RFID 보안 프로토콜 취약성 분석 및 설계

오정현° 김현석 최진영
고려대학교 컴퓨터학과
{jhoh°, hskim, choi}@formal.korea.ac.kr

Vulnerability analysis and Design of RFID Security Protocols

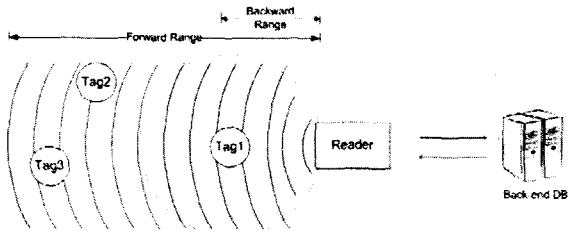
Junghyun Oh° Hyunsuk Kim Jinyoung Choi
Dept. of Computer Science and Engineering, Korea University

요 약

RFID기술은 RF를 이용하여 자동적으로 사물의 정보를 획득할 수 있는 매우 편리한 기술이다. 하지만 RF라는 매체를 사용하는 무선통신 환경에서 데이터를 주고받기 때문에 악의적인 공격자에 의해 사물의 의도적으로 노출이 될 수 있는 취약점을 지니고 있다. 이러한 RFID 시스템의 보안적 취약점을 보안하기 위해 RFID시스템에서 사용할 수 있는 강력한 보안 프로토콜의 요구가 높아졌다. RFID 시스템에서 사용될 보안 프로토콜을 설계하기 위해서는 비밀성, 익명성 등 반드시 고려해야할 요구사항이 있는데, 기존에 제안되었던 RFID 보안 프로토콜들은 이러한 요구사항들을 모두 완벽하게 만족시키지 못하였다. 본 논문에서는 RFID 시스템 프로토콜 모델을 제시하고, 정형기법을 사용하여 보안적 문제점들을 확인한 후, 문제점들을 보완하여 설계한 효율적인 RFID 보안 프로토콜을 제시하고자 한다. 또한 설계된 보안 프로토콜의 보안성을 정형기법을 통해 신뢰할 수 있는 검증을 실시하였다.

1. 서 론

RFID 시스템은 Radio Frequency를 사용하여 물리적인 접촉이 필요 없이 물품의 정보를 자동적으로 읽어 들이는 유비쿼터스 기술 중에 하나이다. 일반적으로 RFID 시스템은 태그와 리더 그리고 데이터베이스 서버로 구성되어 있다 (그림 1).



[그림 1] RFID 시스템

하지만 시스템을 구성하는 요소간의 통신 채널이 무선환경이라는 특수성 때문에 도청이 매우 용의하여 이를 통한 정보 누출(Data Leakage)과 누출된 정보를 통해 정보 추적(Traceability)이 가능하다는 취약점을 갖고 있다. 더하여 태그를 구성하는 하드웨어의 제약으로 유선통신 환경에서 사용하는 암호화 방법들을 사용할 수 없기 때문에 보안성과 경제성을 모두 만족시키는 인증 프로토콜의 개발이 필요하게 되었다.

이에 다양한 보안 프로토콜들이 제안되었으나, 보안성 및 경제성을 완벽하게 만족시키지 못하였고, 또한 직관적인 방법에 의해 프로토콜들의 보안성을 검증하여 명확한 검증이 이뤄졌다고 할 수 없다.

본 논문에서는 기존에 제안된 프로토콜들의 보안적 취약점을 분석하고, 문제점을 수정한 프로토콜을 제안하고자 한다. 또한 설계된 프로토콜의 보안성을 직관적인 방법을 사용하여 보안성을 분석한 기존의 프로토콜들과는 달리, 정형기법을 사용하여 제안 프로토콜의 보안성을 정형적으로 분석하고자 한다.

정형기법[1]은 수학적 논리나 이론을 바탕으로 하여 HW 또는 SW 시스템이 주어진 요구사항에 맞게 설계되었고, 안전하게 개발되었는지 확인 및 검증하는 방법론으로, 일반적으로 시스템의 동작 및 특성을 정형적으로 명세하는 정형명세와 정형명세된 시스템이 주어진 요구사항을 만족하는지 정형적으로 검증하는 방법인 정형검증으로 나뉜다.

