# A Technical Comparison of IPSec and SSL

AbdelNasir Alshamsi * Takamichi Saito †
Tokyo University of Technology

## Abstract

*IPSec (IP Security) and SSL (Secure Socket Layer) have been the most robust and most potential tools available for securing communications over the Internet. Both IPSec and SSL have advantages and shortcomings. Yet no paper has been found comparing the two protocols in terms of characteristic and functionality. Our objective is to present an analysis of security and performance properties for IPSec and SSL.*

## 1 Introduction

Securing data over the network is hard and complicated issue while the threat of data modification and data interruption is rising. The goal of network security is to provide *confidentiality*, *integrity* and *authenticity*.

The combination of these properties is the pillar of the security protocols. How to combine them is the question with many answers. With the recent development of the security tools, so many protocols and powerful tools have been proposed, but the most famous, secure and widely deployed are IPSec (IP Security) [1] and SSL (Secure Socket Layer) [2].

In this paper we will provide a technical comparison of IPSec and SSL; the similarities and the differences of the cryptographic properties. The results of performance are based on comparing OpenSWAN [3] as IPSec and Stunnel [4] as SSL.

## 2 IPSec

IPSec [1] is an IP layer protocol that enables the sending and receiving of cryptographically protected packets of any kind (TCP,UDP,ICMP,etc) without any modification. IPSec provides two kinds of cryptographic services. Based on necessity, IPSec can provide confidentiality and authenticity (1) or it can provide authenticity only (2):

1. ESP (Encapsulated Security Payload) [5]
2. AH (Authentication Header) [6]

ESP provides confidentiality, authenticity and integrity protection for the communication. AH on the other hand ensures that authenticity and integrity of the data is protected. Establishing IPSec connection requires two phases: Phase 1 (ISAKMP SA) [7] and Phase 2 (IPSec SA) [8] (see table 1).

### 2.1 Phase 1

Phase 1 performs mutual authentication and produces the encryption key required to protect Phase 2. Phase 1 has two modes: Main Mode and Aggressive Mode. The differences between these two modes are the number of messages exchanged and the ID protection.

### 2.2 Phase 2

Phase 2 negotiates the cipher and authentication algorithm required to protect further transactions. Phase 2 has one mode, Quick Mode.

### 2.3 Key Exchange

Various methods of Key Exchange mechanism and Authentication methods are supported by IPSec.

**Key Exchange Method:**

1. DH
2. KINK[1] [9]

## 3 SSL

SSL (Secure Socket Layer) [2] is an Application layer protocol. SSL is mostly utilized to protect HTTP transactions, and has been used for other purposes like IMAP and POP3, etc. SSL is compatible with applications running only over TCP, but some modifications

*Graduate School of System Electronics, 1404-1 Katakurcho, Hachiouji City ,Japan. alshamsi@aqua.ts.it.teu.ac.jp

†Department of Computer Science, 1404-1 Katakurcho, Hachioujicity ,Japan. saito@cc.teu.ac.jp

[1] KINK is a Kerberos based protocol that provides Key Exchangeand authentication mechanism, Not standardized yet.



Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

are required for the applications to run over SSL. Recent development of SSL software like Stunnel [4] has added an easiness of use to SSL. SSL is composed of the following protocols:

1. Handshake protocol
2. Change Cipher Spec protocol
3. Alert protocol
4. Application Data protocol

Handshake protocol is used to perform authentication and key exchanges. Change Cipher Spec protocol is used to indicate that the chosen keys will now be used. Alert protocol is used for signaling errors and session closure. Application Data protocol transmits and receives encrypted data.

### 3.1 Key Exchange

**Key Exchange Method:**

1. RSA
   Client sends the *pre_master_secret* after encrypting it with Server's public key.
2. DH
   Client and Server exchange DH public values and produce the pre_master_secret independently.

## 4 Comparison of IPSec and SSL

### 4.1 Authentication Algorithm

IPSec supports the use of Digital Signature and the use of a Secret Key Algorithm, where SSL supports only the use of Digital Signature. The use of a random 2048 bit Secret Key is considered as strong as any other authentication methods. In the absence of Digital Signature algorithm, IPSec can still be implemented using the Secret Key but SSL can't be implemented.

### 4.2 Authentication Method

IPSec supports one type of authentication method while SSL supports a various types of authentication. They are described in table 1 and 2.

Table 1: IPSec Authentication Method

| Authentication Method | Authentication Algorithm |
|---|---|
| Mutual Authentication | PSK |
| | RSA/DSA Digital Signature |
| | RSA Public Key |
| | KINK |

Table 2: SSL Authentication Method

| Authentication Method | Authentication Algorithm |
|---|---|
| Server Authentication | RSA (Challenge/Response) |
| | DSA Digital Signature |
| Client Authentication | RSA/DSA Digital Signature |
| Anonymous | none |

### 4.3 MAC

MAC (Message Authentication Code) is used for authenticating the exchanged messages after the connection is established. Both IPSec and SSL require the implementation of HMAC-SHA-1 and HMAC-MD5. HMAC is a hash function that requires a secret key to produce message digest. The strength of the Hash Algorithm is based on the length of the output (see table 3).

