자바 무선 보안

상명대학교 장혜진*

1. 서 론

무선으로 인터넷에 연결될 수 있는 페이저(pager), PDA(Personal Digital Assistant Profile), 이동 전화(mobile phone) 등의 휴대형 무선 장비들이 급속히 보급되고 있다. 인터넷 지원 무선 장비들(Internet-enabled wireless devices)의 개수는 조만간 PC와 같은 전통적인 인터넷 클라이언트의 개수를 훨씬 능가할 것으로 예측되고 있다[1].

휴대형 무선 장비들을 통해 언제 어디서나 뉴스나 기상 정보를 보고, 전자 메일, 온라인 뱅킹(banking), 주식 거래, 일정 관리, 게임, 쇼핑(shopping) 등의 작업을 수행할 수 있다면 매우 편리할 것이다. 하지만 그런 유용한 서비스들을 제공하려면 PDA, 이동 전화 등의 휴대용 무선 장비들이 가진 다음과 같은 제약들을 고려해야만 한다.

- 휴대용 무선 장비들은 배터리를 전원으로 사용하는 경우가 많으며, 처리 속도, 기억 용량, 입력 방식, 화면 크기 등에 대한 제약을 갖는다.
- 휴대용 무선 장비들은 유선에 비해 낮은 대역폭, 느린 네트워크 속도, 낮은 접속 안정성, 높은 네트워크 연결 비용의 제약을 갖는다.
- 휴대용 무선 장비들의 규격 및 프로토콜들이 매우 다양하며 서로 호환하지 않는 경우가 많다.

Sun Microsystems사가 1999년 JavaOne 컨퍼런스에서 발표한 J2ME(Java 2 Micro Edition)는 페이저, 휴대 전화, PDA, 디지털 셋톱박스와 같은 소형 장비들을 위해 설계되었다. J2ME가 지원하려는 소형 장비들에는 다양한 휴대형 무선 장비들이 포함된다. 이 글은 자바 기술 중에서 특히 J2ME를 중심으로 무선 보안에 관련된 주제를 다룬다. 제 2 장에서는 무선 환경에서의 자바 기술의 필요성과 J2ME에 대해 살펴본다. 제 3장에서는 무선 환경에서의 단대단(end-to-end) 보안 문제에 대하여 살펴본다. 제 4장에서는 J2ME가 지원하는 컨피규레이션(configuration)의 하나인 CLDC (Connected, Limited Device Configuration)와 무선 보안에 관련된 내용을 다룬다. 제 5장은 결론을 맺는다.

2. 무선 환경에서의 자바 기술

많은 종류의 이동 전화, PDA 등의 무선 단말들이 자바 기능을 채택하고 있다. 3Com사의 Palm, COMPAQ 사의 iPAQ Pocket PC 등의 대부분의 PDA들이 자바 환경을 지원하고 있으며, 이동 전화 쪽에서도 LG 텔레콤, SK 텔레콤과 같은 국내 통신 서비스 사업자들뿐 아니라 노키아(Nokia), 모토롤라(Motorola) 등의 해외 통신업체에서도 자바를 지원하는 다양한 이동 전화기들을 출시하고 있다. 아래의 표 1은 국내의 이동 통신 업체들이 지원하는 이동형 전화기들에서 사용된 가

표 1 국내 이동 통신 업체와 가상 기계

가상 기계	GVM	SKVM	MAP	LGT	BREW
서비스 사업자	SK 텔레콤	SK 텔레콤	KTF	LG 텔레콤	KTF
개발 환경	Mobile C 기반	Java	C	Java	C
개발사	신지소프트	XCE	모빌탑	벨록스소프트	퀄컴
방식	스크립트	스크립트	스크립트	스크립트	바이너리

* 종신회원

상 기계(virtual machine)들이다[2]. 5개의 대표적인 가상 기계들 중 2가지가 자바를 사용함을 볼 수 있다.