보안 프로토콜을 정형검증 하는 방법은 BAN[2], GNY Logic[3] 등을 통한 정리증명과 SPIN, SMV 또는 FDR 등을 이용한 모델체킹 등 다양한 방법들이 있으나, 본 논문에서는 CSP 언어로 명세하고, FDR을 사용하여 모델체킹을 실시하여 보안 취약점 존재여부를 확인하고, 확인된 취약점을 보완한 프로토콜을 제안하고자 한다.

논문은 2장에서 RFID 시스템에 대해서 간단하게 소개하고, 3장에서 기존에 제안된 보안 프로토콜 및 정형기법에 대해 간략히 소개하였다. 그리고 4장에서 제안 프로토콜의 모델이 되는 기본 방법에 대해 소개하고, 이 방법의 보안적 취약점을 정형적 분석한 후, 5장에서 취약점을 보완한 프로토콜을 제시하였다.

2. RFID 시스템

RFID 시스템은 일반적으로 물품에 부착되어 사용되는 태그(Transponder), 태그 내부의 정보를 읽어 들이는 리

더(Transceiver) 그리고 태그 관련정보를 저장하여, 리더로부터 받은 태그의 정보를 인증하고 리더에게 관련정보를 보내주는 서버로 구성되어 있다.

태그는 내부에 자신 고유의 식별자를 갖고 있으며, 이 값은 태그가 수명을 다할 때 까지 고정적이거나, 서버와의 약속된 방법에 의해 가변적일 수도 있다. 서버는 모든 태그의 정보를 저장하고 있으며, 각각의 태그를 식별 및 인증할 수 있는 기능을 지니고 있다.

RFID 시스템 구성요소들 중에서 리더와 서버의 통신 채널은 유선통신이므로 기존의 유선망에서 사용되는 보안 프로토콜들을 사용이 가능하기 때문에 문제가 되지 않지만, 태그와 리더간의 통신채널은 무선통신이기 때문에 무선통신이 갖는 도청가능이라는 취약점을 갖고 있다. 다음은 대표적인 보안적 취약점을 정리한 것이다.

- 태그 정보 누출 - 태그 내에 저장된 정보의 누출
- 태그 위치 추적 - 태그 정보를 통한 위치 추적

위와 같은 이유로 RFID 시스템은 보안 프로토콜을 필요하게 되었으며, 보안 프로토콜은 설계 시 다음과 같은 보안 요구사항을 만족해야 한다.

- 자료의 비밀성 - 태그 관련 자료는 악의적인 공격자에게 노출되어서는 안 된다
- 자료의 무결성 - 태그 관련 자료는 악의적인 공격자에 의해 임의적으로 수정되어서는 안된다
- 자료의 익명성 - 프로토콜에서 사용되는 자료에 의해 태그가 추적되어서는 안 된다

3. 관련 연구

위와 같은 RFID 시스템의 보안 취약점을 제거하기 위해서 다양한 보안 프로토콜들이 제안되었는데, 대표적인 보안 프로토콜에는 Weis가 제안한 해쉬 락 기법 및 Randomized 해쉬 락 기법[4], Okubo가 제안한 해쉬 체인 기법[5], 그리고 Henrici가 제안한 해쉬 기반 ID Variation 기법[6]등이 있다. 하지만 이들은 태그의 정보의 익명성을 제대로 보장해주지 못해 위치추적이라는 취약점을 제거해주지 못하였다.

최근 이러한 태그의 익명성을 보장하기 위한 보안 프로토콜들이 다시 제안되었는데, 대표적인 것으로는 Tsudik이 제안한 YA-TRAP[7] 프로토콜과 Chatmon이 제안한 O-TRAP[7] 등이 있다. 이들은 Challenge-response 기법을 기본으로 사용하고, 리더와 태그간 Time-stamp 또는 서로 약속된 난수값을 사용하는 데이터를 통해, 태그와 리던간 상호인증을 이끌어 낼 수 있도록 설계되어 있다. 하지만 공격자에 의한 광범위한 Time-stamp 공격에 매우 취약하고, 태그와 리더가 사용하는 난수값을 수정하면, 서버에 저장되어 있는 해쉬테이블 전체의 내용이 모두 다시 갱신되어야 하기 때문에 효율성에도 문제가 있다.

