# WAP에서 Java 기반 종단간 보안 구현

김원진[0] 강태호 방 훈 원대희 이재영
한림대학교 컴퓨터 공학과
{wjkim[0], Lamius, hooni, dhwon, jylee}@isul.ce.hallym.ac.kr

# Implementation of End-to-End Security on Based Java in the WAP

Won-Jin Kim[0] Tae-Ho Kang Hoon Bang Dae-Hee Won Jae-Young Lee
Dept. of Computer Engineering, Hallym Univ.

요 약

무선 인터넷의 산업 표준으로 제시된 WAP은 많은 부분에 있어 연구되어 구현되어지고 있으나, 기존의 WAP의 문제점으로 전송계층에서의 종단간 보안 문제는 서비스 제공자에게 부담을 가중시키는 또 다른 문제점이 제기되었다. 본 논문에서는 구조적인 전송계층의 문제를 상위 계층인 응용 계층에서 J2ME를 이용하여 전송되는 데이터를 암호화시켜 종단간 보안 문제를 해결하여 기존의 구조적인 WAP형태를 변경하지 않도록 구현하였다.

## 1. 서 론

인터넷의 발전으로 인하여 많은 정보들이 인터넷을 통하여 공유되고, 전자상거래와 같은 많은 작업들이 수행되어져 왔다. 인터넷이 무선인터넷으로 발전함에 따라 이에 대한 연구가 전세계적으로 진행되고 있고, 많은 회사들 또한 이 연구에 집중하고 있다.

현재 무선 인터넷의 산업 표준으로 자리잡고 있는 WAP은 무선 통신 환경과 기존의 유선 통신 환경과의 호환성을 위해 WAP 프로토콜과 HTTP 프로토콜을 이용한다. 유,무선 간의 네트워크를 연결하는 WAP 게이트웨이를 사용하고, 게이트웨이는 사용자와 서비스 제공자 사이에서 연결 서비스를 제공한다. 이 게이트웨이가 종단간 보안에 있어서 문제점을 발생하며, 이 문제를 전송계층에서 해결하기 위해 나온 방안은 서비스 제공자에 프록시서버와 네트워크 접속점을 두어 종단간 보안을 해결하도록 하고 있다[1, 2]. 이러한 방안은 서비스 제공자에게 많은 부담을 갖게 하며, 더 많은 서비스 제공자가 생겨남에 따라 보안성이 중요한 서비스 제공자들이 프록시 서버에 보안 유지를 위한 관리와 대책에 대한 많은 비용을 감수해야 할 것이다.

단지 전송계층에서의 보안 문제점을 단순한 하드웨어적인 방안이 아닌 소프트웨어적인 구현으로 제공한다면 다양한 환경에서의 서비스 제공 및 새로운 기능의 개선이나 추가가 용이할 것이다.

Sun사는 자바2 에 J2ME이란 가전기기와 임베디드 장치를 위한 차세대 애플리케이션을 사용할 수 있는 플랫폼을 추가하였고, J2ME의 CLDC(Connected Limited Device Configuration)와 MIDP(Mobile Information Device Profile)는 휴대 전화나 양방향 무선 호출기 같은 소형 단말기용 무선 애플리케이션 개발을 위한 간편하고 확장성 있는 플랫폼을 제공하게 되었다.

자바가상머신(KVM)을 통하여 휴대폰 상에서 사용할 수 있게 하는 자바플랫폼을 개발하였고, 이를 통하여 기존 서비스의 문제점을 개선한다. 다소 제한적인 자바의 여러 특성을 휴대폰에서 이용하게 함으로 어떤 서비스를 필요로 할 때, 필요한 서비스에 대한 모듈을 장치의 변경 없이 서버로부터 필요한 모듈을 다운로드 받아 제공되게 된다. 이를 보안에 관련된 사항에 적용하여 해결이 가능해진다.

본 논문에서는 WAP 서비스에서의 보안 문제점을 분석하고, J2ME에서의 보안성을 확인하고, WAP에서의 종단간 보안 기능을 제공하기 위해 J2ME를 이용한 응용 레벨에서의 보안 모듈을 설계 및 구현하여 좀 더 안전한 고 신뢰성 있는 무선 인터넷 서비스 환경을 구축하고자 한다.

