PERBANDINGAN KINERJA IP SEC DAN SSL

Dessyanto Boedi P, ST., MT.
Teknik Informatika
UPN "Veteran" Yogyakarta

Abstract

Ipsec (IP Security) and SSL (Secure Socket Layer) is the most widely used techniques to secure data communications over the Internet. Both of these techniques has advantages and disadvantages of each. The purpose of this study is to present an analysis of the two techniques above in terms of security and performance.
Key words : IP Sec, SSL, Internet

Ipsec (IP Security) dan SSL (Secure Socket Layer) merupakan teknik yang paling banyak digunakan untuk mengamankan komunikasi data melalui Internet. Kedua teknik ini memiliki keunggulan dan kelemahan masing-masing. Tujuan dari penelitian ini adalah untuk menyajikan analisis terhadap kedua teknik di atas dalam segi keamanan dan kinerja.
Kata kunci : IP Sec, SSL, Internet

1 Pendahuluan

Pengamanan data yang berlalu lalang di jaringan Internet merupakan masalah yang sulit dan rumit sementara ancaman terhadap penyadapan dan pembajakan data semakin hari masing mengkhawatirkan. Tujuan pengamanan data dalam jaringan adalah untuk memberikan kerahasian (*confidentiality*), integritas (*integrity*) dan keautentikan (*authenticity*) baik terhadap data maupun komunikasinya.

Confidentiality merupakan cara untuk menjaga data agar tetap bersifat rahasia bagi pihak-pihak yang tidak berhak terhadap data tersebut. *Integrity* merupakan cara yang digunakan untuk menyakikan pengguna data bahwa data tersebut benar-benar data yang dikirim. Sedangkan *authenticity* digunakan untuk membuktikan identitas dari masing-masing pihak yang terlibat dalam komunikasi data.

Ketiga hal tersebut merupakan pilar-pilar yang digunakan dalam protokol keamanan jaringan. Di dalam masing-masing pilar tersebut terdapat berbagai macam algoritma yang bisa digunakan. Penggunaan berbagai algoritma tersebut memerlukan berbagai pertimbangan. Penggunaan kunci kriptografi yang kuat dengan algoritma autentikasi yang lemah dapat membuat *hacker* dengan mudah mengacaukan data. Pengunaan algoritma autentikasi yang kuat dengan algoritma kriptografi yang lemah dapat mempermudah *hacker* dalam membongkar data. Penggunaan algoritma kriptografi dan algoritma autentikasi yang kuat akan semakin baik dalam melindungi data namun akan berdampak pada kecepatan pengiriman data dan konsumsi CPU.

Berbagai teknik dan tool pengamanan jaringan telah dikembangkan, namun teknik yang paling banyak digunakan adalah IPSecdan SSL (Frankel, 2001) . Tulisan ini akan membahas perbandingan teknis dari IPSec dan SSL baik itu kemiripan maupun perbedaannya dalam mengamankan data dan jaringan.

2 IPSec

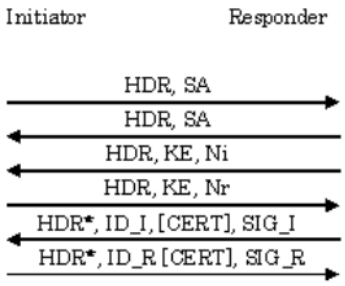
IPSec (Frankel, 2001) merupakan protokol yang bekerja pada layer IP yang digunakan untuk memproteksi data yang dikirimkan baik menggunakan TCP, UDP atau ICMP. Dalam IPSec terdapat dua macam layanan kriptografi yaitu:

- ESP (*Encapsulated Security Paylaod*)
- AH (*Authentication Header*)

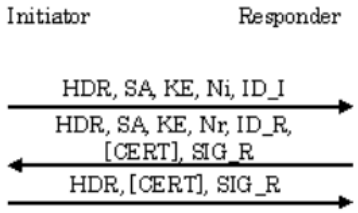
ESP digunakan untuk menyediakan layanan *confidentiality, authentication* dan *integrity* terhadap komunikasi data. Sedangkan AH hanya digunakan untuk *authentication* dan *integrity*. Di dalam ESP *header* terdapat informasi yang diperlukan untuk dekripsi dan autentikasi data. Pembentukan koneksi IPSec memerlukan dua fase yaitu *Phase* 1 (ISAKMP SA) dan *Phase* 2 (IPSec SA).

