

RFID을 위한 해시 체인 기반을 이용한 경량화 상호 인증 프로토콜

이기성*

*호원대학교한대학교 컴퓨터·게임학부
ygslee@howon.ac.kr

A Lightweight Mutual Authentication Protocol based Hash Chain for RFID Systems

Gi-Sung Lee*

*Dept of Computer & Game, Howon University

요 약

이 시스템은 무선 주파수와 RFID 태그 사용으로 불법적인 위변조, 도청, 추적, 프라이버시 침해 등은 불가피하다. 따라서 본 논문에서는 리더와 태그 및 데이터베이스 간에 해시 체인을 이용하여 키를 생성한다. 키 정보를 전송 메시지에 추가하지 않기 때문에 공격자는 키를 획득할 수 없으며 간단한 논리 연산과 카운터만을 사용하기 때문에 저전력 RFID 시스템에 적합하다.

1. 서론

IT 839 전략 중에 하나인 RFID/USN(Radio Frequency Identification/Ubiquitous Sensor Network) 기술은 한 차원 높은 IT 기술 도약을 위해 필수적인 인프라이다. 전파 식별 기술인 RFID(Radio Frequency Identification)는 무선 주파수를 이용하여 리더와 태그 간에 데이터 통신을 하는 ADC(Automatic Data Collection) 기술로서, 빠르고 신속성 있고 노출되어 있지 않고 가려져 있거나 이동 중에 있어도 통신이 가능한 시스템이다[1]. 이는 이미 다양한 분야에서 사용되고 있으며, 특히 유통 및 물류 분야는 자동 인식 기능으로 가시성(Visibility)를 확보하여 전체 공급망 관리의 효율성과 경쟁력을 극대화 시켜줄 수 있는 기술로 부상하고 있다.

RFID 시스템은 3가지로 구성되는데, 리더의 질의에 대하여 사물, 사람 등의 식별 정보를 무선 통신을 사용하여 전송하는 태그(Tag) 또는 트랜스폰더(Transponder)와 태그가 전송하는 데이터를 수신하여 태그를 인식하거나 태그에 새로운 정보를 다시 쓰는 역할을 수행하는 리더(Reader) 또는 트랜시버(Transceiver) 마지막으로 데이터베이스는 태그에 관련된 정보를 저장하고 관리하는 역할을 한다[2].

RFID는 리더와 태그 간에 무선 통신을 사용하기 때문에 사용자의 개인정보 유출 문제 및 군에서 병력의 위치 추적(프라이버시 침해), 재전송 공격과 스푸핑 공격과 같은 위조 공격, 도청과 같은 공격의 위협에 노출될 수 있다. RFID 시스템을 사용하기 전에 위의 공격에 대응해야 한다. 또한 RFID 시스템은 전력과 계산 능력에 제한적이기

때문에 위의 공격을 방지하기 위해 경량화된 암호 알고리즘은 필수적이다.

기존에 제안된 S.Lee가 제안한 동기화된 비밀 정보를 이용하는 상호인증 프로토콜[2]은 공격자가 악의적인 임의의 수(random number)를 전송함으로써 정당한 태그로 위장하여 리더를 속일 수 있는 스푸핑 공격에 취약하다는 것이 Ha 등에 의해 밝혀졌다[3].

Peris-Lopez의 LAMP(Lightweight Mutual Authentication Protocol)[4], M2AP(Minimalist Mutual Authentication Protocol)[5] 및 EMAP(Efficient Mutual Authentication Protocol)[6]은 간단한 논리 연산에 의해 높은 구현 효율성을 제공하도록 설계되었다. 그러나 비동기화공격을 통해 항상 공격자가 비밀 정보를 획득할 수 있는 문제점이 존재한다[7].

제안하는 프로토콜은 RFID 시스템의 제약조건인 계산능력과 제한적인 수명을 고려하여 간단한 논리연산과 해시 체인을 이용하였으며 기존의 여러 공격에 안전하도록 프로토콜을 설계하였다.

