http://dx.doi.org/10.3745/KIPSTC.2011.18C.5.331

RFID와 리더간의 안전성이 검증된 키 분배 프로토콜의 개선

배 우 식[†] · 이 종 연[††]

요 약

RFID 시스템은 리더와 태그간의 무선 통신 구간이 존재한다. 이 부분은 항상 보안 취약성으로 공격자의 목표가 되며 기밀누설, 프라이버시 침해등 보안적인 문제가 발생한다. 이와 관련하여 기존에 여러 가지의 프로토콜이 제안된 바 있으나 구현이 까다로워 대부분 이론적 정리증명의 수준이서 머물러 보안 프로토콜의 안정성이 정확히 증명되지 못했다. 따라서 본 논문에서는 특히 기존에 제안된 Kenji *et al.* 의 프로토콜이 보안 속성을 만족하는지 검사하고 ID 및 메시지의 노출 등 취약성을 확인하였다. 이 문제의 해결책으로 공개키 및 난수를 줄여 개선한 RFID 보안 프로토콜을 제안한다. 주요 특징으로는 불필요한 계산을 없애고 보안상으로 취약성이 없도록 구성했다. 안전한 프로토콜을 개발 및 검증하기 위해 Casper 및 FDR(Failure Divergence Refinements) 도구를 이용하여 확인하고 제안한 프로토콜이 보안상으로 안전함을 확인했다. 아울러 본 연구의 학술적 기여는 다음과 같이 요약된다. 첫째 정리증명에서 벗어나 모델 체킹으로 보안성을 검증하였다. 둘째 FDR 검증으로 프로토콜의 개발을 좀 더 효과적으로 할 수 있는 방법을 제시하였다.

키워드 : RFID 보안, 보안 프로토콜, 프라이버시, 정형검증, 키 분배

Improvement of a Verified Secure Key Distribution Protocol Between RFID and Readers

Woo Sik BAE[†] · Jong Yun LEE[††]

ABSTRACT

The RFID system includes a section of wireless communication between the readers and the tags. Because of its vulnerability in terms of security, this part is always targeted by attackers and causes various security problems including the leakage of secret and the invasion of privacy. In response to these problems, various protocols have been proposed, but because many of them have been hardly implementable they have been limited to theoretical description and theorem proving without the accurate verification of their safety. Thus, this study tested whether the protocol proposed by Kenji *et al.* satisfies security requirements, and identified its vulnerabilities such as the exposure of IDs and messages. In addition, we proposed an improved RFID security protocol that reduced the number of public keys and random numbers. As one of its main characteristics, the proposed protocol was designed to avoid unnecessary calculations and to remove vulnerabilities in terms of security. In order to develop and verify a safe protocol, we tested the protocol using Casper and FDR(Failure Divergence Refinements) and confirmed that the proposed protocol is safe in terms of security. Furthermore, the academic contributions of this study are summarized as follows. First, this study tested the safety of a security protocol through model checking, going beyond theorem proving. Second, this study suggested a more effective method for protocol development through verification using FDR.

Keywords : RFID Security, Security Protocol, Privacy, Formal Validation, Key Distribution

1. 서 론

RFID(Radio Frequency Identification) 시스템은 일반적으로 태그(Tag), 리더(Reader) 및 데이터베이스(Database)로 구성되어있다. 이 기술은 무선 주파수를 이용하여 사물에 부착된 태그로부터 물리적 접촉 없이 정보를 읽고 처리할 수 있는 기술이다. 아울러 바코드에 비해 태그에 많은 데이터를 저장 할 수 있고 수동형 태그의 경우 반영구적으로 사

† 정 회 원 : 충북대학교 컴퓨터교육과 박사수료
†† 종신회원 : 충북대학교 컴퓨터교육과 교수(교신저자)
논문접수 : 2011년 7월 5일
수 정 일 : 1차 2011년 9월 15일, 2차 2011년 10월 5일
심사완료 : 2011년 10월 6일

용할 수 있다. 이러한 장점으로 기존의 바코드를 대체하여 물품관리를 함으로써 최근 의료, 유통, 물류, 환경, 보안 등 산업계에서 많은 도입이 진행되어지고 있다. 그러나 RFID는 리더와 태그 구간의 통신이 무선주파수를 이용하여 이루어짐으로 인하여 공격자의 공격에 취약점이 있다. 이러한 보안상 문제로 사생활침해, 산업기밀 유출 등 취약점이 드러나게 되었다. 따라서 보안문제점을 극복하기 위하여 암호학적으로 보안성을 높이는 다양한 연구가 진행되고 있다. 기존 제안된 프로토콜들을 보면 해시락 기법[1,2,3], 암호화 기법[4,5]등의 인증기법이 있는데 제안된 기법들은 각종 보안 취약점을 가지고 있음이 많은 연구자[6-11]에 의해 발견되었다. 보안상으로 취약한 RFID 시스템은 프라이버시 침해문제 등으로 향후 산업계에서 제한적으로 사용할 수밖에 없으며 고부가가치를 창출할 수 없게 된다.

