情 報 保 護 學 會 誌
第 14 券 第 4 號, 2004. 8

SSL/TLS, WTLS의 현재와 미래

이 진 우*, 남 정 현*, 김 승 주**, 원 동 호***

요 약

최근, 인터넷 사용이 증가함에 따라 해킹이나 바이러스, 개인 정보 유출 등의 발생 빈도도 비례하여 증가하고 있는 실정이다. 이에 이러한 역기능들을 방어·통제하기 위한 웹 보안의 중요성도 함께 증대되고 있다. SSL/TLS와 WTLS는 전송 레벨 웹 보안 프로토콜로서 현재 유·무선 네트워크 환경에서 사용자의 인증, 데이터의 무결성 및 기밀성을 보장하기 위하여 보안 시스템 개발 시 그 활용도가 매우 크다. 본 고에서는 SSL/TLS와 WTLS의 변천사를 살펴보고, 각 프로토콜의 주요 동작과정 및 안전성을 분석한다. 또한, 사용자의 익명성을 제공하기 위하여 SSL/TLS을 확장한 SPSL(Secure and Private Socket Layer) 프로토콜을 분석하며, 향후 SSL/TLS와 WTLS의 발전 방향에 대하여 모색한다.

Ⅰ. 서 론

최근, 유·무선 네트워크의 발전으로 인해 다수의 사용자들은 가정이나 회사에서 PC는 물론 핸드폰, PDA 등과 같은 무선 단말기를 사용하여 언제 어디서나 쉽게 인터넷 서비스를 이용할 수 있게 되었다. 인터넷은 정보의 자유로운 상호 접근 및 공유를 가능하게 하였을 뿐만 아니라 금융, 주식 거래 등 다양한 웹 서비스 및 전자상거래와 같은 많은 e-비지니스를 창출하였다.

그러나, 개방성, 공유성의 특징을 가지고 있는 인터넷은 이러한 순기능만 가지고 있는 것은 아니다. 인터넷 사용이 증가함에 따라 해킹이나 바이러스, 개인 정보 유출 등의 발생 빈도도 비례하여 증가하고 있는 실정이다. 이러한 역기능들을 방어·통제하기 위한 웹 보안의 중요성도 함께 증대되고 있다.

본 고에서는 전송 레벨에서의 웹 보안 프로토콜인 SSL/TLS와 WTLS의 변천사를 살펴보며, 각 프로토콜의 구조 및 동작과정 그리고 안전성을 진단한다. 또한, 현재 SSL/TLS와 WTLS의 활용도와 향후 발전 방향에 대하여 제시한다.

본 고의 구성은 다음과 같다. 제 2장에서는 전송 레벨에서의 웹 보안 프로토콜인 SSL/TLS와 WTLS에 대하여 분석하며 동작과정 및 안전성에 대하여 다룬다. 또한, 사용자의 익명성을 제공하기 위하여 SSL/TLS을 확장한 SPSL 프로토콜을 분석한다. 그리고 제 3장에서는 SSL/TLS, WTLS의 활용 분야를 살펴보고, 발전 방향을 제시하며 마지막으로 4장에서 결론을 맺는다.

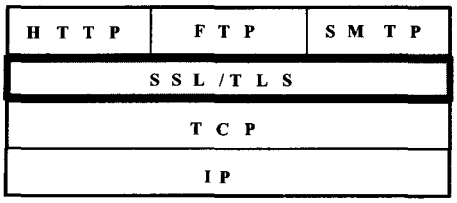
Ⅱ. 전송 레벨 웹 보안 프로토콜

본 장에서는 웹 보안 서비스의 전송 레벨 보안 프로토콜인 SSL/TLS와 WAP(Wireless Application Protocol) 포럼에서 제정한 WTLS의 구조와 동작과정을 비교하며 안전성을 진단한다. 또한, 사용자의 익명성을 제공하는 SPSL 프로토콜에 대해 분석한다.

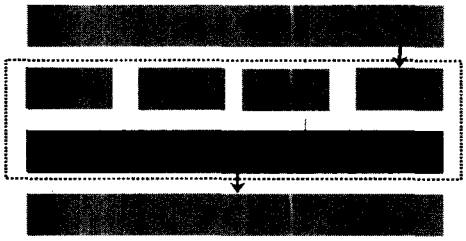
1. SSL/TLS 개요

SSL(Secure Sockets Layer)은 1994년 Netscape사에 의해서 Netscape 웹 브라우저를 통한 안전한 통신을 위하여 처음으로 제안되었으며, 1996년 Internet Engineering Task Force(IETF)에서 SSL v3.0을 제안하였다. 이후에도 SSL v3.0은 지속적으로 수정·보안되었으며 1999년에는 TLS(Transport Layer Security)로 명칭이 바뀌어 RFC 2240(TLS v1.0)으로 표준화되었다[1].

* 성균관대학교 정보통신공학부 정보통신보호연구실({jwlee, jhnam}@dosan.skku.ac.kr)
** 성균관대학교 정보통신공학부 조교수(skim@ece.skku.ac.kr)
*** 성균관대학교 정보통신공학부 정교수(dhwon@ece.skku.ac.kr)



〔그림 1〕 SSL/TLS의 위치



〔그림 2〕 SSL/TLS의 구조

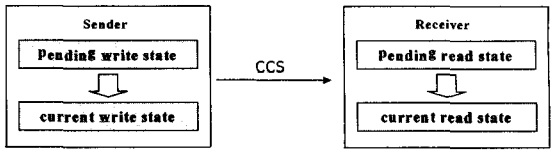
SSL/TLS은 〔그림 1〕과 같이 전송계층과 응용계층 사이에 위치함으로서 다양한 응용계층의 프로그램들과 쉽게 보안설정을 할 수 있으며, SSL/TLS는 〔그림 2〕와 같이 레코드 레이어와 Handshake, Change cipher spec, Alert, Application data 프로토콜로 이루어져 있다.

