모델체킹을 이용한 RFID 시스템 보안분석

김현석 ○ 오정현 최진영

고려대학교 컴퓨터학과

{hskim○, jhoh, choi}@formal.korea.ac.kr

Cryptanalysis of RFID System using Model Checking

Hyun-Seok Kim○ Jung-Hyun Oh Jin-Young Choi

Dept. of Computer Science and Engineering, Korea University

요 약

RFID 기술은 유비쿼터스 환경의 편리성을 위한 목적을 쉽게 이룰 수 있는 기술로 현재 주목 받고 있으며 많은 장점을 가지고 있다. 이러한 RFID 환경에서 가장 보안적인 문제를 갖는 것은 RFID 태그에 저장 관리되어지는 식별 정보에 대한 프라이버시 침해 문제의 보호이다. 본 논문에서는 정형검증 방법인 모델체킹을 이용하여 RFID 태그와 리더기간의 상호 인증 및 태그의 식별 정보에 대한 안전성을 분석하기 위해 RFID 보안 프로토콜의 취약성 분석 및 수정된 프로토콜을 제안함으로써 안전한 유비쿼터스 네트워크 환경을 구축하고자 한다.

1. 서 론

RFID (Radio Frequency Identification)[1][2]는 모든 사물에 RFID 태그를 부착하고 이를 통해 사물의 인식정보를 기본으로 주변의 모든 정보를 탐지 및 이를 실시간으로 네트워크에 연결하여 정보를 관리하는 것으로 먼저 인식 정보를 제공하는 RFID를 중심으로 발전하고 이에 감지기능이 추가되어 이들간의 네트워크가 구축 되는 USN 형태로 발전할 것으로 전망되고 있다. 그러나 이러한 RFID 태그의 사용에 있어서 사용자 개인의 프라이버시 문제(위치정보)와 RFID 태그의 ID가 쉽게 식별되는 문제 및 태그가 사용자가 알지 못하는 사이에 모든 리더에게 자동적으로 응답하는 문제가 인식된다.

이러한 궁극적인 보안문제를 해결하기 위한 방법으로서 보안프로토콜이 제안되어 왔으며 시스템의 설계단계에서부터 사용자와 개발자에게 안전성과 신뢰성을 제공하기 위해 대표적으로 정형기법이라는 연구를 안전성 및 보안성을 검증하고자 하였다.

이러한 방법론은 정형 명세와 정형 검증의 두 가지 방법으로 구분된다.

정형 명세는 개발하고자 하는 시스템의 동작 및 시스템이 만족해야 하는 특성을 정형적인 표현방법을 이용해 모델링하는 방법이고, 정형 검증은 정형적으로 명세된 시스템을 대상으로 그 시스템이 정확한지 혹은 그 시스템의 요구사항으로 주어지는 특성을 만족하는지를 논리적으로 증명하는 방법이다.

그 중 정형 검증은 정리증명과 모델체킹 기법으로 구분되며, 전자는 보안 로직을 이용하여 특정한 논리식으로 시스템을 명세하고 정확한 논리 증명단계로써 정확성을 증명하는 방식이고, 후자는 프로토콜의 인증과정을 유한상태기계의 형식으로 모델링하고 그 모델이 만족해야 하는 요구사항이나 특성을 모델에서 만족되는지를 검증도구를 이용해 자동으로 증명하는 방식으로 ESTEREL, Murphi, FDR[6]과 같은 방법이 있다.

본 논문에서는 정형검증 도구 중 FDR 이라는 모델체킹 도구를 이용, RFID 보안프로토콜인 해쉬기반 프로토콜들[3]의 취약성을 분석하여 보안 프로토콜의 안전성을 향상시키고 수정된 프로토콜을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 RFID 시스템의 개요와 보안요구사항에 대해 설명하고, 3장에서는 RFID 시스템의 보안문제의 해결에 관한 관련 연구에 대해 살펴보고, 4장에서는 각 프로토콜을 명세하고 검증하기 위한 Casper[4] 및 FDR 도구에 대해 소개하며, 5장에서는 CSP, Casper 와 FDR을 이용하여 해쉬기반 보안프로토콜들의 분석 및 결과에 대해 살펴봄과 동시에 이러한 보안 프로토콜의 취약성을 해결한 새로운 프로토콜을 제안한다. 마지막 6장에서는 결론 및 향후 연구방향을 제시하고자 한다.

