「정보학연구」 제10권 제3호 2007년 9월(pp.77-92)
The Journal of Information Technology, Vol.10, No.3

모바일 클라이언트의 데이터 무결성 보장을 위한 보안모듈에 관한 연구

A Study on the Security Module for Data Integrity of Mobile Client

주 해 종 (Joo, Hae Jong)* · 홍 봉 화 (Hong, Bong Hwa)**

목 차

Abstract

This study aims to suggest an implementation methodology of security module for data integrity of mobile internet terminal. This is based on the WTLS(Wileless Transport Layer Security) of WAP Protocol. This security module is expected to achieve central role in conversion of wireless internet environment and emphasis of encryption technology and safe and calculable wireless communication environment construction.

Key words: Power amplifier, distortion, electro thermal, memory effect, pre-distorter

* 인덕대학 산학협력단
** 경희사이버대학교 정보통신학과

Ⅰ. 서론

국내외적으로 이동통신이 급격한 발전을 이루고, 개인의 정보통신에 대한 수요가 증가하면서 사업자, 장비 및 단말기 제조회사에서 경쟁적으로 첨단 기능이나 부가 서비스를 개발하고 있다. 국내에서는 SMS(Short Message Service)가 이미 범용 서비스로 정착되었고, 현재 한정적으로 사용하고 있는 무선 데이터 서비스가 곧 일반화될 것으로 기대된다.

이와 같이 무선 데이터서비스에 대한 중요성이 강조되고 있는 가운데, 여러 가지 다양한 무선 인터넷 솔루션이 개발되고 있다. 이와 같은 무선 인터넷 솔루션은 크게 2가지 부류로 구분할 수 있다[2,3,4]. 첫째는 기존 유선 인터넷에서의 프로토콜인 HTTP에 기반해 무선 데이터 서비스를 제공하는 경우이며, 다른 하나는 무선 네트워크 환경에 적합한 새로운 프로토콜을 개발해 무선 데이터 서비스를 제공하는 방법이다. 현재 HTTP에 기반한 방식은 마이크로소프트사의 ME와 NTT 도코모의 i-mode 서비스가 대표적이며, 프로토콜을 새로 개발하는 방식은 WAP 포럼에서 개발을 주도하고 있는 WAP(Wireless Application Protocol)이 대표적이다.

WAP 포럼에서는 TCP/IP와는 별도의 무선 환경에 적합한 프로토콜을 정의하는 작업을 진행중인데, WTLS(Wireless Trnasport Layer Security)가 바로 그것이다. WTLS는 SSL과 TLS에 디반해 작성되었다. WTLS는 통신을 하는 두 응용 프로그램사이에 안전한 채널을 형성해 통신 내용의 보안을 보장하는 방법이다[2,3,4,6].

WTLS는 DES, IDEA 같은 관용 암호방식을 사용해 두 애플리케이션간의 기밀성 서비스를 제공하며, RSA와 같은 공개키 암호방식과 X.509 인증서를 사용해 클라이언트와 서버의 상호 인증을 제공하고, 내부적으로 누군가 데이터 전송을 방해할 수 없도록 하거나 재전송 공격에 이용할 수 없도록 데이터의 무결성을 제공하는 장점을 가진다[2,3].

본 논문에서는 무선인터넷단말기의 안전한 데이터 보안을 위해 제 2장에서 WAP 프로토콜의 보안 프로토콜인 WTLS에 대하여 논하고 제 3장에서 무선인터넷단말기의 데이터 무결성 보장을 위해 WAP 프로토콜의 WTLS 보안 프로토콜에 추가하기 위해 고안된 보안모듈에 대하여 논하고 제 4장에서 결과를 제시하였다.

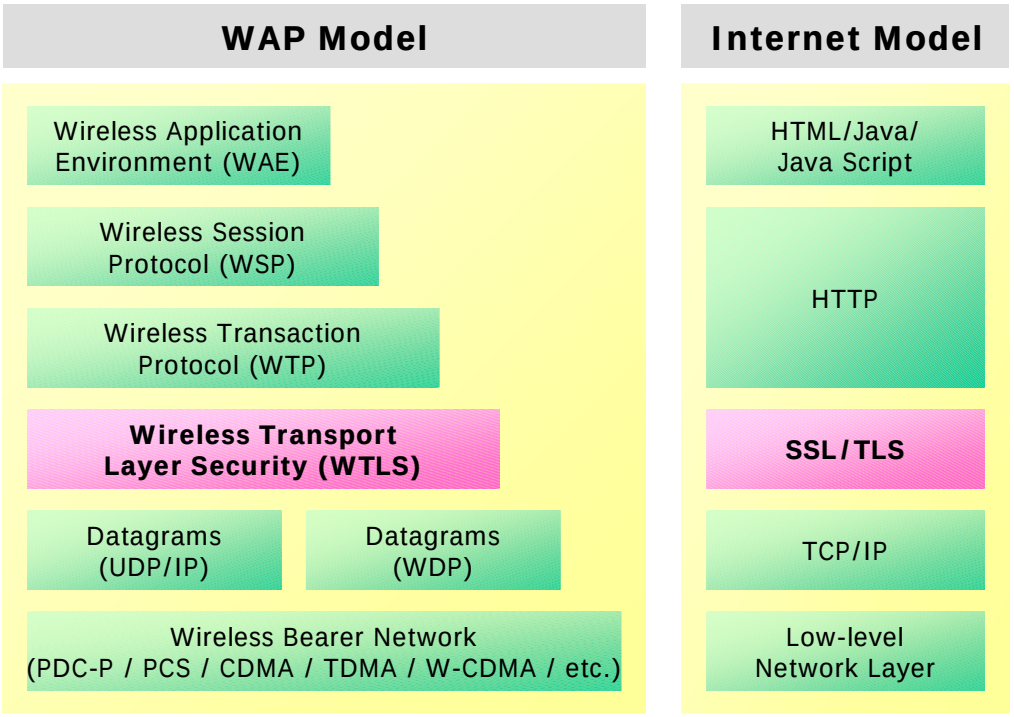
Ⅱ. WTLS(Wireless Transport Layer Security)