## 2. 관련 연구

### 2.1 WAP에서의 종단간 보안

유선 인터넷에서는 이미 정보보호에 대한 인프라가 구축되어 발전하고 있는데, 이 유선 인터넷을 새로운 무선 인터넷 환경으로 전환에 따른 해결로 유선에서의 SSL/TLS와 유사한 전송 보안으로 WTLS(Wireless Transport Layer Security)를 제공한다[3].

SSL/TLS가 유선 인터넷에서의 TCP/IP 상위에서 종단간 보안(End-to-End Security)를 제공한다. 이에 비해 무선 인터넷의 WAP 모델에서는 단말기와 서버사이에 프로토콜 변환을 위하여 WAP 게이트웨이가 필요로 하며, 이 WAP 게이트웨이에서 프로토콜간 변환을 위하여 일시적으로 암호화와 복호화를 제공해 보안에 문제점을 발생한다.

전송계층에서의 보안을 해결하기 위해 3가지 방식이 제시되었다.

첫 번째 방식은 Secure WAP Gateway를 서비스 제공자의 Web 서버가 있는 안전한 도메인 안에 위치시키는 방식으로, 무선 단말기와 Secure WAP Gateway가

WTLS를 이용하여 단대단 통신을 하여 보안을 해결하는 방식이며, 현재 WAP forum에서 명세화되었다[4].

두 번째 방식은 무선 단말기와 Web 서버의 응용 계층에서 암호화시키는 보안 모듈을 이용하여 보안성을 제공하는 방식이다. 본 논문에서 첫 번째 방식의 단점인 추가적인 비용 없이 제공될 수 있는 이 방식을 선택하였다.

세 번째 방식은 WTLS를 직접적으로 이용하는 방식으로 기존의 유선 통신망에 추가적인 비용과 유선 네트워크를 변형하여야 한다는 것으로 지금 현재의 유선 통신망을 그대로 이용할 수 있다는 WAP의 기존 취지에서 벗어나는 문제점이 있다.

### 2.2 J2ME의 보안 모델

일반적으로 컴퓨터에서 보안의 기본적인 목적은 악의적이거나 예기치 않은 접근으로부터 시스템을 보호하기 위한 것이다. CLDC 사양에서 지원되는 컨텐츠와 애플리케이션의 동적 다운로드는 네트워크 보안에 대한 중요한 문제이다.

자바는 보안을 고려하여 개발되었다. J2SE에서는 바이트코드 검증(bitecode verification)이나 시큐리티 매니저(Security Manager)와 같은 보안 기능이 풍부하게 지원한다. 하지만 J2SE 코드 중 보안을 위해 작성된 부분의 크기가 자원이 제한적인 무선 장치의 메모리 범위를 벗어난다. 따라서 제한적인 환경을 가진 무선 장치에 적합한 요구사항을 충족시키기 위해서 J2ME가 수정되었다[5, 6].

클래스 파일 검증 및 사전검증은 각각의 자바 클래스 파일들이 로드되면 클래스 파일은 먼저 유효한지 검사된다. 일반적으로 J2SE에서는 이런 검증 과정이 실행 시에 JVM에서 수행된다. 하지만 J2ME가 적용되는 무선 장치에서는 사용 자원이 제한적이기 때문에 실행 성능의 향상을 위해 클래스 파일의 검증은 장치 외부에서 일부가 수행되고 나머지 일부가 장치에서 수행된다. 장치 외부에서 수행되는 검증 과정을 사전 검증(preverification)이라고 부른다.

사전 검증된 클래스 파일은 장치 내부의 검증을 위해 버추얼 머신에 로드된다. 장치 외부의 사전검증과 장치 내부의 사전검증이 합쳐져서 안전성과 실행시의 무결성을 보장하게 된다.

J2ME의 Sandbox 모델은 J2SE에서 빌려와 사용한다. J2ME의 Sandbox 모델의 기초는 자바 애플리케이션이 이미 정의되어 있는 Configuration, Profile, 라이센스가 필요없는 오픈 클래스들에만 접근이 가능한 폐쇄적인 환경에서 실행되어야 한다는 것이다. Sandbox 모델은 다음과 같은 내용을 지원한다.