따라서 본 논문에서는 RFID 보안문제를 해결하기 위한 프로토콜을 제안한다. 기존에 제안된 Kenji *et al.*의 제안프로토콜[12]의 프로토콜 동작을 정형 명세언어로 설계한 후 보안속성을 만족하는지 검사하고 ID 및 메시지의 노출 등의 취약성을 분석하여 개선한 프로토콜을 제안한다. 여기서 보안 프로토콜의 검증은 크게 정리증명과 모델검증으로 연구되고 있는데 정리증명은 논리식을 사용하여 요구되는 특성을 정리하여 표현하는 방법이다. 모델검증은 시스템에 대한 유한상태 모델과 검사하고자 하는 시스템이 요구속성을 만족하는지 정형명세한 후 자동으로 검사하는 방법이 모델검증이다. 본 논문에서는 모델검사기법으로 효율성을 인정받고 있는 Casper[13,14,15]와 FDR[16]을 이용하여 제안한 프로토콜이 보안상으로 안전함을 증명한다[17]. 아울러 세부적인 연구내용은 다음과 같다. 첫째, 기존의 Kenji *et al.*의 제안프로토콜[12]을 알아보고 FDR 검증을 실시하여 문제점을 확인한다. 둘째, 제안프로토콜을 검토하고 프로토콜을 명세한다. 셋째, Casper FDR 도구로 정형검증을 실시하여 제안프로토콜의 보안성을 검증한다. 끝으로 본 논문의 연구결과는 기존의 정리증명에서 벗어나 정형검증기법으로 제안프로토콜의 보안성을 검증하여 실제시스템에 적용할 수 있는 기틀을 마련한다.

본 논문의 구성은 다음과 같다. 2장에서는 모델검사 도구에 대해 설명하고, 3장에서는 관련연구에 대해 기술하고, 4장에서는 제안프로토콜의 명세하고, 검증 및 안전성을 비교한다. 마지막으로 5장에서는 결론을 요약한다.

2. 모델 검사 도구

2.1 CSP

CSP(Communicating Sequential Processes) 언어는 병렬성을 갖는 통신프로토콜의 동작을 효율적으로 명세하기위해 제작되었다.[18] 초창기 일반적인 통신프로토콜과 제어시스템을 명세하기위해 사용되었지만 점차 보안프로토콜을 명세하기 위한 영역으로 확대되고 있다. CSP에서는 순수 동기화(pure synchronization:|||)와 인터리빙 패럴리즘(interleaving parallelism:||)의 개념을 사용하여 분산시스템 환경에서 동작하는 서버·클라이언트 및 공격자 모델을 정형적으로 표현할 수 있다는 장점이 있다. 예로써 다음과 같이 분산시스템 환경에서 동작하는 보안시스템은 다음과 같이 표현될 수 있다.

$$SYSTEM = CLIENT1 \;|||\; CLIENT2 \;|||\; SERVER \;||\; INTRUDER$$

2.2 Casper

CSP로 프로토콜을 명세하기 쉽게 개발되어진 컴파일러이다. Casper(a Compiler for the Analysis of Security Protocols)[7]에서 명세하기위해 8개의 세부항목으로 분류하는데 각 항목의 헤더부분은 #으로 시작하며 다음과 같다.

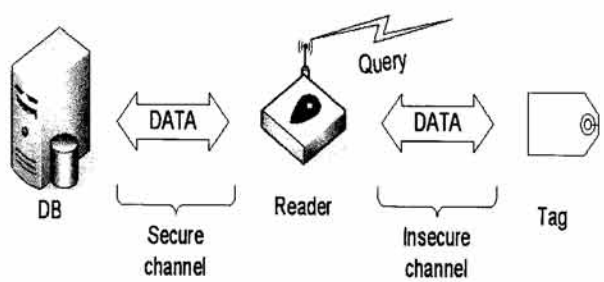
#Free variables : 변수타입 및 함수선언
#Process : 통신 에이전트의 초기상태
#Protocol description : 에이전트 간의 메시지교환 순서나열
#Specification : 보안 속성선언
#Actual Variables : 실제 데이터타입 및 이름선언
#Functions : 함수선언
#System : 에이전트의 초기상태표현
#Intruder Information : 공격자의 초기상태정보

2.3 FDR

CSP를 입력언어로 받아 모델속성을 만족하는지 검사하는 모델검사도구이다.[16] 만일 만족하지 않을 경우 반례(counterexample)를 보여주어 보안취약점을 분석하기에 용이하다. 안전성 검증, 교착상태 검증, 라이브락 검증의 3가지 검증방법을 지원한다. 보안프로토콜의 요구사항인 비밀성, 무결성, 인증, 부인방지 등의 속성을 만족하는지 검사하며 추적모델, 실패모델, 실패/분기모델을 지원한다.

3. 관련 연구

RFID 시스템은 크게 데이터베이스(DB), 리더, 태그의 세 부분으로 구성되며 (그림 1)은 일반적인 RFID 인증프로토콜 모델이다. 데이터베이스와 리더구간의 통신은 일반적으로 안전한 유선방식이며 리더와 태그구간은 무선구간이며 보안상 취약한 부분으로 이를 보강하기 위한 다양한 연구가 진행되고 있다.