1. 개요

기존의 유선 인터넷에서는 많은 양의 데이터를 빠른 시간에 전송할 수 있지만, 무선 환경에서는 이러한 서비스가 어렵다. 특히 이미지나 동영상과 같은 경우에는 상당히 많은 양의 데이터 처리가 필요한데, 무선 환경에서는 이러한 대량의 데이터 위주의 서비스를 제공하는데 무리

가 있다[1,3]. WAP은 이와 같이 기존의 인터넷 프로토콜을 사용할 경우에 발생하는 문제들을 해결하고, 기존 인터넷 중심의 데이터 서비스를 무선 환경에서 효율적으로 처리하기 위해 제안된 프로토콜이다. 국제적으로 WAP 정의를 위해 표준화 기구인 WAP 포럼이 설립되어 표준화 작업이 진행되어, 1997년에 Nokia, Motorola, Ericsson, Phone.com 등 4개의 단말기 업체를 중심으로 구성되었으며, 현재 약 200여개의 업체가 참여중이다.

WAP 포럼에서는 그림1과 같이 기존 TCP/IP와는 별도의 무선 환경에 적합한 프로토콜을 정의하는 작업을 진행중인데, 이 가운데 무선 환경의 보안 프로토콜이 WTLS이다. WTLS의 역할은 인터넷에서의 SSL/TLS와 동일하며, SSL/TLS를 기반으로 해서 설계되었다.



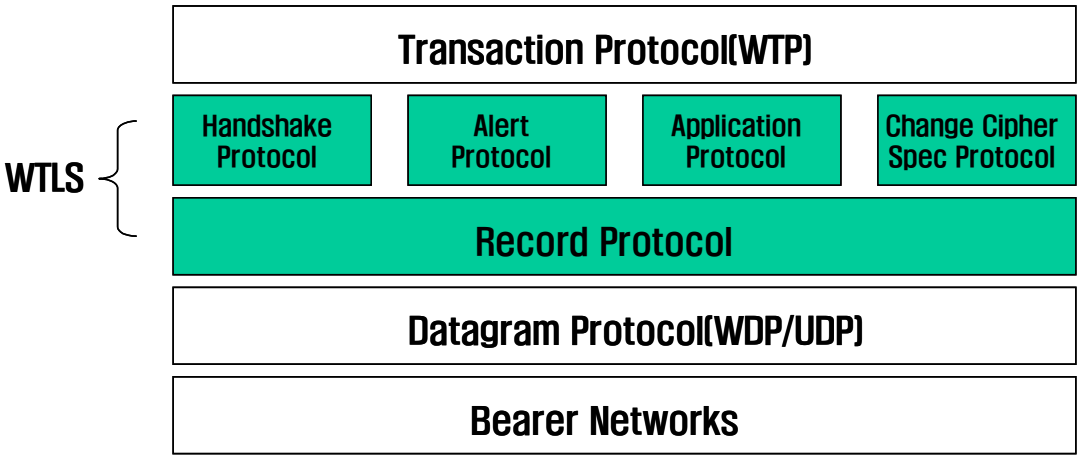
<그림 1> WAP 프로토콜 스택

WTLS와 SSL/TLS와의 가장 큰 차이점은 SSL/TLS가 connection-oriented 프로토콜인 TCP 상위에서 동작하는데 비해, WTLS는 datagram 프로토콜인 UDP 또는 WDP 상위에서 동작한다는 점이다. 따라서 SSL/TLS가 재전송, 중복된 데이터 처리 등의 기능을 수행하지 않아도 되는 반면에 WTLS에서는 통신의 신뢰성을 보장할 수 있는 방법이 추가되어야 한다. WTLS는 SSL/TLS와 마찬가지로 기밀성, 데이터 무결성, 사용자 인증 등의 정보보호 서비스

를 제공한다[3].

2. 구조

WTLS의 구조는 그림2와 같은데, 이 가운데 핸드쉐이크 프로토콜(Handshake Protocol), 얼라트 프로토콜(Alert Protocol), 사이퍼 스펙 프로토콜(Cipher Spec Protocol)은 WTLS의 동작에 대한 관리를 위해 사용되며, 실질적인 보안서비스는 레코드 프로토콜(Record Protocol)에서 제공된다[2,3]. 무선인터넷단말기와 서버가 WTLS를 통해 연결할 경우, 먼저 핸드쉐이크 프로토콜을 수행해 한 세션동안 보안 서비스 제공에 사용되는 세션키, 암호 알로리즘, 인증서 등과 같은 암호 매개변수를 서로 공유하게 된다. 여기서 생성된 세션 정보는 레코드 프로토콜에서 보안 서비스를 제공하는 이용된다. WTLS 얼라트 프로토콜은 핸드쉐이크 프로토콜, 체인지 사이퍼 스펙 프로토콜, 레코드 프로토콜이 수행중일 때 발생하는 모든 오류 메시지를 처리하는 프로토콜이다.