본 논문의 구성은 다음과 같다. 2절에서는 관련연구에 대해 기술하고, 3절에서는 제안하는 프로토콜에 대해 설명한다. 4절에서는 프로토콜의 효율성 및 안전성을 검증한다. 마지막으로 결론과 향후 관련연구 방향에 대해 제시한다.

2. 관련연구

1.1. Choi 외 4명[2]

이 프로토콜에서 back-end 서버인 데이터베이스와 태그는

미리 T와 B간의 동기화된 비밀정보 또는 비밀키인 k 를 공유하고 있다. 또한 데이터베이스는 각 태그들의 비밀 정보를 저장하는 k 필드를 가지고 있으며 이전 세션의 비밀 정보를 저장하는 k_{last} 필드를 가지고 있다. 프로토콜은 아래와 같이 4단계로 구성된다.

먼저 첫 번째 단계에서 리더는 Query와 첫 번째 랜덤수인 r_R^1 을 태그에 보낸다. 두 번째 단계에서 태그는 메시지를 수신한 후에 r_T^1 를 생성하고 r_T^1 와 최초 키 값인 k^1 및 수신한 r_R^1 을 해시한 $P = h(r_R^1 \oplus k^1 \oplus r_T^1)$ 값을 생성하고 리더에게 P 값과 r_R^1 값을 전송한다. 리더는 다음단계로 이 메시지에 r_R^1 을 추가하여 데이터베이스에 전송한다. 3번째 단계에서 데이터베이스는 태그를 인증하기 위해 P 의 정당성을 다음과 같이 체크한다.

$$P = ?h(r_T^1 \oplus k^1 \oplus r_R^1) \quad (1)$$

이때 r_T^1 와 r_R^1 은 수신한 파라미터이며 k^1 는 데이터베이스 필드에 저장한 키 값이다. 인증 단계가 끝나면 데이터베이스는 리더를 경유하여 태그에게 $Q = h(P \oplus k^1 \oplus r_R^1)$ 를 전송하고 데이터베이스를 갱신한다. Q 를 수신한 태그는 데이터베이스를 검증하기 위해 다음과 같이 확인한다.

$$Q = ?h(P \oplus k^1 \oplus r_R^1) \quad (2)$$

검증 후, 태그는 데이터베이스와 같이 키 값 $k^2 = h(k^1)$ 를 갱신한다.

이 프로토콜에서 공격자는 스푸핑 공격이 가능하다. 먼저 공격자는 리더와 태그간의 메시지에서 r_R^1 와 P 및 r_T^1 를 도청한다. 그런 후에 다음 세션에서 리더가 태그에게 Query를 전송하면 공격자는 r_R^2 를 도청하고 $r_T^2 = r_R^1 \oplus r_R^2 \oplus r_T^1$ 와 $P' = P$ 를 생성하여 리더에게 전송하면 리더는 r_R^2 를 포함한 메시지를 안전한 채널을 통해 데이터베이스에게 전송한다. 문제는 이때 발생하는데 데이터베이스가 태그 검증을 위해 r_T^2, r_R^2 와 P' 만을 이용하기 때문이다. 다시 말해서 다음과 같이 스푸핑 공격이 가능하다[3].

$$\begin{aligned} P' &= h(r_T^2 \oplus k^1 \oplus r_R^2) \\ &= h(r_T^1 \oplus r_R^1 \oplus r_R^2 \oplus k^1 \oplus r_R^2) \\ &= h(r_T^1 \oplus k^1 \oplus r_R^1) \\ &= P \end{aligned} \quad (3)$$

더불어 이전 세션에서 데이터베이스의 k^1 값이 갱신되었고 해도 데이터베이스의 k_{last} 필드에 k^1 값이 있을 것이기 때문에 스푸핑 공격은 불가피하다. 즉, 데이터베이스에 이전 세션의 키 값을 저장하지 말아야 하며, 매번 키를 계산하는 것이 힘들어야 한다. 또한 \oplus 연산 때문에 생기는 위장 공격을 피해야 한다.