더하여 이제까지 제안된 RFID 보안 프로토콜들은 직관적인 방법에 의한 보안성 검증을 실시하여, 프로토콜의 설계가 완벽하다고 볼 수 없으며, 실제 여러 보안적 취약점이 발견되고 있다. 이러한 이유로 정형기법을 통한 프로토콜의 설계 및 보안성 검증은 매우 중요하다고 볼 수 있다.

본 논문에서는 모델체킹이라는 정형기법을 사용하여 프로토콜의 보안성을 검증하였다. 모델체킹은 정형적으로 명세된 시스템을 모델체킹 툴에 입력하여, 시스템이 도달할 수 있는 모든 상태(State)에 의도적으로 도달하게 하여, 원하지 않는 상태로의 도달 여부에 대해 확인하고, 그럴 경우 어떠한 방법으로 도달하게 되는지 반례를 보여주는 방법론이다.

3.1 CSP(Communicating sequential Process)

CSP[8]는 프로세스 알지브라 언어로 시스템을 정형적으로 명세하는데 쓰이는 언어 중의 하나이다. CSP는 병렬성을 갖는 통신 프로토콜을 효율적으로 명세할 수 있어 통신프로토콜 및 제어 시스템을 명세하는데 많이 사용되었고, 분산 시스템 환경하에서 동작하는 클라이언트 서버와 공격자 모델을 정형적으로 명세할 수 있는 장점으로 보안 프로토콜을 명세하는 분야까지 확대되어 사용되고 있다. 분산시스템 환경하에서 동작하는 보안 시스템은 다음과 같이 표현할 수 있다.

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

3.2 CASPER(A Compiler for the Analysis of Security Protocols)

CSP는 통신 프로토콜을 효율적으로 명세할 수 있지만, 명세가 매우 어렵고 복잡하여 숙련된 프로그래머에 대한 의존도가 매우 높다. 이에 따라 프로그래머에 의한 오류코드가 포함될 수 있는 문제점이 존재한다.

CASPER[9]는 보안 프로토콜을 명세하는데 사용되는 도구로써, 프로토콜을 명세하는 방법이 매우 명료하고 간단하다. 그리고 CASPER는 간단한 문법을 통해 명세된 코드를 CSP 언어로 자동적으로 변환해주기 때문에 위에서 언급한 프로그래머에 의해 발생될 수 있는 문제점을 제거시켜준다.

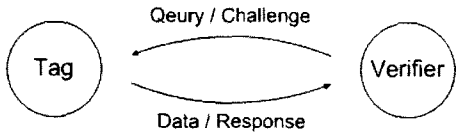
3.3 FDR(Failure Divergence Refinement)

FDR[10]은 CSP를 입력언어로 받는 도델체킹 도구로서, CSP로 명세된 프로토콜이나 보안 시스템이 그들이 제시하는 비밀성 또는 인증과 같은 보안속성을 만족 하는지 자동적으로 확인해준다. 이를 통해 해당 속성을 만족시키지 못할 경우에는 반례를 제시하여, 가능한 공격 시나리오 분석을 도와준다. 보안 프로토콜의 경우, 반드시 갖추어야 하는 비밀성, 무결성, 인증성, 부인방지와 같은 보안속성의 만족여부를 검증해주고, 만족시키지 못

할 경우 반례를 제시해준다

4. 기본 모델

일반적으로 보안 프로토콜을 설계할 때 서버와 리더는 유선망을 사용하기 때문에 기존에 사용하는 보안 알고리즘을 그대로 사용할 수 있으므로, 서버와 리더를 하나로 묶어서 생각하는 경우가 많다. 그림 2는 위와같은 가정을 전제로 RFID 시스템 구성요소를 추상화 한 것이다.