2.1 왜 자바의 사용을 고려하는가

자바 기술은 휴대형 무선 장비들을 위한 유력한 기반 기술들 중의 하나이다. 자바가 무선 서비스 업체 및 장비 생산 업체들에게 유력한 기술로 평가되는 이유들은 다음과 같다[3].

• 응용 프로그램 및 서비스의 동적인 배포 가능

자바 기술을 사용하면 PDA, 이동 전화 등의 무선 장비들이 실시간으로 필요한 응용 프로그램을 다운로드 하는 것이 가능하다. 예를 들어, 심심할 때 이동 전화로 자바 게임을 다운로드 하여 즐길 수 있을 것이며, 여행 정보가 필요할 때 도시 안내 정보를 다운로드 하여 근처의 호텔을 찾아낼 수 있을 것이다. 또한 단말기의 교체 없이 단말기 내장 프로그램을 다운로드 하여 업그레이드 할 수 있을 것이다.

• 플랫폼간의 교차 호환성 제공

단말기가 다르다고 같은 프로그램을 다시 개발하는 것은 매우 번거로운 일이다. 자바로 작성된 응용 프로그램은 재개발이나 수정 또는 재컴파일 없이 다양한 단말기들에서 수행될 수 있다. 예를 들어, 이론적으로 모토롤러의 이동 전화를 위해 작성한 프로그램이 동일한 컨피큐레이션(configuration)과 프로파일(profile) 명세를 갖는 노키아의 이동 전화에도 다운로드 되어 수행될 수 있다. 플랫폼 교차 호환성(cross-platform compatibility)은 단말기, 장비 제조업체, 컨텐츠 제공자들 모두에게 매우 중요하다.

• 보다 유용하고 풍부한 기능을 사용자에게 제공 가능

WAP이나 NTT의 i-mode와 같은 브라우저 기반 서비스 보다 자바 기술을 사용하는 서비스가 보다 풍부한 그래픽 효과, 보다 신속한 반응 속도 등을 제공할 수 있다. 예를 들어, 자바 기술을 사용하면, 예를 들어, 현재 위치를 중심으로 도시 지도를 다운로드 하여 보여주는 서비스, 테트리스와 같은 게임, 콘서트 입장권 경매 등의 서비스와 같은 동적이며 사용자 교감적인 서비스들을 제공할 수 있다.

• 네트웍 단절 상태에서 동작 가능

WAP 방식에서와 달리, 이동형 장비가 서비스 영역을 벗어나거나 온라인 상태가 아닌 상태에서도 단말기로 다운로드 된 자바 응용 프로그램들은 수행될 수 있다. 접속되지 않은 상태에서 동작하며 나중에 다시 접속이 된 상태에서 동기화를 수행하면 충분한 다양한 응용 분야들이 존재한다.

• 보안성 강화

무선 인터넷 분야에서는 아직 해결되지 않은 보안 문제점들이 많다. 예를 들어, WAP, Parm.net, i-mode 등의 방식은 무선 구간과 유선 구간을 연결하는 어떤 프록시(proxy) 또는 게이트웨이(gateway)를 필요로 하며, 그 게이트웨이는 무선 구간과 유선 구간과의 프로토콜 변환을 수행하기 위하여 통신 내용에 대하여 상당히 이해해야만 하므로, 구조적으로 게이트웨이에 보안상의 틈이 존재하게 된다. 즉, WAP과 같이 게이트웨이를 사용하는 방식은 구조적으로 단말기와 서버 간의 단대단 보안(end-to-end security)을 지원하기 힘들다. 하지만, 차세대 이동 전화와 같은 미래의 휴대형 무선 장비들은 TCP/IP 상에서 동작하게될 가능성이 많으며, 자바로 TCP/IP를 사용하는 이식성이 강한 응용 프로그램을 작성하는 것은 매우 쉽기 때문에 자바 기술을 사용하는 휴대형 무선 장비들은 서버와의 단대단 보안을 보다 용이하게 구현할 수 있을 것이다.

2.2 J2ME(Java 2 Micro Edition)

J2ME는 호출기, 휴대 전화, PDA, 셋톱박스와 같은 제한된 자원들을 갖는 소형 장치들을 위한 자바이다.