2. RFID 시스템 및 보안요구사항

2.1 RFID 시스템 개요

RFID 시스템 (그림 1)은 다음 세가지 구성요소로 이루어져있다.

- RFID labels (트랜스폰더) : 객체 식별 정보전송
- RFID label 리더 (트랜시버) : Tag 정보 수집
- Application system (데이터베이스) : 리더에 의해 수집되는 Tag 확인 정보제공

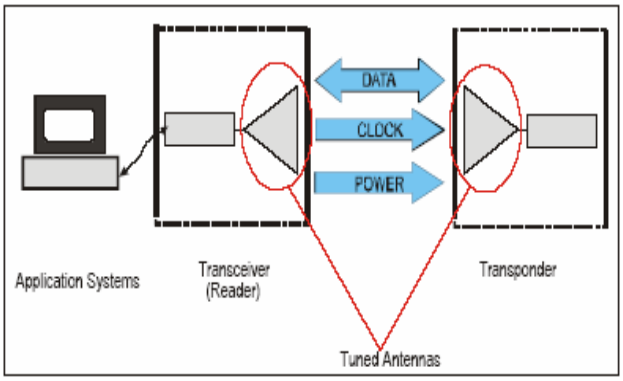

그림 1. RFID 컴포넌트 상호작용도

이러한 환경하에서 RFID의 보안 요구사항은 다음과 같이 정리할 수 있다.

2.2. RFID 시스템의 보안 요구사항

위 환경하에서 RFID 시스템의 RF태그와 리더 등 구성환경에 대해 다음과 같은 사항을 고려해 보안 요구사항을 설정할 수 있다. 특히 여러 가지 보안 요구사항 중 물리적인 방안을 통해 Tag 비용을 높이기 보다, 보안프로토콜과 같이 암호기술적 해결방안을 적용하기 위한 요구사항들로 아래와 같이 4가지를 제시하였다.

A. 인증이 되지 않은 리더에게 정보유출이 되지 않아야 하며, 태그와 그 소유자 사이에 긴 시간 동안의 추적(long-term tracking)이 불가능해야 한다.
B. 태그의 내용은 근제한기법 (access control)에 의해 질의채널 (interrogation channel)이 안전하지 않다고 예상되면 암호화되어야 한다.
C. 태그와 리더 사이에는 상호인증(mutual authen -tication)이 제공되어야 한다.
D. 태그와 리더 모두 재생공격(replay attack) 및 공격자 중간공격(man-in-the-middle attack)에 저항력이 있어야 한다.

3. RFID 시스템 보안문제 해결에 관한 관련연구

RFID 시스템에서 사용자 프라이버시의 보호를 위한 많은 연구들이 진행되어 오고 있다. 현재 진행되어 왔던 연구결과 중, Kill 명령어의 접근법, Blocker 태그 기법, 해쉬락(Hash-Lock)기법[1], 랜더마이즈 해쉬락 기법[3], XOR 기반 원타임 패드 기법[7], 외부재암호화 기법, 해쉬 체인(Hash-Chain) 기반 기법등이 있다.
RFID 네트워크 시스템의 보안을 위한 공개키 알고리즘의 하드웨어적인 구현은 2004년 Rabin, NTRU, ECC등의 공개키 알고리즘에 대한 구현 결과 제시에 의해 NTRU의 경우 20μ W의 저전력에 3000개의 게이트만 필요하며, 경량화된 센서 노드에 탑재 가능한 것으로 알려져 있다[8]. 그러나 현재는 기존의 알고리즘을 개선하여 사용하고 있으며 향후에는 새로운 알고리즘 개발이 요구된다. 그룹키 관리를 위해 대칭키 방식 적용시 리더와 태그간의 키를 공유해야 하며, 각 태그마다의 유일한 키를 관리하는 등의 많은 계산량 때문에 사용하기 어려우며, 키의 유출에 의한 태그 무력화, 장기간의 사용에 따른 노출 가능성등의 문제가 있다. 또한, 태그에 암호키를 탑재하는 방식은 에칭(etching), 탐침 등의 물리적공격에 취약하며 암호키의 노출 가능성이 있다. 버클리 대학의 SmartDust 프로젝트에서 채택한 센서네트워크의 보안 프로토콜인 SPINS(Security Protocol for Sensor Network)[9]은 μ TESLA와 SNEP로 구성되어 있으며 메시지 인증, 무결성, 기밀성, 적시성 등의 서비스를 제공하고 있다. 랜덤키 사전 분배방식은 키 DB을 선택하고 무작위로 키를 선택하여 센서 노드에 할당하며, 두 개의 노드는 자신의 키 DB를 탐색하여 상대방이 같은 공통키를 소유하고 있으면 이 키를 세션키로 사용하는 방식이다. 본 논문에서는 물리적 레벨의 보호 기법이 아닌 암호 기술을 중심으로 한 RFID에서의 보안 프로토콜을 분석한다.