1.1 SSL/TLS 동작

SSL/TLS는 크게 세션(session) 상태와 커넥션(connection) 상태로 나누어서 이루어진다. 실제로 통신은 하나의 세션 안에 여러 개의 커넥션이 포함되는 형태로 이루어진다〔2〕〔3〕.

상태는 예비 상태(pending state)와 현재 상태(current state)로 나뉘어 지며 예비 상태는 서버와 클라이언트가 통신을 수행하면서 설정된 알고리즘과 키를 임시로 저장하는 상태이며, 현재 상태는 레코드 레이어에서 실제 데이터가 처리될 때의 상태를 의미한다. 각각의 상태는 다시 read 상태와 write 상태로 이루어지며 read 상태는 상대방이 전송한 데이터를 읽기위한 상태이며, write 상태는 상대방에게 데이터를 전송할 때 사용되는 상태를 말한다.

Handshake 프로토콜에서는 알고리즘을 포함한 보안 파라미터(security parameters)가 설정되고, 이것은 레코드 레이어로 전달되어 키 블록을 생성하고 예비 상태로 저장된다. 저장된 예비 상태는 Change cipher spec 프로토콜에 의해 현재 상태로 옮겨진다. 이렇게 옮겨지면 예비 상태는 NULL 상태로 다시 초기화 된다.



〔그림 3〕 상태의 변화

이때, Change cipher spec 메시지를 보내면 예비 write 상태가 현재 write 상태로 바뀌고, 다시 Change cihper spec 메시지를 받게 되면 예비 read 상태가 현재 read 상태로 변경된다. 위의 〔그림 3〕은 예비 상태와 현재 상태로 변화하는 과정을 나타낸 것이다.

Full Handshake에 의해 세션이 한번 생성하게 되면 이후 통신은 Abbreviated Handshake 프로토콜에 의해 하나의 세션 상태를 공유하면서 커넥션 상태만을 재생성하여 이루어질 수 있다.

〔표 1〕은 SSL/TLS 세션 상태의 파라미터를 나타내며 하나의 세션이 종료될 때까지 유지된다.

〔표 1〕 SSL/TLS 세션 상태의 파라미터

session identifier	임의의 바이트로서 세션 구분 담당
peer certificate	x.509 v3 형식의 인증서
compression method	압축 알고리즘 명시
cipher spec	암호통신에 사용할 알고리즘 및 키 길이 명시
master secret	서버와 클라이언트가 공유하는 48바이트의 비밀 값
is resumable	새로운 커넥션을 생성할 수 있는지를 나타내는 플래그

〔표 2〕 SSL/TLS 커넥션 상태의 파라미터

server and client random	Hello 메시지에서 교환되는 서버와 클라이언트의 32 바이트의 비밀 랜덤 값
server write MAC secret	서버가 MAC 생성 시 사용할 비밀 값
client write MAC secret	클라이언트가 MAC 생성 시 사용할 비밀 값
server write key	서버의 암호화 키
client write key	클라이언트의 암호화 키
initialization vectors	블록 암호 알고리즘 사용 시 초기벡터(IV) 값
sequence numbers	Change ciper spec 메시지를 보내거나 받으면 0으로 초기화되고 메시지마다 1씩 증가

커넥션 상태는 [표 2]와 같은 파라미터를 포함하고 있다. 각 비밀키는 세션 상태의 master secret과 서버와 클라이언트의 랜덤 수로 생성되므로 각 커넥션 상태는 다르게 설정된다.

1.1.1 Handshake 프로토콜

레코드 계층 상위에서 클라이언트는 Handshake 프로토콜을 이용하여 보안 파라미터를 서버에게 요청하게 된다. 이때, 서버는 클라이언트의 요청에 응답하여 통신에 필요한 파라미터들을 설정한다. Handshake에 의하여 설정된 파라미터들은 Change ciper spec 프로토콜로 사용할 수 있도록 활성화되고, 데이터들은 Application data 프로토콜을 통하여 SSL/TLS에 의해 보호되어 전송된다. 통신과정에서 발생한 오류들은 Alert 프로토콜이 처리하게 된다.

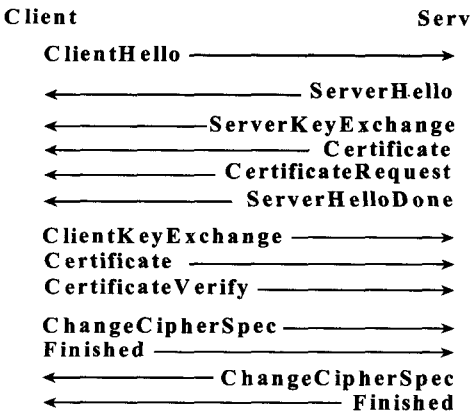
Handshake 프로토콜 수행 과정에서 클라이언트와 서버를 상호인증하며 암호 알고리즘, 암호키, MAC 알고리즘 등의 세션 상태를 유지할 수 있는 요소들을 설정하게 된다. Handshake 프로토콜은 크게 Full Handshake 프로토콜과 Abbreviated Handshake 프로토콜로 나뉘며 Full Handshake 프로토콜의 동작과정은 [그림 4]와 같이 진행되며 다음과 같이 요약될 수 있다.