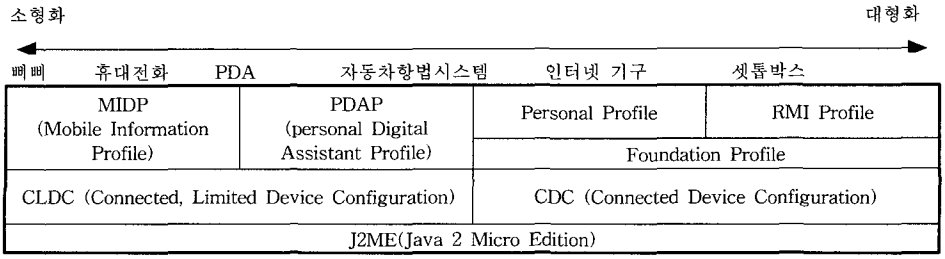

그림 1 J2ME의 개요

J2ME는 무선 환경상의 장비들만을 대상으로 개발된 것은 아니지만 인터넷 지원 무선 장비들에게 적합한 기능들을 제공한다.

J2ME는 다양한 종류의 단말들에 대응하기 위하여 컨피규레이션(configuration)과 프로파일(profile)의 개념을 사용한다. 다음의 그림 1은 J2ME의 개요를 보여준다[4]. 현재 개발중인 컨피규레이션은 CDC(Connected Device Configuration)와 CLDC(Connected, Limited Device Configuration)로 나누어진다.

CDC는 32 비트 마이크로프로세서와 2M 바이트 이상의 메모리를 가진 페이저, PDA, 대화형 디지털 TV 셋톱박스와 같은 가전이나 내장 장비(embedded device)들을 위한 자바 가상 머신 CVM과 기본 클래스 라이브러리들을 제공한다[5]. CLDC는 16/32 비트 마이크로프로세서와 160 KB 정도의 메모리를 갖는 보다 제약적인 장비들에게 가상 머신 KVM과 기본 클래스 라이브러리들을 제공한다[6].

MIDP는 이동 전화, 적은 자원을 갖는 PDA 등을 위한 완전한 J2ME 응용 런타임(runtime) 환경을 제공하는 자바 API들의 집합이다. MIDP는 사용자 인터페이스, 지속성 저장소(persistence storage), 네트워킹, 응용 생명 주기 등을 규정한다[7].

3. 무선 보안

무선 환경에서의 보안 방식은 그 수준에 따라 프록시 기반 구조에서의 보안 방식, 단말과 서버간의 단대단 보안 방식, 그리고 단말과 단말간의 단대단 보안 방식으로 분류할 수 있다.

3.1 프록시 기반 구조에서의 보안 방식

프록시 기반 구조(proxy-based architecture)란 유선 구간과 무선 구간의 프로토콜 변환을 위한 프록시 또는 게이트웨이를 사용하는 구조를 의미한다.

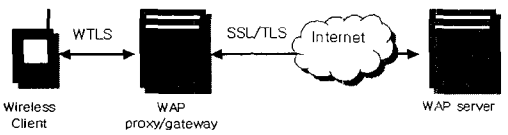

그림 2 프록시 기반 보안 구조

위 그림 2는 무선 구간과 우선 구간을 연결하는 프록시/게이트웨이를 사용하는 구조에서의 통신 보안 방식을 보여준다. 무선 단말기와 게이트웨이간에는 보안을 위하여 WTLS(Wireless Transport Layer Security)[8] 등의 보안 프로토콜이 사용된다. WTLS는 무선 구간의 보안을 위하여 SSL을 경량화시킨 프로토콜이다. 게이트웨이와 서버간의 보안을 위해서는 SSL/TLS[9]등의 보안 프로토콜이 사용된다. 하지만 WAP 게이트웨이는 무선과 유선 구간의 프로토콜간의 변환 뿐 아니라 어떤 경우에는 내용 변환도 수행해야 하므로 게이트웨이에서 보안적 틈이 발생한다. WAP 게이트웨이를 사용하면서 단말과 서버간의 종단간 보안을 보장받는 한 방법은 응용 계층(application layer)에서 담당하는 것이다. 즉, 단말과 서버의 WAP 방식의 인코딩/디코딩 수행과 무관하게 응용 계층에서 데이터 부분을 암호화/복호화 하는 방식이다. 하지만 이 방식은 비표준 방식이며 응용 계층에서의 프로그래밍 및 처리 부담이 크다. 프록시 기반 구조의 대표적인 문제점들은 다음과 같다[1].