Table 3: HMAC Algorithm Type

| Protocol | MAC Algorithm | Hash Length |
|---|---|---|
| IPSec | HMAC-SHA-1-96 [10] | 12 Byte |
| | HMAC-MD5-96 [11] | 12 Byte |
| SSL | HMAC-SHA-1 | 20 Byte |
| | HMAC-MD5 | 16 Byte |

### 4.4 Around Transport Layer

IPSec Phase 1 negotiations are exchanged over the UDP (port 500 only). Thus, *Retransmit Timer* must be maintained. SSL Handshake is exchanged over TCP and unlike IPSec; the port can be changed according to the application.

As a Server, both IPSec and SSL are bound to specific ports where as a Client, IPSec is bound to specific ports but SSL is not.

SSL works only over the TCP since UDP can cause data to be arbitrarily lost or re-ordered. IPSec avoids the UDP problem by adding a new TCP header to the original packet's field, which allow UDP or TCP based applications to work with IPSec. Supporting only TCP application is a shortcoming of SSL.

### 4.5 Order of Cryptographic Operations

IPSec encrypts the data first then creates MAC for the encrypted data. If a modified data were inserted in the middle of transaction, IPSec would verify the MAC before performing any decryption process [1].

SSL is the opposite; it creates the MAC for the plaintext first then encrypts the data. SSL on the other



Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

hand, is obligated to decrypt it first then verifies the MAC which could result in wasting CPU over decrypting modified packets.

### 4.6 Interoperability

IPSec doesn't integrate well with other IPSec vendors [12]. Some cases require some modification. SSL is trouble free and well integrated.

### 4.7 Overhead Size

One disadvantage of IPSec is the extra size added to the original packet. SSL needs less overhead than IPSec. The extra required bytes for each protocol are described in table 4.

Table 4: Overhead Size

| Protocol | Mode | Byte Size |
|---|---|---|
| IPSec Tunnel Mode | ESP | 32 |
| | ESP and AH | 44 |
| IPSec Transport Mode | ESP | 36 |
| | ESP and AH | 48 |
| SSL | HMAC-MD5 | 21 |
| | HMAC-SHA-1 | 25 |

IPSec Tunnel Mode requires adding another 20 Byte IP header. All the data above don't include the padding bytes and the pad length.

### 4.8 Residing Layer

Because IPSec resides in the IP layer, it allows multi-users to use one tunnel between two endpoints while SSL allow multi-users to have individual connections and different encryption key for each connection. The merit of using one tunnel for multi-users, as with IPSec, is to lower the overhead caused by establishing individual connections. The merit of using independent connection, as with SSL, is that each user has individual session. Consequently, compromising one connection doesn't compromise the other connections.

### 4.9 Time of Handshake Process

The time to establish a session is another element. Table 5 shows how much time is needed to establish a session for IPSec . The results are based on the use of a 2048 bit RSA key and 1536 bit DH.

Table 6 shows the time required for establishing SSL session. The results are based on the use of a 2048 bit RSA key and 768 bit DH. Using 1536 bit in DH has consumed 1648 msec in the Client Authentication. That is considered extremely slow when it is compared with 768 bit.

Table 5: IPSec Handshake Time

| Mode | Establishing |
|---|---|
| Main Mode (PSK) | 97 msec |
| Aggressive Mode (PSK) | 56 msec |
| Main Mode (RSA) | 170 msec |

Table 6: SSL Handshake Time

| Mode | Establishing |
|---|---|
| Server Authentication | 41.7 msec |
| Client Authentication | 74.8 msec |
| Server Authentication (Diffie-Hellman) | 66.1 msec |
| Client Authentication (Diffie-Hellman) | 118.6 msec |

### 4.10 Compression Algorithm

Compression is utilized by IPSec through a compression protocol called IPComp [13]. Unfortunately compression is used in a small range with SSL. Only *OpenSSL* [14] supports compression.

We have examined the compression in two different environments (low and high bandwidth). Compressin in a low bandwidth topology increased the throughput for IPSec and SSL. Compression in a high bandwidth topology decreased the throughput for IPSec except with 3DES and increased the throughput for SSL.

The increase and decrease of throughput is based on a combination of elements; the residing layer, the overhead size generated by each protocol and the relevant speed between the compression, the encryption and the trasnfer.

### 4.11 Performance

The experiments were conducted on two machines with the following:

1. Fedora Core 1 (Kernel-2.4.22)
2. Pentium 4, 2.4 GHz, RAM 512 MB
3. NIC 1000 Mbps
4. Openswan 1.8 as IPSec
5. Stunnel 4.06 as SSL
6. Ethereal as time measuring tool
7. Iperf 2.0.1 as performance measuring tool
8. Cyclesoak as CPU measuring tool

The results have shown a variation of throughput speed and CPU consumption.



Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**

#### 4.11.1 IPSec ESP

Table 7 illustrates the perfomance of IPSec with compression and without. CPU consumption varied between 91.5% and 93.3%.

Table 7: IPSec Perfomance

| Algorithm | Throughput (Mbps) | | | |
|---|---|---|---|---|
| | No Compression | | Compression | |
| | MD5 | SHA-1 | MD5 | SHA-1 |
| No Algorithm | 427 | | | |
| 3DES | 77.1 | 67.4 | 109 | 96.3 |
| DES | 131 | 109 | 111 | 104 |
| AES256 | 153 | 134 | 112 | 102 |
| AES128 | 191 | 155 | 114 | 103 |

The reason the perfomance falls down in IPSec when applying compression is the relation between the encryption speed and the compression. Most of the encryption algorithms are faster than the compression except for 3DES. Therefore, applying the compression to a higher speed encryption algorithm in IPSec will cause the throughput to fall down.

#### 4.11.2 SSL

Our experiment includes the case of using exportable keys like EXP1024 and EXP512 RSA keys[2] , the throughput rate and CPU consumption don't show any change. For this reason, it will not be included in this paper.

Table 8 illustrates the perfomance of SSL. CPU consumption varied between 89.7% and 93%.

Table 8: SSL Performance

| Algorithm | Throughput (Mbps) | |
|---|---|---|
| | No Compression | Compression |
| 3DES-CBC-SHA | 87.9 | 260 |
| DES-CBC-SHA | 154 | 282 |
| AES256-SHA | 140 | 247 |
| AES128-SHA | 162 | 280 |
| RC4128-MD5 | 234 | 280 |
| RC4128-SHA | 214 | 285 |

## 5 Conclusion

We presented the resemblance and the differences between IPSec and SSL. Each of the protocols has unique properties. Choosing IPSec or SSL depends on the security needs. If a specific service is required and is supported by SSL, it is better to select SSL. If over all services or Gateway-to-Gateway communications are needed then IPSec is a good choice considering the following: IPSec uses a shorter form of HMAC than SSL, thus SSL data integrity is more secure. SSL is more compatible with firewall than IPSec, unless IPSec and Firewall are integrated in the same device. Unlike SSL, IPSec clients need a special IPSec software for remote access. In low bandwidth networks or dial-up networks using compression is beneficial, SSL doesn't support that. Pre-Shared scheme is easier to configure and doesn't require any PKI infrastructure,IPSec supports compression but unfortunately SSL doesn't support it. IPSec is capable of protecting wireless networks. In most cases IPSec doesn't interoperate well, so both sides of the connection are required to have the same vendor's devices (see table 9).

Table 9: IPSec vs. SSL

| Function | IPSec | SSL |
|---|---|---|
| Configuration | hard | easy |
| Client Authentication | must | option |
| Pre-Shared Key | yes | no |
| Interoperability Problem | yes | no |
| TCP Application Support | all | some |
| UDP support | yes | no |
| Throughput Rate | high | high |
| Compression Support | yes | OpenSSL only |
| Handshake Time | slow | fast |

[2] US law requires using limited RSA encryption key length for exportable application. The export control was relaxed.

## References


[1] Sheila Frankel, *"Demystifying the IPSec Puzzle"*, Artec House Publisher, 2001

[2] Eric Rescorla, *"SSL and TLS, Designing and Building Secure Systems"*, Addison-Wesley, 3rd Printing,Aug. 2001.

[3] `www.opwnswan.org`

[4] `www.stunnel.org`

[5] S. Kent, R. Atkinson, *"IP Encapsulating Security Payload (ESP)"*, RFC 2406. Nov. 1998.

[6] S. Kent, R. Atkinson, *"IP Authentication Header"*, RFC 2402. Nov. 1998.

[7] D. Maughan M., Schertler M., Schneider J. Turner, *"Internet Security Association and Key Management Protoc ol (ISAKMP)"*, RFC 2408, Nov 1998.

[8] D. Harkins, D. Carrel, *"The Internet Key Exchange( IKE)"*, RFC 2409, Nov 1998.

[9] M. Thomas,J. Vilhuber, *"Kerberized Internet Negotiation of Keys (KINK)"*, Internet Draft. Jan 2003.

[10] C. Madson, R. Glenn, *"The Use of HMAC-SHA-96 within ESP and AH"*, RFC 2404, Nov. 1998.

[11] C. Madson, R. Glenn, *"The Use of HMAC-MD5-1-96 within ESP and AH"*, RFC 2403, Nov. 1998.

[12] `www.ipa.go.jp/security/fy12/report /ipsec.html`

[13] A. Shacham, B. Monsour, R. Pereira. M. Thomas, *"IP Payload Compression Protocol (IPCOMP)"*, RFC 2393, Dec. 1998.

[14] `www.openssl.org`




Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA'05)
1550-445X/05 $20.00 © 2005 **IEEE**