① 클라이언트는 서버에게 ClientHello 메시지를 전송하고 서버의 응답을 기다린다. 서버는 바로 ServerHello 메시지를 보내지 않으면 에러가 발생하면서 커넥션이 중단된다.

② 클라이언트와 서버는 Hello 메시지에 프로토콜 버전, 세션 ID, cipher suite, 압축 방법들을 명시하고 각각 난수를 생성하여 교환한다. 또한 서버의 Certificate, SecureKeyExchange, 클라이언트의 Certificate, ClientKeyExchange를 통하여 키 생성에 필요한 pre-master secret과 master secret을 서로 공유하게 된다. 서버는 클라이언트에게 인증서를 요청할 수 있으며 ServerHelloDone 메시지를 전송하여 클라이언트의 요청을 기다린다.

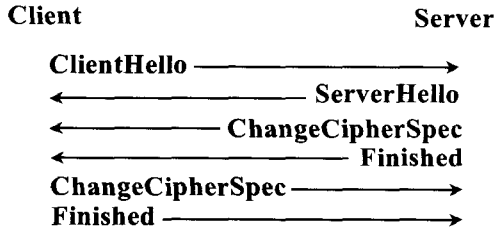
③ 서버가 클라이언트의 인증서를 요청하였을 경우에 클라이언트는 반드시 Certificate를 전송하여야만 한다. ClientKeyExchange 메시지의 Hello에서 선택된 키 교환 알고리즘의 pre-master secret에 대한 정보를 서버에게 전송 한다. 또한 클라이언트는 설정된 파라미터들을 Change cipher spec 프로토콜로 수행 한 후 Finished 메시지를 전송하여 마치게 된다.

④ 클라이언트에게 모든 응답을 받았을 경우, 서버는 클라이언트와 같은 방법으로 Change cipher spec 프로토콜을 수행한 후 Finished 메시지를 클라이언트에게 전송하고 모든 Handshake 과정을 마치게 된다.



[그림 4] Full Handshake 동작 과정

클라이언트가 새로운 세션을 원할 경우에는 empty 세션 ID를 전송하며, 이미 생성된 세션 상태를 이용하여 새로운 커넥션 상태를 생성할 경우에는 ClientHello 메시지에 재 새용 하기를 원하는 세션 ID를 서버에게 전송하면 된다. 하지만, 클라이언트가 전송한 세션 ID를 서버가 찾을 수 없다면 서버는 에러 메시지와 함께 empty 세션 ID를 전송하게 되고, 일치하는 세션 ID가 있을 경우에는 클라이언트와 서버는 Abbreviated Handshake 프로토콜을 이용하여 Change cipher spec을 교환하게 된다. [그림 5]는 Abbreviated Handshake 프로토콜의 동작과정을 나타내고 있다. Abbreviated Handshake프로토콜을 이용하여 세션 상태를 재 사용할 경우에는 기존의 세션 상태는 그대로 유지하며 서버와 클라이언트가 교환한 Change chiper spec을 이용하여 예비 상태를 유지하게 된다.



[그림 5] Abbreviated Handshake 동작과정

1.1.2 Change ciper spec 프로토콜

이 프로토콜은 Handshake 프로토콜 과정에서 설정된 예비 상태를 현재 상태로 바꾸는 프로토콜이다.

1.1.3 Alert 프로토콜

Alert 프로토콜은 모든 통신 과정에서 발생하는 에러 메시지를 전달한다. 메시지에는 warning level 과 fatal level로 구분되며, 만약 fatal level의 메시지를 전송받은 경우에는 커넥션이 중단되고 세션이 닫히게 된다.

〔표 3〕, 〔표 4〕은 SSL/TLS의 에러 메시지의 종류와 할당값, 그리고 에러 레벨을 명시하고 있다.

〔표 3〕 SSL Alert의 메시지 종류

에러 메시지 종류	할당값
close_notify(warning level)	0
bad_record_mac(fatal level)	20
handshake_failure(fatal level)	40
bad_certificate	42
certificate_revoked	44
certificate_unknown	46
unexpected_message(fatal level)	10
decompression_certificate (fatal level)	30
no_certificate	41
unsupported_expired	43
certificate_expired	45
illegal_parameter(fatal level)	47

〔표 4〕 TLS Alert의 메시지 종류

에러 메시지 종류	할당값
decryption_failed(fatal level)	21
unknown_ca(fatal level)	48
decode_error(fatal level)	50
export_restriction(fatal level)	60
insufficient_security (fatal level)	71
user_canceled(warning level)	90
record_overflow(fatal level)	22
access_denied(fatal level)	49
decrypt_error	51
protocol_version(fatal level)	70
internal_error(fatal level)	80
no_renegotiation(warning level)	100

1.2 SSL/TLS 차이점

TLS는 SSL을 보안하여 다음과 같은 여러 사항에서 수정 · 보안이 이루어졌다.

1.2.1 키 생성

SSL은 해쉬 함수를 직접 이용하나, TLS에서는 의사 난수 함수(PRF)를 이용하여 생성한다.

〔SSL〕

master_secret =
MD5(pre_master_secret + SHA('A' + pre_master_secret + ClientHello.random + ServerHello.random)) +
MD5(pre_master_secret + SHA('BB' + pre_master_secret + ClientHello.random + ServerHello.random)) +
MD5(pre_master_secret + SHA('CCC' + pre_master_secret + ClientHello.random + ServerHello.random));

〔TLS〕

PRF(secret, label, seed) = P_MD5(S1, label + seed) XOR P_SHA-1(S2, label + seed) 정의

master_secret =
PRF(pre_master_secret, 'master secret', ClientHello.random+ServerHello.random)〔0 .. 47〕

1.2.2 Alert 메시지 추가

SSL에서는 위의 〔표 3〕과 같이 총 12개의 에러 메시지가 존재하나 TLS에서는 SSL에서의 no-certificate 메시지를 제외한 12가지가 추가되었다..