4. Casper 와 FDR 도구

4. 1 CSP (Communicating Sequential Process)

CSP[5]는 프로세스 알제브라 언어로서, 병렬성을 갖는 통신프로토콜의 동작을 효율적으로 명세하기 위한 언어이다. 최초 일반 통신 프로토콜 및 제어 시스템의 명세를 위해 사용되어졌으나, 점차 보안 프로토콜의 명세를 위한 영역으로 확대되어 가고 있다. CSP 에서 제공하는 pure synchronization(|||)과 Interleaving parallelism(||) 개념을 사용하여 분산 시스템 환경하에서 동작하는 클라이언트 서버와 공격자 모델을 정형적으로 표현할 수 있는 장점을 갖고 있다. 예를 들어, 분산시스템 환경하에서 동작하는 보안 시스템은 다음과 같이 간략히 표현할 수있다.

SYSTEM = CLIENT1 ||| CLIENT2||| SERVER || INTRUDER

4.2 Casper (A Compiler for the Analysis of Security Protocols) [4]

CSP(Communication Sequential Process)[4]언어 를 이용하여 보안프로토콜 행위를 명세하고 FDR[6] 정형검증 도구를 이용하여 보안속성을 검증하는 연구가 진행되었다. 하지만, CSP 언어를 이용한 정형명세과정은 정형적 설계 방법에 익숙치 않은 보안프로토콜 설계자에게는 매우 복잡한 명세언어라는 단점을 갖고 있었다. 이에 따라, 보안프로토콜의 행위를 간략히 명세할 수 있도록 Casper 도구가 개발되었다. Casper 도구로 보안프로토콜의 행위와 검증속성을 명세하게 되며, 자동변환기능을 이용해 CSP 명세코드를 생성할 수 있다. 결국, 자동 생성된 CSP 명세코드를

FDR 정형검증도구에 입력하여 보안프로토콜을 검증하게 된다.

4.3 FDR (Failure Divergence Refinement)

FDR [6]도구는 CSP 명세언어를 입력으로 받아들이는 모델체킹 도구로서, CSP 명세언어로 기술된 보안프로토콜 모델이 보안성 및 인증속성과 같은 보안속성들을 만족하는지 검증하게 되며, 만일 만족하지 않을 경우에는 CSP 이벤트로 기술된 반례(counterexample)을 보여주어 보안상 취약점 분석을 용이하게 한다.

FDR 도구는 3 가지의 검증방법을 지원하고 있다.

- Trace refinement : 안전성 (safety) 검증
- Failures refinement : 교착상태 (deadlock) 검증
- Failures - Divergence : 라이브락 (livelock) 검증

5. Casper/FDR 을 이용한 해쉬-언락킹 프로토콜분석 및 수정된 프로토콜 제안

5.1 해쉬 언락킹 프로토콜 분석 및 결과

태그는 해쉬 메커니즘을 처리할 수 있는 H/W 기반의 암호화 모듈로서 보안적 요구사항을 처리할 수 있다. Tag 에는 MetaID 정보만을 보관할 수 있는 저장 공간을 보유하고 있어야 하며 Lock 과 Unlock 처리기능만 동작하면 된다. Unlock 이 된 Tag 만이 Tag Reader 장비와 운영이 가능하다.