• 규모 확장성(scalability) 부족 문제

프록시 기반 구조는 프록시를 사용하는 많은 무선 클라이언트에서 송신되거나 수신되는 데이터 패킷들을 처리해야 하므로 병목 현상을 발생시킬 수 있다. 또한 일반적으로 유선 구간의 밴드폭(bandwidth)이 무선 구간의 밴드폭 보다 크므로 프록시는 많은 데이터들을 버퍼링(buffering)해야만 하는 부담을 갖게 된다.

• 법적 문제

일반적으로 프록시(또는 게이트웨이)는 통신 업체에서 제공하고 관리한다. 사용자가 프록시를 스스로 선택할 수 없다(또는 어렵다)는 문제로부터 법적 문제들이 발생될 수 있다.

• 보안 문제

무선 서비스를 위하여 무선 서비스 주체가 스스로 프록시를 관리하는 경우는 드물다. 중요한 기밀들을 제 3자가 관리하는 프록시를 통해 통신한다는 것은 심각한 문제가 된다. 또한, 무선쪽 구간에서는 암호 등의 보안 기법이 사용되지 않거나 아주 약한 강도의 암호 알고리즘들이 사용되는 경우가 많다는 점도 문제이다.

3.2 단대단 보안 방식

단대단 보안(end-to-end security) 방식이란 통신의 쌍방이 동일한 어떤 통신 보안 프로토콜을 사용하여 논리적으로 직접 통신 보안을 확보하는 방식을 의미한다. 통신의 쌍방은 동일한 통신 보안 프로토콜을 사용하여 통신하므로 WAP 게이트웨이와 같은 프로토콜

변환을 위한 장치가 필요하지 않다. 이 때 이론적으로는 통신 쌍방 간에 다양한 새로운 프로토콜들이 사용될 수 있겠지만 다음과 같은 이유 때문에 현실적으로는 TCP/IP에 호환하는 보안 프로토콜들이 바람직하다.

• SSL과 같은 기존의 TCP/IP 호환 프로토콜들은 이미 그 보안성 및 효과들이 철저히 검증되었다.

• 무선 보안을 필요로 하는 대부분의 응용 분야가 유선 인터넷과의 연동을 필요로 하므로, 무선 단말에서도 TCP/IP와 같은 유선 인터넷 프로토콜에 호환하는 보안 프로토콜의 사용이 바람직하다.

TCP/IP에 호환하는 SSL과 같은 기존의 인터넷 보안 프로토콜을 직접 무선 단말에 사용하려고 할 때 가장 문제가 되는 것은 무선 단말들은 일반적으로 자원의 제약이 심하여 SSL과 같은 인터넷 보안 프로토콜을 그대로 사용하는 것이 현실적이지 않다는 점이다. 특히 다음과 같은 요소들이 문제가 된다.

• 신원확인을 위해 전자 서명이 사용되는 경우 전자 서명에 사용되는 RSA와 같은 비대칭 암호 알고리즘들의 복잡도가 문제가 된다. 보다 효과적인 비대칭 암호 알고리즘으로 타원 곡선 암호(Elliptic Curve Cryptography)등이 사용되기도 하지만 아직 휴대 전화와 같은 휴대형 무선 단말에서는 비대칭 암호 알고리즘은 너무 느리다.