2.2 Peris-Lopez의 3명[4,5,6]

Peris-Lopez가 제안한 LMAP(Lightweight Mutual Authentication Protocol), M²AP(Minimalist Mutual

Authentication Protocol) 및 EMAP(Efficient Mutual Authentication Protocol)은 고정된 ID($m=96$ bits), 태그 구별을 위해 사용되며 유효 세션 후 갱신되는 index-pseudonym(IDS), 인증을 위한 4종의 96bit 비밀키 K_1, K_2, K_3, K_4 , 인증 및 키 갱신에서 $+, \wedge, \vee, \oplus$ 의 연산만 사용한다는 특징을 가지고 있어 경량 및 고속 구현에 적합하다.

리더가 태그를 인증하는 경우, 리더는 난수 n_2 값을 이용하여 $IDS^{(n)}$ 와 비밀키를 갱신하고 태그는 $n_2 \oplus I_j$ 를 이용하여 $IDS^{(n)}$ 과 비밀키를 갱신하므로 양측 사이의 동기가 깨지게 된다.

3. 제안하는 프로토콜

3.1 표기법

본 프로토콜에서 사용하는 표기법은 아래와 같다.

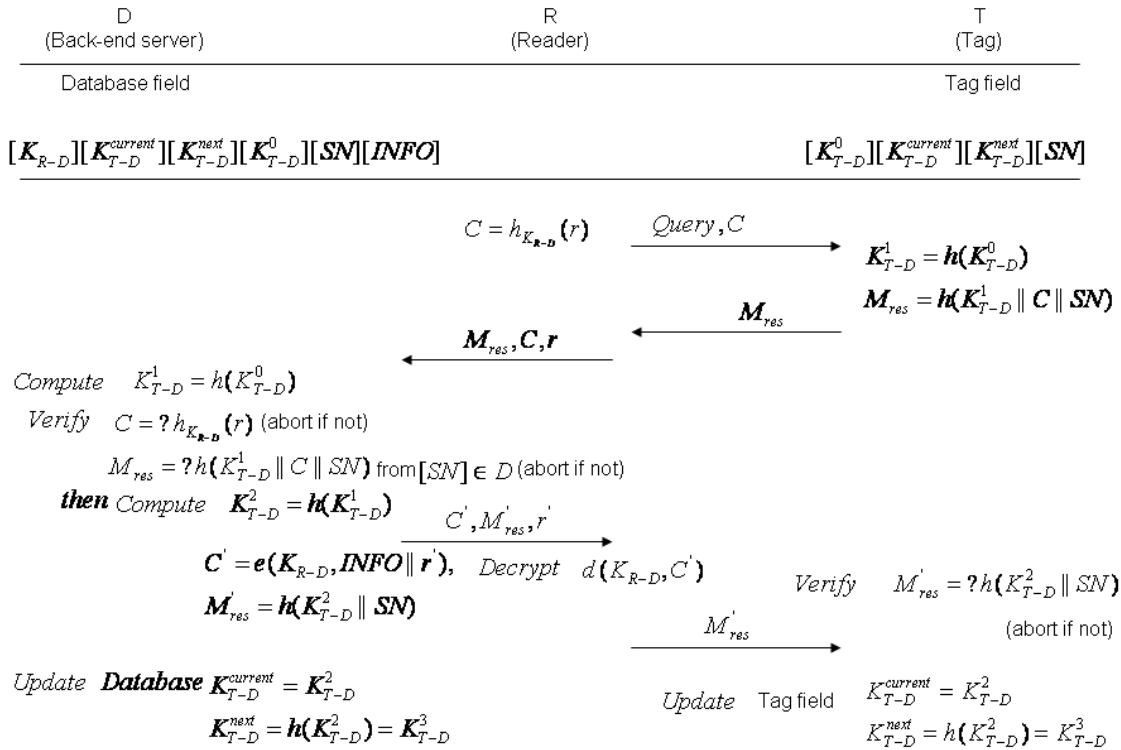
- T : RFID 태그 또는 transponder.
- R : RFID 리더 또는 transceiver.
- D : 데이터베이스를 가진 Back-end 서버.
- SN : 태그의 serial number.
- $e(k, m)$: 비밀키 k 를 이용하여 메시지 m 을 암호화하는 대칭키 암호 함수.
- $d(k, m)$: 비밀키 k 를 이용하여 메시지 m 을 복호화하는 대칭키 암호 함수.
- $h()$: 일방향 해시 함수.
- $h(k, m)$: 비밀키 k 를 이용한 해시 함수.
- k_{R-D}^i : 리더와 데이터베이스 i 번째 비밀키.
- k_{T-D}^i : 태그와 데이터베이스 간의 i 번째 비밀키.
- r : 랜덤 수.
- $m_1 \parallel m_2$: 메시지 m_1 과 m_2 의 비트 결합.