표 1. 해쉬-락 스킴의 표현법

T	RF 태그의 식별자
R	RF 리더의 식별자
DB	백엔드 데이터 베이스의 식별자
Xkey	통신참여자 X 의 세션키
metaID	키를 해쉬값으로 처리한 값
ID	태그의 정보값
Xn	통신참여자 X 에 의한 난수값
H	해쉬함수

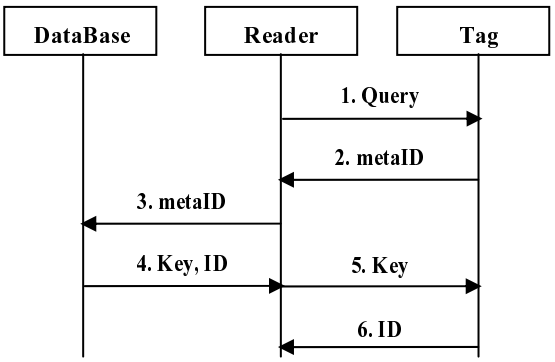

그림 2. 해쉬-언락킹 프로토콜

- 해쉬-락의 locking 프로토콜
  ① 리더는 랜덤한 키 key 를 선택하고, meta ID 값으로 hash(key)를 계산한다.
  ② 리더는 metaID 를 태그에 기록한다.
  ③ 태그는 잠긴 상태(locked state)에 들어간다.
  ④ 리더는 metaID, key 를 저장한다.

- 해쉬-락의 unlocking 프로토콜(그림 2. 참조)
  ① 리더는 태그에게 태그의 metaID 를 질의한다.
  ② 리더는 데이터베이스에서 metaID 와 key 를 조사한다.
  ③ 리더는 태그에게 key 를 전송한다.
  ④ 만일 hash(key)와 metaID 가 일치하면, 태그는 잠긴 상태에서 빠져 나온다(unlock).

본 논문에서는 해쉬-언락킹 프로토콜을 Casper 도구를 이용해 모델링하였는데 그림 3 은 해쉬-언락킹 프로토콜을 Casper 표현방식으로 모델링한 것으로 8 가지 항목 중 자유변수 영역과 프로토콜 기술영역, 침입자 영역에 대한 표현이다.

```
#Free variables
R, T : Agent
DB   : Server
key  : SessionKey
Id   : Text
H    : HashFunction

InverseKeys = (key, key)

#Protocol description
0.      -> T  : R
1.  T   -> R  : (H(key)) % metaID
2.  R   -> DB : metaID % (H(key))
3.  DB -> R   : key, Id
4.  R   -> T  : key
5.  T   -> R  : Id

#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag, Reader, DataBase}
```

그림 3 . Casper 를 이용한 해쉬-언락킹 프로토콜 명세

먼저 자유변수 영역에서, R 은 리더, T 는 태그로서 각각 Agent 로 나타내고, DB 는 백엔드 서버의 역할을 한다. key 는 Session 키, Id 는 Tag 의 정보를 표현하고, InverseKeys 는 Session 키에 대한 암호화 복호화를 표현하며, H 는 해쉬함수를 뜻한다. 다음으로 프로토콜 기술 영역은 해쉬-언락킹 프로토콜을 명세한 부분으로 여기서 % 표현은 메세지 1 에서 T 가 H(key) 값을 metaID 로서 수신자인 R 에게 복호화의 목적이 아닌 단지 다른 수신자 DB 에게 전달하는 목적을 지니고 있다. 따라서 메세지 2 에서 이 메세지가 DB 에게 전달되어 복호화된다.

마지막으로 침입자 영역에 대한 정보가 제시되어 있다.

