ASK 모바일 프로토콜 취약점 분석 및 수정
(Analyzing and Fixing the Vulnerabilities of ASK Protocol)

류 갑 상 † 김 일 곤 †† 김 현 석 ††
(Gab Sang Ryu) (Il Gon Kim) (Hyun Seok Kim)
이 지 현 †† 강 인 혜 ††† 최 진 영 ††††
(Ji Yeon Lee) (In Hye Kang) (Jin Young Choi)

요 약 지금까지의 많은 보안프로토콜들은 비정형화된 설계 및 검증 방법을 통해 개발되었다. 그 결과 유.무선 네트워크 분야에서 보안상 안전하다고 여겨왔던 많은 보안 프로토콜들의 보안 취약점들이 하나둘씩 발견되어오고 있다. 이에 따라, 통신 프로토콜을 개발하기 이전에 설계단계에서부터, 수학적인 기호 및 의미에 바탕을 둔 설계 언어로, 프로토콜의 안전성을 분석하기 위한 정형적 설계 및 검증 방법의 중요성이 점차 증대되고 있다. 현재, 모바일 통신 네트워크의 확산과 더불어 다양한 모바일 프로토콜들이 제안되고 있다. 본 논문에서는 정형적 검증 방법을 이용해서 ASK 모바일 프로토콜의 보안 취약점을 지적한다. 또한, 보안 취약점을 개선한 새로운 ASK 모바일 프로토콜을 제안하고 검증하였다.

키워드 : ASK 프로토콜, 정형검증, 모바일 프로토콜

Abstract Security protocols have usually been developed using informal design and verification techniques. However, many security protocols thought to be secure was found to be vulnerable later. Thus, the importance of formal specification and verification for analyzing the safety of protocols is increasing. With the rise of mobile communication networks, various mobile security protocols have been proposed. In this paper, we identify the security weakness of the ASK mobile protocol using formal verification technique. In addition, we propose a new ASK protocol modifying its vulnerability and verify its robustness.

Key words : ASK protocol, formal verification, mobile protocol

1. 서 론

지금까지의 보안프로토콜의 대부분은 비정형화된 설계 및 검증 방법을 통해 개발되었다. 그 결과 유.무선 네트워크 분야에서 보안상 안전하다고 여겨왔던 많은 보안 프로토콜들의 보안 취약점들이 하나둘씩 발견되어 오고 있다[1]. 이에 따라, 설계단계에서 보안프로토콜의 보안성, 인증 및 무결성과 같은 보안 속성들을 검증하기 위해 정형기법이 활용되어 왔다. 특히, Casper 및 FDR 도구를 이용한 정형적 설계 및 검증 방법은 많은 보안 프로토콜의 취약점을 밝혀낼 수 있었다[1,2].

기존의 정형적 설계 및 분석 방법론은 대부분 유선 네트워크 상에서 동작하는 보안프로토콜의 보안성을 분석하는데 중점을 두었다. 현재 무선 네트워크의 활성화와 더불어 다양한 모바일 프로토콜들이 등장하고 있다. 그리고 모바일 디바이스를 이용한 소프트웨어 다운로드 및 전자상거래와 같은 유료화 서비스가 각광을 받고 있다. 이에 따라, 모바일 사용자 및 서비스 제공자간의 안전한 통신 및 서비스 보장을 위해 모바일 프로토콜이 제안되고 있으며, 보안프로토콜의 안전성 보장은 중요한 연구과제로 부각되고 있다.

무선 네트워크의 낮은 대역폭, 모바일 디바이스의 낮은 연산속도 및 배터리 소모량등과 같은 모바일 통신 환경을 고려해야 하기 때문에, 모바일 보안프로토콜을 설계하는 것은 매우 어렵다. 게다가, 무선 네트워크의

† 종신회원 : 동신대학교 멀티미디어학과 교수
gsryu@dsu.ac.kr
†† 학생회원 : 고려대학교 컴퓨터학과
igkim@formal.korea.ac.kr
jylee@formal.korea.ac.kr
hskim@formal.korea.ac.kr
††† 종신회원 : 서울시립대학교 컴퓨터학과 교수
inhye@uos.ac.kr
†††† 종신회원 : 고려대학교 컴퓨터학과 교수
choi@formal.korea.ac.kr
논문접수 : 2005년 6월 7일
심사완료 : 2005년 10월 1일

특성상 유선 네트워크에 비해 보안성이 떨어지기 때문에 안전한 키 교환 및 사용자 인증을 보장하는 모바일 보안프로토콜을 설계하는 것은 더욱 어려운 과정이다. 따라서 정형화된 설계 및 검증 방법을 보안프로토콜 개발이 요구되고 있다.