- 자바 클래스 파일들은 검증 과정을 거쳐 유효한 자바 애플리케이션임을 보장받아야 한다.
- CLDC, 프로파일, 라이센스가 필요없는 클래스들에서 정의된 것처럼 제한적이고 미리 정의된 자바 API들만을 이용해서 애플리케이션을 개발할 수 있다.
- 장치에서 자바 애플리케이션을 다운로드하고 관리하는 것은 버추얼 머신 내부에서만 가능하다. 표준 클래스 로딩 메커니즘이나 버추얼 머신을 프로그래머들이 오버라이드하는 것을 방지하기 위하여 어떠한 사용자 정의 클래스 로더도 사용할 수 없다.
- 버추얼 머신에서 native function에 접근할 수 없다. 이것은 애플리케이션 개발자들이 CLDC, 프로파일, 라이센스가 필요 없는 클래스에서 제공하는 자바 라이브러리가 아닌 native functionality를 포함한 새로운 라이브러리들을 다운로드 하거나 또는 native function에 접근할 수 없다는 것을 뜻한다.

## 3. WAP에서 Java기반의 종단간 보안 모듈 설계

종단간 보안 모듈은 그림 1과 같이 클라이언트의 요청에 의하여 WML 문서가 서버 보안 모듈에 의해 암호화되고 웹서버로부터 WAP 게이트웨이로 전송된 후 WAP 프로토콜을 위해 각 헤더 내용들이 변환된 후 클라이언트에게 전송되어 클라이언트 보안 모듈에 의해 복호화되고 사용자에게 보여지게 된다.

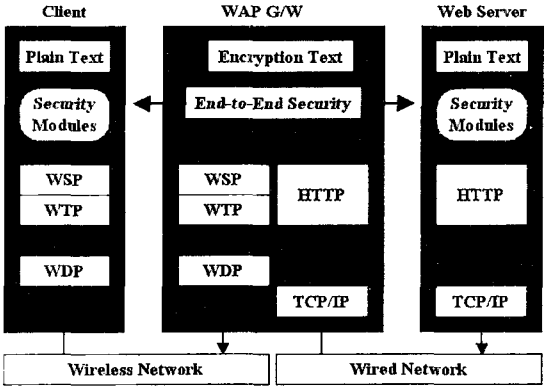


그림 1 종단간 보안 구현 개요

### 3.1 클라이언트 보안 모듈

휴대폰이라는 제한된 메모리에 적당한 응용프로그램을 구동시킬 수 있는 CLDC/MIDP 기반 위에 MIDlet 형태의 보안 모듈을 설계하여, 메시지에 대한 무결성과 기밀성, 부인방지, 사용자 인증을 적용하였다.

클라이언트 보안 모듈의 수행과정은 클라이언트가 메시지 입력하고, 입력한 메시지에 대한 무결성을 확인하기 위한 HASH 값을 생성한다. 서버와 이전에 연결 시 생성된 클라이언트와 서버간 세션변수를 키 값으로 추출하여 클라이언트 메시지와 생성된 메시지 HASH 값을 세션변수 키 값을 이용하여 암호화 시켜, 암호화된 데이터를 생성하여 서버로 전송하게 된다.

### 3.2 서버 보안 모듈

서버 보안 모듈은 Servlet에 의해 구현되어지고, 모듈 자체의 환경에는 제한적인 사항이 없기 때문에 클라이언트 보안 모듈과 유사한 모듈로 구성될 수 있다.

JSP 환경에서 구현되어지기 때문에 모듈 자체가 주변 환경과 호환성이 중요하였다.

서버 보안 모듈의 수행과정은 다음과 같다.

클라이언트로부터 암호화된 데이터 수신하고, 수신된 데이터를 클라이언트와 이전 연결에 저장된 클라이언트와 동일한 세션변수 키 값으로 복호화하여, 복호화된 데이터로부터 메시지에 대한 HASH 값 생성, 생성된 HASH 값과 데이터에 포함된 HASH 값 비교하여 메시지에 대한 무결성을 검토한 후 메시지를 처리한다.

클라이언트로 전송 시에는 클라이언트의 수행과정과 유사하나 연결 시 생성되는 세션 키 값을 저장하여 그 값을 유지해야 한다.

그림 2는 클라이언트/서버 보안 모듈에 대한 구성도이다.