5.1.1 해쉬-언락킹 프로토콜 검증 결과

해쉬-언락킹 프로토콜에서는 metaID 의 값을 중간자 공격 및 재생 공격에 이용함에 따라 태그 정보의 노출 및 추적이 가능하게 하였을 뿐만 아니라 리더기와 태그의 인증에 실패하는 결과를 초래하였다. 이를 Casper script 를 이용하여 명세하기 위해 해쉬-언락킹 프로토콜의 두 개체간 사용된 정보에 대한 비밀성과 개체간 상호 ID 에 대한 인증을 만족해야 하며 이는 다음과 같이 표현할 수 있다.

Secret(R, key, [T])
Secret(R, Id, [T])
Agreement(T, R, [Id, key])

첫번째 표현은 " R 은 key 정보를 오직 T 와만 알고 있다" 라고 풀이할 수 있고 두번째 표현은 " R 은 Id 정보를 오직 T 와만 알고 있다" 로 풀이할 수 있다. 세번째 표현은 " T 는 Id, key 정보를 통해 R 로부터 자신의 개체를 인증받는다" 라고 풀이할 수 있다

모델 체커를 이용해 비밀성과 개체인증 속성의 만족 여부를 확인한 결과 첫번째 표현에서 R 이 전달하는 key 에 대해 T 와의 비밀성 속성을 만족하지 않았고 이에 따라 결국 두 개체간의 데이터가 누설되었다. 또한 Id 의 정보도 비밀성 속성을 만족하지 않았으며 마지막 속성인 개체 인증에서도 Id, key 의 정보를 이용해 두 개체간의 인증에 실패했다.

위 비밀성 요구사항의 반례에 대해 FDR 의 interpret 기능을 통해 분석한 결과는 그림 4 와 같다.

```
0.                -> Tag          : Reader
1.  Tag           -> I_Reader     : H(Key)
2.  I_Tag         -> DataBase     : H(Key)
1.  I_Tag         -> Reader       : H(Key)
3. DataBase       -> I_Tag        : Key, ID
2.  Reader        -> I_DataBase   : H(Key)
Reader believes ID is a secret shared with Tag
The intruder knows ID
```

그림 4. FDR 을 이용한 반례의 분석결과

2.2 에서 제시된 보안 요구사항에 대해 위 분석 결과를 토대로 다음과 같이 정리할 수 있다.

A. 인증 통한 long-term tracking 방지 : R 입장에서 T 로부터 정상적인 데이터를 전송받았다고 간주했으나, I_Tag 나 I_DataBase 와 같은 악의적인 개입이 가능했으며. metaID 정보는 tracking 에 사용될 수 있다.
B. 암호화를 통한 채널 안전성 확보 : 위 반례에서 알수 있듯이, 세번째 메시지의 Key 와 ID 는 암호화되지 않은 채 전송되어 노출되었다.
C. 상호 인증 : R 입장에서 T 로부터 정상적인 데이터를 전송받았다고 간주했으나, I_Tag 나 I_DataBase 와 같은 악의적인 개입이 가능했다.
D. Tag 정보의 유출방지를 통한 재생공격 및 중간자 공격 방지 : T 가 R 에게 정상적인 데이터 전송을 했다고 간주했으나 I_Reader 에 의해 H(key)정보가 노출되었다. 결과적으로 T 의 metaID 정보는 중간자 공격에 이용되었다

5.2 랜더마이즈 해쉬-락 스킴 분석 및 결과

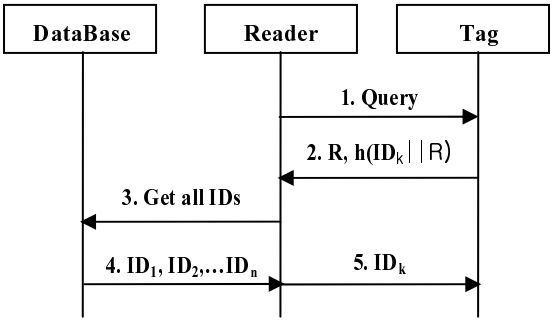

그림 5. 랜더마이즈 해쉬-언락킹 프로토콜

해쉬-락기법에서 가능한 사용자 추적을 방지하기 위한 방식이다. 이 기법에서는 태그에 일방향 해시 함수와 난수발생기(PRNG)가 구축되어있어야 한다.

