WTLS 클라이언트 설계

김동주° 김상욱*
경북대학교 정보통신학과
*경북대학교 컴퓨터과학과
{djkim, swkim}@woorisol.knu.ac.kr

Design of WTLS Client

Dong-Ju Kim° Sang-Wook Kim*
Dept. of Information and Communication
*Dept. of Computer Science

요 약

최근 무선 인터넷 사용이 보편화됨에 따라 인터넷 비즈니스가 장소의 제약없이 이루어지고 무선 전자상거래가 일상적으로 발생하고 있다. 이러한 무선 전자상거래가 이루어질 때 노출되기 쉬운 사용자 신용정보를 보호하기 위한 보안 기능이 필요하다. 본 논문은 WAP 스택에서 이동 단말기와 WAP 게이트웨이간의 통신에 보안 기능을 제공하는 계층인 WTLS의 클라이언트 구조를 설계하는데, 이는 암호화 처리와 이벤트 처리 등을 위한 여러 모듈과 테스트 프로그램으로 구성된다.

1. 서론

최근 무선 전화망에서의 빠른 기술 발전과 휴대용 이동 단말기의 엄청난 보급으로 무선 인터넷 사용이 보편화됨에 따라 인터넷 비즈니스가 장소의 제약 없이 이루어지고, 무선 인터넷을 통한 물품 구매 또한 일상적으로 발생하고 있다. 이러한 무선 전자상거래가 이루어질 때 노출되기 쉬운 사용자 신용정보를 보호하기 위한 보안 기능이 필요하다.

본 논문은 WAP Forum에서 정의한 표준인 WAP(Wireless Application Protocol) 스택에서 이동 단말기와 WAP 게이트웨이간의 통신에 보안 기능을 제공하는 계층인 WTLS(Wireless Transport Layer Security) 클라이언트 모델을 제시한다. 이동 단말기에 탑재되는 WTLS 클라이언트는 WAP 게이트웨이의 WTLS 서버와 연동하여 사용자 데이터를 보호하게 된다.

본 논문의 구성은 다음과 같다. 제 2 절에서 WTLS의 구조와 기능을 간략하게 살펴보고, 제 3 절에서 WTLS 클라이언트 구조의 여러 모듈들과 실험 프로그램에 관해 기술하고 제 4 절에서 결론을 맺는다.

2. WTLS 구조와 기능

WTLS의 주된 목적은 두 통신 애플리케이션간의 기밀성(privacy), 데이터 무결성(data integrity) 그리고 인증(authentication)을 제공하는 것이다. WTLS는 기본적으로 TLS 1.0 (Transport Layer Security)과 유사한 기능을 제공하면서 동적 키 갱신과 같은 새로운 기능을 포함하고 있다.[1]

WTLS는 그림 1에서 볼 수 있듯이 크게 handshake 프로토콜과 record 프로토콜로 이루어지는데 handshake 프로토콜은 change cipher spec, alert, handshake 세 개의 부 프로토콜로 구성된다.

2.1 Handshake 프로토콜

Handshake 프로토콜은 서버와 클라이언트가 서로를 선택적으로 인증하고, 세션 정보와 연결 정보를 생성하는 프로토콜이다. WAP 게이트웨이와 클라이언트가 데이터를 주고받기 전에 수행되며 프로토콜 버전 확인, 암호 알고리즘 선택, 선택적인 인증, 공개키 암호를 이용한 비밀 정보 공유 등이 이루어진다. Change cipher spec 프로토콜은 뒤따르는 메시지들이 새로이 협의된 암호 스펙에 의해 보호됨을 수신측에게 알리고, alert 프로토콜은 handshake 프로토콜 실패, 암호 오류, 인증서 오류 등에 대한 메시지를 전송한다.