(그림 1) RFID 인증프로토콜 모델

3.1 해시락 프로토콜의 문제점

이 방법은 태그의 식별 값인 metaID가 고정되어있어, 출력되는 데이터가 같아 해당 태그로부터 데이터가 전송되었는지 확인할 수 있게 된다. 그리고 리더기와 태그사이의 통신채널은 도청이 가능하기 때문에 악의적인 공격자는 키(Key)를 획득한 후, 해시연산하고 metaID를 산출하여 인증을 받을 수 있다. 또한 제 3자가 고정된 metaID를 재전송함으로써 인증 받을 수 있으며, metaID가 식별자처럼 사용되기 때문에 스푸핑 공격 및 사용자 추적이 가능하다.

프로토콜 명세는 다음과 같이 표현되어진다.

(1) Tag → Reader : metaID
(2) Reader → DB : metaID
(3) DB → Reader : Key
(4) Reader → Tag : Key
(5) Tag → Reader : ID

3.1.1 Casper를 이용한 해시락 프로토콜의 명세

(그림 2)는 해시락 프로토콜을 Casper로 명세한 8개 영역 중 중요한 3개 영역이다. Free variables에서 R, T는 클라이언트, DB는 서버이며 프로토콜의 Agent를 나타낸다. 키(key)는 한 번의 세션에서만 사용하기 때문에 세션키로 명세했다. InverKey는 Session키에 대한 암호화를 표현하며, H는 해시함수를 나타낸다.

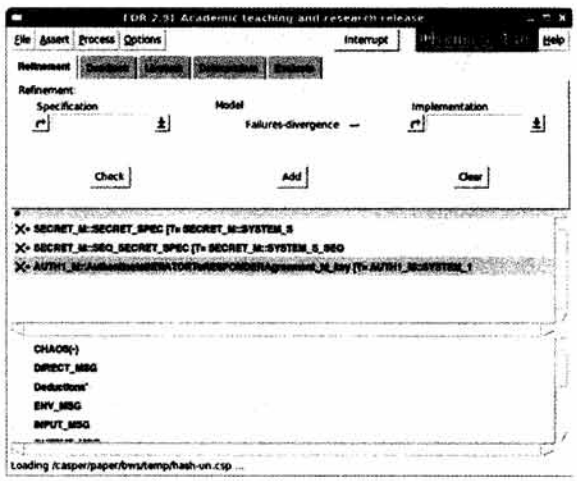
```
#Free variables
R, T : Agent
DB : Server
key : SessionKey
Id : Text
H : HashFunction
InverseKeys = (key, key)
#Protocol description
0.    -> T : R
1.  T -> R : (H(key)) % metaID
2.  R -> DB : metaID % (H(key))
3.  DB -> R : key, Id
4.  R -> T : key
5.  T -> R : Id
#Intruder Information
Intruder = Mallory
IntruderKnowledge =
      {Tag, Reader, DataBase}
```

(그림 2) 해시락 프로토콜의 명세

해시락 프로토콜을 FDR 검증결과 (그림 3)과 같이 취약성이 발견되었다. 이는 metaID의 값을 중간자 공격 및 재생공격에 이용함에 따라 태그정보의 노출 및 추적이 가능한 문제가 있었고, 때문에 리더와 태그의 인증이 보안 상 실패하게 되었다

3.2 Kenji et al. 프로토콜[6]의 문제점

Kenji *et al.*은 One-time ID를 사용하고 TTP(Trusted Third Party)에 기반으로 한 키 분배 프로토콜을 제안하였다.



(그림 3) 해시락의 FDR 검증결과

〈표 1〉 프로토콜에서 사용하는 기호정의

표현	의미
A,B,S	통신참여자
x,y	A와 B가 생성한 난수
g^x, g^y	C와 R이 생성한 Diffie-Hellman 키
{M}K	키 K로 암호화된 메시지 M
kAB	A 와 B 의 상호 인증키
O_{AS}	One-time ID
N_b	랜덤값

이 프로토콜은 시스템에서 서버가 사용자에게 사전에 배포한 것으로 가정하여 사용자와 서버간의 비밀값을 이용하여 One-time ID를 생성한다. 리더와 서버가 동기화된 연속된 숫자 1을 매번 프로토콜이 동작될 때마다 1씩 증가시킨다. 리더와 서버는 1씩 증가시킨 i와 공유된 비밀값을 공격자가 계산할 수 없도록 복잡한 공식에 적용하여 계산하고 이 값을 세션에서 One-time ID로 사용한다. 서버는 One-time ID를 수신한 후 자신이 생성한 값과 비교를 통해 식별하게 된다. 공격자는 복잡한 공식으로 One-time ID를 역으로 계산할 수 없어 공격하지 못하게 되는 방식이다. 사용할 기호의 정의는 <표 1>과 같다.

프로토콜 명세는 다음과 같이 표현되어 진다.

(1) A → B : $S, O_{AS}, (B, g^x, O_{AS})_{kAS}$
(2) B → S : $O_{AS}, (B, g^x, O_{AS})_{kAS}, (g^y, O_{AS})_{kBS}$
(3) S → B : $S, (A, g^x, g^y, (g^x, g^y)_{kAS})_{kBS}$
(4) B → A : $g^x, (g^x, g^y)_{kAS}, (g^x, N_b)_{kAB}$
(5) A → B : $(N_b - 1)_{kAB}$

프로토콜의 명세에 대한 설명은 다음과 같다.

(1) A는 S와 공유되어 있는 O_{AS}를 공개키로 암호화하여 B에게 전송한다.
(2) B는 자신이 생성한 g^y를 O_{AS}와 함께 암호화하여 S에게 전송한다.

