전방위 안전성이 향상된 RFID 인증 프로토콜 제안

김헌욱*, 조종근*, 이장춘*, 이은유*, 임수윤*, 이훈재**
*동서대학교 디자인 & IT 전문대학원
**동서대학교 컴퓨터정보공학부
e-mail:hunwookkim@paran.com

A RFID Authentication Protocol with a Strong Forward Security

Hun-Wook Kim*, Jong-Kuen Jo*, Jang-Chun Lee*,
Ern-Yu Lee*, Shu-Yun Lim*, Hoon-Jae Lee**
*Graduate School of Design & IT, Dongseo University
**Div. of Computer & Information Eng., Dongseo University

요 약

RFID(Radio Frequency Identification) 기술의 발전은 편의성과 경제성으로 전 세계에 확산되어 사용되고 있다. 비록 세계적인 보안 표준안이 정해 지지 않았지만 RFID 보안 분야는 각 나라에 맞게 표준을 선정하여 적용되고 있다. 한편, RFID 기술의 발전으로 RFID의 칩이 소형화되고, 장거리 통신이 가능해짐에 따라 RFID 시스템에 내장된 정보를 다른 누군가가 활용하여 개인 또는 기업 등의 프라이버시를 불법으로 수집하는 피해가 발생할 수 있다. 본 논문은 RFID 시스템에서 정보의 누출을 막기 위해 해쉬 알고리즘과 초경량 스트림 암호알고리즘을 사용하여 전방위 안전성이 향상된 RFID시스템을 제안, 분석하였다.

1. 서론

RFID 기술의 발전은 편의성과 경제성으로 전 세계에 확산되어 사용되고 있다. 비록 아직 세계적인 표준안이 정해지지는 않았지만 RFID 보안 분야는 각 나라에 맞게 표준을 선정하여 사용하고 있다. 한편, RFID 기술의 발전으로 RFID 핵심 칩은 소형화되었고, 통신거리는 더 길어지게 되었다. 칩의 소형화로 동전 크기의 리더와 점 크기의 태그가 개발되었고, 넓어진 통신 범위로 누구나 언제 어디서든 태그의 정보를 읽을 수 있고, 위장 태그를 만들 수 있게 되었다. 개인 또는 기업 등의 RFID 시스템을 사용하는 모든 곳에 프라이버시 침해라는 문제에 노출되어있다.

그래서 최근 프라이버시 침해 문제를 해결하기 위하여 프라이버시를 보호하기 위한 많은 아이디어들이 개발되고 있다. 그러나 RFID태그 자원의 한계 때문에 DES[1], AES[2], SHA-1[3] 등의 암호 알고리즘은 사용이 힘든 실정이다. M.Feldhofer et al[4]가 고안한 RFID 적용 AES 알고리즘을 제시하였지만 128비트를 암호화하기 위해서는 약 1,000 클럭의 많은 클럭이 요구된다.

※본 연구는 정보통신부지원 대학 IT 연구센터 육성지원사업에 의하여 수행되었습니다.

<표 1> 각 표준별 최소 TagID 길이

구 분	ISO18000-7	EPC	ucode
ID길이(bit)	48	64	128
전방위 안전성	2^{48}	2^{64}	2^{128}

한편 기존에 제안된 RFID인증 프로토콜의 문제점[5][6]인, 해쉬 알고리즘을 사용시 짧은 Tag ID (<표 1> 참조)에 대해서는 전방위 안전성(Forward Security)이 보장되지 않는점 및 Collision 문제[7][8]가 있다. 본 논문에서는 보완하기 위하여 초경량 스트림암호와 기존에 제안된 프로토콜에서 사용된 해쉬 알고리즘을 사용하여 최대한 적은 자원에서 구현이 가능하도록 하였고, RFID 인증 프로토콜 설계시 고려할 사항[9][10][11]을 반영하여 성능 향상된 프로토콜을 제안한다.

2. 제안 프로토콜

본 논문에서는 리더와 데이터베이스 서버는 안전한 통신을 하며, 리더와 태그간의 통신은 안전하지 안다고 가정한다. 제안된 프로토콜의 특징은 짧은 ID 길이에 관계없이 전방위 보안에 강하고 해쉬 알고리즘의 충돌공격을 방지하며 데이터베이스의 성능이 뛰어나다.

전방위 안전성은 기존 해쉬 알고리즘을 사용하는 인증 프로토콜에서는 2^{80}의 복잡도를 유지하기 위해서는 80bit 이상의 ID를 사용하는 시스템에서만 사용이 가능하다. 본 논문에 제안된 프로토콜은 ID는 128bit의 키를 가지는 초경량 스트림 암호에서 생성되는 키 수열에 의해서 암호화되고, T 값은 128bit 키를 해쉬 알고리즘을 사용하므로 전방위 보안에 강하다.