본 논문에서는 Casper/FDR 도구를 이용한 정형적 설계 및 검증방법을 사용하여, ASK 모바일 보안 프로토콜의 보안 취약점을 지적한다. 그리고 안전한 상호 키 교환 및 사용자 인증을 제공하는 새로운 ASK 모바일 프로토콜을 검증하고 제안하고자 한다.

본 논문의 구성은 다음과 같다. 제2장에서는 정형기법을 이용하여 보안프로토콜의 취약점을 분석하는 관련연구를 소개하고, 제3장에서는 Casper 및 FDR 도구를 이용한 보안프로토콜 정형적 설계 및 검증 방법론에 대해 간략히 설명한다. 제4장에서는 모바일 ASK 프로토콜의 메시지 교환절차를 소개한다. 제5장에서는 ASK 프로토콜에서 검증한 보안속성을 정의하고 검증결과를 보여준다. 제6장에서는 새로운 ASK 모바일 프로토콜을 소개하고 검증결과를 설명한다. 마지막으로 제7장에서 결론을 맺고자 한다.

2. 관련연구

보안 프로토콜의 안전성을 검증하기 위한 방법은 크게 모델체킹과 정리증명 방법으로 나누어진다. 모델체킹의 장점은 자동화 검증도구가 지원된다는 사실이다. 즉, 사용자가 시스템의 모델을 입력하고 요구 사항 명세를 나타내는 속성들을 입력하면 도구는 자동적으로 모델의 상태를 검사하여, 속성을 만족하지 못하는 경우, 반례를 보여주어 모델의 어느 부분이 잘못되었는지를 쉽게 알 수 있게 해 준다는 것이다. 반면에 정리증명 방식은 증명과정에 사람의 개입이 필요하기 때문에, 보안 프로토콜을 논리적으로 증명하기에 앞서 가정을 세우고, 논리추론 규칙에 따라 보안 취약점을 추론해 내기가 쉽지 않다.

Roscoe와 Goldsmith는 CSP 언어를 이용하고 FDR 모델체킹 도구를 이용하여 보안 프로토콜의 안전성을 검증하는 연구의 기반을 마련하였다[3]. 대부분의 보안 프로토콜 검증 연구는 유선 네트워크상에서 사용되는 프로토콜의 취약점을 검증하는데 중점을 두어 왔으며, 상대적으로 모바일 프로토콜의 보안 취약점을 검증한 논문은 그다지 많지 않다.

Ghezzi와 Kemmerer은 ASTRAL 모델체킹 도구를 이용하여 TMN 모바일 프로토콜의 취약점을 발견하였다[4]. Coffey와 Dojen은 정리증명 방법에 기반을 둔 GNY 로직을 이용하여, BCY 모바일 프로토콜의 보안 취약점을 지적하고 새로운 CDF-BCY 프로토콜을 제안하였다[5]. 하지만, GNY 로직을 이용한 정리증명 방법의 경우 수학적 논리 전문가가 아니면, 보안 취약점을 분석해 내기가 어렵다는 단점을 보여주고 있다.

본 논문에서는 모델체킹 방법을 이용하여 ASK 모바일 프로토콜의 취약점을 분석하고, 보안성이 강화된 새로운 프로토콜을 제안한다.

3. Casper와 FDR 도구

3.1 Casper(A Compiler for the Analysis of Security Protocols)

CSP[6] 언어를 이용하여 보안프로토콜 행위를 명세하고 FDR 정형검증 도구를 이용하여 보안속성을 검증하는 연구가 진행되었다[7]. 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[8]. 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. 다음은 Casper 명세에서 사용되는 기본적인 7개의 섹션헤더와 의미를 간략히 보여주고 있다.

- #Free variables : 변수의 타입 및 함수 선언
- #Process : 통신 에이전트의 초기 상태 표현
- #Protocol description : 통신 에이전트간의 메시지 교환 표현
- #Specification : 검증하고자 하는 보안속성 선언
- #Actual variable : 통신 에이전트가 사용하는 실제 데이타 타입 및 이름 선언
- #Function : 프로토콜에서 사용하는 함수선언
- #System : 통신 에이전트의 초기 상태정보 표현
- #Intruder information : 공격자의 초기 상태정보 표현

3.2 FDR(Failure Divergence Refinement)

FDR 도구는 CSP 명세언어를 입력으로 받아들이는 모델체킹 도구로서 옥스포드 대학에서 개발되었다[9]. 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 3가지의 검증방법을 지원하고 있다.

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

그림 1은 Casper 및 FDR 도구를 이용하여, 보안프로

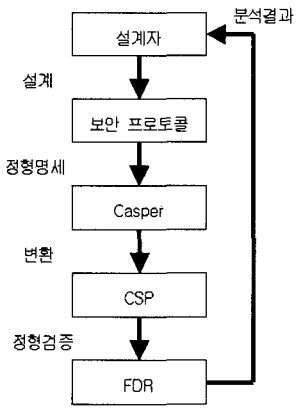

그림 1 Casper 및 FDR 도구를 이용한 보안프로토콜 설계 및 검증

토콜을 정형적으로 설계하고 검증하는 과정을 보여주고 있다.