• 인증서의 유효성 검증은 휴대형 무선 단말이 수행하는 데 부담스런 작업이다. 휴대형 무선 단말은 기억 공간에 제약을 가지므로 여러 장의 인증서들을 보관하기 어려우며 인증서 폐기 목록 등의 처리가 어렵다. 이런 문제점을 해결하기 위하여 단기 인증서(Short Lived Certificate) 또는 OCSP(Online Certificate Status Protocol) 등이 사용되지만 보다 효과적인 인증서 관련 메커니즘이 요구되고 있다.

단대단 보안 방식도 그 통신 주체에 따라 단말과 서버의 단대단 보안과 단말과 단말의 단대단 보안으로 구분될 수 있다. 서버는 단말과 달리 충분한 자원을 갖는다.

3.2.1 단말과 서버간의 단대단 보안

단말과 서버간의 단대단 보안 방식이란 그림 3에서와 같이 무선 단말이 서버와 어떤 프로토콜(예를 들어 KSSL[10])을 사용하여 논리적으로 직접 통신 보안을 확보하는 방식을 의미한다. 단말과 서버는 동일한 프로토콜을 사용하여 통신하므로 WAP 게이트웨이와 같은 프로토콜 변환을 위한 장치가 필요하지 않다. 예를 들어, HTTPS는 PC와 웹 서버간의 단대단 보안 프로토콜이다. HTTPS는 SSL을 통해 단말과 서버간의 단대단 유선 보안을 확보한다. SSL은 오랜 시간동안 검증되었으며 현재 대부분의 웹 브라우저와 웹 서버가 지원하는 보안 프로토콜이다. SSL은 통신 쌍방의 신원 확인, 암호화, 완전성 보장 등의 보안을 제공한다. 무선 환경에서도 무선 단말기와 웹 서버간에 SSL과 같은 보안 프로토콜이 제공된다면 바람직하다.

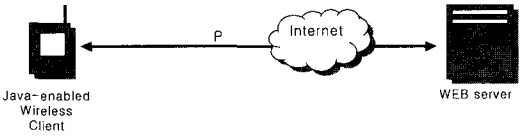

그림 3 웹 서버와 무선 클라이언트간의 단대단 보안

그림 3과 같은 단대단 보안 구조에서는 프록시 방식에서의 보안 문제가 존재하지 않으며, 프록시가 통신의 병목이 되는 현상도 없어질 수 있다. 하지만 일반적으로 무선 단말기들은 처리 능력이나 메모리 등의 자원이 충분하지 않다는 제약을 가지므로 유선 통신에서 사용되는 SSL과 같은 보안 프로토콜을 그대로 무선 환경에서 사용하는 것은 비현실적이다.

무선 단말과 서버간의 단대단 보안을 위한 흥미로운 연구의 하나는 KSSL[10]이다. KSSL은 무선 단말이 웹 서버와 단대단 보안 통신을 하기 위한 보안 프로토콜이다. KSSL은 무선 단말 쪽에만 적용되는 프로토콜이다. 즉, 충분한 자원을 갖는 웹서버는 SSL을 그대로 사용하며 부족한 자원을 갖는 무선 단말은 KSSL 프로토콜을 사용한다. KSSL은 SSL 프로토콜에서 가장 복잡하며 시간이 많이 걸리는 신호 변경(handshake) 부분을 간략하게 만든 것이다.

일반적으로 무선 휴대 전화와 같은 무선 단말에서는 SSL과 같은 프로토콜이 현실적이지 않다고 판단되어 왔으나 최근의 연구 결과는 SSL을 수정하여 복잡성을 줄인 KSSL이 휴대 전화나 PDA 등에서 현실적으로 사용될 수 있음을 보여준다[1, 10].

3.2.2 단말과 단말간의 단대단 보안

단말과 서버간의 단대단 보안의 확보가 단말과 단말간의 단대단 보안의 보장을 의미하지는 않는다. 예를 들어, 단말 A와 단말 B가 각각 웹 서버와 SSL 통신을 할 수 있다 하더라도, 단말 A의 메시지 M_A를 웹 서버를 통하지만 웹 서버가 모르게 단말 B로 안전하게

전달하는 것은 어렵다. 웹 서버를 중간에 두고 단말들이 웹 서버를 통해 SSL 통신을 하는 경우 WAP 게이트웨이에서와 유사하게 웹 서버에 보안 틈이 발생할 수 있기 때문이다.