3.2 가정

제안하는 프로토콜에서 태그는 수동적인 태그로서, 해시 함수와 하나의 세션동안 상태를 유지할 수 있는 능력을 가지고 있다. 위에서 소개한 프로토콜과 달리 리더와 태그 및 리더와 데이터베이스 구간은 무선 네트워크로서 안전하지 않은 채널이라고 가정한다. 리더와 데이터베이스 사이에는 비밀 값 k_{R-D}^0 을 태그와 데이터베이스 사이에는 해시 체인 초기값 k_{T-D}^0 값을 공유하고 있다고 가정한다. 제안하는 프로토콜에서 RFID 태그에 대한 물리적인 공격은 고려하지 않는다.

3.3 프로토콜

위에서 분석한 내용을 토대로 저전력 RFID 시스템에 적합한 경량화된 해시 체인 기반의 상호 인증 프로토콜을 그림 1과 같다.



[그림 1] 제안하는 상호 인증 프로토콜

단계 1: 먼저 리더 R은 난스 r를 생성한 후, 데이터베이스 D와 공유하고 있는 비밀키 K_{R-D} 를 이용하여 r을 해시한다. 이때, 해시함수는 키를 이용한 일방향 해시 함수이다. R은 $C = h_{K_{R-D}}(r)$ 을 포함하는 질의 Query를 태그 T에게 전송한다. 이때, K_{R-D} 는 D에서 R를 인증하기 위해 사용되며 C를 통해 능동 공격자에 대한 중간자 공격(man-in-the middle attack)을 방지할 수 있다.

단계 2: C를 수신한 태그 T는 D와 공유한 해시 체인 초기 값 K_{T-D}^0 을 이용하여 $K_{T-D}^1 = h(K_{T-D}^0)$ 을 계산한다. 그런 후에 $M_{res} = h(K_{T-D}^1 || C || SN)$ 을 계산하고 R에게 전송한다. M_{res} 는 D에게 합법적인 R이라는 것을 확인시키고 능동적 공격인 도청으로부터 M_{res} 를 위조 못하게 하기 위함이다.

단계 3: 이 메시지를 수신한 R는 C와 r를 추가하여 전송한다. r은 중간자 공격을 방지하고 C와 함께 D에서 합법적인 R로부터 온 메시지인지를 검증하기 위해 사용된다. 데이터베이스 D에서는 먼저 정당한 R인지 수신한 S와 r을 이용하여 $C = ? h_{K_{R-D}}(r)$ 을 계산한다. 이때 C값이 올바르게 없을 경우에는 프로토콜을 중지한다. 동시에 D는 K_{T-D}^0 와 SN을 데이터베이스로부터 추출한다. 그런 후에 $K_{T-D}^1 = h(K_{T-D}^0)$ 를 계산하고 $M_{res} = ? h(K_{T-D}^1 || C || SN)$ 을 검증한다. D는 R과 T의 정당성을 확인하고 $K_{T-D}^2 = h(K_{T-D}^1)$ 을 계산한다. 합법적인 R이라는 것이 검증되면 K_{R-D} 를 이용하여 태그의 정보 INFO를 암호화해서 R에게 전송한다. 이는 중간자 공격과

도청을 방지하기 위해 비밀키를 이용하여 암호화한다. 또한 D는 T와 상호인증을 위해 $M'_{res} = h(K_{T-D}^2 || SN)$ 을 전송한다. 마지막으로 D는 현재 업데이트 된 T와의 비밀키 K_{T-D}^2 를 데이터베이스 $[K_{T-D}^{current}]$ 필드에 저장하고 다음 세션에 사용할 키를 미리 해시 체인을 통해 계산한 후 $[K_{T-D}^{next}]$ 필드에 저장해 둔다.