해쉬 알고리즘의 충돌공격방법[6]은 다른 평문을 해쉬 알고리즘에 입력하여 동일한 암호문을 만들어 내는 방법으로 위치 트래킹 공격을 막기 위해 ID를 변경하는 인증 프로토콜에서는 데이터베이스에 중복된 ID를 가지게 하여 시스템에 오류를 유발 시킬 수 있다.

데이터베이스의 성능에서는 A_t의 값을 flag의 값에 따라 CE, LE에서 검색하여 중복된 값이 있을 경우 중복된 값의 Ckey, Lkey으로 해쉬 알고리즘을 사용하여 T와 비교를 하므로 데이터베이스의 성능이 뛰어나다.

2.1 시스템 계수

- ID : 태그의 고유 값
- E : 초경량 스트림 암호 알고리즘
- h() : 해쉬 알고리즘
- Ckey : 데이터베이스에 저장된 128bit 길이의 초경량 스트림암호 key, 현재의 key값을 저장
- Lkey : 데이터베이스에 저장된 128bit 길이의 초경량 스트림암호 key, 이전의 key값을 저장
- RR() : Right rotate 함수
- LR() : Left rotate 함수
- CE : 데이터베이스에 저장된 $E_{Ckey}(ID_{RR(1)})$ 값
- LE : 데이터베이스에 저장된 $E_{Ckey}(ID_{RR(2)})$ 값
- key : 태그가 보유한 key 값
- flag : 데이터베이스와 태그의 세션 상태를 나타내는 1bit 값 초기값 0, 정상 0, 비정상 1
- cnt : 카운트 값으로 리더로부터 Q,S를 받으면 1씩 증가
- S : 리더에서 생성되는 랜덤수열발생기(PRNG pseudo random number generator)
- R : 1 ~ (ID_{length} -1) 값
- A : 초경량 스트림 암호로 암호화된 ID

2.2 사전 준비단계

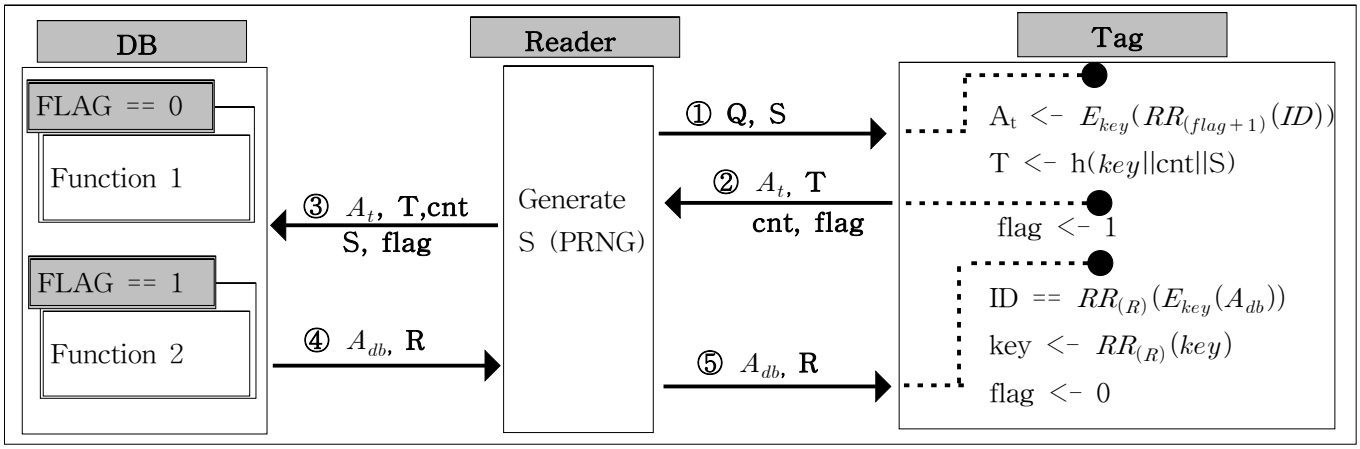
태그를 발행할 때 태그의 ID, key를 데이터베이스의 ID, Ckey에 저장하고 Ckey와 ID를 이용하여 CE를 미리 계산하여 저장한다. 그리고 태그의 flag와 cnt를 0으로 저장한다.

