안전한 유비쿼터스 컴퓨팅 환경을 위한 RFID 시스템의 프라이버시 및 인증 분석

송영소*, 김현석**, 김진우**, 최진영**
*고려대학교 컴퓨터정보통신대학원
**고려대학교 컴퓨터학과
e-mail : *songys@bai.go.kr, **{hskim, choi}@formal.korea.ac.kr, **pkm311@gmail.com

Privacy and Authentication Analysis of the RFID System for Secure Ubiquitous Computing Environment

Young-So Song*, Hyun-Seok Kim**, Jin-Woo Kim**, Jin-Young Choi**
* Dept. of Computer & Information Technology, Korea University
** Dept. of Computer Science & Engineering, Korea University

요 약

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 이러한 유비쿼터스 컴퓨팅 환경내에서 발생할 수 있는 보안 취약점과 보안 요구사항을 도출하여 제시하고, RFID(무선주파수 식별자)를 이용한 인증기술을 중심으로 보안 서비스 방향을 제안함으로써, 유비쿼터스 컴퓨팅 환경이 현실화될 수 있는 기반기술인 보안 및 인증기술에 대해 연구하고자 한다

1. 서론

유비쿼터스 컴퓨팅 환경은 전방위적인 기술을 응용하여 구축될 수 있으며, 이러한 기술은 다양한 시각에서 분류될 수 있다. 이중에서 유비쿼터스 컴퓨팅 환경을 이루는 가장 기반이 되는 기술로서 보안 및 인증기술을 분류하고 있다. 보안및 인증은 유비쿼터스 컴퓨팅 환경이 가지는 특징으로부터 발생하는 보안 취약사항과, 이로부터 도출될 수 있는 보안 요구사항에 대한 것이다. 이러한 보안 및 인증기술을 응용하여 안전한 유비쿼터스 컴퓨팅 환경의 적용에 기여한 기술로서 기존의 바코드를 대신하여 기업 물류 활동에 중대한 변화를 가져올 킬러 어플리케이션 및 유비쿼터스 네트워크의 센서기능을 담당하는 핵심 기술인 RFID(Radio Frequency IDentification)가 있다. 예를 들어, 태그[2]가 부착된 소비자의 물건에 대한 추적을 통해 소비자의 위치 추적이 가능하며, 개개인이 가지고 다니는 물건들을 소비자 모르게 비밀리에 목록화하여 악용할 수 있다. 또한 태그가 출입 통제 시스템에 사용될 경우 악의적인 리더가 태그의 정보를 쉽게 읽어 들이고, 여기서 얻은 정보를 이용하여 태그를 위조하는 것이 가능하다. 이것은 태그의 정보에 대한 인증되지 않은 접근에서 비롯된다. 만약 태그의 메모리에 민감한 데이터가 저장되어 있다면, 이것은 심각한 보안 문제를 야기시킬 수 있다. 현재 RFID 에서의 이러한 보안 문제를 해결하기 위한 다양한 연구가 진행되고 있다. 이러한 연구의 일환으로 RFID 시스템에서 사용자 프라이버시를 보호를 위해 kill tag, faraday cage, 방해 전파(active jamming), blocker tag 등과 같은 물리적레벨의 대응 기법과 hash lock[1][3]등과 같이 암호 기술을 이용한 보호 기법이 제안되고 있다. 본 논문에서는 물리적 레벨의 보호 기법이 아닌 암호 기술을 중심으로 한 RFID 에서의 보안 프로토콜을 분석한다.

이러한 보안프로토콜을 구현하기 전에 설계단계에서부터 사용자와 개발자에게 안전성과 신뢰성을 제공하기 위한 기술이 요구되고 있다. 그러한 요구를 만족시키기 위해 진행되는 노력 중 대표적으로 정형 기법이라는 연구가 있으며 이는 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 BAN[8], GNY 와 같은 보안 로직을 이용하여 특정한 논리식으로 시스템을 명세하고 정확한 논리 증명단계로써 정확성을 증명하는 방식이고, 후자는 프로토콜의 인증과정을 유한상태기계의 형식으로 모델링하고 그 모델이 만족해야하는 요구사항이나 특성을 모델에서 만족되는