첫째, 검증하고자 하는 보안프로토콜을 Casper 도구로 명세한다.

둘째, Casper의 자동변환기능을 이용해 CSP 명세코드를 생성한다.

셋째, FDR 도구에 CSP 명세코드를 입력한다.

넷째, FDR 도구의 검증결과를 분석한다.

마지막으로 보안프로토콜의 취약점이 밝혀지면, 문제점을 수정하여 동일한 설계 및 검증절차를 반복 수행한다.

4. ASK 프로토콜

ASK(Aydos, Sunar, Koc) 모바일 프로토콜은 Aydos에 의해 처음으로 제안되었다[10]. ASK 프로토콜은 모바일 폰과 같은 소규모 배터리를 사용하는 통신 단말기와 서비스 제공자 사이의 인증 및 안전한 상호 키 교환을 보장하기 위해 개발되었다. 또한 ASK 프로토콜은 BCY[11] 및 Aziz-Diffie[12] 프로토콜 보다 속도 및 안전성 측면에서 더 우수한 것으로 알려져 있다[10]. 하지만, 아직까지 ASK 프로토콜의 보안성을 모델체킹 방법을 이용하여 정형적으로 검증한 연구는 진행되지 않았다.

그림 2는 ASK 모바일 프로토콜의 메시지 교환 순서를 보여주고 있으며, 표 1은 ASK 프로토콜에서 사용되는 기호의 의미를 나타내고 있다.

그림 2에서, S는 제3의 키 분배 서버이고, V는 사용자 U에게 컨텐츠 데이타를 제공하는 서비스 제공자를 의미하고 있다. 예를 들어, PDA와 같은 모바일 통신 디바이스를 소지하고 있는 사용자가 고스톱 게임과 같은 유료 컨텐츠를 다운로드 받고자 할 때, ASK와 같은 프로토콜을 사용하여 인증을 받을 수 있다. 이런 경우, ASK 프로토콜의 메시지 교환 절차는 다음과 같이 요약된다.

```
Msg 1. S -> V : CertV
Msg 2. S -> U : CertU
Msg 3. V -> U : Kv+ [U computes : K1 = {Kv+}Ku-
Msg 4. U -> V : Ku+ [U computes : K2 = {Ku+}Kv-
Msg 5. V -> U : {CertV, Rv}K [U and V compute : SK = {K}Rv]
Msg 6. U -> V : {CertU, Rv}K
```

그림 2 ASK 프로토콜

Msg 1 : S는 V에게 공개키 Kv+를 전달하기 위해, V와 Kv+ 정보를 포함하는 인증서 CertV를 V에게 전송한다.

Msg 2 : S는 U에게 공개키 Ku+를 전달하기 위해, U와 Ku+ 정보를 포함하는 인증서 CertU를 U에게 전송한다.

Msg 3 : V는 S로부터 수신한 자신의 공개키 Kv+를 U에게 전달한다. 사용자 U는 Diffie-Hellman 키로 사용할 K1 키(K1 = {Kv+}Ku-)를 생성한다.

Msg 4 : U는 S로부터 수신한 자신의 공개키 Ku+를 V에게 전달한다. 서비스 제공자 V는 Diffie-Hellman 키로 사용할 K2 키(K2 = {Ku+}Kv-)를 생성한다. Msg3과 Msg4 과정을 통해, 결국 U 와 V는 동일한 공유키 K(K1 = K2)을 생성하게 된다.

Msg 5 : V는 Diffie-Hellman 키로 암호화한 인증서 CertV와 난수 Rv를 U에게 전송한다. 그런 다음, V와 U는 Rv를 이용하여 세션키 SK를 각각 생성하게 된다.

Msg 6 : 마지막으로 U는 K키로 암호화한 인증서 CertU와 Rv를 V에게 전송한다. 사용자에 대한 정상적인 인증이 이루어지고 난 후, 사용자는 유료 컨텐츠를 다운로드 받게 된다.