1.2.3 CertificateRequest

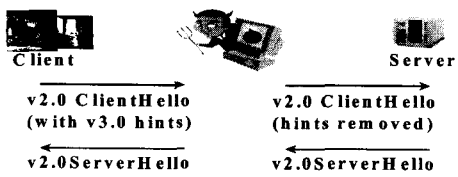
TLS에서는 rsa_ephemeral_dh, dss_emhemeral_dh, fortezza 방식이 제외되었다.

1.2.4 Cipher suite

TLS에서는 Fortezza 알고리즘이 제외되었다.

1.3 SSL/TLS 취약성 분석

SSL/TLS에서 발생하는 공격으로는 Ciphersuite



〔그림 6〕 Version rollback 공격

rollback 공격, Dropping Change cipher spec 공격, Key exchange algorithm rollback 공격, Version rollback 공격이 있다[1][4].

1.3.1 Ciphersuite rollback 공격

Ciphersuite rollback 공격은 클라이언트와 서버간의 Hello 메시지가 평문으로 전송됨으로서 공격자는 Hello 메시지를 가로채어 클라이언트가 전송하는 Ciphersuite 메시지의 목록을 수정하여 공격자가 원하는 알고리즘으로 대체하여 공격하는 방식이다. 이는 Handshake 프로토콜 과정에서 Finished 메시지에 Handshake 프로토콜의 모든 메시지에 대한 해쉬값을 생성하여 메시지 인증을 함으로써 공격을 방지할 수 있다.

1.3.2 Version rollback 공격

공격자는 〔그림 6〕과 같은 방법으로 보안성이 좋은 v3.0을 지원함에도 불구하고 v2.0으로 통신이 이루어지도록 방해할 수 있다.

1.3.3 Dropping Change cipher spec 공격

Dropping Change cipher spec 공격은 공격자가 Change cipher Spec을 가로채어 cipher suite를 갱신할 수 없도록 방해하는 공격이다. 이러한 공격으로부터 Change cipher Spec을 보호하기 위하여 TLS에서는 구현 시 Finished 메시지를 검증하기 전에 Change cipher Spec 메시지를 체크하여 예비 read 상태를 반드시 현재 read 상태로 바꾸도록 명시하였다.

1.3.4 Key exchange algorithm rollback 공격

Key exchange algorithm rollback 공격은 Hello 메시지가 평문으로 전송되는 것을 이용하여 공격자는 Hello 메시지를 가로채어 클라이언트가 전송하는 Ciphersuite 메시지에 자신이 임의로 선택한 키 교환 알고리즘으로 대체하여 서버에게 전송한다. 또한, 공격자는 서버가 클라이언트로 전송하는 것을 똑같은 방법으로 가로채어 자신이 임의의 키 교환 알고리즘으로 대체시킨다. 이럴 경우 클라이언트와 서버는 서로 다른 키 교환 알고리즘을 사용하게 되는 것이다. 이를 방지하기 위하여 파라미터들을 서명하여 전송함으로서 공격자는 임의대로 키 교환 알고리즘을 대체 할 수 없도록 한다.

1.3.5 RSA의 키의 크기

대부분의 인증서는 RSA 방식을 사용한다. 512 비트의 합성수가 인수분해 되었기 때문에 1024비트를 사용하는 것이 바람직하다[5].

2. WTLS의 개요

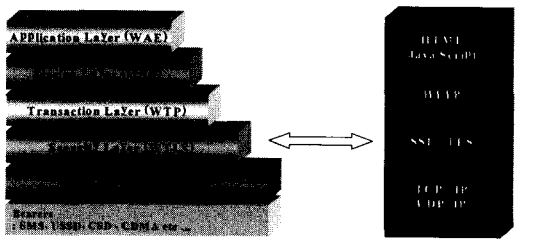
Wireless Application Protocol(WAP)은 1997년 6월 Phone.com, Nokia, Motorola, Ericsson사가 주축이 되어 만든 WAP 포럼에서 무선 환경과 인터넷을 연계하기 위하여 제안된 프로토콜이다. 무선 환경은 유선 환경과 달리 동작 특성을 고려하여 패킷의 중복이나 손실 등을 처리할 수 있어야 한다[5].

WAP은 무선 단말기의 저전력 CPU, 저 메모리 등, 계산 능력이 제한된 환경에 적합한 프로토콜을 정의하고 있으며, 전송(WDP), 보안(WTLS), 트랜잭션(WTP), 세션(WSP), 응용(WAE) 등 5개의 계층으로 구성되어 있다. 그 중 보안 레이어에서 사용되는 보안 프로토콜이 WTLS(Wireless Transport Layer Security)이다.

〔그림 7〕은 WAP의 계층별 구조를 나타내고 있다. WTLS는 TCP/IP의 보안에 사용되는 SSL/TLS를 기반으로 무선 환경에 적합하도록 개선한 프로토콜이다.