<그림 2> WAP의 보안프로토콜 WTLS

1) Handshake Protocol

핸드쉐이크 프로토콜은 Handshake Protocol, Alert Protocol, Change Cipher Spec Protocol 등 3개의 하위 프로토콜로 구성되며, Record Protocol에서 사용될 보안 파라미터 결정, 클라이언트와 서버 인증, 오류 처리 등에 이용된다.

Alert Protocol에서는 오류 메시지가 정의되며, 클라이언트나 서버에서 오류가 발생했을 때 오류 메시지를 보내서 오류가 발생할 사실을 상대방에게 알리는 역할을 한다. Change Cipher

Spec Protocol은 하나의 메시지로 구성되며, 이 메시지가 전송된 이후의 메시지는 새로운 보안 파라미터에 의해서 암호화되어 전송됨을 알리는 역할을 한다.

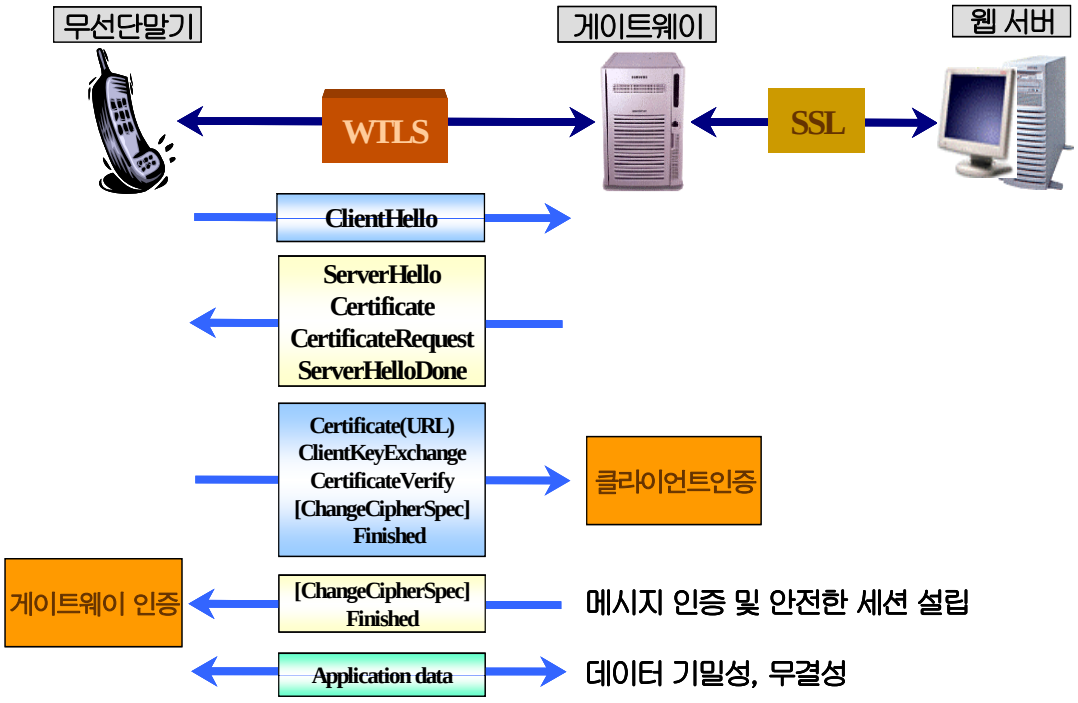
WTLS를 이용해 무선인터넷단말기와 서버를 연결할 경우, 한 세션동안 보안 서비스 제공에 사용되는 세션키, 암호 알고리즘, 인증서 등과 같은 암호 매개변수를 서로 결정하여야 하는데, 이는 핸드쉐이크 프로토콜을 통해서 이루어진다. 이를 위해서 교환되는 정보는 표 1과 같다.

Compression method, cipher spec, master secret와 같은 암호 매개변수를 결정하는 핸드쉐이크 과정은 크게 full handshake, abbreviated handshake, optimized full handshake 등 3가지로 구분할 수 있다. 이 가운데 full handshake와 abbreviated handshake는 SSL/TLS에서도 사용되는 방법으로 full handshake는 새로운 세션을 시작할 때 사용되는 것이며, abbreviated handshake는 기존의 세션을 재개해서 다시 이용할 경우에 사용된다. full handshake의 동작 과정은 그림3과 같다.