표 1 ASK 프로토콜 기호 및 의미

기호	의미
U	모바일 사용자의 식별자
V	서비스 제공자의 식별자
S	인증기관의 식별자
Rx	X 호스트에서 생성한 임의 난수
Kx+	X 호스트의 공개키
Kx-	X 호스트의 개인키
K	Diffie-Hellman 키 (K1 = K2)
SK	세션키(SK1 = SK2)
CertV	S에서 발행한 V의 인증서
CertU	S에서 발행한 U의 인증서

5. ASK 프로토콜 검증

4장에서 앞서 언급한 ASK 프로토콜에서 사용하는 기호 및 메시지 교환 절차를 바탕으로 Casper로 명세하였다. 본 논문에서는 Casper 명세 중 보안프로토콜의 행위와 검증속성과 같은 일부 섹션헤더 부분만보여주고 있다; #Free variables, #Protocol Description, #Specification, #Intruder information와 #Equivalences. 다른 명세 부분은 매우 간단하고 명확해서 자세한 설명은 생략하고자 한다.

```
#Free variables
v, u : Agent
s : Server
pkv, pku : PublicKey
skv, sku : SecretKey
SPK : Server > ServerPublicKey
SSK : Server > ServerSecretKey
rv : Nonce
InverseKeys = (pkv, skv), (pku, sku), (SPK, SSK)
```

#Free variable 섹션 헤더에서는 #Protocol description에서 사용되는 자유 변수의 타입 및 함수를 정의하고 있다. v와 u는 각각 V와 U 호스트의 식별자를 의미하고 s는 S 호스트 서버의 식별자를 나타낸다. pkv와 pku는 각각 V와 U의 공개키 Kv+와 Ku+를 가리킨다.

skv와 sku는 각각 V와 U의 개인키 Kv-, Ku-를 의미한다. 그리고 SPK와 SSK는 서버 S의 공개키 및 개인키를 반환하는 함수를 표현하고 있다. rv는 v의 임의 난수이며, InverseKeys는 각 호스트의 암호 및 복호화 키 쌍을 가리키고 있다.

```
#Protocol description
0. > v : u
1a. s > v : {v, pkv}{SSK(s)} % digV
1b. s > u : {u, pku}{SSK(s)} % digU
2. v > u : pkv
3. u > v : pku
4. v > u : {{digV % {v, pkv}{SSK(s)}, rv}{pku}}{skv}
5. u > v : {{digU % {u, pku}{SSK(s)}, rv}{pkv}}{sku}
```

#Protocol description 섹션 헤더는 프로토콜의 메시지 교환 동작을 나타내기 위해 사용된다. 표현식 {m}{k}은 키 k로 암호화한 메시지 m을 나타낸다. 그리고 표현식 m%d에서, m은 전달하고자 하는 메시지를 의미하고, d는 메시지를 저장하기 위한 변수로 사용된다. 이 기호는 메시지를 수신한 호스트가 메시지 m을 복호화 할 수 없고, 단지 다른 호스트에 전달만 하는 기능을 표현하기 위해 사용된다. 이와 유사한 표현식 d%m은 d 변수에 저장되어 있는 메시지 m을 의미한다. 예를 들어, 메시지 1a와 4에서 제3의 인증기관 s는 인증서 {V, Kv+}Ks-를 v에게 전송하고, 다시 v는 동일한 메시지를 u에게 전달하기 위한 과정을 표현하고 있다.

```
#Specification
Secret(v, rv, [u])
Secret(u, rv, [v])
Agreement(v, u, [rv, pku, skv])
Agreement(u, v, [rv, pkv, sku])
```

#Specification 섹션 헤더는 검증하고자 하는 보안 속성을 정의한다. 본 논문에서는 비밀성(confidentiality)과 인증(authentication), 두 가지의 보안 속성을 검증하였다. Casper에서 'Secret' 표현식은 비밀성 속성을 정의하기 위해 사용된다. 비밀성은 악의적인 공격자가 어떤 중요한 비밀 정보를 가로채지 못한다는 속성을 나타낸다. 예를 들어, 앞의 #Specification 섹션 헤더에 쓰여진 'Secret(v, rv, [u])' 표현식은 다음과 같이 해석된다."서비스 제공자 v는 중요한 비밀 정보 rv를 v와 u 사이에만 공유하고 있다고 믿는다." 'Agreement' 표현식은 인증 속성을 정의하기 위해 사용된다. 예를 들어, 'Agreement(v, u, [rv, pku, skv])' 표현식은 다음과 같이 해석된다. "서비스 제공자 v는 rv, pku 및 skv 정보를 이용하여 사용자 u에게 인증을 받는다."

인증 속성은 단방향 인증 및 양방향 인증 측면으로 나누어 검증할 수 있다. 본 논문에서는 사용자 U와 서비스 제공자 V간의 상호 인증관계를 검증하기 위해 'Agreement(v, u, [rv, pku, skv])'와 'Agreement(u, v, [rv, pkv, sku])' 표현식을 사용하였다.

```
#Intruder information
Intruder = Mallory
IntruderKnowledge = {Vendor, User, Sam, Mallory,
Rm, PKv, PKu, PKm, SKm}
```