지를 검증도구를 이용해 자동으로 증명하는 방식으로 ESTEREL, Murphi, NRL protocol Analyzer[9]와 FDR[6][7]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 향상시키고자 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 해쉬기반 보안프로토콜들에 대해서 설명하고 3 장에서는 프로토콜을 명세하고 검증하기 위한 Casper[5] 및 FDR 도구에 대해 소개하며, 4 장에서는 CSP[4], Casper 와 FDR 을 이용하여 해쉬-언락킹 보안프로토콜의 분석 및 결과에 대해 살펴보고, 5 장에서는 이러한 보안프로토콜의 취약성을 해결한 새로운 프로토콜을 제안한다. 마지막 6 장에서는 결론 및 향후 연구 방향을 제시하고자 한다.

2. 해쉬 기반 보안프로토콜

2.1 해쉬-락 스킴

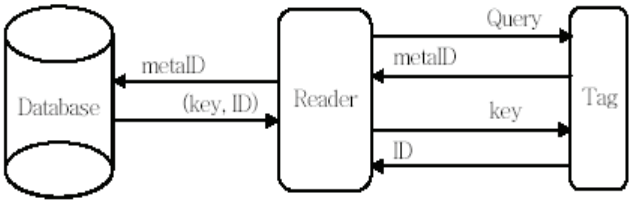

그림 1. 해쉬-언락킹 프로토콜

- 해쉬-락의 locking 프로토콜
  ① 리더 R 은 랜덤한 키 key 를 선택하고, meta ID 값으로 hash(key)를 계산한다.
  ② R 은 metaID 를 태그 T 에 기록한다.
  ③ T 는 잠긴 상태(locked state)에 들어간다.
  ④ R 은 (metaID, key)를 저장한다.
- 해쉬 락의 unlocking 프로토콜(그림 1. 참조)
  ① 리더 R 은 태그 T 에게 T 의 metaID 를 질의한다.
  ② R 은 데이터베이스에서 (metaID, key)를 조사한다.
  ③ R 은 T 에게 key 를 전송한다.
  ④ 만약 hash(key)와 metaID 가 일치하면, T 는 잠긴 상태에서 빠져 나온다(unlock).

일방향 해시 함수의 역함수 계산 어려움에 기반한 해쉬-락 스킴은 인가받지 않은 리더기가 태그 콘텐츠 읽는 것을 방지할 수 있다. 위장(spoofing)은 방지하지 못하지만 탐지는 가능하다. 공격자는 태그에게 metaID 를 요구한 후에 재생 공격(replay attack)에서 합법적 리더기에게 태그를 위장하는 것이 가능하다. 그러면 합법적 리더기는 위장된 태그에게 키를 주게 된다. 그러나 리더기는 태그의 콘텐츠(일반적으로 태그의 ID)를 체크하여 백엔드 데이터베이스로부터 적절한 metaID 인지를 검증할 수 있다. metaID 가 부적절한 경우, 리더기는 적어도 위장이 발생했음을 경고할 수 있다. 해쉬-락 스킴은 태그에 해시 함수의 구현만을 요구하고, 백앤드에 키관리를 요구한다. 이러한 요구조건은 가까운 장래에 경제적인 것이 될 수 있다. 그러나, 위 방식에서는 metaID 가 식별자처럼 사용되기 때문에 사용자 추적(tracking of individual)이 가능하다.

2.2 랜더마이즈 해쉬-락 스킴

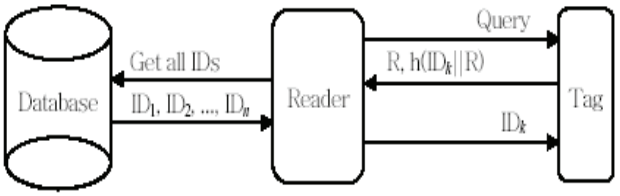

그림 2. 랜더마이즈 해쉬-언락킹 프로토콜

해쉬-락기법에서 가능한 사용자 추적을 방지하기 위한 방식이다. 태그는 인가되지 않은 사용자에 의한 질의에 대하여 예상 가능한 응답을 하지 않지만, 합법적인 리더기에 의해서는 여전히 식별 가능해야 하는 방식이다. 이 기법에서는 태그에 일방향 해시 함수와 난수발생기(PRNG)가 구축되어있어야 한다. 합법적인 리더기는 태그를 스캔하기전에 "knows what she owns"를 가정한다. 태그를 lock 상태로 만드는 것은 프로토콜이 필요 없는 간단한 과정이나, 태그를 unlock 상태로 하는 프로토콜은 필요하다. 태그를 unlock 상태로 하는 프로토콜은 다음과 같다(그림 2. 참조).