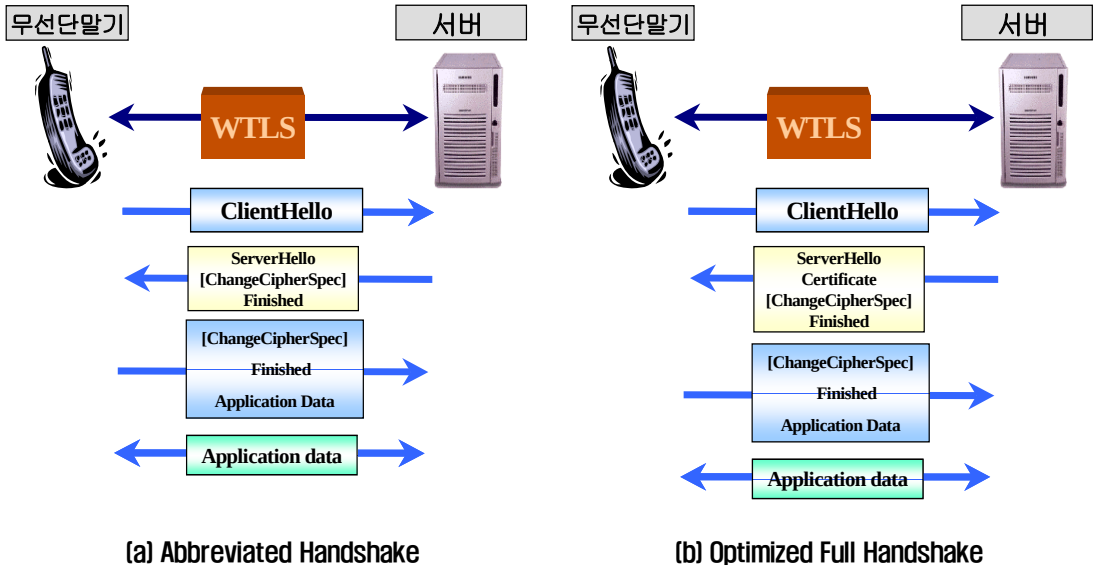
<표 1> Handshake 과정에서 교환되는 정보

정보	설명
Session Identifier	서버가 세션을 식별하는데 사용되는 임의의 수
Protocol Version	WTLS 프로토콜 버전
Peer Certificate	서버 및 클라이언트 인증서
Compression Method	데이터 암호화에 앞서 사용되는 압축 방법
Cipher Spec	사용되는 관용 암호 알고리즘 및 MAC 알고리즘
Master Secret	클라이언트와 서버에 의해서 공유되는 20바이트 비밀 정보
Sequence Number Mode	현재 세션에 사용되는 일련번호 사용방법(off, implicit, explicit)
Key Refresh	보안 서비스 제공에 사용되는 정보(암호키, MAC 정보, IV) 등의 교체 주기
Is Resumable	현재 세션이 새로운 세션을 시작하는데 사용될 수 있는지 여부를 나타내는 표시자

무선인터넷단말기는 ClientHello 메시지를 전송함으로써 서버에게 연결을 요청한다. 이때 무선인터넷단말기가 사용할 수 있는 관용 알고리즘의 목록, 공개키 알고리즘의 목록, 압축 방법들의 목록들을 전송한다. ClientHello 메시지를 수신한 서버는 이에 대한 응답으로 ServerHello 메시지를 전송한다. 이때 서버는 무선인터넷단말기가 전송한 암호 매개변수들의 목록에서 세션에서 사용할 것을 결정하며, 서버 인증을 위해서 서버의 인증서를 전송하고 클라이언트 인증을 위해서 클라이언트 인증서를 요청한다. 이 과정을 통해서 서로를 인증하고 필요한 암호 매개변수들을 생성한 무선인터넷단말기와 서버는 Finished 메시지를 보내서 Handshake 과정을 종료하고, 실제 데이터를 교환한다.



<그림 3> WTLS Full Handshake



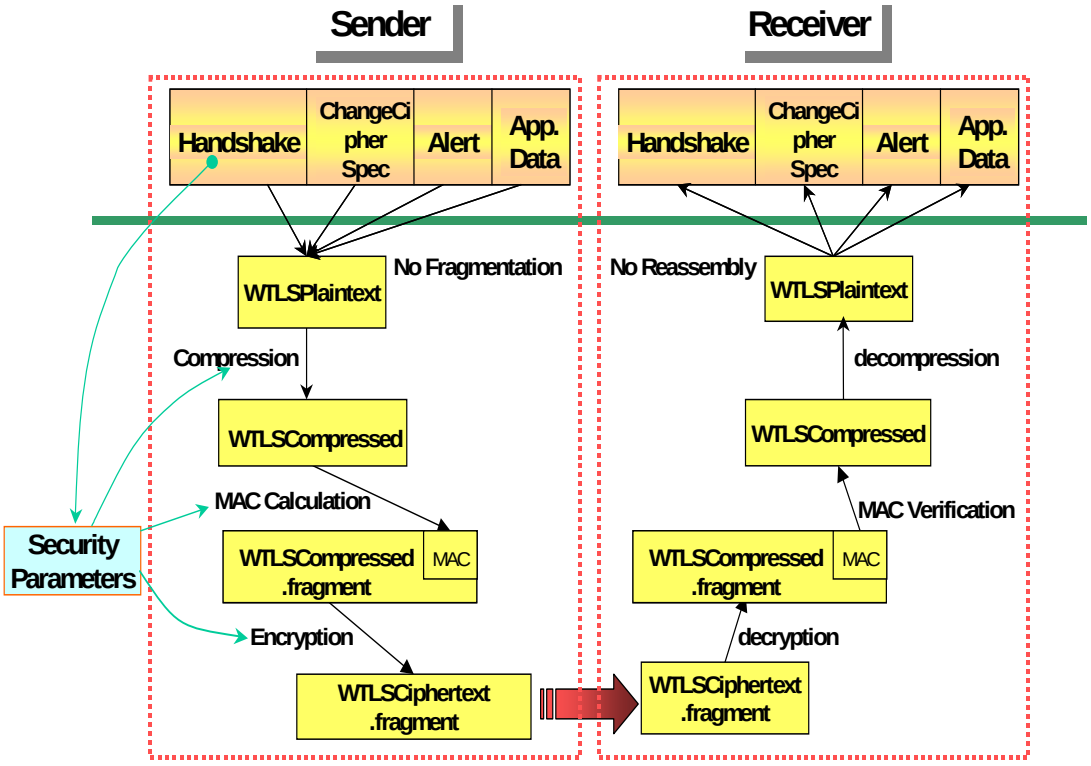
<그림 4> WTLS Abbreviated Handshake 및 Optimized Full Handshake

