WTLS 프로토콜의 하드웨어 모듈

김진봉, 김동규
부산대학교 컴퓨터공학과

Hardware Modules for the WTLS Protocol

Jin Bong Kim, Dong Kyue Kim
Dept. of Computer Engineering, Pusan National University

요 약

오늘날 정보통신 기술의 급속한 발전과 함께 인터넷은 그 용도를 점차 넓혀가고 있으며 기존의 유선망에서 무선망을 이용한 무선데이터 서비스가 점차 활성화되고 있다. 현재 무선망에서 사용되는 프로토콜은 WAP(Wireless Application Protocol)이 가장 널리 사용되고 있다. 그런데 이 무선망에서 사용되는 무선 단말기의 낮은 데이터 전송속도 및 연산능력에 의하여 기존의 소프트웨어로 구현된 WAP으로는 기존 무선단말기의 성능 향상에 한계가 있다. 특히 WAP의 한 계층인 WTLS(Wireless Transport Layer Security)는 보안의 기능을 담당하는 계층으로 복잡한 연산과정을 수행한다. 본 논문에서는 WTLS 프로토콜 중에서 실질적인 보안 서비스를 제공하는 레코드 프로토콜(record protocol)) 을 하드웨어 모듈로 구현함으로써 기존의 시스템과 비교하고 앞으로의 방안을 제안하고자 한다.

1. 서론

최근 무선 통신 시장은 매우 빠르게 성장하고 있으며, 이를 기반으로 하는 새로운 서비스들이 제공되고 있다. 그러나 현실적으로 무선단말기는 낮은 데이터 전송속도, 낮은 연결 신뢰성, 제한된 대기시간 및 핸드오프와 같은 취약성을 가지며, 화면과 메모리, 배터리 및 연산능력에서도 유선의 단말기보다 매우 불리한 환경을 가지고 있다.

이러한 무선통신 환경의 응용 개발을 위해 Ericsson, Nokia, Motorola, Unwired Planet 주도로 결성된 WAP 포럼에서 WAP 프로토콜을 제정하였다. 현재 WAP 포럼에는 약 200여 회원사가 참가하고 있으며, Ericsson, Motorola, Nokia, 삼성전자, LGIC등의 무선 하드웨어 및 소프트웨어 제조업체와 Unwired Planet, Qualcomm 등 infrastructure 제공업체, SK-telecom, LGIC, Bell South등 무선 사업자가 참여하고 있다.

WAP(Wireless Application Protocol)은 Transport, Security, Transaction, Session 및 Application 계층 프로토콜의 집합을 정의하고 있으며 계층화된 프로토콜로 구성되어져 있다.

특히, 이러한 무선통신 환경에서 데이터 서비스가 원활하게 제공되기 위해서는 인터넷에서와 마찬가지로 정보보호 문제가 반드시 선결되어야 한다. 그러나 현재 무선 단말기의 낮은 CPU속도, 적은 메모리, 전력소비의 제한 및 무선 네트워크의 낮은 대여폭, 많은 대기지연시간, 비안정적 연결상태 등의 제약이 있어 무선 단말기에서 암호화/복호화를 수행하는 것이 무선 단말기에 상당히 많은 부담을 준다.

본 논문에서는 WAP 계층 중 보안에 관련된 프로토콜을 하드웨어 모듈로 구현함으로써 기존의 무선 단말기의 제한된 리소스의 부담을 줄이고 더욱더 빠른 연산처리를 통하여 안정적인 보안 서비스를 제공하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 WAP의 구조에 대해서 설명하고 3장에서는 WAP 계층 중 보안 기능을 담당하는 WTLS에 대하여 구체적으로 기술한다. 4장에서는 WTLS 프로토콜의 하드웨어 모듈 설계에 대한 설명을 하고, 5장에서는 WTLS 프로토콜에 대한 실험결과 및 성능을 기술한다. 6장에서는

*본 연구는 한국과학재단 목적기초연구 (R01-2002-000-00589-0)지원으로 수행되었음.