2.1 WTLS의 Handshake 프로토콜

WTLS의 Handshake 프로토콜은 SSL/ TLS와 같은 방식으로 이루어지나, 각각의 메시지들은 약간의 차이가 있다. SSL/TLS의 ClientHello에는 클라이언트의 버전, 키 교환 알고리즘, 비밀키 알고리즘과 MAC 알고리즘을 하나의 cipher suite로 정의하고 있지만, WTLS



< Wireless APPlication Protocol > < Internet Protocol >

〔그림 7〕 WAP의 계층별 구조

에서는 키 교환 알고리즘은 key exchange suite로 정의하고 cipher suite에 비밀키 알고리즘과 MAC 알고리즘을 각각 분리하여 정의한다.

또한, WTLS의 Hello 메시지에 SSL/TLS에서는 없었던 key refresh 와 sequence number mode를 정의하였다. key refresh에서는 키 블록 교환 주기를 나타내고 sequence number mode에서는 WTLS가 datagram transport를 지원하기 위한 데이터의 sequence number를 붙이는 방법을 나타낸다.

2.2 WTLS의 Alert 프로토콜

WTLS의 Alert 프로토콜에서는 SSL/TLS와 달리 critical level의 에러 메시지가 추가 되었다. 이 critical level의 에러 메시지를 수신한 경우, fatal level의 에러 메시지와 마찬가지로 즉시 접속이 중단된다. 또한 alert 메시지가 올바른 alert 메시지임을 확인할 수 있도록 checksum을 함께 전송하게 된다.

2.3 WTLS의 알고리즘

① 의사 난수 함수(PRF :Pseudo Random Function)

TLS에서 사용하는 PRF와 다르게 정의되어 있으며 그 정의는 다음과 같다.

```
P_hash (secret, seed)
= HMAC_hash(secret, A(1) || seed) ||
  HMAC_hash(secret, A(2) || seed) ||
  HMAC_hash(secret, A(3) || seed) || ...

A(0) = seed,
A(1) = HMAC_hash(secret, A(i-1))
```

해쉬 함수를 이용하여 난수열을 생성하는 P_hash의 값은 위에서 정의한 바와 같다. 하지만 TLS에서의 PRF는 P_SHA-1과 P_MD5를 XOR하여 구하는 반면에 WTLS의 PRF는 정해지지 않은 해쉬함수로부터 PRF의 값을 생성한다.

```
[TLS]
PRF( secret, label, seed )
= P_MD5(S1, label || seed) xor P_SHA_1(S2,
          label || seed).
[WTLS]
PRF( secret, label, seed )
= P_hash( secret, label || seed).
```

[표 5] Ciphe suites 지원 알고리즘 비교

Key Exchange Algorithm.	DH RSA DSA Fortezza	DH RSA DSA	RSA DSA ECDH
Symmetric Algorithm.	IDEA RC2 RC4 DES 3DES Fortezza	IDEA RC2 RC4 DES 3DES	IDEA RC5 DES 3DES
Hash Function.	MD5 SHA	MD5 SHA	SHA SHA_XOR MD5

② Cipher Suites의 비교

초창기 SSL에서부터 TLS, 무선 환경에 적합한 WTLS로 발전함에 따라 지원하는 cipher suite도 변화가 이루어졌다. cipher suite의 변화를 살펴보면 다음 [표 5]와 같다.

2.4 WTLS의 암호 안전성 분석

WTLS는 SSL/TLS를 기반으로 하기 때문에 SSL/TLS가 가지고 있는 취약점이 그대로 상속된다. WTLS는 TLS와 마찬가지로 RSA_anon, DH_anon, 등을 이용하여 익명으로 키 교환 알고리즘 사용 시에는 잘 알려진 man-in-the-middle 공격에 취약하며, RSA_anon을 사용하는 Handshake 프로토콜인 경우에 공격자는 서버의 Hello 메시지를 수신한 후 pre_master_secret을 생성하여 서버의 공개키로 암호화하여 서버로 전송함으로서 위장 공격이 가능하다. 또한, 전송되는 암호문은 WAP gateway를 통하여 복호화 되고 SSL/TLS로 다시 암호화 하여 서버에게 전송되기 때문에 서버와 무선 단말기 사용자 간에 종단간 안전성이 보장되지 않는다는 점이 가장 큰 문제점으로 지적된다.

3. SPSL(Secure and Private Socket Layer)

Pino Persiano와 Ivan Visconti가 제안한 SPSL 프로토콜은 사용자의 프라이버시를 보호하기 위해 Anonymous Identification Protocol을 이용하여 사용자의 익명성을 제공한다[7].

3.1 Anonymous Identification Protocol :AIP

사용자의 익명성을 제공하는 프로토콜로서, RSA방식

을 이용하여 AIP의 동작과정을 분석하면 다음과 같다.

[파라미터]

U_i : 사용자, $1 \le i \le l$

S : 서버

N_l : $p_i \times q_i$, $1 \le i \le l$

e_i : i 번째 사용자의 공개키

d_i : i 번째 사용자의 개인키

s_i : i 번째 사용자의 서명 값

m_i : i 번째 사용자의 메시지

m : 서버의 메시지

서버 S는 사용자 U_i의 공개키 리스트를 가지고 있다.

RSA 공개키 리스트 : $(N_1, e_1), \ldots, (N_l, e_l)$

다음과 같은 절차를 통하여 사용자들의 익명성이 제공된다.