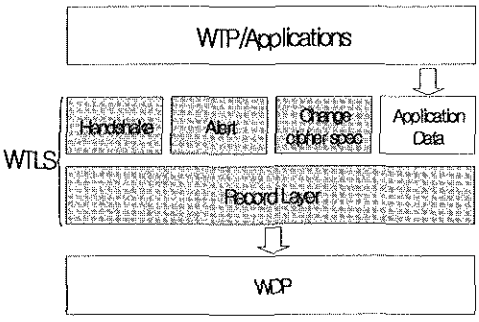

그림 1. WTLS 구조

2.2 Record 프로토콜

Record 프로토콜은 handshake 프로토콜에서 설정된 암호 스펙을 이용하여 전송되는 메시지에 대해 기밀성과 무결성을 제공하는 프로토콜이다. 전송 과정을 살펴보면, 먼저 상위 응용 계층으로부터 임의의 크기의 데이터를 전달받고, 경우에 따라서 압축을 한 후, MAC(Message Authentication Code)을 붙이고 암호화하여 WDP(Wireless Datagram Protocol) 계층으로 전달한다. 수신측에서는 수신된 데이터를 복호화하고, MAC의 유효성을 검증한 후, 압축을 해제하여 응용 계층으로 넘겨준다.

MAC용 알고리즘은 SHA, MD5 등이 사용된다. 클라이언트는 client write MAC secret을 사용하여 MAC을 생성하고, server write MAC secret을 사용하여 수신된 메시지의 MAC을 검증한다. 한편 MAC 생성 시 전송순서 번호를 포함시켜 메시지 오류 또는 변경을 감지할 수 있게 한다.[3]

3. WTLS 클라이언트 설계

WTLS 클라이언트 구조는 그림 3에서 볼 수 있듯이 WTLS 서버나 실험 프로그램과 메시지를 주고 받는 메시지 송수신 모듈과 이벤트를 저장하는 이벤트 큐, 저장된 이벤트를 처리하기 위한 이벤트 처리 모듈과 메시지를 암/복호화하거나 MAC을 생성, 검증하는 암호 모듈로 구성된다.

3.1 WTLS 클라이언트 세션 상태

WTLS 클라이언트는 NULL, CREATING, EXCHANGE, COMMIT1, COMMIT2, CREATED, OPENING, OPEN의 8가지 보안 세션 상태를 가진다(그림 2). NULL로 초기화되는 WTLS 클라이언트 세션 상태는 WTLS 서버로부터 수신되는 이벤트와 자체적으로 생성하는 이벤트의 종류에 따라 상태전이가 일어난다.

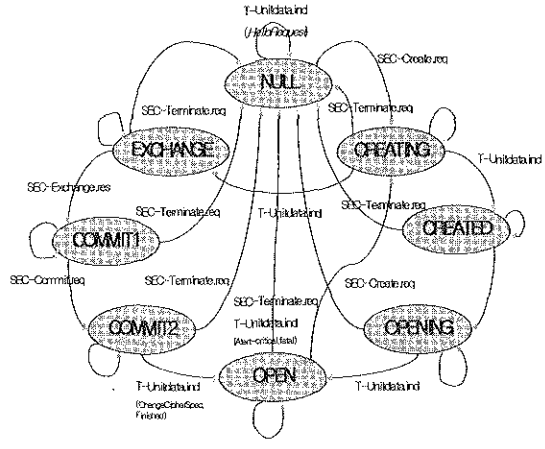

그림 2. WTLS 클라이언트 세션 상태전이도

3.2 이벤트 처리 모듈

WTLS 클라이언트는 서버로부터 hello request 메시지를 수신하거나 서버로 client hello 메시지를 보냄으로써 서버와의 보안 연결을 시작한다. WTLS 프로토콜 동작 시 발생하는 이벤트들은 모두 이벤트 큐에 삽입하고, 이벤트 처리 모듈은 이들 이벤트를 순서대로 읽어들여 이벤트의 유형을 분석한다. 분석된 이벤트 유형에 따라 클라이언트 세션 상태 테이블에서 현재의 세션 상태에 해당되는 처리 동작으로

써 새로운 이벤트를 만들어 이벤트 큐에 삽입하기도 하고, 메시지를 서버로 보내기도 한다. 이 과정에서 WTLS 클라이언트 세션 상태는 변화하게 된다.