결론을 맺는다.

2. WAP(Wireless Application Protocol)

WAP은 WAP 포럼에서 제정하고 있는 무선 환경에서 동작하는 표준 프로토콜로서 무선인터넷 위한 프로토콜로써 계층화된 구조를 가지고 있으며, 각각의 계층은 서로 독립적으로 개발할 수 있다. 즉, 새로운 bearer에 접목하는 것이 가능하고 상위 계층이 다른 계층을 바꾸지 않고 새로운 전송 프로토콜을 사용하는 것이 가능하다. WAP 프로토콜의 구조는 [그림1]과 같다.

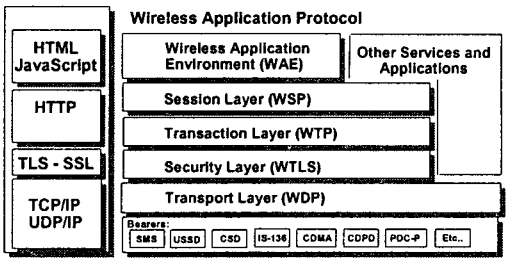

그림1. WAP 프로토콜의 구조

각 계층을 간략히 설명하면, 먼저 WDP는 모든 무선 bearer 네트워크에 WAP 프로토콜을 적용시킬 수 있도록 근본적인 전송서비스를 위한 인터페이스를 제공한다. WTLS는 유선 인터넷상의 세션보안을 위한 TLS(Transport Layer Security)를 기반으로 보안기능을 제공한다. WTP는 WDP와 WTLS계층의 상위 계층에 존재하며, 클래스 등급에 따른 데이터 품질 서비스를 제공하여 주로 트랜잭션을 유지하는 기능을 제공하는 계층이다. 그리고, WSP는 클라이언트와 서버간의 세션설정을 위해 동작하는 프로토콜이다. 마지막으로 WAE는 모바일 장비에 의해 요구된 인터넷의 WAP 컨텐츠를 조회하는 기능을 제공한다.

3. WTLS

3.1장에서는 WTLS 구조와 기능, WTLS의 내부 프로토콜을 간략히 설명하고 3.2장에서는 WTLS 내부 프로토콜 중 Record 프로토콜을 상세히 설명한다.

3.1 WTLS의 구조와 기능

WTLS는 [그림2]와 같이 WDP와 WTP사이에 위치하고 있으며, 무선 환경에 적합한 보안프로토콜이다. 즉, 유선인터넷 구간보다 느린 속도와 낮은 전송율, 무선 단말기의 부족한 자원등을 충분히 고려하였고, 여러 가지 암호 메커니즘을 선택적으로 사용할 수 있으며, 하위 계층인 WDP와 잘 연동되도록 구성되어져있다. WTLS 프로토콜은 SSL과 마찬가지로 응용계층의 필요에 따라 선택적으로 사용된다.

WTLS는 클라이언트와 서버가 인증을 선택적으로 교환하여, 통신하고 있는 주체가 적법한 사용자인지를 확인할 수 있는 사용자 인증(authentication)서비스, 인증코드를 이용한 무결성(integrity) 서비스, 그리고 대칭키 암호 알고리즘을 사용하여 기밀성 서비스를 제공한다. 그리고 WTLS 프로토콜은 근본적으로 종단간 통신 채널을 암호화하는 프로토콜이므로 서명기법을 이용한 부인 방지 서비스를 제공하지는 않는다. 기밀성 서비스를 제공하기 위하여 데이터를 대칭키 암호 알고리즘을 사용하여 암호화하므로 WTLS 프로토콜은 고속 동작이 가능하고, 공격자가 중간에 메시지를 변조하는 경우 인증 코드를 이용하여 메시지가 중간에 수정되었음을 수신자가 쉽게 확인할 수 있다.