Abbreviated Handshake에서는 그림4(a)와 같이 이전 세션 정보를 이용하여 세션을 시작하기 때문에 인증서 교환과 같은 서버와 무선인터넷단말기 인증을 위한 정보는 교환되지 않으며, 이전 세션에서 사용한 암호 매개변수로부터 새로운 세션에서 사용될 매개변수들을 생성한다.

마지막으로 Optimized full handshake는 WTLS에서 새롭게 추가된 것으로 그림4(b)와 같이 서버는 무선인터넷단말기 인증을 위해 인증서를 요청하지 않고, 서버내에 보관하거나 저장소를 통해 제공되는 무선인터넷단말기 인증서를 통해 인증을 수행한다.

2) Record Protocol

레코드 프로토콜은 데이터를 압축하고, 해쉬 및 암호화를 수행하여 전송하거나, 수신한 데이터를 복호화 및 검사하는 역할을 한다. 이때 데이터 압축, 해쉬 계산, 암호화 등에 사용되는 매개변수들은 핸드쉐이크 과정에서 결정된다. 즉 데이터를 전송할 때는 그림5와 같은 과정을 통해 전송할 데이터를 생성하며, 데이터를 수신할 경우에는 이 반대 과정을 거친다. 다만 데이터 단편화는 SSL/TLS에서는 수행하는 기능이지만, WTLS에서는 수행하지 않는다. 이는 WTLS 하위에 위치하는 UDP 혹은 WDP 계층에서 데이터에 대한 단편화가 이미 이루어지기 때문이



<그림 5> Record Protocol의 동작

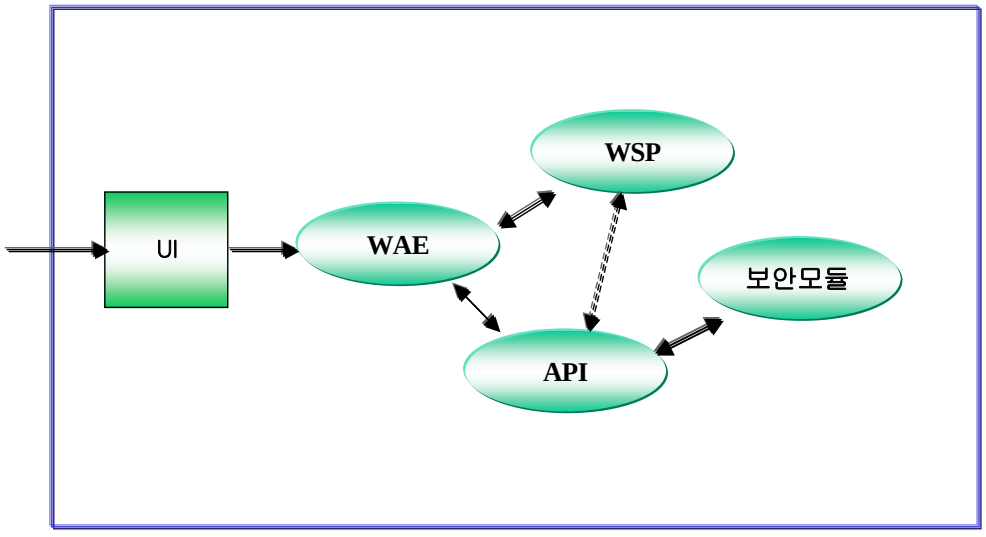
다. 또한 데이터 압축 역시 현제는 지원하지 않는데, 이는 향후 개발이 계속되면서 지원할 예정이다.

이때 전송되는 데이터는 SDU(Service Data Unit)라 불리는 단위로 전송된다. 예를 들어 Abbreviated Handshake의 ServiceHello 단계에서 전송되는 ServerHello와 Finished, Change Cipher Spec 메시지는 하나의 SDU로 합쳐져서 전송된다. 이는 WTLS가 datagram 프로토콜의 상위에서 동작함에 따라 신뢰성을 부여하기 위함이다.

Ⅲ. 무선인터넷단말기의 데이터 무결성 보장을 위한 보안 모듈

1. 보안 블록 다이어그램

현재 인터넷에서 전자상거래의 주류는 사이버 쇼핑몰을 통한 거래이다. 즉 구매자는 인터넷을 통해 사이버 쇼핑몰을 방문하여 물품 검색 등을 통해 구매하고자 하는 물품을 선택하며, 물품 구매후 지불은 신용카드를 이용하는 방법이 가장 많이 사용되고 있다. 그러나 이와 같은 형태는 네트워크, 디스플레이 장치, CPU 및 메모리의 한계 등으로 인해 무선통신 환경에서는 적합하지 않은 실정이다. 따라서 무선 인터넷 환경에서는 비교적 간단한 형태의 전자상거래 솔루션이 주류를 이룰 것으로 예상되며, 이를 위한 데이터 보안 솔루션이 제공되어야 할 것이다.