Common Input : $(N_1, e_1), \ldots, (N_l, e_l)$

U's Private Input :

(i, d_i) *such that* $d_i \cdot e_i \equiv 1 \bmod N_i$

S.1 Pick a random message $m \in \{0,1\}^n$, **send m to U**

U.1 For each $1 \le j \le l$ and $j \ne i$

U.1.1 Pick a random *signature* $s_j \in \{0,1\}^n$

U.1.2 Compute $m_j = s_j^{e_j} (\bmod N_j)$

U.2 Compute

$m_i = m \oplus m_1 \oplus \ldots m_{i-1} \oplus m_{i+1} \oplus \ldots m_l$

Compute $s_i \equiv m_i^{d_i} (\bmod N_i)$

Send $(m_1, \ldots, m_l, s_1, \ldots, s_l)$ **to S**

S.2 Verify that

$m_1 \oplus m_2 \oplus \ldots \oplus m_{l-1} \oplus m_l = m$

For $i = 1, \ldots, l$ **if** $s_i^{e_i} \ne m_i$ **then ABORT**

[AIP 예제]

가정) 사용자의 집합은 3이며, 그 중 한 사용자 U_1이 자신의 익명성을 서버로부터 보장받으려 한다.

RSA 공개키 리스트는 다음과 같다.

$$(N_1, e_1), (N_2, e_2), (N_3, e_3)$$

① 서버 S는 메시지 m을 랜덤하게 선택한 후 각 사용자 U_i들에게 전송해 준다.

② 서버 S로부터 메시지 m을 받은 사용자 U_1는 랜덤하게 s_j를 선택한 후 m_2, m_3를 다음과 같이 계산한다.

$$s_2^{e_2} \bmod N_2 = m_2$$

$$s_3^{e_3} \bmod N_3 = m_3$$

③ U_1은 $m_1 = m \oplus m_2 \oplus m_3$ 계산한 후 자신의 서명 s_1을 생성한다.

$$s_1 \equiv m_1^{d_1} (\bmod N_1)$$

④ U_1은 $(m_1, m_2, m_3, s_1, s_2, s_3)$을 서버로 전송한다.

⑤ 서버 S는 다음과 같은 방법으로 자신이 초기에 사용자 U_i에게 전송한 메시지 m임을 검증한다.

$$m_1 \oplus m_2 \oplus m_3 = m$$

⑥ 서버 S는 전송받은 서명들을 모두 검증한다.

서명이 모두 올바르다면 서버 S는 3명 중 한명이 보낸 메시지임을 확인한다. 하지만 3명의 사용자 중 누구로부터 전송된 메시지인지는 알 수 없다.

3.2 The SPSL Handshake 프로토콜

SPSL Handshake 프로토콜은 서버와 클라이언트 사이에 다음과 같이 크게 4가지의 익명성 레벨을 정의하고 있다.

① **NULL**

이 레벨은 익명성 레벨이 NULL인 상태 즉, 사용자는 공격자나 서버로부터 익명성을 전혀 보장 받지 않은 상태를 말한다(일반적인 SSL/TLS).

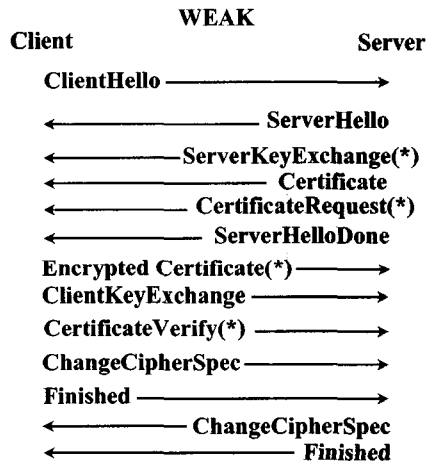
② **WEAK**

만약, 서버가 클라이언트의 인증서를 요청 하였을 경

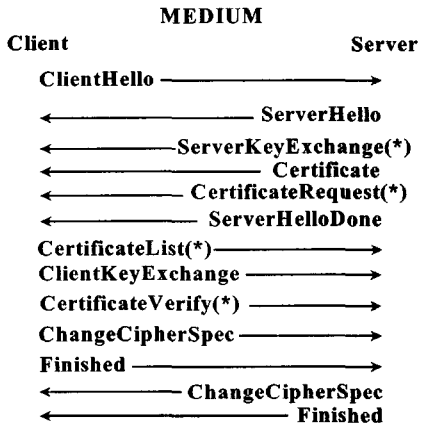
우, 클라이언트는 서버의 암호화키로 인증서를 암호화 하여 전송한다(단 서버의 인증서에 암호화키가 반드시 포함되어 있어야 함). [그림 7]은 WEAK 레벨인 경우의 Handshake 프로토콜을 보여주고 있다. WEAK 레벨인 경우 사용자의 인증서가 서버의 암호화키로 암호화한 후에 전송되므로 사용자는 공격자로부터 익명성이 보장되지만 서버로부터의 익명성은 제공되지 않는다.

③ MEDIUM

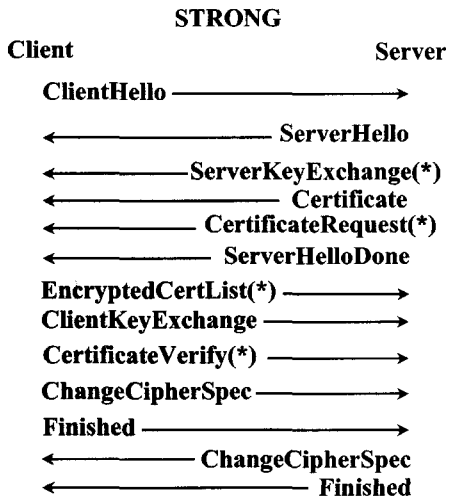
만약, 서버가 클라이언트 인증서를 요청 하였을 경우, 클라이언트는 인증서 리스트 설정하고 AIP를 사용하여 서버와 공격자로부터 익명성이 보장된다. [그림 8]은 익명성 MEDIUM 레벨인 경우의 Handshake 프로토콜을 보여주고 있다.