최종적인 통신 보안의 목표의 하나는 그림 4와 같이 단말간의 단대단 보안을 제공하는 것일 것이다. 그림 4는 그림 3과 달리 서버와 단말간의 단대단 보안이 아니라 단말과 단말간의 단대단 보안 개념을 보여준다. 물론 이 때 단대단의 개념은 논리적인 것이다. 물리적으로는 단말과 단말 사이에 여러 중계기나 서버들이 존재할 수 있다. 단말과 단말간의 단대단 보안 문제는 단말과 서버간의 단대단 보안 문제보다 훨씬 어려운 문제이다. 특히, 무선 휴대형 단말간의 단대단 보안은 단말들의 부족한 자원 때문에 더욱 어려운 문제가 된다.

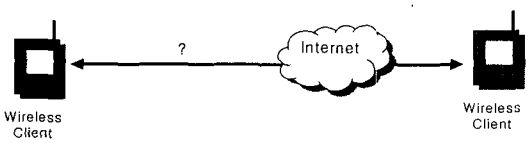

그림 4 무선 클라이언트들간의 단대단 보안

그림 4와 같은 개념의 단대단 보안을 제공하는 실용적인 프로토콜의 구현은 현재로는 비현실적일 수 있다. 하지만, 아주 빠른 속도로 이동 전화, PDA등의 무선 단말들의 프로세서, 메모리, 통신 속도 등이 발전하고 있으므로 단말과 단말간의 단대단 보안 구조는 일반적인 예상보다 빨리 현실성을 갖게 될 수 있다. 무선 단말에서의 자바의 장점들을 고려한다면, 자바를 사용하는 단말들간의 단대단 보안이 우선적으로 현실화될 수도 있을 것이다.

4. CLDC/MIDP와 무선 보안

J2ME의 컨피규레이션과 프로파일 중에서 휴대형 무선 장비들과 보다 밀접한 관련이 있는 것은 CLDC/MIDP이다. J2SE(Java 2 Standard Edition)의 보안 모델은 CLDC/MIDP에 적합하지 않다. 왜냐하면 J2SE의 보안 모델에 요구되는 자원, 그 중에서도 특히 주기억장치의 크기가 CLDC/MIDP에는 너무 크기 때문이다.

4.1 J2SE 보안 모델

J2SE의 보안 모델은 자바 언어 자체, 자바 컴파일러와 런타임 시스템, 보안 관리자(즉, SecurityManager 객체)의 3계층으로 구성된다[11]. J2SE의 응용 수준 보안은 보안 관리자(SecurityManager 객체), 접근 제어기(access controller), 보안 정책(security policy)등에 의존한다. 하지만 J2SE의 보안 모델은 부족한 자원을 갖는 장비들을 지원하기 위한 CLDC의 보안 모델로는 적합하지 않다. 따라서 CLDC는 J2SE와 다른 보안 모델을 사용한다.

4.2 CLDC/MIDP 보안 모델

CLDC의 보안 모델은 저수준 보안(low-level KVM security)과 응용 수준 보안(application-level securioty)으로 구분될 수 있다. 저수준 보안 모델은 장비에 다운로드 되어 KVM에 의해 수행되는 MIDP 응용 프로그램이 자신이 수행되고 있는 장비에 아무런 해를 주지 않도록 하기 위한 것이다. 저수준 보안은 클래스 파일 검증기(class file verifier)에 의해 보장된다. 클래스 파일 검증은 매우 복잡한 작업이므로 소형 무선 장비에서는 수행하기 어렵다. 따라서 CLDC/MIDP에서는 클래스 파일 검증을 그림 5와 같이 두 단계로 나누어 수행한다[2, 11].