2.3 인증과정

인증과정에 대한 설명은 그림 1의 데이터베이스, 리더와 태그의 인증과정에서 보내는 데이터를 기준으로 하여 설명한다.

flag가 0인 경우(그림 1의 Function 1) 인증과정을 보면,

(1) 리더가 PRNG S와 Q(Query)인 ①의 데이터를 태그에서 전송한다. 이때 S는 매번 다른 값이 나온다고 가정한다.



(그림 1) 제안프로토콜 인증과정

(2) ①의 데이터를 받은 태그는 S값과 태그의 정보를 이용하여 A_t <- $E_{key}(RR_{(flag+1)}(ID))$, T <- h($key$||cnt||S)를 계산하여 리더에게 ②를 전송하고 flag <- 1로 저장한다.

(3) 태그로부터 ②를 받은 리더는 ②의 정보에 S를 추가하여 데이터베이스로 ③을 전송한다.

(4) ③의 데이터를 받은 데이터베이스는 CE필드에서 A_t를 검색한다. A_t값은 아주 랜덤한 값이기 때문에 중복된 값이 데이터베이스에 존재할 수 있다. A_t로 검색된 값이 0 이상이면 검색된 필드의 정보를 이용하여 h($Ckey$||cnt||S)를 만들고 T와 비교한다. 이때 h($Ckey$||cnt||S) == T 값이 해쉬 알고리즘 충돌 문제로 중복되더라도 이후에 ID를 검사하므로 문제가 없다. 검색된 필드의 Ckey를 이용하여 A_t를 복호화하여 데이터베이스의 ID와 복화화된 ID가 같은지 비교한다. ID가 같으면 데이터베이스는 LE <-$E_{Ckey}(ID_{RR(2)})$, Lkey <- Ckey로 저장하고 R을 생성하여 ID를 다시 암호화하고, Ckey <- $RR_R(Ckey)$, CE <- $E_{Ckey}(RR_{(1)}(ID))$로 저장하고 ④를 전송한다.

(5) 리더는 데이터베이스로부터 받은 ④를 태그에게 전송한다.

(6) ⑤의 데이터를 받은 태그는 ID == $RR_{(R)}(E_{key}(A_{db}))$를 비교하여 같으면 key <- $RR_{(R)}(key)$, flag <- 0 으로 업데이트하고 인증은 완료된다. flag가 1인 경우(그림 1의 Function 2) 인증과정을 사용하며 flag 0일때와 동일한 통신 구조를 가진다.

```
if flag == 0
 Search CE == A_t
 if CE.count > 0
 {
  for(i=0 ; i <= CE.count ; i++)
   if h(Ckey_i||cnt||S) == T
    ID = RL_(flag+1)(E_Ckey(A_t))
    if ID_db == ID
     LE <-E_Ckey(RR_2(ID))
     Lkey <- Ckey
     R = random(1~ID_length -1)
     A_db <- E_Ckey(ID_RL(R))
     Ckey <- RR_R(Ckey)
     CE <- E_Ckey(RR_(1)(ID))
     Send A_db, R
    else
     halt
   else
    halt
 }
 else
  halt
```

(그림 2) Function 1

```
else flag ==1
 Search LE == A_t
 if LE.count > 0
 {
  for(i=0 ; i <= LE.count; i++)
   if h(Lkey_i||cnt||S) == T
    ID = RL_(flag+1)(E_Lkey(A_t))
    if ID_db == ID
     R = random(1~ID_length -1)
     A_db <- E_Lkey(RL_R(ID))
     Ckey <- RR_R(Lkey)
     CE <- E_Lkey(RR_(1)(ID))
     Send A_db, R
    else
     halt
   else
    halt
 }
 else
  halt
else
 halt
```

(그림 3) Function 2

3. 제안 프로토콜 분석

1장에서 언급한 인증 프로토콜 설계시 고려사항과 해쉬 함수의 문제점에 대해 제안 프로토콜을 분석한다.

· 도청(Eavesdropping) : 공격자는 태그와 리더사이의 통신내용을 쉽게 도청하여, Q,S,A_t,T,cnt,flag, A_{db},R을 알 수 있지만 초경량 스트림 암호 알고리즘으로 암호화되는 A_t, 해쉬 알고리즘를 사용하여 만든 T로부터 ID, key 등의 어떠한 정보도 알아낼 수 없다.

· 트래픽 분석(Traffic Analysis) : 공격자는 도청을 통하여 얻은 정보를 이용하여 다른 공격 수단으로 사용하려 하지만 태그는 리더의 질의에 언제나 다른 T를 전송하므로 트래픽 분석에 안전하다.