#Intruder information 섹션 헤더에서는 통신 프로토콜을 공격하기 위한 공격자의 사전 정보를 표현한다. 예를 들어, 위의 명세 코드를 보면 공격자 호스트의 이름은 Mallory라고 설정하였으며, Vendor, User 그리고 Sam은 각각 V, U 그리고 S 호스트에 대한 도메인 이름을 나타내고 있다. 본 논문에서는 공격자는 모든 호스트의 식별자 및 공개키 정보를 알고 있고, 자신의 임의

난수 Rm과 자신의 개인키 키 SKm을 알고 있다고 가정한다.

```
#Equivalences
forall rv, pkv, pku, skv, sku . {{rv}{pku}}{skv} =
{{rv}{pkv}}{sku}
```

#Equivalences 섹션 헤더는 메시지 상호간의 수학적 동치관계를 나타내는데 사용된다. 위의 표현식은 Diffie-Hellman 키 교환 알고리즘의 키 동치성을 나타내고 있으며, 다음과 같은 수학적 기호로 표현할 수 있다.

$$\forall rv, pkv, pku \{\{rv\}pku\}skv\} = \{\{rv\}pkv\}sku\}$$

V와 U 호스트 사이의 상호키 교환을 위해 사용되는 Diffie-Hellman 키 교환 알고리즘의 동치성은 다음과 같이 정의할 수 있다. Diffie-Hellman 알고리즘에 대한 보다 상세한 내용은 [13]의 70페이지 내용을 참조하기 바란다.

정의 1. Diffie-Hellman 키 분배 알고리즘

$$\begin{aligned} K &= (Y_V)^{X_U} \bmod q \\ &= (a^{X_V} \bmod q)^{X_U} \bmod q \\ &= (a^{X_V})^{X_U} \bmod q \\ &= a^{X_V X_U} \bmod q \\ &= (a^{X_U})^{X_V} \bmod q \\ &= (a^{X_U} \bmod q)^{X_V} \bmod q \\ &= (Y_U)^{X_V} \bmod q \end{aligned}$$

위의 식에서, U의 공개키는 $Y_U = a^{X_U} \bmod q$이고, 개인키는 X^U를 나타내고 있다. 이와 마찬가지로 V의 공개키는 $Y_V = a^{X_V} \bmod q$이고, 개인키는 X^V를 나타내고 있다. 이에 따라, U와 V의 공유키 K는 $(Y_V)^{X_U} \bmod q = (Y_U)^{X_V} \bmod q$라는 식이 성립하게 되며, 이 식은 본 논문에서 명세한 ASK 프로토콜의 V와 U 각각의 키 K1과 K2의 동치성(K1 = K2)을 보여주고 있다(K1 = {Kv+}Ku-, K2={Ku+}Kv-).

궁극적으로 ASK 프로토콜에서 세션키 SK1 = {{Rv}{Kv+}}Ku-은 세션키 SK2 = {{Rv}{Ku+}}Kv-와 동일한 세션키임을 나타내기 위해 #Equivalences 섹션을 작성하였다.

앞에서 언급한 Casper 명세를 통해, CSP 코드를 생성한 후, FDR 검증 도구를 동작시킨 결과 ASK 프로토콜이 앞의 #Specification 섹션헤더에서 정의한 'Secret (v, rv, [u])', 'Secret(u, rv, [v])' 및 'Agreement(v, u, [rv, pku, skv])' 의 보안 요구사항을 만족시키지 못함을 확인하였고, FDR 도구에서 생성한 반례(counter-example) 이벤트 분석결과 다음과 같은 공격 시나리오를 찾아낼 수 있었다.

ASK 프로토콜을 이용하여, 사용자 U는 자신이 소지한 모바일 통신 디바이스를 이용하여, 고스톱 게임과 같은 유료 컨텐츠를 다운로드 받고자 하는 실제 시나리오를 가정해 볼 수 있다. 이때, S는 인증서를 발급하는 공인인증기관이고, V는 유료 컨텐츠를 제공하는 업체의 서버가 된다. 모바일 보안 프로토콜의 인증 취약점으로 인해, 만일 악의적인 목적을 가진 공격자가 정상적인 컨텐츠 제공 서버로 위장하여, 개인정보 유출 및 네트워크 장애를 초래하는 웜 바이러스를 모바일 통신 디바이스에 대량으로 유포하게 된다면, 개인의 중요 정보 유출, 통신 기기 고장, 네트워크 장애와 같은 문제점들이 발생할 수 있다.