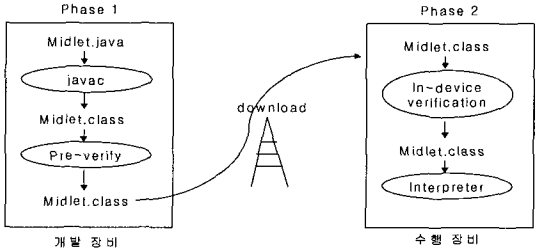

그림 5 두단계 검증

• 선검증(pre-verification)

선검증은 PC와 같은 개발 장비에서 수행된다. 선검증의 결과로 생성된 클래스 파일에는 선검증의 입력으로 주어진 클래스 파일과 검증에 관련된 부가 정보 등이 추가된다.

• 장비내 검증(in-device verification)

장비내 검증은 그 클래스 파일이 검증된 것인지 검증후 수정되지 않았는가를 확인하기 위한 몇 가지 조건만을 검증한다. KVM은 선검증되지 않았거나 선검증되었지만 검증후 수정된 클래스 파일의 수행을 거부한다.

J2SE의 모래 상자 보안 모델(sandbox security model)은 충분한 자원을 필요로 하는 보안 모델이므로, CLDC에서는 응용 수준 보안을 위하여 새로운 모래 상자 모델, 즉 CLDC 모래 상자(sandbox) 모델이 사용된다. CLDC 모래 상자 모델[2, 11]은 다음과 같다.

- 자바 클래스 파일들은 적절히 검증되어야 한다.
- 응용 프로그래머에게는 제한적이고 미리 정의된 API 집합만이 제공된다.
- 자바 프로그램의 다운로드와 관리는 KVM 내의 네이티브 코드(native code) 수준에서 일어나며, 응용 프로그래머가 클래스 적재기(class loader)를 치환(overriding)하거나 사용자 정의 클래스 적재기를 사용할 수 없다.
- 네이티브 코드를 포함하는 새로운 라이브러리를 다운로드 하거나 CLDC와 MIDP에 의해 주어지는 라이브러리의 일부가 아닌 네이티브 기능에 접근할 수 없다.
- 리플렉션(reflection)을 지원하지 않는다.
- 쓰레드 그룹이나 데몬 쓰레드를 지원하지 않는다.
- 약참조(weak reference)를 지원하지 않는다.
- 가상 머신의 시스템 클래스 즉, java.* 와 javax.microedition.* 클래스들을 치환할 수 없다.

4.3 자바 무선 네트워크 데이터 보안

CLDC/MIDP의 모래 상자 보안 모델은 무선 단말기를 악성 코드로부터 보호하는 것을 주된 목적으로 한다. 무선 전자상거래와 같은 무선 응용을 위해서는 별도로 축약, MAC, 암호, 전자 서명과 같은 데이터 보안 체계가 필요하다. 일반적으로 무선 환경은 서론에서 기술된 제약들을 가지므로 유선에서의 데이터 보안 체계 보다 더 가볍고 효율적인 데이터 보안 체계를 필요로 한다. 자바로 구현된 데이터 보안 체계를 구현하려는 경우, J2ME는 J2SE의 일부 기능만을 지원하므로 J2SE를 이용하여 구현된 데이터 보안 체계를 J2ME에서 그대로 사용하기는 어렵다는 점을 고려해야 한다.

Sun Microsystems 사는 J2SE에서 JCA(Java Cryptography Architecture)와 JCE(Java Cryptography Extension)을 통해 데이터 보안을 제공하였다. 하지만, JCA나 JCE는 MIDP 환경과 같은 제약적인 환경에서 사용하기에는 매우 무겁다. 현재까지 Sun Microsystems사는 J2ME 플랫폼에 어떠한 데이터 보안도 지원하고 있지 않다.

Sun Microsystems 사는 1999년 JavaOne 컨퍼런스에서 MIDP 환경에서의 SSL 지원을 위하여 KSSL[10]을 제안하였지만 아직 KSSL과 관련된 어떤 제품도 선보이지 않았다.