단계 4: R은 수신 메시지 중 C' 값을 복호화하고 태그 정보 INFO를 획득한다. 이때 복호화가 되지 않을 경우에는 프로토콜을 중지한다. r' 는 중간자 공격을 방지하기 위해 포함된 파라미터이다. 그런 후에, R은 T에게 M'_{res} 메시지를 전송한다.

단계 5: T는 M'_{res} 메시지의 정당성을 검증하기 위해 $M'_{res} = ? h(K_{T-D}^2 || SN)$ 을 계산한다. 이를 통해 D를 인증할 수 있다. 그런 후에 D와의 비밀키를 갱신하기 위해 $K_{T-D}^{current} = h(K_{T-D}^2) = K_{T-D}^3$ 을 계산한 후 자신의 메모리 필드 $[K_{T-D}^{current}]$ 에 저장한다.

4. 성능분석

4.1 안전성 분석

능동적인 공격자에 의한 도청 가능 메시지는 리더 R이 태그 T에게 전송하는 C값과 T가 R에게 응답하는 M_{res} , R이 D에게 전송하는 r, 기타 M'_{res}, C', r' 가 있다. 그러나

제안하는 프로토콜에서 리더 R 이 발생하는 랜덤수를 제외한 모든 정보는 일방향 해시함수나 암호화 연산을 거쳐 전송되는 정보이므로 공격자가 전송 메시지를 도청하더라도 비밀키 값이나 태그 정보는 획득할 수 없기 때문에 안전하다.

다음으로 공격자가 합법적인 리더, 태그나 데이터베이스로 위장하여 상대방을 속이거나 유용한 정보를 획득하려는 스푸핑 공격이 있다. 제안하는 프로토콜에서 공격자는 합법적인 리더로 가장하여 태그를 속이기 위해서는 M'_{res} 값을 계산해야 한다. 그러나 공격자는 데이터베이스 D 와 T 가 공유하고 있는 비밀키 K_{T-D}^2 를 계산할 수 없기 때문에 안전하다. 또한, 태그로 위장하고자 할 경우, 공격자는 R 이 생성한 C 값은 도청할 수 있지만 M_{res} 값과 태그의 SN 은 계산해야 할 수 없기 때문에 스푸핑 공격에 안전하다. 마지막으로 비동기화 공격은 공격자가 메시지를 블로킹하여 정상적인 인증 과정을 방해하는 공격으로 태그는 최소한 이전 세션에서 사용된 비밀 정보를 저장하고 있어야 한다. 제안하는 프로토콜에서는 데이터베이스 필드와 태그 필드에 모두 $K_{T-D}^{current}$ 와 K_{T-D}^{xt} 를 가지고 있기 때문에 공격자가 메시지를 블로킹하여도 동기를 회복할 수 있다. 마지막으로 기존 프로토콜[2,3]과 달리 모든 구간을 무선 구간으로 간주하고 프로토콜을 설계했기 때문에 더욱 현실적이고 강건할 수 있다.