<그림 6> 무선인터넷단말기의 보안 블록 다이어그램

본 논문에서 제시하는 무선인터넷단말기의 데이터 무결성 보장을 위한 종단간(End-to-End) 보안모듈의 블록 다이어그램은 그림6과 같다. API를 call하는 위치는 무선인터넷단말기 개발자의 편의에 따라 WAE 또는 WSP에 위치할 수 있으며, WAP 서버에서 무선인터넷단말기의 종단간 보안을 처리하기 위해 "Application/vnd.wap.ter_wap_sec_wml"과 같은 콘텐츠 타입을 정의해야 한다. 또한, 무선인터넷단말기의 보안 모듈을 실행시킬때에는 WSP layer에서는 Connection Oriented로 세션을 연결시켜야 한다.

이때 무선인터넷단말기에서 사용하는 관용 보안 알고리즘은 Message Authentication Code (SHA-1: Secure Hash Algorithm), Symmetric Cryptography Algorithm(SEED), Asymmetric Cryptography Algorithm(RSA : Rivest Shamir Adeleman), Random-DES 등이다.

2. 보안 모듈의 구현

1) 레코드 정의

무선인터넷단말기의 데이터 무결성 보장을 위한 보안 모듈에서 사용되는 레코드를 표2와 같이 정의한다. 레코드는 레코드 헤더(Record header)와 단편(Fragment)으로 구성되며, 레코드 헤더는 레코드 타입(1 바이트)과 순서(2 바이트), 그리고 레코드 크기(2 바이트)를 포함한다. 또한, 단편은 n 바이트의 크기를 가지며, 데이터와 해쉬 값을 포함하고 있다.

<표 2> 레코드 정의

<table>
<tr><th rowspan="2">의미</th><th colspan="3">Record Header</th><th rowspan="2">Fragment</th></tr>
<tr><th>Record Type</th><th>Sequence #</th><th>Record Size</th></tr>
<tr><td>길이</td><td>1 바이트</td><td>2 바이트</td><td>2 바이트</td><td>n 바이트</td></tr>
<tr><td rowspan="2">값</td><td>*상위 4비트: 암호화여부
0010 : 암호화된 레코드
0000 : 비암호화 레코드</td><td rowspan="2">연속된 레코드를 전송할 경우 사용 (레코드순번)</td><td rowspan="2">Fragment의 크기(n)</td><td>*일반 record
: data + hash 값</td></tr>
<tr><td>*하위 4비트 : Record Type
0001 : alert
0010 : handshake
0011 : application data</td><td>*암호화된 record
: 암호화 data
+ hash 값</td></tr>
</table>

① Client Hello Record

본 레코드는 무선인터넷단말기에서 서버로 송신하기 위한 레코드로서 난수발생기에 의해 "Client Random#1"을 발생시키고, 표 3과 같은 레코드를 서버로 송신한다.

<표 3> "Client Hello" Record

Rec type	Seq#	Size	Type	Data Size	Ver.	Mode	Client Random#	User_id	Hash값
1바이트	2	2	1	2	1	1	16	n	m
00000010	0		1(Client Hello)		1	1/2			

② Server Hello Record

본 레코드는 서버에서 무선인터넷단말기로 송신하기 위한 레코드로서 서버에서 버전을 체크하고 "Client Random#와 Mode정보"를 서버에 저장하여 난수 발생기에 의해 "Server Random#"를 발생시킨 뒤, 표 4와 같은 레코드를 무선인터넷단말기로 송신한다.

<표 4> "Server Hello" Record

Rec type	Seq#	Size	Type	Data Size	Ver.	Random#	Public Key	Hash값
1바이트	2	2	1	2	1	16	128	m
00000010	0		2(Server Hello)		1			

③ Xfer Key Info Record

본 레코드는 패스워드와 무선인터넷단말기의 난수 생성 정보를 암호화하여 서버로 전송하는 레코드로서 무선인터넷단말기의 버전 검사, "Server Random#2"를 단말기에 저장, "Client Random#2" 생성, password와 Client Random#2 해쉬 처리, 그리고 서버의 Public Key로 정보를 암호화한 후, 표 5와 같은 레코드를 서버로 송신한다.

<표 5> "Xfer Key Info:Ep(Password+Client Random#2)" Record

Rec type	Seq#	Size	Type	Data Size	Data(Ep(...)	Hash값
1바이트	2	2	1	2	n	m
00000010	1		3(Xfer Key Info)			

④ Verification Record

본 레코드는 미리 정의된 데이터를 암호화하고 검사하여 서버로 전송하는 레코드로서 서버의 Private Key로 복호화를 수행하고 "Client Random#1, Client Random#2+Password, Server Random#"로 Key Block을 생성하여 미리 정의된 데이터를 암호화한 후, 표 6과 같은 레코드를 서버로 송신한다.