(3) S는 B의 메시지를 수신한 후 복호화하고 g^x, g^y를 kBS로 암호화하여 B에게 전송한다.

(4) B는 메시지를 복호화하여 암호화된 g^x와 N_b를 kAB로 암호화하여 A에게 전송한다.

(5) A는 메시지를 복호화하고 B가 생성한 N_b를 암호화하여 전송한다.

3.2.1 Casper를 이용한 Kenji *et al.* 프로토콜의 명세

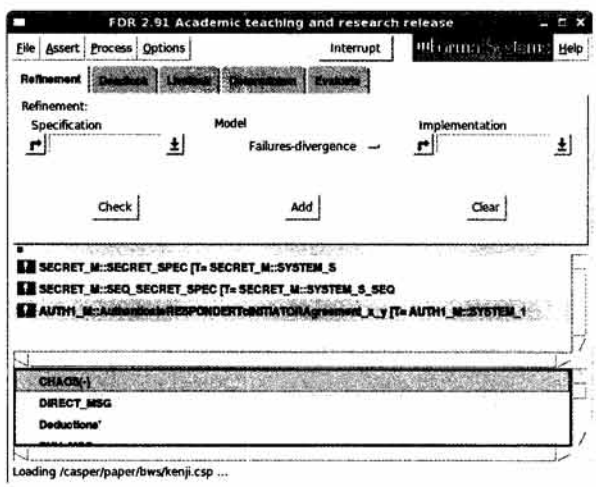
(그림 4)는 Kenji *et al.*의 프로토콜을 Casper로 명세한 8개 영역중 중요한 3개 영역이다. Free variables에서 A, B는 클라이언트, S는 서버이며 프로토콜의 Agent를 나타낸다. pkas, pkbs는 공개키이며, skas, skbs는 암호화된 비밀키이며 암·복호화를 나타낸다[19].

```
#Free variables
A,B : Agent
S : Server
x,y,z,id : Nonce
pkas,pkbs : PublicKey
skas,skbs : SecretKey
InverseKeys = (pkas,skas),(pkbs,skbs),(x,x),(y,y)
#Protocol description
0.   -> A : B
1. A -> B : S,id,{B,x,id} {pkas}%encl
2. B -> S : id,encl%{B,x,id} {pkas},{y,id} {pkbs}
3. S ->B : S, {A,x,y, {x,y} {pkas}%enc2} {pkbs}
4. B ->A : x,enc2% {x,y} {pkas} , {{x,z} {x}} {y}
5. A ->B : {{z} {y}} {x}
#Intruder Information
Intruder = Mallory
IntruderKnowledge =
      {Alice,Bob,Sam,Mallory,M,IDm,PKms,SKms}
```

(그림 4) Kenji, Kouichi의 Casper 프로토콜 명세

3.2.2 Kenji et al. 프로토콜의 FDR 검증

FDR 도구를 이용하여 Kenji *et al.*이 제안한 프로토콜의 안전성에대한 여부를 확인한 결과, (그림 5)와 같이 취약한 부분이 발견되었다.



(그림 5) Kenji *et al.*의 제안프로토콜 취약점

이 프로토콜은 One-time ID(OID)를 암호화하지 않은 형태로 전송하여 공격자가 첫 번째 메시지를 가로채서 다음 세션에서 두 번째 메시지를 위조할 수 있다. 이는 A의 일회성 ID와 메시지가 외부에 노출되어 공격자가 자신이 만들어낸 동일한 세션키를 이용하여 정상적인 인증 참여자로 위장이 가능하다. 서버는 공격자가 OID를 사용하여 세션키를 생성하고 인증을 시도할 경우 막을 수 없게 된다. 또한 이 프로토콜은 리더가 전송하는 OID에 대한 확인을 통해 승인되지 않은 사용자의 인증시도를 차단하는 프로토콜의 목표를 달성하지 못한다. 이로 인하여 공격자가 메시지를 재작성하여 공격을 할 수 있기 때문에 무결성(Integrity)에 취약하다. 그리고 자신의 공유키 수령자가 실제로 공유키를 소유하는지 확신하지 못해 암호화키 확인(Key confirmation)에 취약하다. 아울러 서비스의 안전성을 확보하기 위해 불필요한 요청을 식별하고 차단할 수 없게 되어 DoS공격에 취약함을 알 수 있다.

4. 제안하는 프로토콜

따라서 본 논문에서는 Kenji *et al.*의 프로토콜 방식을 개선하여 One-time ID, 난수를 사용하고 키 분배를 기반으로 새로운 프로토콜을 제안한다. 시스템에서 서버가 사용자에게 사전에 배포한 것으로 가정하여 사용자와 서버간의 비밀값을 이용하여 One-time ID를 생성한다. 이 프로토콜은 리더와 서버가 난수를 이용한다. 서버는 One-time ID를 수신한 후 자신이 생성한 값과 비교를 통해 식별하게 되고, 공격자는 One-time ID를 역으로 계산할 수 없어 공격하지 못하게 되는 방식이다. 사용할 기호정의는 <표 2>와 같다.