① 리더 R 은 태그 T 에게 질의를 보낸다.
② T 는 랜덤한 난스(nonce)R 을 생성하고, hash(ID || R) 값을 계산한다.
③ T 는 R 에게 (R, hash(ID || R))을 전송한다.
④ R 은 모든 알려진 IDi 값에 대해 hash(IDi || R) 을 계산한다.
⑤ 만약 hash(IDi || R) == hash(ID || R)을 만족하는 IDi 를 찾는다면, R 은 T 에게 IDi 를 전송한다.
⑥ 만약 IDi 와 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. 소매 상점은 일반 사용자에 비해서 위치 프라이버시와 연관성이 적기 때문에 소매 상인들은 해쉬-락 기법을 적용하고, 구매하는 소비자에게는 랜더마이즈 해쉬-락 기법을 적용한다. 한 가지 문제는 합법적인 리더기들이 어떻게 그들의 태그를 알게 되느냐는 것이다. 물건이 팔렸을 때, 그것의 ID 도 반드시 같이 전송되어야 한다. 그렇지 않으면 새로운 소유주가 태그를 읽을 수 없다. 새로운 소유주가 자신의 태그에 접근하는 한 가지 방식이 printed master key 를 사용하는 것이다. 위 방식이 충분히 현실적이지만 이론적으로 완벽하지는 않다. 이는 일방향 함수의 정의가 역함수 계산의 어려움만을 의미하기 때문이다. ID 비트가 노출되지 않음을 보장하기 위해서 보다 강한 프리미티브의 사용이 가능하다. 각 태그는 리더기와 유일한 비밀키를 공유한다고 하고, PRF(Pseudo-Random Function) 앙상블(ensemble)을 지원한다고 가정하면, 이론적으로 ID 비트 노출 방지가 가능하다. 구현상의 문제로 PRF 앙상블을 대칭키 암호화보다 아주 적은 자원으로 구현 가능하냐의 문제가 발생하는데,PRF 앙상블의 최소 하드웨어 복잡도는 open problem 이다.

3. CSP, Casper and FDR

3. 1 CSP(Communicating Sequential Process)[4]

CSP 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 수 있다.

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

3.2 Casper(A Compile for the Analysis of Security Protocols)[5]

CSP 와 FDR 을 이용한 보안프로토콜 명세시 명확하고 세부적인 표현에 있어 machine-based 이 아닌 수작업에 전적으로 의존한다는 점에서 매우 방대한 시간이 소요된다는 단점을 가지고 있어 비용 대 효과면에서 다소 비효율적인 방법론이라 할 수 있다. 이러한 점을 개선한 도구로서 추상적인 표현만으로 CSP 명세소스를 자동으로 생성해주는 개발도구가 바로 Casper 이다.

3.3 FDR(Failure Divergence Refinement)[6]

모델체킹 도구로서, CSP 언어로 생성된 파일을 통해 구현된 보안 모델에 대해 비밀성, 인증성과 같은 보안 속성의 만족여부를 체킹하는 도구이다. 이를 통해 해당 속성을 만족시키지 못할 경우 반례를 보여주어, 공격 시나리오의 가능형태를 분석해 준다. 즉 보안 프로토콜이 반드시 갖추어야 할 요구사항인 비밀성, 무결성, 인증, 부인방지와 같은 보안속성의 만족여부에 대한 검사 도구이다.

4. Casper/FDR 을 이용한 해쉬-언락킹 프로토콜 분석 및 결과

4.1 해쉬-언락킹 프로토콜 분석

본 논문에서는 해쉬-언락킹 프로토콜을 모델체킹 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
InverseKeys = (key, key)
H : HashFunction

#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.