<표 6> "Verification Ep(Predefined data)" Record

Rec type	Seq#	Size	암호화된 Rocord Body			
			Type	Data Size	Data(Ep(...)	Hash값
1바이트	2	2	1	2	n	m
00000010	2		4(Verification)			

⑤ Required Sign Key

본 레코드는 사용자의 Sign Key를 서버에 가지고 있지 않을 경우 서버에서 무선인터넷단말기로 Sign Key를 요청하는 레코드로서 표 7과 같은 레코드를 단말기로 송신한다.

<표 7> "Required Sign Key" Record

Rec type	Seq#	Size	Type	Data Size	Data	MAC값
1바이트	2	2	1	2	1	m
00000010	2		5(Required Sign Key)	1	1	

⑥ Exchange Sign Key Record

본 레코드는 무선인터넷단말기가 “Required Sign Key” 레코드를 서버로부터 수신한 뒤, 단말기에서 Sign Key를 생성하여 서버로 송신하는 레코드로서 표 8과 같은 레코드를 서버로 송신한다.

<표 8> "Exchange Sign Key Eks(Sign Key)" Record

Rec type	Seq#	Size	Eks {Record Body}				MAC값
			Type	Data Size	단말기#	Public Key	
1바이트	2	2	1	2	11	n	m
00100010	2		6(Verification)				

⑦ Finished Record

본 레코드는 서버에서 암호화된 데이터를 복호화하여 미리 정의된 데이터와 비교한 뒤, 두 데이터가 같으면 무선인터넷단말기로 송신하는 레코드로서 표 9와 같은 레코드를 무선인터넷단말기로 송신한다.

<표 9> "Finished" Record

Rec type	Seq#	Size	Type	Data Size	Data	Hash값
1바이트	2	2	1	2	1	m
00000010	0		14(Finished)			

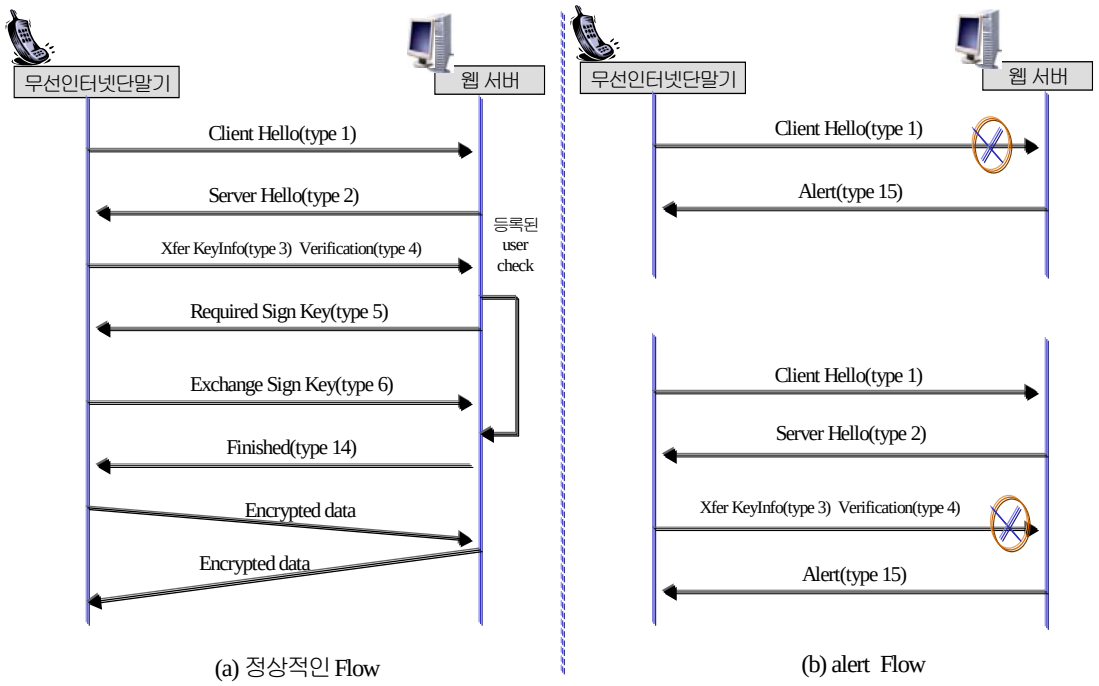
⑧ Alert Record

본 레코드는 무선인터넷단말기에서 서버로 보낸 데이터인 User ID와 Password가 일치하지 않는 경우 사용하는 레코드로서, 표 10과 같은 레코드를 무선인터넷단말기로 송신한다.

<표 10> "Alert" Record

Rec type	Seq#	Size	Type	Data Size	Data	Hash값
1바이트	2	2	1	2	1	m
00000010	0		15(alert)			

※ Data : User ID Mismatched 0x01
Password Mismatched 0x02



<그림 7> 보안 모듈의 처리 흐름도

위에서 정의한 레코드를 이용하여 무선인터넷단말기의 데이터 무결성 보장을 위한 보안모듈의 처리 흐름을 그림7과 같이 도식화할 수 있으며, 그림7(a)는 정상적인 처리 흐름을 나타내고 그림 7(b)는 비 정상적인 처리 흐름을 나타내고 있다.