〈표 2〉 제안프로토콜의 기호정의

기 호	설 명
T	태그
R	리더
S	서버
x, k	T와 R 이 생성한 난수
{M}K	키 K로 암호화된 메시기 M
a1, a2	상호인증키
OAS	One-time ID

4.1 Casper 명세

(그림 6)은 제안하는 프로토콜의 Casper 명세코드이다. 보안프로토콜에서 사용되는 변수유형 선언, 동작절차, 공격자모델 등 중요한 3개의 영역을 나열하였다. 변수들과 함수타입은 #Free variables에 정의된다. T, R은 태그와 리더이며 S는 서버를 표현한다. 변수 x, k는 Nonce 타입이며 이는 유효한 메시지의 타입을 정의한다. a1, a2는 태그와 리더가 생성한 One-time ID이며, 한 번의 세션에서만 사용하기 때문에 세션키로 명세한다. Inversekeys (k,k),(a1,a1),(a2,a2),(x,x)는 각 함수별 서로의 역의 키들을 반환 한다는 의미로 선언

```
#Free variables
T, R : Agent
S : Server
x, k : Nonce
a1, a2 : SessionKey
InverseKeys = (k,k),(a1,a1),(a2,a2),(x,x)
#Protocol description
0.   -> T : R
1. T -> R : {a1,x}{a1}%enc1
2. R -> S : {enc1%{a1,x}{a1},a2,k}{a2}
3. S -> R : {T,x,{k}{a1}%enc2}{a2}
4. R -> T : enc2%{k}{a1},{{x,k}{x}}{k}
5. T -> R : {k}{x}
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {Tag, Reader, DB, Mallory, M, A3}
```

(그림 6) 제안하는 프로토콜의 Casper명세

되었다. #Protocol description에는 프로토콜에서 메시지들에 대한 정의이다. 정수 0. 1. 2 등은 전달되는 메시지의 순서를 나타낸다. #Intruder Information은 공격호스트의 이름과 초기정보를 정의한다. 공격자의 이름은 Mallory이며 공격자는 호스트가 Tag, Reader, DB, Mallory, M, A3 라는 것을 알고 있다는 가정이다.

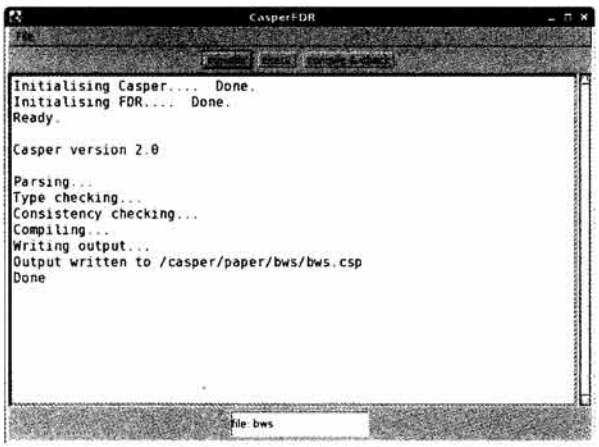
4.2 CSP 명세

CSP(Communication sequential Processes)는 통신프로토콜의 행위를 정형적 명세를 위해 개발된 프로세스 알제브라 언어의 일종이다[17]. CSP 언어는 프로세스의 동시성을 표현하기에 적합하여 보안프로토콜을 명세하고 FDR 모델검사 도구를 이용하여 취약성을 분석한다. 제안하는 프로토콜의 중요한 메시지 부분의 명세를 <부록>으로 나타내었다. Type of principals는 주요 함수타입을 나열하였으며, Channel declarations 항목은 INPUT, OUTPUT, DIRECT 등의 선언부분이다. Define type of signals, and declare signal channel 항목은 신호의 데이터타입을 정의하며 또한 채널의 신호를 차례대로 선언하여 신호 순서대로 연산되는 메시지를 나타내었다.

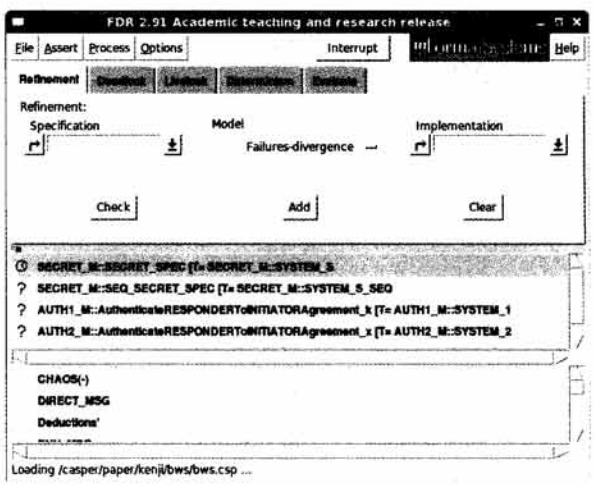
4.3 실험 및 결과

CSP 언어는 병렬성을 갖는 통신프로토콜의 동작을 명세하기위해 사용된다. 하지만 CSP 언어를 명세하는 방식이 매우복잡하고 실수할 가능성이 높아 프로토콜을 정확히 분석하기 어려운 단점이 있다. 이런 문제를 해결하기 위해 CasperFDR의 컴파일 명령을 이용하여 컴파일을 실행하면 (그림 7)와 같이 CSP 파일의 생성이 이루어진다.