[그림 7] WEAK 레벨의 Handshake



[그림 8] MEDIUM 레벨의 Handshake



[그림 9] STRONG 레벨의 Handshake

④ STRONG

MEDIUM 레벨의 경우와 다른 점은 클라이언트가 선택한 인증서 리스트를 전송할 때 서버의 암호화키로 인증서 리스트를 암호화 하여 전송한다. MEDIUM과 마찬가지로 서버와 공격자로부터 익명성이 보장되며, 또한 공격자는 인증서의 리스트조차 전혀 알 수 없다.

Ⅲ. SSL/TLS와 WTLS의 발전방향

지금까지 SSL/TLS와 WTLS의 변천사 및 주요 동작과정과 안전성에 대하여 살펴보았다. 본 장에서는 다양한 분야에서의 SSL/TLS와 WTLS을 이용한 최근 보안 동향을 살펴보고 향후 발전방향에 대하여 논의한다.

1. VPN 분야

VPN(Virtual Private Network: 가상사설망)은 저렴한 공공의 인터넷망을 이용하여 고비용의 사설전용선을 사용하는 효과를 얻는 것이다. 즉, 공중망을 마치 자신의 전용망처럼 사용하는 서비스로, VPN 상에 소속된 사용자는 VPN을 Private Network으로 인식한다.

최근, VPN의 표준처럼 쓰였던 IPSec VPN 방식에서 SSL을 이용한 SSL VPN이 급부상하고 있다. 이는 SSL은 주요 웹 브라우저에 기본적으로 탑재되어 있으므로 SSL VPN은 웹 브라우저에서 SSL을 이용하여 쉽게 VPN을 구성하는 장점을 가지고 있다. IPSec을 이용한 VPN 구성에서는 원격 사용자의 데스크톱이나 노트북에 반드시 클라이언트 소프트웨어가 설치되어야 하지만,

SSL VPN은 기본 웹 브라우저만 있으면 어디서나 간편하게 VPN에 접속할 수 있다. 즉, 사용자들은 별도의 장비나 소프트웨어가 없이 인터넷 주소만 입력함으로서 손쉽게 VPN을 구성할 수 있는 것이다. 따라서 SSL VPN은 IPSec VPN에 비해 설치 및 관리가 편리하고 비용도 절감되는 효과를 얻을 수 있다.

2. 무선 랜 분야

인터넷 서비스 사업자들은 핫스팟(HOT SPOT)이라고 불리는 특정 장소에서 가입자들에게 무선 랜 서비스를 시작하였다. 이러한 공중 무선 랜은 다른 사업자간의 로밍, 신호 간섭 등 여러 가지의 요구사항을 가지고 있으며 이에 IEEE 802. 11 워킹 그룹은 무선 랜의 보안 향상을 위하여 IEEE 802.1x에서 사용자 인증 프로토콜인 EAP(Extensible Authenfication Protocol)을 정의하고 있다. EAP의 여러 가지 인증 방법 중에 PKI를 기반으로 하는 인증서 방식의 EAP-TLS 방식이 있다. 이는 SSL/TLS 보안 솔루션을 무선 랜에 적용한 한 예로 볼 수 있다. 그러나, SSL/TLS 보안 솔루션을 무선 랜에 적용하였을 경우, 단말에서의 많은 연산량을 요구하게 되므로 저전력 CPU 및 저메모리를 사용하는 무선 단말기에는 적합하지 않다. 따라서, 무선 환경에 적합한 WTLS에서 지원하는 알고리즘과 WTLS 인증서를 이용한 인증 방법을 고려하는 것이 바람직하다.

3. 방송·통신 분야

최근, 융합 현상이 방송·통신·금융 분야에서 활발히 이루어져 서로 다른 분야의 경계가 무너지고 새로운 통합 서비스가 제공되기 시작하였다. 특히, 방송·통신 분야에서는 1100만 가입자에 달하는 케이블 TV에서의 주문형 비디오(VOD)서비스와 핸드폰과 같은 소형 단말기에서의 위성 방송을 이용한 TV 수신 서비스가 상용화 되고 있다. 이러한 방송·통신 융합 서비스들은 새로운 네트워크 솔루션을 필요로 하고 있는 실정이며, 특히 VOD와 같은 그룹 지향적인 서비스들은 강도 높은 보안 솔루션이 함께 요구되고 있다. 현재 보안 솔루션으로 웹 기반의 SSL/TLS가 크게 주목 받고 있다. 이는 SSL/TLS가 앞장에서 설명하였듯이 전송계층과 응용계층 사이에 위치함으로서 다양한 응용계층의 프로그램들과 쉽게 보안설정을 할 수 있으며 확장성이 용이하고, 또한 웹 기반으로서 시스템 개발자뿐만 아니라 사용자도 쉽게 시스템을 구축하고 사용할 수 있다는 장점을 가지고 있기 때문이다.

4. SSL/TLS와 WTLS의 발전 방향

최근, 인터넷 상에서 웹 보안과 함께 사용자의 프라이버시 향상을 위한 익명성에 관한 논쟁이 대두되고 있다. 일반적으로 인증서를 기반으로 하는 SSL/TLS 방식에서 인증서는 근본적으로 사용자의 익명성을 보장하지 못하므로 Pino Persiano와 Ivan Visconti는 사용자의 익명성을 보장하는 확장된 SSL/TLS 프로토콜을 제안하였다. 하지만, 제안된 SPSL 프로토콜은 사용자의 익명성을 제공하기 위하여 많은 양의 서명 및 검증을 수행해야 하는 단점을 지니고 있다.