그림 3에서 보는 바와 같이, 통신을 시작하게 되면 1번 메시지를 통해서, 인증기관 S는 V의 인증서({V, Kv+}Ks-)를 V에게 전송하는데, 이때 공격자 I는 이 인증서를 가로챌 수 있다. I(V) 기호는 V 호스트로 가장하여, 메시지를 가로채거나 정상적인 호스트로 위장하는 공격자 I를 나타낸다. 모바일 통신이 주로 이루어지는 무선네트워크 상태에서는 AirSnort 와 같은 패킷 캡처 도구를 이용하여 손쉽게 인증서를 도청할 수 있게 된다. 그런 다음 공격자는 인증기관인 것처럼 가장하여, 1번 메시지에서 가로챈 인증서를 V에게 재 전송하게 된다.

```
Msg 1. S → I(V) : {V, Kv+}Ks-
Msg 2. I(S) → V : {V, Kv+}Ks-
Msg 3. V → I(U) : Kv+
Msg 4. I(U) →V : Ku+
Msg 5. V → I(U) : {{V, Kv+}{Ks-}}, Rv}{Ku+}}Kv-
Msg 6. S → I(U) : {U, Ku+}Ks-
Msg 7. I(S) → U : {U, Ku+}Ks-
Msg 8. I(V) → U : Kv+
Msg 9. U → I(V) : Ku+
Msg 10. I(V) → U : {{{V, Kv+}{Ks-}, Rv}{Ku+}}Kv-
Msg 11. U → I(V) : {{{U, Ku+}{Ks-}, Rv}{Kv+}}Ku-
```

그림 3 ASK 프로토콜 공격 시나리오

3번 메시지에서 서비스 제공자 V는 사용자 U에게 자신의 공개키(Kv+)를 전송하게 되며, 공격자 I는 패킷 캡처도구를 이용하여 이 메시지를 가로채게 된다. 4번 메시지에서 공격자 I는 사용자 U로 위장하여 U의 공개키(Ku+)를 V에게 전달하게 된다. 1번과 2번 그리고 3번과 4번 메시지에서 보이는 것처럼 공격자가 정상적인 호스트 상호간의 통신 메시지를 가로채어, 재사용하는 공격을 man-in-the-middle 공격이라고 한다.

결국 ASK 프로토콜은 man-in-the-middle 공격에 취약하여, 10번째 메시지에서 공격자는 V로 가장하여 5번째 메시지에서 가로챈 정보 {{V, Kv+}{Ks-}, Rv}{Ku+}} Kv-를 사용자 U에게 전달하게 된다. 11번째 메시지에서 사용자의 모바일 통신 디바이스는 서비스 제공자 V가 정상적인 호스트로 인증하게 되며, Diffie-Hellman 키를 이용한 세션키 SK(그림 2 참조)를 생성하게 된다. 마지막 11번째 메시지에서 공격자 I는 V로 위장한 상태에서, 사용자가 전송한 정보를 이용하여 동일한 세션키 SK를 생성할 수 있게 된다.

결국, 서비스 제공자로 위장한 공격자는 악의적인 웜 바이러스 기능을 가진 유료 컨텐츠를 사용자에게 배포할 수 있게 된다. 이렇게 되면, 유선네트워크 상에서의 바이러스 유포에 의한 피해가 모바일 통신 환경에도 동일하게 발생할 수 있게 된다.

앞의 공격 시나리오를 통해, ASK 프로토콜에서 서비스 제공자 V와 사용자 U와의 상호 인증에 보안상 취약점이 존재함을 알 수 있다. FDR 모델체킹 도구를 이용한 검증 결과 ASK 모바일 프로토콜은 다음과 같은 보안 요구사항을 만족시키지 않음을 확인할 수 있었다.

- 그림 2의 4번째 메시지에서 사용자의 공개키(Ku+)가 평문으로 전달되기 때문에, 그림 3의 4번 메시지에서 공격자는 사용자의 공개키를 가로챌 수 있게 된다. 이는 결국 사용자 인증의 신뢰성을 보장하지 못하는 결과를 야기시킨다.
- 그림 2의 5번째 메시지({CertV, Rv}K)를 공격자가 가로채거나 재사용할 수 있기 때문에 서비스 제공자의 신분을 보장할 수 없게 된다. 이는 결국 서비스 제공자와 사용자 사이의 상호 인증을 보장해 주지 못하는 결과를 야기시킨다.
- 그림 3의 5번, 10번 및 11번 메시지에서 공격자는 사용자와 서비스 제공자가 사이의 세션키를 가로채어 정상적인 서비스 제공자처럼 위장할 수 있게 된다. 즉, 세션키의 신성성(freshness)을 보장할 수 없기 때문에 사용자와 서비스 제공자의 상호교환 키의 안전성을 신뢰할 수 없다.

6. 수정된 ASK 프로토콜

5장에서 언급한 ASK 프로토콜의 보안 취약점을 해결하기 위해서 다음과 같은 수정사항들을 반영하였다.

첫째, ASK 프로토콜의 메시지 3번과 4번은 메시지 5번 및 6번 항목과 중복되고 사용자 공개키의 비밀성이 공격자에 의해 침해될 수 있기 때문에 그림 2의 3번과 4번 메시지를 삭제하였다.