[그림 2] RFID 시스템

제안하고자 하는 프로토콜의 기본 모델이 되는 프로토콜은 Ari Juels[11]의 프로토콜이다. Ari Juels는 RFID 태그의 메모리가 태그 가격에 민감한 영향을 주기 때문에, 태그의 가격을 낮추기 위해 기존의 암호화 알고리즘을 사용하지 않고 메시지를 상호 공유하고 있는 비밀값과 XOR 계산하여 암호화하는 방법으로 태그와 검증자간 상호 인증하는 프로토콜을 제안하였다. 제안된 프로토콜에서 사용된 알고리즘은 One-Time Padding기법으로써, 태그에 Padding Factor를 두어 태그와 검증자간의 인증세션이 성공적으로 종료되었을 경우 이를 사용하여 태그와 리더에 저장된 비밀키와 Padding Factor를 가지고 XOR 계산을 시켜서 업데이트시키는 방법이다.

4.1 초기 가정(Assumptions)

주어진 프로토콜은 RFID 기술이 계속적으로 개발되고 있는 시점에서 볼 때 아직은 중간자 공격과 같은 강력한 공격이 현실적으로 불가능하다고 가정하고, 감청공격과 같은 수동적인 공격을 전제로 하여 설계되었다. 그러한 이유로 Ari Juels가 제시한 보안 프로토콜은 다음과 같은 2가지 가정을 기본 전제로 하고 있다.

- 태그에 대한 Query의 제한
- 태그와 검증자의 통신 중 공격자의 간섭 제한

RFID 시스템에서 태그와 검증자간의 통신은 매우 짧은 시간동안에 이루어지기 때문에 공격자는 이 짧은 인증시간 내에 공격을 실시해야하며, 실패했을 경우 태그에 대한 공격을 처음부터 다시 시작해야한다. 또한 태그와 검증자간 인증세션이 성공적으로 완료되면, 사용되었던 비밀키는 태그와 검증자간의 정해진 어떠한 절차에 의해 그 값이 바뀔 수 있으므로, 프로토콜 실행간 공격자의 간섭은 제한적일 수밖에 없다.

4.2 패딩(Padding) 기법

하나의 인증자(ID)가 하나의 태그를 대변하게 된다면, 태그 복제라는 문제가 일어날 수 있기 때문에, 하나의 태그가 여러 개의 인증자를 사용하여 인증 받도록 한다면 효율적일 것이다. 하지만 여러 개의 인증자를 사용하더라도 사용되는 인증자의 값들이 일정하다면 공격자에 의한 인증자 획득은 시간문제일 수 있다. 때문에 성공적인 인증과정이 종료되면 태그의 인증자 값을 Padding 기법을 사용하여 모두 갱신 시킨다면 이러한 문제점들을 해결할 수 있다.

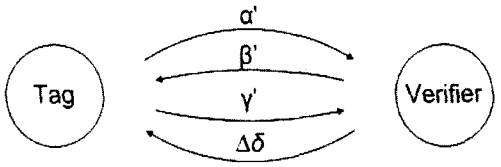
Padding 기법은 아주 간단한 고전 암호화 방법으로써, 메시지 M을 통신 객체간 공유하고 있는 Padding Factor와 XOR 계산을 하여 메시지를 암호화 하는 것이다.

주어진 프로토콜에서는 이와 동일하게 검증자와 태그가 상호 Padding Factor δ 값을 공유하고 있고, 태그와 검증자가 사용했던 인증자를 상호인증과정이 종료되면 Padding Factor δ와 XOR 함수를 사용하여 그 값을 갱신토록 하였다. 이로써, 인증자의 비밀성과 익명성이 보장 받게 되고, 더하여 동일한 Padding Factor에 의해 암호화되지 않은 경우 인증을 받을 수 없기 때문에 무결성까지 보장받게 되는 것이다. 태그는 인증자의 익명성을 높이기 위해 한 개가 아닌 여러 개의 Padding Factor 저장하여 사용할 수 있으며, 한 번 사용된 Padding Factor는 삭제되거나 또는 배열의 마지막으로 이동하여 재사용될 수 있다.

A Set of Padding Factors $\delta = \{\delta_1,\delta_2,\ldots,\delta_m\}$.

m값은 Padding Factor의 개수로써, 공격자의 도청공격에 대한 저항력의 정도를 가리키는 수치이다. 주어진 프로토콜에서는 태그 인증자의 익명성과 Padding Factor의 효율성을 더욱 높이기 위해 인증세션이 종료되면 검증자로 하여금 $\Delta\delta$ 값을 태그에게 보내어 기존의 δ값과 XOR 계산을 통해 갱신토록 하였다.