4.2 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 만족여부를 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

T 가 R 에게 정상적인 데이터 전송을 했다고 간주했으나 I_Reader 에 의해 H(key)정보가 노출되었다. 결과적으로 문제점은 T 의 metaID 정보는 중간자 공격에 이용되었으며, 또 다른 공격시나리오로서 R 입장에서 T 로부터 정상적인 데이터를 전송받았다고 간주했으나, I_Tag 나 I_DataBase 와 같은 악의적인 개입이 가능했다. 이러한 해쉬-언락킹 프로토콜의 문제점으로 분석되었던 부분은 태그 내에 저장된 정보를 해쉬 기반기법으로 태깅할 수 있게 함으로써 발생되었으며 이는 태그의 정보를 인증된 리더기에 의해 데

이터베이스에 접근함으로써 태그정보를 가져갈 수 있도록 함으로써 중간자 공격과 재생공격을 방지할 수 있었다. 즉 리더기는 태깅하고자 하는 태그로부터 랜덤한 값을 받아 데이터 베이스로부터 인증을 받고 이 데이터 베이스는 리더기와 태그에게 각각 인증코드와 태그 정보를 부여한다. 이렇게 인증코드를 부여받은 리더기는 태그에게 인증을 요청하고 태그는 허가된 인증코드를 확인함으로써 데이터베이스로부터 받은 태그정보를 리더기에게 제공함으로써 해쉬 기반 기법의 문제점을 해결할 수 있었다.

5. 해쉬기반 프로토콜의 취약성을 수정한 제안프로토콜

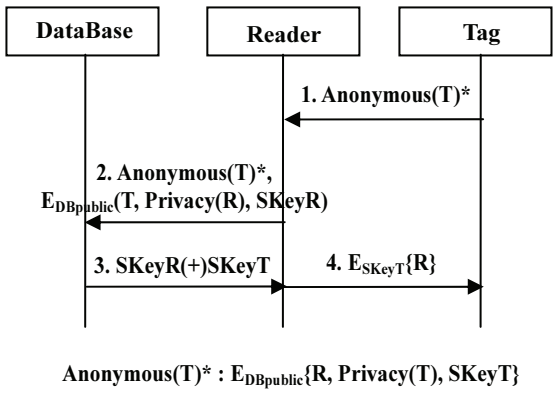

그림 5. 프라이버시 보호를 위한 제안프로토콜

프로토콜은 다음과 같은 절차로 인증이 이루어지며 앞서 언급된 취약성은 인증과정에서 도입된 2 가지 기술에 의해 극복할 수 있다.

메시지 1. 리더의 식별자, 태그의 식별자를 익명의 값으로 처리된 값, 태그의 식별자를 데이터베이스와의 세션키값으로 암호화하여 리더에게 전송한다.
메시지 2. 태그로부터 전송받은 값이 메타값으로 처리되어 있으므로 이를 데이터베이스에게 재전송만 가능하고 데이터베이스의 공개키로 암호화된 태그의 식별자와 리더의 식별자를 익명으로 처리한 값, 그리고 데이터베이스와의 세션키를 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 값으로 리더의 식별자를 암호화하여 태그에게 전송하며 이를 통해 태그는 리더를 인증할 수 있게 된다.

위 프로토콜에서는 2 가지 새로운 기술이 적용된다.

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

6. 결론 및 향후 연구방향

유비쿼터스환경에서 RFID 보안프로토콜은 중요한 역할을 한다. 본 논문에서는 RFID 환경에 적용할 수 있는 암호 기술을 분석하였다. RFID 환경을 위한 암호 기술로 키설정 기술 및 개발된 보안 프로토콜 기술을 분석 및 문제점을 제시하였고 이러한 문제점을 해결한 새로운 프로토콜을 제안하였다. 향후 연구과제로서 대칭적인 세션키 기반 경량화 보안프로토콜의 설계 및 검증을 하고자 한다.

참고문헌

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

[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] C.A.R. Hoare, Communicating Sequential Processes, Prentice-Hall, Englewood Cliffs, NJ, 1985.

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

[6] Formal Systems Ltd. FDR2 User Manual, Aug. 1999.

[7] P. Y. A. Ryan and S. A. Schneider, Modelling and Analysis of Security Protocols: the CSP Approach, Addison-Wesley, 2001.

[8]M. Abadi, M. Burrows, and R. Needham, “A Logic of Authentication”. In Proceeding of the Royal Society, Series A,426, 1871,pp.233-271, December 1989.

[9] Philip E.Varner, “Formal Methods as and Environmental Catalyst for Emergent Security in System Design and Construction”, December 12, 2002.