둘째, 사용자와 서비스 제공자의 인증서, CertV와 CertU의 재사용 공격을 방지하기 위해서 인증서의 만기 시간을 나타내는 변수 TSv와 TSu를 각각 첨가하였다. 만일 공격자가 사용자 및 서비스 제공자 사이의 메시지를 가로채어 재사용 하게 되면, 인증서에 표기된 타임스탬프를 통해 사용자와 서비스 제공자는 메시지의 재사용 여부를 확인할 수 있게 된다. 이에 따라, 서비스 제공자와 사용자는 자신들의 공개키가 공격자에 의해 재사용되지 않음을 보장할 수 있게 된다.

셋째, 세션키의 안전성을 강화하기 위해 사용자의 임의 난수 Ru를 첨가하였다. 수정된 ASK 프로토콜의 새로운 세션키는 h(ru, rv)가 된다. h는 MD4 또는 MD5와 같은 단방향 해쉬 함수를 나타낸다. 단방향 해쉬함수는 DES 및 RSA와 같은 복잡한 암호 알고리즘에 비해 연산 수행 속도가 더 빠르고 에너지를 덜 소모하기 때문에 이동통신용 소형단말기에 적합하다.

그림 4는 수정된 ASK 프로토콜의 메시지 교환 절차를 보여주고 있다

```
Msg 1. S → V : CertV
Msg 2. S → U : CertU
Msg 3. V → U : {CertV, Rv}Kv- [U computes : Ru]
Msg 4. U → V : {CertU, {Rv, Ru}Ku-}Kv+ [V
               computes : SK = h(Rv, Ru)]
Msg 5. V → U : {{Rv, Ru}Ku+}Kv- [V computes :
               SK = h(Rv, Ru)]
```

그림 4 수정된 ASK 프로토콜

수정된 ASK 프로토콜을 이용하여, 앞의 5장에서 언급한 바와 같이, 사용자가 모바일 통신 디바이스를 이용하여 유료 컨텐츠를 다운로드 하는 경우를 가정해 보자. 1번과 2번 메시지를 이용하여 인증기관은 서비스 제공자와 사용자의 모바일 통신 디바이스에 각각 인증서를 분배 하게 된다. 만일 공격자가 V와 U의 인증서를 가로챈다 하여도, 인증서에 포함된 타임 스탬프를 이용하여 일정한 시간이 지나면 인증서를 재사용할 수 없다. 3번 메시지에서 서비스 제공자는 사용자에게 자신의 인증서 CertV와 임의 난수 Rv 전송하게 된다. 만일 공격자가 3번 메시지를 가로 챈다고 해도, 서비스 제공자의 공개키 Kv+를 모르기 때문에, 세션키 SK를 생성하는데 필요한 Rv를 알 수 없다. 하지만, 공개키의 특성상 공격자가 만일 서비스 제공자의 공개키 Kv+를 알 수 있다고 가정하더라도, 4번 메시지에서 서비스 제공자의 개인키 Kv-를 알지 못하기 때문에, 세션키 SK를 생성할 수 없게 된다. 이렇게 되면, 사용자와의 정상적인 인증 절차가 이루어지지 않게 되어, 악의적인 바이러스 코드의 다운로드를 사전에 막을 수 있다. 서비스 제공자가 정상적인 컨텐츠 제공자라면, 수정된 ASK 프로토콜을 이용하여 기존의 프로토콜 보다 안전한 통신을 가능하

게 하며, 이와 더불어 간소화된 메시지 교환 절차 및 경량화된 암호 알고리즘의 사용을 통해 소형 모바일 통신 디바이스의 밧데리를 보다 장시간 사용하는데 기여하게 된다.

Casper와 FDR 도구를 이용하여, 수정된 ASK 프로토콜의 비밀성 및 인증 속성을 검증하였다. Casper에서 명시한 두 가지 보안 속성은 다음과 같다.

```
#Specification
Secret(v, ru, [u])
Secret(u, ru, [v])
Agreement(v, u, [rv, ru, pku, skv])
Agreement(u, v, [rv, ru, pkv, sku])
```

FDR 도구를 통해 검증해 본 결과, 수정된 ASK 프로토콜은 위에서 정의한 보안 속성들을 모두 만족하고 있음을 확인할 수 있었다. 공격자는 사용자의 임의 난수 Ru 정보를 가로챌 수 없기 때문에, 결국 사용자와 서비스 제공자는 세션키와 Diffie-Hellman 키를 이용해서 상호 호스트를 인증하게 된다.

7. 결 론

모바일 통신 네트워크의 활성화와 더불어 모바일 사용자와 서비스 제공자 상호간의 안전한 통신 서비스를 제공하기 위해 다양한 보안프로토콜들이 제안되고 있다. 하지만, 대부분의 경우 비정형화된 설계 및 검증방법을 통해 개발되기 때문에 나중에 보안상 취약점이 발견되고 있다. 본 논문에서는 모델체킹 기법을 이용하여 ASK 모바일 프로토콜의 보안 취약점을 지적하였다. 뿐만 아니라, 보안 취약점을 개선한 새로운 ASK 프로토콜을 제안하고 검증하였다.