4.2 효율성 분석

본 프로토콜에서 프로토콜의 효율성 분석을 위해 태그와 데이터베이스간의 계산량과 데이터 저장 공간만을 고려한다. [4-6]번의 프로토콜은 단순한 논리연산만을 이용하였기 때문에 효율성을 높일 수 있지만 안전성 측면에서는 취약하다. [3]의 경우에서 I 프로토콜은 DB 에 저장된 ID 수가 m 이라고 할 때 $\left\lceil \frac{m}{2} \right\rceil + 2$ 의 DB 해시 연산이 필요하며 비동기화 발생시 동기회복을 위해서는 평균 $m + \left\lceil \frac{m}{2} \right\rceil + 2$ 번의 DB 해시 연산이 필요하고 II의 경우에는 정상상태에서 3번의 해시연산과 비동기화가 발생시에는 다음 세션에서 정당한 태그를 찾아 동기를 회복하는데 평균 $\left\lceil \frac{m}{2} \right\rceil \cdot 2 + 3$ 번의 해시 연산이 필요하다. 제안하는 프로토콜에서는 동기화시에는 태그에서 4번의 해시 연산과 DB 에서 6번의 해시 연산과 1번의 암호화 과정이 수행된다. 비동기화일 경우에 DB 에서는 $\left\lceil \frac{m}{2} \right\rceil \cdot 2 + 3$ 번의 해시 연산과 1번의 암호화 과정이 수행된다. 비교적 [3]에서 제안한 프로토콜보다 약간의 계산량(암호화 연산)이 추가되지만 DB 에서는 별 무리가 되지 않는다. 단, 검색 시간이 약간 지연된다는 단점이 있다. 마지막으로 필수 저장 공간을 비교해보면 [3]의 경우 태그에서 SN 을 저장

하기 위해서는 $2i$ (모든 정보들은 i 비트로 구성되어 있다고 가정)비트의 저장 공간이 필요하고 데이터베이스에서는 $3i \cdot m$ 의 저장 공간이 필요하다. 그러나 II의 경우에는 k_{xt} 를 저장해야 하므로 태그당 i 비트씩 늘어나며 태그에서는 $SYNC$ 정보를 저장해야 하므로 1비트씩 증가한다. 제안하는 프로토콜에서 태그는 4비트의 저장 공간이 필요하고 데이터베이스에서는 $6i \cdot m$ 의 저장 공간이 필요하다. 데이터베이스의 경우 메모리에 제약이 태그보다 유연하기 때문에 가능하며 태그의 메모리 공간은 [3]과 비슷하다.

5. 결론

본 논문에서는 모든 구간의 안전성을 고려한 RFID 상호인증 프로토콜을 제안했다. 먼저 무선 통신상에서 쉽게 발생할 수 있는 도청과 스푸핑 공격에 강건함을 보였으며 비동기화공격 역시 태그와 데이터베이스가 동일한 비밀키를 저장하면서 동기화를 맞추기 때문에 공격자가 메시지를 블로킹 하여도 동기를 회복할 수 있었다. 따라서 대용량 보다는 저용량 RFID 시스템이면서 보안이 더욱 필수적인 군사목적 RFID 시스템에 적합하다. 단, DB 에서 태그 정보 검색에 대한 지연 시간을 줄이는 것은 향후 연구 과제로서 더 연구해야 할 부분이다.

참고문헌

- [1] 이병길, 강유성, 박남제, 최두호, 김호원, 정교일, “능동 및 모바일 RFID 서비스 환경에서의 정보보호 기술”, 한국정보보호학회 학회집, 제 15권, 제3호, pp. 40-47, 2005
- [2] 최은영, 이수미, 임종인, 이동훈, “분산시스템 환경에 적합한 효율적인 RFID 인증 시스템”, 한국정보보호학회논문집, 제16권, 제6호, pp.25-35, 2006.
- [3] 하재철, 김환구, 하정훈, 박제훈, 문상재, “비밀정보 동기화에 기반한 Strong RFID 인증 프로토콜”, 한국정보보호학회 논문집, 제17권, 제5호, pp.99-109, 2007.
- [4] P. peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapaidor, and Ribagorda, “LAMP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags,” Workshop on RFID Security 2006, pp.137-148, 2006.
- [5] P. peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapaidor, and Ribagorda, “M²AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID tags,” Proceedings of UIC 2006, pp.912-923, 2006.
- [6] P. peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapaidor, and Ribagorda, “EMAP: An Efficient Mutual-Authentication Protocol for Low-cost RFID tags,” Proceedings of OTM Federated Conferences and Workshop 2006, pp.352-361, 2006.
- [7] 권대성, 이주영, 구분옥, “경량 RFID 상호인증 프로토콜 LMAP, M²AP, EMAP 대한 향상된 취약성 분석”, 한국정보보호학회 논문집, 제 17권, 제 4호, pp.104-113, 2007.