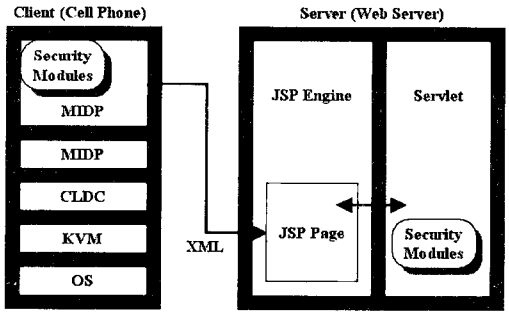


그림 2 클라이언트/서버 보안 모듈 구성도

## 4. 구현

종단간 보안 모듈을 구현하는 개발 환경은 표1과 같다.

표1 개발 환경

| 구분 | Client | WAP Gateway | Web Server |
|---|---|---|---|
| 사용기종 | Pentium-III 733 | Pentium-II 400 | Pentium-II 400 |
| 운영체제 | Windows 2000 pro. | RedHat Linux 7.0 | RedHat Linux 7.0 |
| 개발도구 | J2ME j2sdk-1_3 | | j2sdk-1_3 |
| 개발환경 | J2ME Wireless Toolkit 1.0.3 | kannel_1.1.5 | Tomcat |

종단간 암호화를 제공하는데는 클라이언트 측에서는 cipherMIDlet을 이용해서 메시지를 입력받아 암호화로 서버에 전송하게 된다. 암호화된 메시지는 서버 측에서 cipherServlet을 이용하여 복호화하고, 동일한 방식으로 cipherMIDlet과 cipherServlet을 통하여 상호 운용되어진다. 즉, 네트워크 사이에는 암호화된 데이터가 전송되는 것이다.

구현을 위해 J2ME에 암호화 알고리즘을 적용시킨 Bouncy Castle 암호 패키지에서 필요한 암호 클래스만을 포함시켜 사용하였다.

아래의 예는 RC4 암호화에 대한 암호/복호화를 보여준다.

암호화 객체로는 org.bouncycastle.crypto.engines.RC4를 사용하였다[7, 8].

```
// Message encoding
byte[] plaintext = mTextBox.getString();
byte[] ciphertext = new byte[plaintext.length];
mOutCipher.processBytes(Plaintext, 0,
                Plaintext.length, ciphertext, 0);
char[] hexCiphertext =
                HexCodec.bytesToHex(ciphertext);
```

암호화 객체는 전송 전에 암호문을 16진수 텍스트로 변환하여 서버로 전송한다.

```
// Message decoding
String hex = new String(ciphertext);
byte[] dehexed =
            HexCodec.hexToBytes(hex.toCharArray();
byte[] deciphered = new byte[dehexed.length];
mInCipher.processBytes(dehexed, 0, dehexed.length,
                deciphered, 0);
```

cipherMIDlet은 송신 시와 동일한 수신된 16진수 문자열을 byte 배열로 전환한 후, 복호화한 후 화면에 출력한다.

클라이언트 측에서는 평문을 디스플레이 되기 때문에 암호화된 내용은 볼 수 없었다.

## 5. 결론 및 향후 연구 과제

무선 인터넷이 기존 인터넷과 동일한 개념으로 가고 있는 현실에서 보안적인 요구사항 역시 필수적인 기술이다. 무선 인터넷에 대한 보안 기술은 무선망의 특성인 낮은 전송률, 높은 에러, 단말기 자원의 제약 등을 구현 환경적인 문제와 유선 인터넷과 연동되는 보안 수준 요구 사이에서 많은 연구가 진행 중이다.

WAP에서의 종단간 보안 문제점을 다양한 플랫폼에 적용할 수 있는 J2ME기반으로 보안 모듈을 작성하여 문제점을 해결하였다.

향후 연구 과제로는 무선 인터넷의 장비에 호환성을 가지고 다양한 환경에 가볍게 구현될 수 있도록 KVM의 소스에 암호화 라이브러리를 포함시켜서 속도에 대한 성능 향상을 기대해 볼 수도 있을 것이다.

## 6. 참고문헌

[1] 고양우, "무선 인터넷상의 보안 기술", 모바일컴아이
[2] WAP forum, "Wireless Application Protocol Architecture Specification", April, 1998.
[3] WAP forum, "Wireless Transport Layer Security", April, 2001.
[4] WAP forum, "End-to-end Transport Layer Security Specification", July, 2000.
[5] http://java.sun.com/j2me/
[6] Yu Feng & Dr. Jun Zhu, "Wireless Java Programming with J2ME", pp.52-53, June, 2001.
[7] Jonathan Kundsen, "Java Cryptography", 1998.
[8] http://www.bouncycastle.org