5. 결 론

무선 인터넷 보안에 관련된 일반적으로 가장 쟁점이 되는 문제들은 무선 인터넷 보안 기술의 경량화 문제와 종단간 보안 제공 문제라고 할 수 있다. 무선 인터넷 환경에서 충분한 성능을 갖는 보안 알고리즘, 인증서 검증 체계 등이 개발되어야 한다. 무선 단말기의 하드웨어적 특성에 맞는 암호, 축약, 전자 서명 등의 보안 알고리즘들에 대한 기초 연구도 필요하다. 또한 통신의 종단간의 보안을 효과적으로 보장하기 위한 소프트웨어적, 하드웨어적인 체계가 개발되어야 한다. 무선 단말에 암호 알고리즘을 효과적으로 처리하기 위한 하드웨어의 결합하기 위한 연구도 이루어지고 있다.

자바를 지원하는 무선 단말은 그렇지 않은 단말들에 비교하여 2장에서 언급한 여러 가지 장점을 갖는다. 따라서 점차로 많은 종류의 무선 단말들이 자바를 지원할 것이라 예상되며 자바 지원 단말들에서의 통신 보안을 위한 보다 많은 연구들이 필요하다.

자바를 지원하는 단말들은 자바(특히 J2ME)의 보안 모델에 의해 그 단말 상에서 동작하는 응용 프로그램들(즉 자바 프로그램들)에 의해 해를 받지 않도록 되어있다. 또한 자바 언어는 TCP/IP를 매우 철저히 지원하는 언어의 하나이므로 자바 언어를 지원하는 단말에서는 TCP/IP에 호환하는 단대단 보안 프로토콜을 구현하는 것이 비교적 용이하다고 할 수 있다.

일반적으로 단대단 보안 기술은 아직 매우 부족하다. 무선 단말과 서버간의 단대단 보안은 실험적 단계에 있다. 무선 단말과 무선 단말간의 단대단 데이터 보안에 대한 연구는 더욱 부족하다고 할 수 있다.

참고문헌

[1] Vipul Gupta, Sumit Gupta, "Securing the Wireless Internet", IEEE Communication Magazine, Dec 2001.

[2] 강성윤, 이경범, 홍성인, "클릭하세요 자바 모바일 프로그래밍", 도서출판 대림, 2002년 1월

[3] Martin Hardee, "Why Wireless needs Java Technology", http://java.sun.com/features/2000/

07/woreless.print.html
[4] Jonathan Knudsen, Wireless Java: Developing with Java 2, Micro Edition, APress, 2001
[5] Connected Device Configuration (CDC) and the CVM Virtual Machine, http://java. sun.com/products/cdc
[6] CLDC and the K Virtual Machine (KVM), http://java.sun.com/products/cldc
[7] Mobile Information Device profile(MIDP); http://java.sun.com/products/midp
[8] WAP Forum, "Wireless Transport Layer Security Specification"; http://wapforum.org/what/technical.html
[9] A. Frier, P. Karlton, and P. Kocher, "The SSL3.0 Protocol Version 3.0"; http:// home.netscape.com/eng/ssl3
[10] Vipul Gupta, and Sumit Gupta, "KSSL: Experiments in Wireless Internet Security", SMLI TR-2001-103, Sun Microsystems, Nov., 2001.
[11] Qusay Mahmoud, "Wireless Java Security", http://wireless.java.sun.com/midp/articles/security/

장 혜 진

1985 서울대학교 사범대학 수학교육과 졸업(학사)
1987 서울대학교 대학원 계산통계학과 졸업(석사; 전산과학 전공)
1987~1989 한국전자통신연구소 연구원
1994 서울대학교 대학원 계산통계학과 졸업(박사; 전산과학 전공)
1994~현재 상명대학교 컴퓨터정보통신학부 부교수
관심분야 데이터베이스 응용, 에이전트 시스템, Java, 유무선 PKI 보안 시스템
E-mail: hjchang@smuc.ac.kr