4.3 프로토콜 절차



[그림 3] 기본 프로토콜

주어진 프로토콜은 Challenge-Response 기법의 종류로써, 태그는 k개의 비밀키 $K= \{\alpha_i\}\cup\{\beta_i\}\cup\{\Upsilon_i\}$(단, $1 \le i \le k$)와 m개의 Padding Factor $\delta = \{\delta_1,\delta_2,\ldots,\delta_m\}$를 검증자와 상호 공유하고 있으며, 카운터(c)를 두어 비밀키 및 Padding Factor 선택에 사용한다. 검증자는 내부

에 모든 태그에 대한 비밀키들과 Padding Factor들을 저장하여, 태그 인증에 이를 사용한다.

가. 태그(Tag)
(ㄱ) i = (c mod k) + 1 을 계산
(ㄴ) c = c + 1
(ㄷ) $\alpha' = \alpha_i$ 선택 후 검증자에게 송신
나. 검증자
(ㄱ) 저장된 값 중에서 α'와 동일한 α_i 검색
(ㄴ) 있으면
대칭되는 β_i값을 찾아 $\beta' = \beta_i$ 송신
(ㄷ) 없으면 세션 종료
다. 태그
(ㄱ) $\beta' = \beta_i$ 이면, 상호 인증 성공
(ㄴ) Padding Factor를 갱신을 위해 Υ_i 값을 송신
라. 검증자
(ㄱ) $\Upsilon' = \Upsilon_i$ 이면, 태그 인증 성공
(ㄴ) $\Delta\delta$ 태그에게 송신
(ㄷ) 확인된 태그의 데이터 갱신
$\delta_i = \delta_i \oplus \Delta\delta$
$k = k \oplus \delta_i$
마. 태그
(ㄱ) $1 \le i \le m$, $\delta_i = \delta_i \oplus \Delta\delta$ 계산(갱신작업)
(ㄴ) $k = k \oplus \delta_1$ 계산 (비밀키 갱신)

4.4 취약점 분석

보안성 분석은 주어진 프로토콜이 정상적으로 상호인증을 이끌어 내는지와 사용되는 비밀키가 공격자에게 노출되지 않는지에 대해서 검증하고자 한다.

4.4.1 프로토콜 정형적 명세

```
#Free variables

T,V : Agent
a,b,c : SessionKey
InverseKeys = (a,a),(b,b),(c,c)

#Protocol description

0.    -> T : V
1.  T -> V : a
2.  V -> T : b
3.  T -> V : c

#Intruder Information

Intruder = Mallory
IntruderKnowledge =
          {Tag, Verifier, Mallory}
```

위는 Casper 도구를 사용하여 프로토콜을 명세한 것이다. #Free variables 부분은 프로토콜에서 사용되는 데이터의 유형 및 함수 등을 정의하는 부분이고, #Protocol description 부분은 어떠한 절차에 의해 객체가 통신이 이뤄지는지 명세된 부분이다. #Intruder Information 부분은 공격자가 누구이고, 어떠한 정보를 알고 있는지 명세하는 부분이다.

```
#Specification

Agreement(T,V,[a,b,c])
Agreement(V,T,[a,b,c])
Secret(T,a,[V])
Secret(T,b,[V])
Secret(T,c,[V])
```

#Sepcification 부분은 주어진 프로토콜이 만족코자 하는 요구사항에 대해서 명세하는 곳으로, 상호인증에 사용되는 비밀키 값들이 태그와 검증자 인증에 정상적으로 사용가능하다 것과 비밀키의 비밀성이 유지되어 공격자에게 노출되지 않는다는 것을 명세한 것이다.

4.4.2 검증결과

프로토콜의 검증은 태그와 검증자가 사용되는 비밀키 값을 비밀성(Secret) 및 이를 통한 효과적인 상호 인증 가능 여부(Agreement)에 대해서 실시하였다. 위의 프로토콜 명세코드를 CSP 언어로 변환하여 FDR 모델체킹 도구로 검증한 결과 보안 취약점을 발견할 수 있었다. 다음은 FDR이 제시한 반례 코드이다.