새롭게 제안된 ASK 프로토콜은 기존의 프로토콜에 비해 적은 메시지 교환 횟수를 갖기 때문에 낮은 연산 속도와 적은 양의 배터리를 사용하는 모바일 통신 환경에 적합한 것으로 사료된다. 또한 보안상 기존의 ASK 프로토콜에 비해 더 안전하다는 것을 검증하였다.

향후 연구과제로는 본 논문에서 새롭게 제안하는 프로토콜과 기존의 ASK 프로토콜의 네트워크 속도에 따른 성능을 비교 및 분석 하고자 한다.

참 고 문 헌

[1] G. Lowe, "Breaking and Fixing the Needham-Schroeder Public-Key Protocol," TACAS 96, pp.147-166, 1996.

[2] I. G. Kim and J. Y. Choi, "Formal verification of PAP and EAP-MD5 Protocols in wireless networks : FDR Model Checking," 18th AINA, pp.264-269, 2004.

[3] A. Roscoe and M. Goldsmith, "The Perfect Spy for Model-Checking Cryptoprotocols," *Proceedings of the 1997 DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

[4] Z. Dang, "Using the ASTRAL Model Checker for Cryptographic Protocol Analysis," Proceedings of the 1997 DIMACS Workshop on Design and Formal Verification of Security Protocols, 1997.

[5] T. Coffey and R. Dojen, "Analysis of a mobile communication security protocol," Proceeding of the 1st international symposium on Information and communication technologies, pp. 322-328, 2003.

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

[7] P. Y. A. Ryan and S. A. Schneider, modelling and analysis of security protocols: the CSP Approach, Addison-Wesley, 2001.

[8] G. Lowe, "Casper: A Compiler for the Analysis of Security Protocols," 10th IEEE Computer Security Foundations Workshop, 1997.

[9] Formal Systems(Europe) Ltd. Failure Divergence Refinement-FDR2 User Manual, 1999.

[10] M. Aydos, B. Sunar, and C. K. Koc, "An elliptic curve cryptography based authentication and key agreement protocol for wireless communication," presented at the 2nd Int. Workshop Discrete Algorithms and Methods for Mobility, Dallas, TX, Oct. 1998.

[11] M. J. Beller, L. -F. Chang and Y. Yacobi, "Privacy and authentication on a portable communications system," Proceedings of the International Computer Symposium, Vol.1, pp.821-829, 1994.

[12] A. Aziz and W. Diffie, "and authentication for wireless local area networks," IEEE Personal Commun., First Quarter 25 31, 1994.

[13] W. Stallings, NETWORK SECURITY ESSENTIALS : Applications and Standards, Second Edition, page 70, Prentice-Hall, 1999.

류 갑 상

1983년 전남대학교 전산통계학과 학사
1985년 전남대학교 전산통계학과 석사
1998 고려대학교 컴퓨터학과 박사수료
1985년~1996년 한국기계연구원 선임연구원. 1996~현재 동신대학교 멀티미디어학과 교수. 관심분야는 공장자동화, 보안 프로토콜

김 일 곤

2000년 경기대학교 영어영문과 학사. 2002년 고려대학교 컴퓨터학과 석사. 2005년 고려대학교 컴퓨터학과 박사. 2005년~현재 고려대학교 컴퓨터학과 연구교수. 관심분야는 정형기법, 소프트웨어공학, 보안프로토콜

김 현 석

2000년 육군사관학교 경제경영학과 학사 2005년 고려대학교 컴퓨터학과 석사과정 관심분야는 정형기법, 무선네트워크보안

이 지 현

1999년 동덕여자대학교 컴퓨터학과 학사 2001년 고려대학교 컴퓨터학과 석사 2002년~현재 동남보건 대학교 인터넷 경영정보통신 학과 조교수. 관심분야는 소프트웨어공학, 컴퓨터 보안, 정형기법

강 인 혜

1987년 서울대학교 컴퓨터공학과 학사 1989년 서울대학교 컴퓨터공학과 석사 1997년 University of Pennsilvania 박사. 현재 서울시립대학교 기계정보공학과 조교수 관심분야는 소프트웨어공학, 컴퓨터 보안, 정형기법

최 진 영

1982년 서울대학교 컴퓨터공학과 학사 1986년 Drexel Univ. 전산학 석사. 1993년 Univ. of Pennsylvania 전산학 박사 현재 고려대학교 컴퓨터학과 교수. 관심분야는 컴퓨터이론, 정형기법. 실시간시스템