3.3 암호 모듈

WTLS는 두 통신 개체간의 기밀성을 보장하기 위해 암호화 알고리즘을 사용한다. 또한 메시지가 전송되는 과정에서 불법적인 제 3 자에 의한 메시지 변조 여부 확인을 위해 송신 메시지에 MAC을 부가하거나 수신 메시지의 MAC을 검증한다. 이러한 MAC을 생성할 때 메시지 작성자와 수신자만이 알고있는 비밀키를 포함하여 MAC을 만든다면 메시지 무결성뿐만 아니라 실제 메시지 작성자를 확인할 수 있다. 암호 모듈을 구현하기 위해서는 OpenSSL 프로그램을 사용할 수 있는데, OpenSSL은 다양한 암호화 함수를 사용할 수 있는 명령어 입력 도구인 동시에 DH, RSA 키 파라미터의 생성과 메시지 다이제스트 계산 등의 기능을 제공한다.[2]

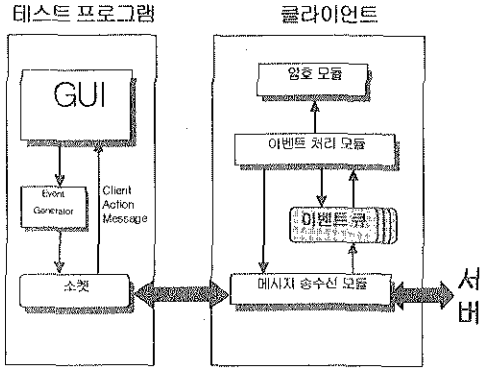

그림 3. WTLS 클라이언트 구조도

3.4 테스트 프로그램

WTLS 클라이언트의 기능을 실험하기 위한 프로그램은 소켓을 사용하여 WTLS 클라이언트와 통신한다. 사용자 인터페이스는 크게 클라이언트에 이벤트를 발생시키는 메뉴와 명령창 그리고 메시지창으로 나누어진다. 사용자가 발생시킨 이벤트는 소켓을 통해 WTLS 클라이언트로 전달되고 클라이언트는 발생한 이벤트를 이벤트 큐에 삽입하여 이벤트 처리 모듈에서 처리하도록 한다. 이벤트를 처리하는 과정에서 클라이언트는 새로 생성된 이벤트나 WTLS 클라이언트 세션 상태의 변화 등을 테스트 프로그램으로 보고하여 사용자가 알 수 있도록 한다. 사용자가 이벤트를 생성하기 위해 사용한 메뉴항목들은 명령창에 출력되고, WTLS 클라이언트로부터 수신된 메시지들은 메시지창에 출력되어, 사용자의 명령에 따른 WTLS 클라이언트의 동작 흐름을 쉽게 파악할 수 있도록 한다.

4. 결론 및 향후과제

본 논문은 WAP을 이용한 두 통신 애플리케이션 간의 기밀성, 데이터 무결성 그리고 인증을 제공하는 WTLS 클라이언트 구조 설계에 관한 것이다. 본 논문에서 설계된 WTLS 클라이언트는 발생하는 각각의 이벤트를 현재의 클라이언트 보안 세션 상태를 기준으로 처리하는데 중점을 두고 있다.

WTLS는 SSL/TLS를 기반으로 하여 무선 환경에 최적화된 채널 보안 프로토콜이지만, WAP 게이트웨이에서 WTLS로 통신하는 무선구간과 SSL로 통신하는 유선구간사이의 프로토콜 변환 시에 메시지의 원문이 그대로 노출된다.[4] 이러한 문제점을 해결하기 위해 향후에는 WAP 클라이언트와 서버의 종단간 보안을 제공할 수 있는 최적의 방안을 선택하여 시스템을 구성해야 한다.

참고문헌

[1] WAP WTLS ver. 18-Feb-2000, http://www.wapforum.org
[2] http://www.openssl.org
[3] 김종훈, "SSL기반의 Apache Web 인증 서버 설계 및 구현", 경북대학교 석사논문, 1999
[4] 양진욱, 김순자, "WAP 마이크로브라우저와 서버의 종단간 보안 매커니즘 설계", 한국정보처리학회 추계 학술발표논문집, 제7권 제2호, 2000
[5] 이정엽, "SSL기반의 신용정보보호 전자지불 프로토콜 제안", 경북대학교 석사논문, 2000
[6] Stephen A. Thomas, "SSL and TLS Essentials: Securing the Web", WILEY, 2000
[7] Eric Rescorla, "SSL and TLS - Designing and Building Secure Systems", Addison-Wesley, 2000