다음으로 FDR 2.91 버전의 모델검증 도구를 이용하여 본 논문에서 설계한 프로토콜의 안전성(safety), 교착상태(deadlock), 라이브락(livelock) 등의 동작을 검증하기위해 FDR을 실행한다. (그림 8)은 소스파일을 로딩하여 기본적인 오류 없이 검증할 준비가 되어있는 상태를 나타낸다. 상태



(그림 7) Casper의 CSP 파일로 변환



(그림 8) 제안프로토콜의 보안 검증과정

표기 ? 상태는 아직 항목별 오류검증이 실행되지 않은 상태로 실행준비중 이라는 표현이다. ◷기호는 검증 실행중인 상태를 나타내는 기호로 표현된다. 각 항목별 검증실행을 하기위해 마우스의 오른쪽버튼을 누른 후 Run 또는 Debug를 선택하여 검증 또는 취약점 수정을 실행한다.

한편, 제안하는 프로토콜을 FDR 도구를 이용한 프로토콜 검증결과 (그림 9)와 같이 모든 보안속성에 대한 만족이 확인되었다.

(그림 9)에는 4가지 검증결과가 제시되며 각 결과의 표현은 다음과 같이 분석된다.

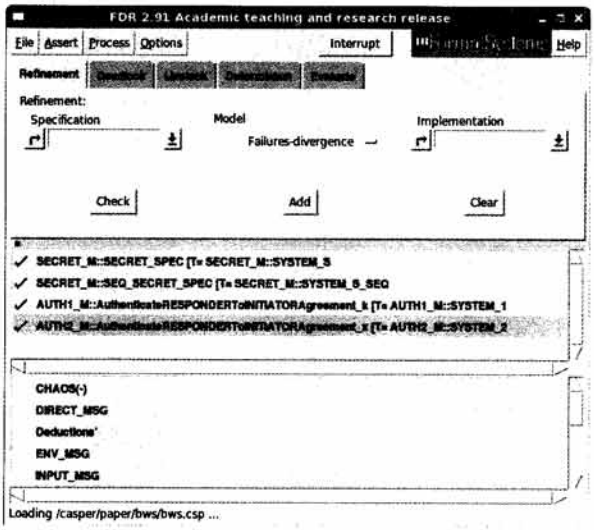
1) SECRET_M::SECRET_SPEC[T=SECRET_M::SYSTEM_S

프로토콜의 보안성 확보로 메시지 앞의 체크표시는 프로토콜이 공격자에게 노출되지 않았음을 표현한다. 검증한 One-time ID와 세션키의 보안성을 확인하였다.

2) SECRET_M::SEQ_SECRET_SPEC[T=SECRET_M::SYSTEM - S_SEQ

이 항목은 프로토콜이 시스템에서 정상적인 프로세스로 동작하는지를 확인한 결과이며 제안한 프로토콜은 안전한 프로세스로 동작함을 확인하였다.

3) AUTH1_M::AuthenticateRESPONDERToINITIATORA



(그림 9) 제안프로토콜의 보안 검증결과

reement_k[T=AUTH1_M::SYSTEM_1

4) UTH2_M::AuthenticateRESPONDERToINITIATORA greement_x[T=AUTH2_M::SYSTEM_2

3),4)는 k, x를 통해서 Responder와 Initiator가 서로 인증할 수 있는지 검증한다는 의미로 서로 안전하게 인증함이 확인되었다.

4.4 안전성 비교

제안한 프로토콜과 해시연산 기반의 해시락 기법, Kenji *et al.* 기법의 안전성을 비교분석하였다. <표 3>에서 보여주는 것과 같이 해시락 기법은 초창기 제안된 프로토콜로 많은 취약점을 나타낸다. Kenji *et al.* 기법은 초기에 암호화하지 않은 형태의 데이터로 위조가 가능하여 재전송 공격, 도청 공격 등에 취약하다. 제안 프로토콜의 경우 Kenji *et al.* 기법의 취약성을 보강하였으며 계산량을 줄이고 안전성은 충족한 시스템으로 FDR 검증결과 안전함이 입증되었다.

〈표 3〉 프로토콜의 안전성비교

	해시락 기법	Kenji et al.기법	제안 프로토콜
스푸핑 공격	취약	중간	안전
재전송 공격	취약	취약	안전
도청공격	취약	취약	안전
상호인증	미제공	제공	제공
트래픽분석 공격	취약	안전	안전
서비스거부 공격	안전	중간	안전