• 랜더마이즈 해쉬-락의 unlocking 프로토콜(그림 5. 참조)
① 리더는 태그에게 질의를 보낸다.
② 태그는 랜덤한 넌스값을 생성하고, hash(ID || R) 값을 계산한다.
③ 태그는 리더에게 (R, hash(ID_k || R))을 전송한다.
④ 리더는 모든 알려진 ID 값들에 대해 hash(ID_k || R) 을 계산한다.
⑤ 만약 hash(ID_k || R)의 만족하는 ID_k를 찾는다면, 리더는 태그에게 ID_k를 전송한다.
⑥ 만약 ID_k와 ID 가 일치한다면, T 는 잠긴 상태에서 빠져나온다.

이 방식은 초당 100~200 개의 태그를 읽어야 하는 많은 개수의 태그를 소유한 환경에서는 비현실적이다. 그러나 상대적으로 적은 수의 태그 사용자를 갖는 환경에서는 가능한 방식이다.

5.2.1 랜더마이즈 해쉬-언락킹 프로토콜 검증 결과

앞서 살펴본 5.1 의 해쉬-언락킹 프로토콜과 같이 Casper 와 FDR 을 통해 검증한 결과, 다음과 같은 결과를 얻을 수 있다.

A. 인증 통한 long-term tracking 방지 : T 가 R 의 식별자 정보를 해쉬하여 R 에게 보냄으로써 T 는 R 을 신뢰하고 IDk 를 전송하고 있으나 IDk 가 고

성능 리더기에 의해 그 값을 식별할 경우 tracking 가능

B. 암호화를 통한 채널 안전성 확보 : DB 로부터 R 에게 전송되는 ID 들은 암호화한 값들이 아니며, 이들의 값은 Tag 의 분석에 이용가능

C. 상호 인증 : R 입장에서 T 로부터 정상적인 데이터를 전송받았다고 간주했으나, I_Tag 나 I_DataBase 와 같은 악의적인 개입이 가능했다.

D. Tag 정보의 유출방지를 통한 재생공격 및 중간자 공격 방지 : T 가 R 에게 정상적인 데이터 전송을 했다고 간주했으나 I_Reader 에 의해 H(key)정보가 노출되었다. 가능하고 R 입장에서 허위 IDk 에 의해 재생 공격이 가능함. 결과적으로 T 의 metaID 정보는 중간자 공격에 이용가능.

5.3 해쉬기반 프로토콜의 취약성을 수정한 제안프로토콜

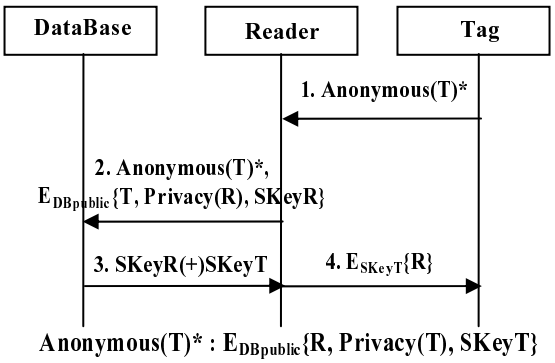

그림 6. 제안프로토콜

이러한 해쉬-언락킹 프로토콜의 문제점으로 분석되었던 부분은 태그 내에 저장된 정보를 해쉬 기반기법으로 태깅할 수 있게 함으로써 발생되었으며 이는 태그의 정보를 인증된 리더기에 의해 데이터베이스에 접근함으로써 태그정보를 가져갈 수 있도록 하여 중간자 공격과 재생공격을 방지할 수 있었다. 즉 제안프로토콜은 다음 (그림 6)과 같은 절차로 인증이 이루어지며 앞서 언급된 취약성은 인증과정에서 도입된 2 가지 기술에 의해 극복할 수 있다.