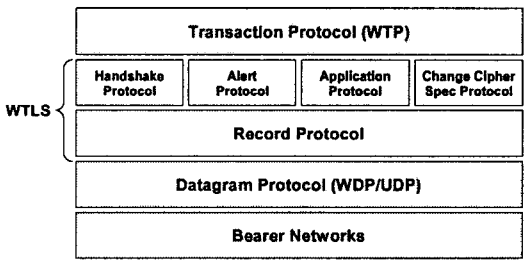

그림2. WTLS 프로토콜의 구조

WTLS 프로토콜의 구조는 [그림2]와 같다. WTLS 프로토콜은 handshake 프로토콜, change cipher spec 프로토콜, alert 프로토콜, 그리고 record 프로토콜로 구성된다. record 프로토콜은 실질적인 보안서비스가 제공되는 프로토콜이고, handshake 프로토콜은 record 프로토콜을 위한 보안 파라미터(parameter)를 서버와 클라이언트간에 공유하는 프로토콜이고, change cipher spec 프로토콜은 이전 handshake 프로토콜로 공유된 대기 연결 상태를 현재의 연결 상태로 변경하기 위한 프로토콜이다. 마지막으로 alert 프로토콜은 상대 WTLS 계층에 경고(alert)를 알리기 위한 프로토콜이다.

3.2 Record 프로토콜

Record 프로토콜은 WTLS 프로토콜 중에서 실질적인 보안 서비스를 제공하는 프로토콜이다. [그림3]은 record 프로토콜의 동작과정을 나타내었다.

WTLS의 동작과정은 다음과 같다. 먼저 record 프로토콜은 전송될 메시지를 수신하여, 선택적으로 데이터를 압축하고, 선택적으로 MAC을 적용시키며 그 결과를 다시 암호화하여 상대 WTLS 계층으로 전달한다. 그리고 암호화된 데이터를 수신한 record 프로토콜은 암호문을 복호하고, 수신된 MAC를 검증하며, 압축된 압축문을 푼다음, 이를 상위 계층으로 넘겨준다. 이때, 데이터 압축, 해쉬 연산, 암호화 등에 사용되는 매개변수들은 handshake 프로토콜의 handshake 과정에서 결정된다. 그리고, 여기서 사용되는 암호 알고리즘은 데이터 암호화시에는 DES, RC5 등이 사용되며 해쉬 알고리즘에는 SHA-1와 MD5를 사용한다.

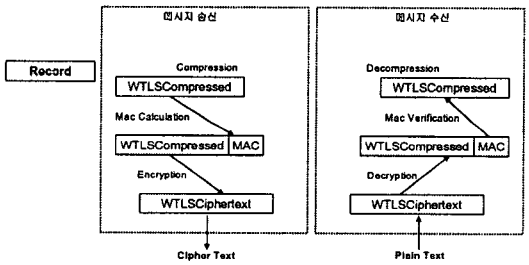

그림3. Record Protocol의 동작과정

4. Record 프로토콜 하드웨어 모듈의 설계

본 논문에서 제시하는 record 프로토콜의 전체적인 하드웨어 모듈의 구성은 [그림4]와 같다.

