있다(그림 4).

$(ID, s_{t,1}, s_{t,n})$
$(ID, s_{t,1}, s_{t,n})$

[그림 4] 백엔드 서버의 데이터베이스 구조

백엔드 서버는 안전한 채널을 통해 리더로 부터 받은 $a_{t,i}$ 값을 개인키 p q 를 통해 복호화 한다.

$$D(a_{t,i})=s_{t,i}||i$$

복호화 하여 얻어진 값은 바로 $s_{t,i}$ 와 i 값이 되는데, 이 값을 가지고 $n-i$ 번의 H 해시함수 연산을 수행하여 $s_{t,n}$ 을 구할 수 있다.

$$H^{n-i}(s_{t,i})=s_{t,n}$$

해시함수 연산 수행으로 나온 $s_{t,n}$ 을 데이터베이스에 저장되어 있는 모든 태그의 $s_{t,n}$ 값들과 비교하여 해당하는 ID 를 찾는다.

4.4 안전성

제안하는 프로토콜은 일방향 해시함수의 특성을 기초로 전방위 보안성을 제공한다. 공격자가 태그에서 내보내는 정보 $a_{t,i}$ 를 도청하였다 하더라도 Rabin 공개키 암호화가 사용되어 소인수분해 문제의 어려움을 기반으로 하기 때문에 $s_{t,i}$ 를 알 수 없다. 또한 태그의 비밀 값 $s_{t,i}$ 를 물리적 공격으로 획득 한다 하여도, 일방향 해시함수의 특성으로 인해 $s_{t,i}$ 를 가지고 $s_{t,j}(1 \le j < i)$ 의 값을 알아내는 것은 어렵다.

4.5 효율성

Ohkubo의 프로토콜에서는 하나의 태그를 판별하기 위해서는 모든 태그의 $s_{t,1}$ 에 대하여 n 번의 해시함수 연산을 수행해야 하기 때문에, 결국 평균적으로 mn 번의 해시함수 연산을 수행해야 한다. 그러나 우리가 제안하는 프로토콜에서는 태그가 준 $a_{t,i}$ 를 복호화 한 후 나온 값 $s_{t,i}$ 와 i 를 가지고 $s_{t,n}$ 을 바로 구하기 때문에 $(n-i)$ 번의 해시함수 연산을 하면 된다. 우리가 제안하는 프로토콜에서 백엔드 서버의 계산 복잡도는 $O(n)$ 이다.

5. 결론

RFID가 산업전반에 걸쳐 곳곳에 적용되기 시작하면서부터 프라이버시 대한 중요성이 크게 부각되고 있다. 기존의 시스템에 비해 전혀 다른 수준의 공격이 가능해질 수 있기 때문에 여러 가지 공격 가능성에 대해 대처하고 생각해 보아야 한다. 그러한 관점에서 지금까지 우리는 기존에 쓰이는 프로토콜에 대해 간략하게 살펴보았고, 안정성과 효율성에 대해 살펴보고 향상된 프로토콜을 제안하였다.

우리가 제안한 프로토콜의 가장 큰 장점은 Rabin의 암호화 알고리즘의 특징을 이용하여 기존 기법보다 서버의 계산량을 줄였다는 점과, 해시함수의 전방위 보안 보장과 위치추적에 대해 보다 안전한 통신을 할 수 있게 한다는 점이다. 또한 위에서 언급했던 재 암호화 기법[2]과 비교 했을 때, 우리가 제안한 프로토콜은 태그에서의 복호화 작업을 수행하지 않기 때문에 재 암호화 기법보다 적용 가능성이 더 높고 할 수 있다.

비록 현재 Rabin 공개키 암호화 방식을 태그에 적용하기에는 무리라 할지라도 수년 내에 태그의 메모리 확장 기술이 향상된다면 충분히 가능한 프로토콜이라 할 수 있다.

6. 참고 문헌

[1] Sarma, Sanjay and Weis, Stephen and Engels, Daniel, Daniel, "RFID system and Security and Privacy implications", *CHES* 2002

[2] Saito Junichio and Ryou, Jae-Cheol and Sakurai, Kouichi, "Enhancing Privacy of Universal Re-encryption scheme for RFID tags", *EUC*, 2004

[3] S. A. Weis, S. Sarma, R. Rivest, and D. Engels. "Security and privacy aspects of low-cost radio frequency identification systems". In *First International Conference on Security in Pervasive Computing*, 2003.

[4] D. Henrici and P. Muller, "Hash-Based Enhancement of Location Privacy For Radio-Frequency Identification Devices Using Varying Identifiers". *PerSec'04*, pp. 149-153, March 2004.

[5] Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshta, "Cryptographic Approach to "Privacy-Friendly" Tags", *RFID Privacy Workshop. MIT*. 15 November 2003.