메시지 1. 리더의 식별자 (R), 태그의 식별자 를 익명의 값으로 처리된 값 (Privacy(T)), 태그의 식별자 (T)를 데이터베이스와의 세션키값 (SkeyT)으로 암호화하여 리더에게 전송한다.

메시지 2. 태그로부터 전송받은 값이 메타값(Anonymous(T))으로 처리되어 있으므로 이를 데이터베이스에게 재전송만 가능하고 데이터베이스의 공개키로 암호화된 태그의 식별자 (T)와 리더의 식별자를 익명으로 처리한 값 (Privacy(R)), 그리고 데이터베이스와의 세션키 (SKeyR)를 전송한다.

메시지 3. 데이터베이스는 리더와 태그로부터 전송받은 각각의 세션키값을 배타적합을 통해 리더에게 전달하고 리더는 SKeyR 값을 가지고 있으므로 SKeyT 값을 복호화 할 수 있게된다.

메시지 4. 태그의 세션키 값 (SKeyT)으로 리더의 식별자 (R)를 암호화하여 태그에게 전송하며 이를 통해 태그는 리더를 인증할 수 있게 된다.

위 프로토콜 (그림 6)에서는 2 가지 새로운 기술이 적용된다.

1. 메시지 1 과 2 에서 태그와 리더가 각각의 식별자를 Privacy 라는 통신참여자의 식별자를 파라미터로 하여 익명의 값을 출력하는 함수를 이용한 기술로써 각각의 프라이버시를 노출시키지 않고자 하였다.
2. 태그에게 리더의 안전한 인증을 위해 사전에 리더와 태그가 데이터베이스와 세션키를 각각 설정하고, 데이터베이스는 리더에게 배타적합 (Exclusive-or) 기술을 이용하여 리더로부터 태그의 세션키 값을 도출할 수 있도록 하여 자신의 식별자를 암호화하는데 이용함으로써 태그로부터 인증 받을 수 있게 되었다.

6. 결론 및 향후 연구방향

RFID 기반의 유비쿼터스 컴퓨팅 환경은 네트워크를 기반으로 한 장치 간의 연결을 기본으로 하고 있다. 특히 무선중심의 근거리 통신기술이 발달함에 따라 유비쿼터스 컴퓨팅 환경을 이루는 무수히 많은 개체들은 유기적으로 연결되어, 서로 데이터를 주고받고, 이를 통해 서비스를 제공하게 된다. 이러한 개체간의 연결은 보안 취약점을 발생시키고 있으며, 이에 따른 정보보호 관점의 보안 요구사항이 도출될 수 있다. 본 논문에서는 암호기술적 관점에서 해쉬 기반의 RFID 보안프로토콜의 취약성을 분석하고 수정된 보안프로토콜을 제시하였다. 향후 연구방향으로서 제안되고 있는 보안프로토콜들간의 성능 분석을 통해 활용가능성에 대해 연구하고자 한다.

7. 참고문헌

[1] S. Sarma, S. Weis, and D. Engels, “ RFID systems and security and privacy implications”, In Workshop onCryptographic Hardware and Embedded Systems (CHES) 2002, LNCS No. 2523, pp. 454-469, 2003

[2] EPCGLOBAL INC.: http://www.epcglobalinc.org.

[3] S. Weis, S. Sarma, R. Rivest and D. Engels, “ Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems”, In 1st Intern. Conference on Security in Pervasive Computing (SPC), 2003.

[4] G. Lowe, “ Casper: A compiler for the analysis of security protocols”, In Proceeding of the 1997 IEEE Computer Security Foundations Workshop X, IEEE Computer Society, Silver Spring, MD, pp. 18-30, 1997

[5]C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, 1985.
[6] Formal Systems Ltd. FDR2 User Manual, Aug. 1999.
[7] A. Juels, Privacy and authentication in low-cost RFID tags, In submission, Available at http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/
[8] G. Gaubatz, J. Kaps, and B. Sunar, " Public Keys Cryptography in Sensor Networks Revisited", Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004).
[9] A. Perrig, et al., " SPINS : Security Protocols for Sensor Networks", Mobile Computing and Networking 2001.