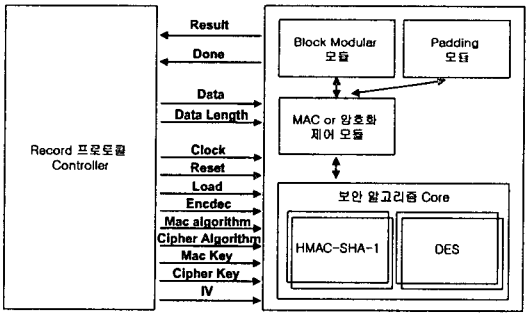

그림4. Record 프로토콜의 하드웨어 모듈의 구성

Record 프로토콜의 모듈을 설명하면 다음과 같다. Record 프로토콜 Controller는 record 프로토콜의 전반적인 제어를 담당하며, handshake 프로토콜에 의하여 사전에 설정되어진 암호 알고리즘 및 각종 파라미터(parameter) 값들을 MAC or 암호화 제어모듈로 전송시켜준다. 그리고 MAC or 암호화 제어모듈은 Record 프로토콜 Controller에서 받은 정보에 정의되어 있는 보안 알고리즘에 따라 보안 알고리즘을 선택하고, MAC (또는 암호화)을 적용시키고자 하는 데이터를 Block Modular 모듈에서 기존에 선택된 MAC(또는 암호화)알고리즘의 입력 데이터의 사이즈만큼 블록단위로 나누고, 순차적으로 각 블록을 해당 알고리즘으로 MAC(또는 암호화)를 적용시킨다. 이때 나누어진 데이터의 제일 마지막 블록의 사이즈가 해당 보안알고리즘의 입력데이터의 사이즈보다 작을 경우에는 Padding 모듈에서 MAC(또는 암호화)알고리즘의 입력 사이즈 크기에 맞게 패딩을 해주고, 연산을 실행한다. 끝으로 모든 실행이 완료되면, MAC or 암호화 제어모듈은 Record 프로토콜 Controller모듈에게 Done 신호를 날려주고 그 결과값을 돌려준다.

5. 실험 결과 및 성능 분석

5.1장에서는 기존의 구현된 WTLS 프로토콜의 전체 속도와 record 프로토콜의 수행속도를 분석 비교하고 5.2장에서는 소프트웨어로 구현된 record 프로토콜과 하드웨어 모듈로 구현된 record 프로토콜의 성능을 비교 설명한다.

5.1 WTLS 프로토콜의 성능 분석

WTLS 프로토콜에는 앞장에서 설명했듯이 handshake 프로토콜, change cipher spec 프로토콜, alert 프로토콜, record 프로토콜로 크게 4가지로 구성되어져있다.

[그림5]는 WTLS 프로토콜이 처리할 입력 데이터의 크기에 따라 WTLS의 전체 수행 속도와 WTLS의 내부 프로토콜인 record 프로토콜의 수행 속도를 각각 측정하여 비교한 결과이다. [그림5]를 보면 WTLS 내부 프로토콜 중 실질적으로 복잡한 연산을 통하여 보안기능을 담당하는 record 프로토콜의 속도가 입력 데이터의 크기가 증가함에 따라 WTLS의 전체 수행 속도에서 차지하는 비중이 점차 높아지는 것을 알 수 있다. 뿐만 아니라 record 프로토콜은 WTLS의 다른 프로토콜와는 달리 데이터 송수신시에 반복적으로 수행되기 때문에 WTLS의 전체 수행 속도를 더욱더 가중시키는 작용을 한다.

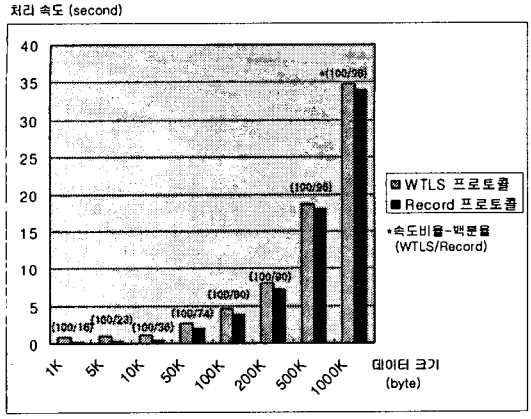

그림5. WTLS프로토콜과 record프로토콜의 속도비교

그러므로, 이러한 복잡한 연산을 수행하는 record 프로토콜을 하드웨어 모듈로 구현함으로써 기존의 시스템 보다 더 빠른 연산처리를 하고 이것을 통하여 전체적인 WTLS의 수행속도를 증가시킬 수 있다.