- Secret(T,a,[V])에 대한 반례코드

```
env.Tag.(Env0,Verifier,<>)
send.Tag.Verifier.(Msg1,A,<>)
receive.Tag.Verifier.(Msg1,A,<>)
send.Verifier.Tag.(Msg2,B,<A,B,C>)
receive.Verifier.Tag.(Msg2,B,<>)
send.Tag.Verifier.(Msg3,C,<A,B,C>)
leak.A
```

반례코드를 해석해보면 다음과 같다.

```
0.              -> Tag : Verifier
1. Tag          -> I_Verifier  :  A
1. I_Tag        -> Verifier    :  A
2. Verifier     -> I_Tag       :  B
2. I_Verifier   -> Tag         :  B
3. Tag          -> I_Verifier  :  C
 The Intruder Knows A
```

위의 해석을 보면 공격자가 중간자 공격을 통하여 태그와 검증자간의 인증 세션을 종료시키지 않으면서, 중간에서 비밀키 값 중에서 α_i값을 획득했음을 볼 수 있다. 위의 결과를 통하여 우리는 공격자가 위와 동일한 방법으로 α_i값뿐만 아니라 β_i값과 Υ_i값까지 획득할 수 있음을 유추할 수 있으며, 실제 FDR은 동일한 반례를 제시해 주었다.

그렇다면, 태그와 검증자가 난수화된 비밀키를 통해 상호 인증을 실시한다 하더라도, 인증 세션동안에는 정해진 비밀키를 사용하여야 하며, 종료되기 전까지는 비밀키 및 Padding Factor 가 갱신할 수 없기 때문에, 이런 방법으로 태그의 비밀키가 모두 노출이 될 수 있다는 것을 의미한다. 더하여 노출된 비밀키를 사용한 공격자의 위장공격으로 태그와 검증자간의 상호인증 또한 정상적으로 이뤄질 수 없다는 것도 알 수 있다. 즉, 주어진 프로토콜은 비밀성과 인증성 그리고 익명성에 있어서 취약점을 보여준다는 것을 확인할 수 있다.

5. 제안 프로토콜

5.1 수정사항

위의 프로토콜의 문제점은 비밀키로 사용되는 난수값이 인증 세션동안에는 노출이 된다는 점이다. 이는 프로토콜을 설계할 때 능동적인 공격자가 아닌 감청에 의존하는 수동적인 공격자에 대한 고려만을 했기 때문이다. 더하여 현실적으로 한 개의 태그에 여러 쌍의 비밀키 값과 여러 개의 Padding Factor를 모두 저장시켜 사용한다는 것은 매우 어렵다. 태그의 메모리는 태그의 가격에 매우 민감한 영향을 미치기 때문에, 용량을 늘릴수록 태그의 가격도 높아지게 된다. 그러기 때문에 여러 쌍의 비밀키를 사용하는 것은 사실상 불가능하다.

위에서 언급한 문제점을 해결하기 위해서 기존 프로토콜의 몇 가지 부분을 수정하였다.

첫째, 사용되는 비밀키를 3개의 난수쌍이 아닌, 2개로 수정하였다. 3개의 난수로 구성된 비밀키를 사용할 경우 상호 인증을 더욱 강력하게 실시할 수 있지만, 절차가 길어지고 메모리 요구가 크며, 수많은 태그와 통신을 실시할 경우 시스템의 효율성이 저하될 수 있다.

둘째, 해쉬함수를 사용하였다. 인증 세션간에 사용되는 비밀키를 해쉬함수를 사용하여 암호화 한다면, 공격자가 암호화된 비밀키를 획득한다 하더라도 내용을 알 수가 없게 된다. 물론 사용되는 비밀키가 일정하면, 내용을 알 수 없다 하더라도 위장공격을 통해 인증과정에 사용할 수 있으며, 복제 및 위치추적도 가능하다. 때문에 한 쌍의 비밀키를 사용하되 검증자간 인증과정에서 사전에 정해진 약속에 의해 Padding Factor를 사용하여 태그의 비밀키를 난수화하고, 해쉬함수를 사용해서 이 난수들을 암호화 한다면, 기존 프로토콜의 인증 알고리즘은 그대로 유지되면서, 비밀키의 비밀성, 무결성, 익명성 그리고 상호인증까지 만족시킬 수 있다.

