유비쿼터스 컴퓨팅환경에서 보안프로토콜의 정형적 분석

김현석° 이송희 최진영

고려대학교 컴퓨터학과
{hskim°, shlee, choi}@formal.korea.ac.kr

Formal Analysis of the Security Protocol in Ubiquitous Computing Environment

Hyun-Seok Kim° Song-Hee Lee Jin-Young Choi

Dept. of Computer Science and Engineering, Korea University

요 약

유비쿼터스 환경에서의 네트워킹은 장소나 시간에 제약없이 최상의 서비스를 받기 위한 편리성을 제공한다. 이에 RFID 기술은 이러한 유비쿼터스의 목적을 쉽게 달성할 수 있는 기술로 주목받고 있으며 많은 장점을 갖는 반면, 프라이버시 침해문제와 같은 보안적인 문제를 갖고 있다. 본 논문에서는 RFID 환경에서 안전하지 않은 Tag 나 Reader 기를 식별하는 기술인 보안프로토콜을 정형적 검증방법인 모델체킹을 이용하여 신뢰성있는 네트워크를 구축하고자 한다.

1. 서 론

RFID[1][2]를 위한 컴퓨팅 환경은 일반적인 인터넷 환경과는 달리 많은 제약사항을 갖는다. 이러한 제약사항은 Cellular Phone 등을 이용한 무선 인터넷보다 더욱 자원 측면적 한계를 갖는다. 즉 유비쿼터스를 위한 RFID 환경을 구축하기 위해서는 모든 상품이나 사람 등 객체에 설치되는 Tag 가격은 5 센트 이하로 구현되어져야 하며 대신에 Reader 장비나 Back End 시스템에서 많은 성능, 자원 측면에서 열악한 Tag 장비의 자원적 한계를 극복할 수 있도록 설계 운영되어져야 한다. 이에 보안 기술 적용에 대한 부분도 이러한 운영, 환경 측면을 충분히 고려해야 한다. 본 논문에서는 물리적 레벨의 보호 기법이 아닌 암호 기술을 중심으로 한 RFID 에서의 보안 프로토콜을 분석한다.

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 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 향상시키고자 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 해쉬기반 보안프로토콜들에 대해서 설명하고 3 장에서는 프로토콜을 명세하고 검증하기 위한 Casper[4] 및 FDR 도구에 대해 소개하며, 4 장에서는 CSP, 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 연구방향을 제시하고자 한다.

2. 해쉬 기반 보안프로토콜

2.1 해쉬-락 스킴

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 운영 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

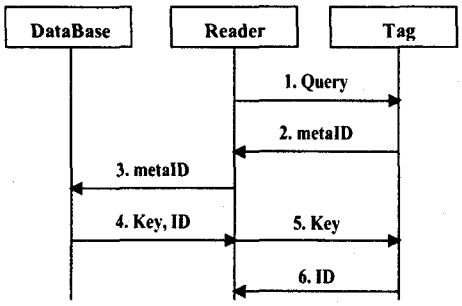

그림 1. 해쉬-언락킹 프로토콜

- 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 1. 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).

2.2 랜더마이즈 해쉬-락 스킴

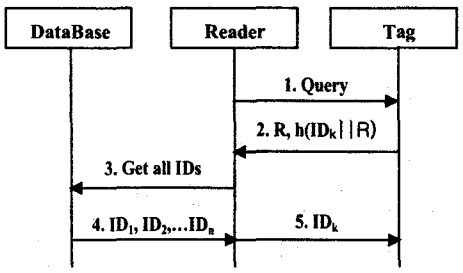

그림 2. 랜더마이즈 해쉬-언락킹 프로토콜

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 2. 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.

3. Casper 와 FDR 도구

3. 1 CSP(Communicating Sequential Process)

CSP[4]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

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

CSP(Communication Sequential Process)[4]언어를 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.

3.2 FDR(Failure Divergence Refinement)

FDR [6]도구는 CSP 명세언어를 입력으로 받아들이는 모델체킹 도구로서 옥스포드 대학에서 개발되었다. 이 도구는 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 도와준다.

FDR 도구는 3 가지의 검증방법을 지원하고 있다.

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

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

4.1 해쉬-언락킹 프로토콜 분석

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, R은 리더, T는 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 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

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 데이터베이스에 접근함으로써 태그정보를 가져갈 수 있도록 함으로써 중간자 공격과 재생공격을 방지할 수 있었다.

5. 해쉬기반 프로토콜의 취약성을 수정한 제안프로토콜

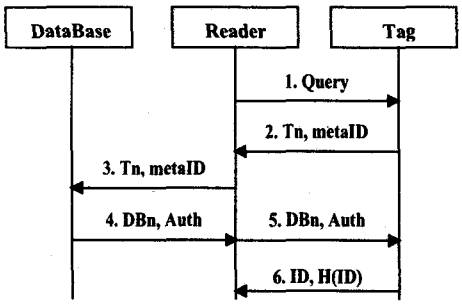

그림 5. 해쉬기반 제안프로토콜

위 프로토콜에서는 2가지 새로운 기술이 적용된다.

1. 첫번째로 태그의 난수와 데이터 베이스의 난수가 데이터 프라이버시와 태그 응답시의 악의적인 리더로부터의 재생공격을 막기 위해 도입된다. 따라서 태그는 난수재생기가 필요한데, 이는 장소추적을 막기 위해서 적어도 하나는 필요하다.

2. 통신참여자들간의 기밀성을 보장하기 위해 배타적합 (Exclusive-or) 기술이 도입되었다.

3. 리더와 태그, 데이터 베이스와 리더간의 안전한 채널을 구축하기 위해 또 다른 metaID 와 같은 타입의 데이터 베이스의 난수값과 리더의 원래 키값으로 이루어진 Auth라는 값이 사용되었다.

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 제시하였하고 이러한 문제점을 해결한 새로운 프로토콜을 제안하였다. 향후 연구과제로서 대칭적인 세션키 기반 경량화 보안프로토콜의 설계 및 검증을 해보고자 한다.

6. 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.