5. 결 론

RFID 시스템의 편리성과 효율성으로 각 산업 분야에 적용이 활발히 진행되고 있다. 그러나 보안성능에 대한 취약성 때문에 도입이 지연되거나 포기하는 분야가 생기고 있다. 이러한 보안문제는 향후 RFID 시스템의 활성화를 지연시키는 중요 요인이 될 수 있다. 시스템에 물리적인 기술로 보완하는 것은 비용 증가 등 한계가 발생하기 때문에 암호 프로토콜 등으로 보안안전성을 확보하는 것이 효과적이다. 본 논문에서 제안한 프로토콜은 기존에 제안된 Kenji *et al.* 프로토콜[12]의 취약성으로 발견된 One-time ID(OID)를 암호화하지 않은 형태로 전송하여 공격자가 여러 가지 공격을 할 수 있는 부분을 보강하였고 제안프로토콜이 FDR 도구의 보안속성을 만족하는지 검증을 실시하였다. 정형기법 중 모델 체킹을 하였으며 Casper, FDR 프로그램을 사용하여 검증결과 모든 보안적인 측면에서 만족함을 보였다. 본 실험의 결과로 향후 다음과 같은 기대효과가 있을 것으로 판단된다. 첫째, 정리증명에서 벗어나 정형검증을 실시함으로 실제시스템의 적용에 필요한 보안성이 확인되었다. 둘째, 검증결과 보안성이 충족되었기 때문에 RFID 시스템에 적용할시 개발자들의 복잡한 시험 및 코딩작업을 줄이고 곧바로 프로그래밍을 할 수 있다. 셋째, RFID 의 취약한 부분을 보강하여 신속한 시스템의 개발에 도움이 될 것이라 확신한다. 앞으로 강력한 해시함수를 이용하여 RFID에 적용할 수 있는 프로토콜을 연구 및 검증함으로 국방용 등에 사용할 프로토콜의 설계가 필요할 것이다.

참 고 문 헌

[1] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. w. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems," Security in Pervasive Computing 2003, LNCS 2802, pp.201-202, Springer-Verlag Heidelderg, 2004.

[2] S. A. Weis, "Security and Privacy in Radio-Frequency Identification Devices" MS Thesis, MIT.May, 2003.

[3] S. E. Sarma, S. A. Weis, D. W. Engels. "RFID systems, Security & Privacy Implications", White Paper MIT-AUTOID-WH-014, MIT AUTO-ID CENTER, 2002.

[4] Yu Tian-tian, Feng Quan-yuan, "A Security RFID Authentication Protocol Based on Hash Function," ieec, pp.804-807, 2009 International Symposium on Information Engineering and Electronic Commerce, 2009.

[5] He Lei, Gan Yong, Sun Tong, Wang Peng-yuan, "A Revised Efficient Authentication Protocol for Low-Cost RFID System," iitaw, pp.116-118, 2009 Third International Symposium on Intelligent Information Technology Application Workshops, 2009.

[6] Hung-Yu Chien and Chen-Wei Huang. "A Lightweight Authentication Protocol for Low-Cost RFID", Springer Science + Business Media, LLC. Manufactured in The United States, 2008.

[7] Shijie Zhou, Zhen Zhang, Zongwei Luo and Edward C. Wong. "A lightweight anti-desynchronization RFID authentication protocol", Springer Science + Business Media, LLC 2009.

[8] 이근우, 오동규, 곽진, 오수현, 김승주, 원동호 "분산 데이터베이스 환경이 적합한 Challenge-response 기반의 안전한 RFID 인증 프로토콜", 한국정보처리학회논문지C, 제12-C권, 제3호, pp.309-316, 2005.

[9] 안해순, 박기동, 윤은준, 남인길 "강력한 보안성을 제공하는 RFID 상호 인증 프로토콜" 정보처리학회논문지C, 제16-C권, 제3호, pp.325-334, 2009.

[10] 원태현, 유영준, 천지영, 변진욱, 이동훈 "온라인 백-엔드 데이터베이스가 없는 안전한 RFID 상호 인증 프로토콜", 정보보호학회논문지, 제20권, 제1호, pp.63-72, 2010.

[11] 박용수, 신주석, 최명실, 정경호, 안광선 "해시된 태그ID와 대칭키 기반의 RFID 인증프로토콜"정보처리학회논문지C, 제16-C권, 제6호., pp.669-680, 2009.

[12] K. Imamoto and K. Sakurai, "Design and Analysis of Diffie-Hellman Based Key Exchange Using ID by SVO Logic",Proc. Electronic Notes in Theoretical Computer Science 2005.

[13] G. Lowe. "Casper: A compiler for the analysis of security protocols." User Manual and Tutorial. Version 1.12 2009.

[14] 김일곤, 김철욱, 김현석, 최진영, 강인혜. "보안 프로토콜의 안전성 분석을 위한 정형적방법론" 정보보호학회논문지, 15(1), pp.17-27, 2005.

[15] C. Kraetzer, "Modelling Watermark Communication Protocols using the CASPER Modelling Language" Proceedings of the 12th ACM workshop on Multimedia and security. pp.107-116, 2010.

[16] Oxford University Computing Laboratory. FDR2 User Manual, 19th October, 2010.

[17] Mihai-Lica Pura, Victor Valeriu Patriciu, Ion Bica, "Formal Verification of G-PAKE Using Casper/FDR2-Securing a Group PAKE Protocol Using Casper/FDR2". SECRYPT 2010: 299-303.

[18] C.A.R Hoare. Communicating Sequential Processes. Prentice-Hall. 1985.

[19] 김주배, 김현석, 최진영. "스마트카드 기반 키분배 인증프로토콜의 정형 검증" 정보과학회 2008 추계학술발표회 35권 2(D)호, pp.45-49, 2008.