셋째, Time-stamp를 사용하였다. 태그에 Tt라는 데이터를 저장하여, 가장 최근에 검증자로부터 받은 Tv값과 비교하여, 공격자에 의한 재생공격에 대응할 수 있도록 한다. 검증자가 Tv값을 보내오면 태그는 공격자 감청의 대응력을 나타내는 수치 m, 즉 저장된 Padding Factor의 개수로 Modulor 계산을 실시하여 사용할 Padding Factor를 선택한다. Padding Factor가 선택되면 태그에 저장된 비밀키 α와 XOR 계산을 실시하여 검증자에게 보낼 값을 생성하는 것이다. $\alpha_i = \alpha \oplus \delta_i$.

이렇게 하면 동일한 세션 내에서도 서로 다른 비밀키를 사용하는 효과를 얻을 수 있으므로, 공격자는 비밀키 획득이 더욱 어렵고, 인증세션이 종료되면 비밀키와 Padding Factor 값도 바뀌기 때문에 기존에 획득한 비밀키 또한 사용할 수 없다.

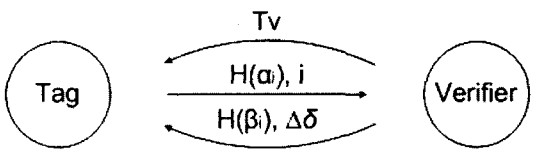
추가적으로 검증자는 내부에 해쉬테이블을 사용하여 단위 태그당 인증에 걸리는 시간을 줄일 수 있다. 다음은 검증자 내부에 저장될 해쉬테이블의 예시이다.

[표 1] 해쉬 테이블

i	T_1	T_2	…	T_n
1	$H(\alpha_{1,1})$	$H(\alpha_{2,1})$	…	$H(\alpha_{n,1})$
2	$H(\alpha_{1,2})$	$H(\alpha_{2,2})$	…	$H(\alpha_{n,2})$
⋮	⋮	⋮	⋮	⋮
m	$H(\alpha_{1,m})$	$H(\alpha_{2,m})$	…	$H(\alpha_{n,m})$

이와 같이 해쉬테이블을 사용하게 되면 단위 태그당 인증에 걸리는 시간이 매우 절약되고, YA-TRAP 프로토콜이나 O-TRAP 프로토콜과 같이 난수값이 바뀔때마다 해쉬테이블 전체를 갱신할 필요 없이 확인된 태그의 데이터만 갱신하면 되기 때문에 효율성이 보다 우수하다.

5.2 프로토콜 절차



[그림4] 제안 프로토콜

가. 검증자(Verifier)

(ㄱ) Tv (Time-stamp)를 태그에게 보낸다

나. 태그(Tag)

(ㄱ) i = (Tv mod m) + 1 을 계산

(ㄴ) i번째 Padding Factor를 검색

(ㄷ) $\alpha_i = \alpha \oplus \delta_i$, $H(\alpha_i)$ 을 계산

(ㄹ) $H(\alpha_i)$ 와 i 를 검증자에게 송신

다. 검증자
(ㄱ) 저장된 값 중에서 α_i와 동일한 값이 있는지 확인
(ㄴ) 있으면 태그 인증 성공
대칭되는 β_i값를 찾아 $H(\beta_i)$ 및 Padding Factor를 갱신하기 위한 $\Delta\delta$를 송신
(ㄷ) 없으면 세션 종료
라. 태그
(ㄱ) $H(\beta_i)$와 $\Delta\delta$를 받으면 상호 인증 성공
(ㄴ) $\delta_i = \delta_i \oplus \Delta\delta$ 계산 실시(갱신작업)

5.3 Casper를 사용한 보안 프로토콜 설계 및 검증

다음은 수정된 보안 프로토콜을 Casper로 명세한 프로토콜 코드이다.

```
#Protocol description
0.    -> T : V
1.  T -> V : H(a)
2.  V -> T : H(b)
```

코드를 보면 기존의 프로토콜에 비해 절차가 매우 간소화되었다는 것을 알 수 있다. 보안성 검증은 동일하게 비밀키의 비밀성과 이를 이용한 효과적인 상호 인증 가능 여부에 대해서 실시하였다.

```
#Specification

Agreement(T,V,[a,b])
Agreement(V,T,[a,b])
Secret(T,a,[V])
Secret(T,b,[V])
```

위와 같이 정형적으로 명세된 프로토콜을 FDR을 통해 보안성을 검증한 결과 문제가 없음을 확인하였다.