2) API 함수

① WAP Gateway에서 Bypass시키기 위해 콘텐츠타입을 미리 정의된 보안 타입으로 변경하는 함수

```
void devGetContentType(U8*Ctype, U8*Ctype2)
{
memcpy(Ctype1, "sec_wml", 7);
memcpy(Ctype2, "sec_wmls", 8);
}
```

* input : First ContentType, Second ContentType

② 종단간 보안 핸드쉐이크 시작 함수

```
void devSecurityStart(U8 mode, UserID, U8 UserIDLength, U8*Pwd,
 U8 PwdLength, U8*Returndata, U8*rdataLength,
 U8*Rstatus, U8*ErrorType)
{
secClientStart(U8 mode, UserID, U8 UserIDLength, U8*Pwd,
 U8 PwdLength, U8*Returndata, U8*rdataLength,
 U8*Rstatus, U8*ErrorType) ;
} ;
mode : mode/sign mode
UserID : USER_ID
UserIDLength : USER_ID length
Pwd : User Password
PwdLength : Password Length
Returndata : record to be sent to server
rdataLength : Return data size
Rstatus : Record type
ErrorType : the state of received record
```

③ 서버로부터 데이터를 받았을 때 이를 파싱(parsing)시키는 함수

```
void devParseRecord(U8*data, UNIT16 dtataLength, U8*Returndata,
 UNIT16*rdataLength, U8*Rstatus, U8*ErrorType)
{
secParseRecord(U8*data, UNIT16 dtataLength, U8*Returndata,
 UNIT16*rdataLength, U8*Rstatus, U8*ErrorType) ;
} ;
data : received data from server
dataLength : received data length
Returndata : record to be sent to server
rdataLength : Return data
Rstatus : Record type
Error Type : the state of received record
```

④ 무선인터넷단말기에서 서버로 데이터를 보내는 함수

```
void devSendRecord(U8*data, UNIT16 dtataLength, U8*Returndata,
 UNIT16*rdataLength, U8*Rstatus, U8*ErrorType)
{
secSendRecord(U8*data, UNIT16 dtataLength, U8*Returndata,
 UNIT16*rdataLength, U8*Rstatus, U8*ErrorType) ;
} ;
data : sending data from server
dataLength : data length
Returndata : record to be sent to server
rdataLength : Return data
Rstatus : Record type
Error Type : the state of received record
```

⑤ 종단간 보안 mode 종료 함수

```
void devSecurityClose()
{
secSecurityClose(void) ;
}
```

⑥ Error 처리 함수

```
if(ErrorType & Fatal_Error)
{
if((ErrorType &0xF0) == sec_USERIDMISMATCHED ||
(ErrorType &0xF0) == sec_HASHCODENOTCORRECT)
{
display error message("잘못된 ID입니다. 다시 입력하세요")
display both UserID input and Password input card
}
else if((ErrorType &0xF0) == sec_PWDMISMATCHED)
{
 display error message("패스워드가 올바르지 않습니다")
 display both UserID input and Password input card
}
}
```

Ⅳ. 결론

이동통신 사용자는 국내에서만 이미 2천만명을 넘어서고 있으며, 전세계적으로는 2007년까지 10억 이상의 사용자를 확보할 것으로 예상되고 있다. 이러한 가운데 무선통신을 통한 데이터 서비스의 제공이 필수적인 요소로 받아들여지고 있으며, 이에 따라 무선통신과 기존 인터넷과의 결합은 향후 정보통신 산업의 중추적인 역할을 할 것으로 기대된다. 이에 대비하여 이미 Mobile IP, IMT2000, WAP 등 무선 인터넷을 겨냥한 다양한 메카니즘이 연구·개발되고 있다.

www의 등장과 함께 혁명적인 전환을 맞았던 인터넷은 이제 무선 인터넷이라는 또 다른 전환기를 맞고 있다. 이와 함께 무선통신 기술에 안전성 및 신뢰성을 보장할 수 있는 정보보호 서비스에 대한 비중 또한 점차 높아지고 있다.

본 연구는 이러한 무선 인터넷 환경의 전환과 암호화 기술의 강조, 그리고 안전하고 신뢰할 수 있는 통신 환경 구축에 있어서 핵심적인 역할을 수행할 WTLS 기반의 무선인터넷단말기의 데이터 무결성 보장을 위한 보안 모듈 구현에 대한 방안을 제시하였다.

앞으로 무선 인터넷의 계속적인 발전은 유·무선 통합 환경의 등장으로 이어질 것으로 예상된다. 유·무선 통합 환경에서는 기존의 인터넷이나 무선 통신에서의 정보보호 서비스와는 다른 새로운 패러다임이 적용될 것이며, 지금부터 이에 대한 연구 및 개발이 시작되어야 할 것이다.

참고문헌

1. 주해종 · 박영배, “모바일 데이터베이스 환경의 신뢰성 보장 질의처리 시스템 설계”, 「한국정보처리학회지」, 제12-D권, 2005. 8.
2. 주해종, “Mobile EC-통신프로토콜과 보안문제”, 한국상무학회, 2000. 12.
3. 김춘길, “전자상거래의 개념과 발전방향”, 「정보과학회지」, 제16권 제5호, 1998. 5.
4. 김현욱, 김영결, 조원득, 김연규, 이성범, “Wireless Application Protocol 서비스 개요”, *SK Telecom Technical Journal,* 제6권 제4호, 1999. 10.
5. WAP Forum, "Wireless Transport Layer Security", 1999. 11.
6. Dierks T., Allen C., "The TLS Protocol", IETF RFC2246, 1999. 1.
7. http://www.wapforum.org
8. http://www.zionwap.net