향후, SSL/TLS와 WTLS 프로토콜은 전송되는 정보의 무결성 및 기밀성만을 고려하기보다는 사용자 프라이버시를 보장해주는 익명성 제어 기술에 관한 연구가 진행되어 나아가야 할 것이다.

또한, 최근 인터넷 활용 분야가 다양한 분야로 확대되면서 여러 명의 사용자들에게 동일한 서비스를 제공해주기 위한 그룹 통신에 대한 요구가 증가하고 있다. 이러한 요구는 현재 방송(유료 영상 서비스), 금융(주식 시세 정보), 교육(원격 사이버 강의), IT(소프트웨어 업데이트), 엔터테인먼트(다중 사용자 게임) 등의 다양한 분야로 확대되고 있는 실정이다. 이에 많은 웹 서비스 업체들과 통신 사업자들은 원격 음성·화상 회의, 커뮤니티 채팅과 같은 그룹 지향적인 응용 어플리케이션들을 개발하고 있다. 하지만 현재의 SSL/TLS와 WTLS는 Handshake 과정에서 그룹키 설정 알고리즘을 지원하지 않으며 그 동작 구조또한 그룹 지향적인 응용 어플리케이션에 적합하지 않다. 이에 SSL/TLS와 WTLS 보안 프로토콜은 향후 웹 기반의 다양한 그룹 지향적인 서비스를 제공할 수 있도록 그룹 키 설정 알고리즘을 지원하도록 설계가 이루어져야 할 것이다.

Ⅳ. 결 론

본 고에서는 SSL/TLS와 WTLS의 변천사와 구조 및 동작과정에 대하여 살펴보았으며, 각 프로토콜을 비교·분석하여 안전성을 진단해 보았다. 2000년도에 발표된 익명성 제어 가능한 SPSL 프로토콜을 분석하였으며 또한, 다양한 분야에서의 SSL/TLS와 WTLS의 활용도를 알아보고 향후, SSL/TLS와 WTLS이 어떻게 발전해 나아가야 하는지에 대해 모색해 보았다.

참 고 문 헌

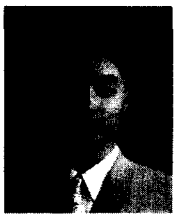
[1] S.Thomas, "SSL snd TLS essential", John

Wiley & Sons, inc, 2000
[2] Alan O. Freier, "The SSL Protocol version 3.0", internet draft, 1996
[3] T.Dierks and C.Allen, "TLS Protocol version 1.0",RFC 2246, 1999
[4] D.Wagner, B.Schneier, "Analysis of the SSL 3.0 Protocol", Proceeding of the Second USENIX Workshop on Electronic Commerce, 1996
[5] WAP Forum, Wireless Transport Layer Security, version 06-Apr-2001, 2001
[6] http://www.itsecurity.com/asktecs/jul701.htm
[7] Pino Persiano, Ivan Visconti, "User Privacy Issues Regarding Certificates and the TLS Protocol", CCS'00, ACM, 2000

〈著 者 紹 介〉

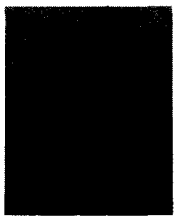
이 진 우(Jinwoo Lee)
학생회원

2003년 2월 : 성균관대학교 정보통신공학부 졸업(공학사)
2003년 3월~현재 : 성균관대학교 컴퓨터공학과 석사과정
〈관심분야〉 암호 프로토콜, 네트워크 보안



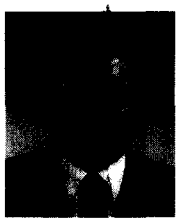
남 정 현 (Junghyun Nam)
학생회원

1997년 2월 : 성균관대학교 정보공학과(공학사)
2002년 5월 : Computer Science, University of Louisiana, Lafayette (M.S.)
2003년 3월~현재 : 성균관대학교 정보통신공학부 박사과정
〈관심분야〉 암호 프로토콜, 암호이론, 네트워크 보안



김 승 주 (Seungjoo Kim)
종신회원

1994년 2월 : 성균관대학교 정보공학과 (공학사)
1996년 2월 : 성균관대학교 대학원 정보공학과 (공학석사)
1999년 2월 : 성균관대학교 대학원 정보공학과 (공학박사)
1998년 12월~2004년 2월 : 한국정보보호진흥원(KISA) 팀장
2001년 1월~현재 : 한국정보보호학회 논문지편집위원
2002년 4월~현재 : 한국정보통신기술협회(TTA) IT 국제표준화 전문가
2004년 3월~현재 : 성균관대학교 정보통신공학부 교수
〈관심분야〉 암호이론, 정보보호표준, 정보보호제품 및 스마트카드 보안성 평가, PET



원 동 호 (Dongho Won)
종신회원

1976년~1988년 : 성균관대학교 전자공학과 (학사, 석사, 박사)
1978년~2003년 : 한국전자통신연구소 전임 연구원, 일본 동경공대 객원연구원, 성균관대학교 교학처장, 전기·전자 및 컴퓨터공학부장, 정보통신대학원장, 연구지원처장, 국무총리실 정보화추진위원회 자문위원, 한국정보보호학회 이사, 부회장, 수석부회장, 회장
현재 : 성균관대학교 정보통신공학부 교수, 한국정보보호학회 명예회장, 정통부지정 정보보호인증기술연구센터 센터장