6. 결론 및 향후과제

RFID 기술은 매우 편리한 기술이지만, 기술이 지니고 있는 보안적 취약점을 제거하지 못한다면, 개인에게 큰 피해를 줄 수 있는 기술이기도 한다.

본 논문에서는 RFID 기술의 보안적 취약점에 대해서 분석하고, 이러한 취약점을 제거하기 위해 기존에 제안된 프로토콜들을 소개하였다. 하지만 기존에 제안된 프로토콜들은 기본적으로 요구되는 세가지 보안요구사항을 모두 만족시키지 못하였으며, 검증 또한 직관적인 방법에 의해 실시되어, 그들의 보안성 검증이 완벽하지 않았음을 정형기법을 통한 검증 결과로 확인 할 수 있었다.

제안하는 프로토콜 기존에 제안된 Ari Juels의 프로토콜이 데이터의 무결성과 익명성은 만족하나, 데이터의 비밀성을 만족시키지 못한다는 것을 모델체킹을 통해 확인하고, 확인된 취약점을 보완하여 설계한 것이다. 또한 설계된 프로토콜을 다시 모델체킹 사용하여 보안성을 검증한 결과 문제점을 보이지 않음도 확인하였다.

향후 과제로써, 제안된 프로토콜에서 사용되는 Padding 기법에 의해 태그 인증시 사용되는 난수 값이 중복 가능문제와, 프로토콜을 사용하기 위해 필요한 태그 메모리의 용량문제에 대해 연구하고자 한다.

7. Acknowledgement

이 연구에 참여한 연구자(의 일부)는 '2단계 BK21사업'의 지원비를 받았음

8. 참조문서

[1] E. M. Clarke and J. M. Wing, Formal Methods: State of the Art and Future Directions, ACM Computing Surveys, vol. 28, No. 4, pp.626-643, 1996
[2] M. Abaid, M. Burrow, and R. Needham. A Logic of Authentication, Proceedings of the Royal Society, Series A,426,1871,pp.233-271, December 1989
[3] L. Gong, R. Needham. R, Yahalom, Reasoning about Belief in Cryptographic Protocols, IEEE, 1990
[4] S.E. Sarma. Weis, and D.W. Engels, RFID systems, Seucrity and Privacy Implications, White Paper IT-AUTOID-WH-014, AUTO-ID CENTER, 2002
[5] M. Ohkubo, K Suzuki, and S. Kinochita, Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low Cost RFID, Proceedings of the SCIS 2004, pp.719-724, 2004
[6] D. Henrici, P. Muller, Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers, Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops(PERCOMW'04), pp.149-153, IEEE, 2004
[7]Christy Chatmon, Tri van Le, and Mike Burmester. Secure anonymous RFID authentication protocols. Technical Report TR-060112, Florida State University, Department of Computer Science, Tallahassee, Florida, USA, 2006.
[8]Stephen Brookes, C. A. R. Hoare, and A. W. Roscoe, "A Theory of Communicating Sequential Processes", Journal of the ACM, vol. 31, no. 3, pp.560-599, Jun 1984
[9]Gavin Lowe, "Casper: A Compiler for the Analysis of Security Protocols", In Proceedings of The 10th Computer Security Foundations Workshop, 1998
[10]Gavin Lowe, "Breaking and Fixing the Needham Schroeder Public Key Protocol using FDR"
[11]Ari Juels. Minimalist cryptography for low-cost RFID tags. In Carlo Blundo and Stelvio Cimato, editors, International Conference on Security in Communication Networks -- SCN 2004, volume 3352 of Lecture Notes in Computer Science, pp. 149--164, Amalfi, Italia, September 2004. Springer-Verlag.