· 위치트래킹(Location Tracking) : 공격자는 태그의 정보를 이용하여 위치트래킹 공격이 가능한데, 제안된 프로토콜에서는 태그가 리더로 정보를 보낼 때 마다 다른 정보를 보냄으로 위치트래킹 공격에 안전하다. A_t는 ID를 암호화한 값인데 이 값은 인증 완료 후 매번 다른 key에 의해 암호화되어 태그가 전송하는 값은 언제나 랜덤한 값을 가진다. T는 key, cnt, S를 해쉬 알고리즘를 사용하여 만들어진 값인데 이 값은 S와 cnt가 매번 바뀌기 때문에 항상 랜덤한 값을 출력한다.

· 스푸핑(Spoofing) : 공격자는 리더와 데이터베이스를 속이기 위해 태그로 위장 또는 태그를 속이기 위하여 리더로 위장하여야 메시지를 보내야 한다. 태그로 위장한 경우, 그림 1의 ②,⑤의 데이터를 가

로채어 수정하여 보내야 하는데 S는 매번 다른 값으로 전송되므로 공격자는 key를 모르는 상태에서 T를 만들어 낼 수 없음으로 태그로 위장할 수 없다. 리더로 위장한 경우, 그림 1의 ①,②,⑤ 데이터를 수집하여 태그에게 수집한 정보와 같은 S를 전송하여 A_t, T, cnt, flag를 얻어 내지만 이 정보로부터 A_{db}와 R을 만들 수 없다.

· 메시지 유실(Message loss) : 공격자가 태그와 데이터베이스간의 key 값의 동기화를 방해하여 태그와 데이터베이스간의 key가 달라지게 하기 위한 공격으로 공격 후에 해당 태그는 사용할 수 없게 된다. 데이터베이스와 태그의 동기화를 위해 flag를 사용하였고, flag 값에 따라 다른 key를 적용하여 메시지가 유실되더라도 동기화가 가능하도록 하였다.

4. 결론

본 논문에서는 초경량 스트림 암호 알고리즘과 해쉬 알고리즘을 사용하여 비교적 적은 자원으로 구현이 가능한 RFID 인증 프로토콜을 제안하였다. 해쉬 알고리즘의 충돌공격 문제를 해결하기 위해 ID는 변경되지 않으며 짧은 ID를 해쉬 알고리즘을 암호화한 값을 이용하여 통신할 경우 공격자가 ID를 쉽게 계산해 낼 수 있는 문제점을 보완하기 위해 ID를 스트림 암호 알고리즘으로 암호화되어 통신되도록 하여 전방향 안전성을 유지했다. 128bit 키를 해쉬 알고리즘으로 암호화하여 교환하고 키를 변경하여 위치트래킹, 스푸핑의 공격이 불가능하게 하였다.

참고문헌

[1] Federal Information Processing Standards (FIPS) , "Data Encryption Standard (DES), " NIST, Technical Report 46-2, January 1988.

[2] Federal Information Processing Standards (FIPS) , "Advanced Encryption Standard (AES), " NIST, Technical Report 197, November 2001.

[3] Federal Information Processing Standards (FIPS) , "Secure Hash Standard SHA-1, " NIST, Technical Report 180-1, April 1995.

[4] M. Feldhofer et al., "Strong Authentication for RFID Systems Using the AES Algorithms," CHES 2004, LNCS 3156, pp.357-370

[5] 유성호, 김기현, 황용호, 이필중, "상태기반 RFID 인증 프로토콜", 한국정보보호학회 논문지 제 14권 6호, 2004년 12월.

[6] 김미주, 최상명, 염호열, "효율적인 동기화를 제공하는 안전한 RFID 인증 프로토콜", 한국정보보호학회영남지부, 학술발표회눈문집, 2006년 2월.

[7] Jie Liang, Xuejia Lai, "Improved Collision Attack on Hash Function MD5, " Cryptology ePrint Archive 425, November2005.

[8] X. Wang, Y. Lisa Yin, H. Yu, "Finding collisions in the Full SHA-1," Crypto 2005, LNCS 3621, pp. 17-36, 2005

[9] Stephen Weis. Sanjay Sarma, Ronald Rivest, and Daniel Engels. "Security and privacy aspects of low-cost radio frequency identification systems," SPC'03, pp 454-469, March 2003.

[10] 주학수,권현조,강달천,윤재호,박배효,전길수,이재일 "RFID/USN 정보보호위협과 대응방안", 한국정보보호학회 논문지 제 14권 5호, 2004년 10월.

[11] 강전일, 박주성, 양대헌 "RFID 시스템에서의 프라이버시 보호기술", 한국정보보호학회 논문지 제 14호 6권, 2004년 12월.