5.2 Record 프로토콜의 H/W와 S/W의 성능 비교

[표1]은 실제로 소프트웨어로 구현되어진 record 프로토콜과 하드웨어 모듈로 구현되어진 record 프로토콜에 동일한 데이터(정보)를 입력시켜서 각각의 수행 속도를 평가, 비교하여 나타난 결과이다.

구현 방법 (구현 환경)	S/W 구현 (평균) CPU-2 66GHz RAM - 1GB	H/W 구현 Xilinx VERTEXE XCV600E-HQ240 chip FGPA Board	속도 증가율
HMAC 단계	25.036 (ms)	0.02518973 (ms)	약 500~1000 배
암호화 단계	0.0949167 (ms)	0.004745174 (ms)	약 20~30 배
Record 프로토콜 전 단계	97.244 (ms)	-	-

표1. Record 프로토콜의 성능 비교

위의 [표1]을 보면 하드웨어 모듈로 구현된 record 프로토콜이 상대적으로 소프트웨어로 구현된 record 프로토콜보다 수행속도가 더 우월함을 짐작할 수 있다.

6. 결론

본 논문에서는 기존의 소프트웨어 구현되어져 있는 WTLS의 내부 프로토콜 중 record 프로토콜을 하드웨어 모듈로 구현하였다. 즉, WTLS에서 복잡한 연산을 수행하며 WTLS의 다른 내부 프로토콜에 비하여 상대적으로 많이 사용되는 record 프로토콜을 하드웨어 모듈로 구현함으로써 WTLS의 전체적인 시스템의 수행속도를 증가시키고자 하였다.

앞으로의 연구는 현재 구현하고 있는 record 프로토콜 모듈을 최적화 시켜 WTLS의 수행속도를 더욱더 향상시킬 것이며 WTLS에서 record 프로토콜 이외에 다른 프로토콜을 하드웨어 모듈로 구현하여 기존의 시스템과 그 수행속도를 비교하여 WTLS의 어떤 부분을 하드웨어 모듈로 구현하는 것이 좋은지 또는 어떤 부분을 소프트웨어로 구현하는 것이 더 좋은지 비교분석하여 최적화된 WTLS 프로토콜 시스템을 개발할 예정이다.

[참고문헌]

[1] WAP Architectures Specification, WAP Forum, 2002-11-6(WAP 2.0)
URL: http://www.wapforum.org

[2] WTLS : Wireless Transport Layer Security Specification V5, WAP Forum, 06-Apr-2001

[3] HMAC : HMAC: Keyed- Hashing for Message Authentication", Krawczyk, H., Bellare, M., and Canetti, R., RFC 2104, February 1997.
URL: ftp://ftp.isi.edu/in-notes/rfc2104.txt

[4] SHA : Secure Hash Standard, NIST FIPS PUB 180-1, National Institute of Standardsand Technology, U.S Department of Commence, Draft, May 1994

[5] DES : "American National Standard for Information Systems -Data Link Encryption", ANSI X3.106, American National Standards Institute, 1983.

[6] MD5 : "The MD5 Message Digest Algorithm", Rivest, R., RFC 1321, April 1992.
URL: ftp://ftp.isi.edu/in-notes/rfc1321.txt

[7] RC5 : "The RC5, RC5- CBC, RC5- CBC- Pad, and RC5 - CTS Algorithms", Baldwin, R. and Rivest R., RFC 2040, October 1996.
URL: ftp://ftp.isi.edu/in-notes/rfc2040.txt

[8] "인터넷 정보 보안", 한국 정보보호진흥원

[9] TLS : "The TLS Protocol", Dierks, T. and Allen, C., January 1999.
URL: ftp://ftp.isi.edu/in-notes/rfc2246.txt

[10] WAP Gateway Project : Kannel
http://www.gnu.org/directory/network/kannel.html