<부 록>

```
-- ***      Messages      ***
-- Message labels
datatype Labels =
  Msg1 | Msg2 | Msg3 | Msg4 | Msg5 | Env0
MSG_BODY = {ALGEBRA_M::applyRenaming(m_) | (_,m_,_,_) <-
SYSTEM_M::INT_MSG_INFO}
-- Type of principals
ALL_PRINCIPALS = Union({Agent, Server})
INTRUDER = Mallory
HONEST = diff(ALL_PRINCIPALS, {INTRUDER})
-- Channel declarations
INPUT_MSG = SYSTEM_M::INPUT_MSG
OUTPUT_MSG = SYSTEM_M::OUTPUT_MSG
DIRECT_MSG = SYSTEM_M::DIRECT_MSG
ENV_MSG = SYSTEM_M::ENV_MSG
channel receive: ALL_PRINCIPALS.ALL_PRINCIPALS.INPUT_MSG
channel send: ALL_PRINCIPALS.ALL_PRINCIPALS.OUTPUT_MSG
channel env : ALL_PRINCIPALS.ENV_MSG
channel error
channel start, close : HONEST.HONEST_ROLE
channel leak : addGarbage_(ALL_SECRETS)
-- Roles of agents
datatype ROLE = INITIATOR_role | RESPONDER_role | SERVER_role
HONEST_ROLE = ROLE
-- Secrets in the protocol
ALL_SECRETS_0 = Nonce
ALL_SECRETS =
addGarbage_(ALGEBRA_M::applyRenamingToSet(ALL_SECRETS_0))
-- Define type of signals, and declare signal channel
datatype Signal =
Claim_Secret.ALL_PRINCIPALS.ALL_SECRETS.Set(ALL_PRINCIPALS) |
Running1.HONEST_ROLE.ALL_PRINCIPALS.ALL_PRINCIPALS.Nonce |
Commit1.HONEST_ROLE.ALL_PRINCIPALS.ALL_PRINCIPALS.Nonce |
  RunCom1.ALL_PRINCIPALS.ALL_PRINCIPALS.Nonce.Nonce |
Running2.HONEST_ROLE.ALL_PRINCIPALS.ALL_PRINCIPALS.Nonce |
Commit2.HONEST_ROLE.ALL_PRINCIPALS.ALL_PRINCIPALS.Nonce |
  RunCom2.ALL_PRINCIPALS.ALL_PRINCIPALS.Nonce.Nonce
channel signal : Signal
Fact_1 =
  Union({
    {Garbage},
    Agent,
    Server,
    Nonce,
    SessionKey,
    {Encrypt.(a1, <a1, x>) |
      a1 <- SessionKey, x <- Nonce},
    {Encrypt.(a2, <T, x, k2>) |
      T <- Agent, a2 <- SessionKey, x <- Nonce,
      k2 <- addGarbage_({Encrypt.(a1, <k>) | a1 <- SessionKey, k
<- Nonce})},
    {Encrypt.(a1, <k>) |
      a1 <- SessionKey, k <- Nonce},
    {Encrypt.(a2, <T, x, Encrypt.(a1, <k>)>) |
      T <- Agent, a1 <- SessionKey, a2 <- SessionKey, k <- Nonce,
      x <- Nonce},    {Encrypt.(k, <Encrypt.(x, <x, k>)>) |
      k <- Nonce, x <- Nonce},
    {Encrypt.(x, <x, k>) |
      k <- Nonce, x <- Nonce},
    {Encrypt.(a2, <k1, a2, k>) |
      a2 <- SessionKey, k <- Nonce,
      k1 <- addGarbage_({Encrypt.(a1, <a1, x>) | a1 <- SessionKey,
x <- Nonce})},
    {Encrypt.(a2, <Encrypt.(a1, <a1, x>), a2, k>) |
      a1 <- SessionKey, a2 <- SessionKey, k <- Nonce, x <-
Nonce},
    {Encrypt.(x, <Encrypt.(k, <k, k>)>) |
      k <- Nonce, x <- Nonce},
    {Encrypt.(k, <k, k>) |
      k <- Nonce}
  })
external relational_inverse_image
external relational_image
transparent chase
```

배 우 식

e-mail : bws@motor.ac.kr

1997년~현 재 아주자동차대학 전산소

2006년 백석대학교 정보기술대학원 (공학석사)

2009년 2월 충북대학교 컴퓨터교육과 박사수료

관심분야 : RFID 보안, 컴퓨터 네트워크, 암호 프로토콜/알고리즘, 정보시스템 등

이 종 연

e-mail : jongyun@chungbuk.ac.kr

1987년 충북대학교 전자계산기공학과 (공학석사)

1999년 충북대학교 전자계산학과 (이학박사)

1990년~1996년 현대전자산업(주) 소프트웨어연구소와 현대정보기술(주) CIM사업부 책임연구원

1999년~2003년 강원대학교 삼척캠퍼스 정보통신공학과 조교수

2003년~현재 충북대학교 컴퓨터교육과 교수

2001년~2009년 IEEE member

2003년~2004년 한국정보처리학회 논문지편집위원 데이터베이스분과, 이사 역임

2007년~2010 한국산학기술학회 이사 역임

현 재 한국정보처리학회, 한국정보과학회, 한국컴퓨터교육학회 종신회원

2010년~현 재 한국컴퓨터교육학회 이사(현)

2010년~현 재 한국융합학회장(현)

관심분야 : 질의처리 및 최적화, 근사질의응(AQA), 시공간 데이터베이스, 데이터 마이닝, 유통물류, GIS, u-Learning과 평가방법