2.1 Phase 1

Phase 1 merupakan fase yang di dalamnya terdapat proses autentikasi dan proses untuk menghasilkan kunci enkripsi untuk memproteksi Phase 2. Phase 1 memiliki dua mode yaitu *Main Mode* dan *Aggressive Mode*. Perbedaan kedua mode tersebut terletak pada jumlah pesan yang dipertukarkan dan proteksi ID.



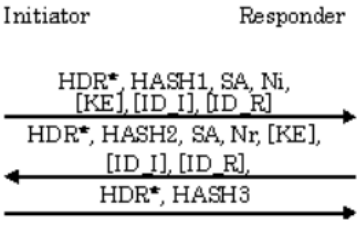
Gambar 1. IPSec *Main Mode*



Gambar 2. IPSec *Aggressive Mode*

2.2 Phase 2

Di dalam fase 2 terdapat proses negosiasi algoritma autentikasi dan enkripsi yang akan digunakan dalam pertukaran data. Phase 2 hanya memiliki satu mode yaitu Quick Mode.



Gambar 3. IPSec *Quick Mode*

Keterangan:

HDR	: ISAKMP *Header*
SA	: *Security Association*
KE	: *Diffie-Hellman exchanged public value*
Ni, Nr	: *nonce*
ID_I,ID_R	: *Initiator, Responder*
CERT	: *Certificate*
SIG_I, SIG_R	: *Signature Initiator, Responder*

2.3 Pertukaran Kunci dan Autentikasi

Berikut ini merupakan mekanisme pertukaran kunci dan metode autentikasi yang digunakan dalam IPSec.

Metode pertukaran kunci:

- Diffie-Hellman

- KINK

Metode autentikasi:

- Pre-shared Key (PSK)
- Digital Signature
- Public Key
- KINK

Metode hash function:

- MD5
- SHA-1

3 SSL

SSL (Secure Socket Layer) merupakan protokol yang bekerja di layer aplikasi. SSL banyak digunakan untuk melindungi pertukaran data pada aplikasi HTTP dan juga digunakan dalam aplikasi IMAP dan POP3. Pengguna dapat dengan mudah mengimplementasikan protokol SSL menggunakan perangkat lunak SSL yang bernama Stunnel. Di dalam SSL sendiri terdapat beberapa protokol seperti:

1. *Handshake Protocol*
2. *Chage Cipher Spec Protocol*
3. *Alert Protocol*
4. *Aplication Data Protocol*

Handshake protocol merupakan proses autentikasi dan pertukaran kunci. *Change Cipher Spec Protocol* digunakan untuk mengindikasikan bahwa kunci - kunci yang dipilih akan digunakan dalam SSL. *Alert Protocol* digunakan untuk mengetahui apakah terjadi kesalahan dalam proses SSL atau tidak dan untuk penutupan sesi. *Application Data Protocol* digunakan untuk mengirim dan menerima data yang terenkripsi (Rescorla, 2001).

3.1 Pertukaran Kunci dan Autentikasi

Metode pertukaran kunci:

1. RSA
   Client mengirimkan *pre_master_secret* setelah mengenkripsikannya menggunakan *public key* milik Server.
2. DH
   Client danserver mempertukarkan infomasi data DH dan menghasilkan *pre_master_secret* secara independen.

Metode Autentikasi:

1. Autentikasi Server
2. Autentikasi Client
3. Anonymous

Contoh SSL Handshake dapat dilihat pada gambar di bawah ini:

Gambar 4. Proses *handshake* dalam SSL

Keterangan:

ClientHello	:	client memberikan informasi ke server mengenai proses enkripsi apa yang bisa digunakan oleh client
ServerHello	:	Server memilih cara enkripsi apa yang digunakan
Certificate	:	Server mengirimkan sertifikat
CertificateRequest	:	Server meminta sertifikat client
ServerHelloDone	:	Server telah mengirimkan prose *Handshake*
Certificate	:	Client mengirimkan sertifikat
ClientKeyExchange	:	Client mengirimkan *pre_master_secret* yang telah dienkripsi menggunakan public key milik server
ChangeChiperSpec	:	Client mengirimkan pesan pada server untuk mengganti proses enkripsi yang baru
Finished	:	Pesan selesai
ChangeChiperSpec	:	Server mengirimkan pesan pada server untuk mengganti proses enkripsi yang baru
Finished	:	Pesan selesai

4 Perbandingan IPSec dan SSL

4.1 Algoritma Autentikasi

Digital Signature dan algoritma Secret Key dapat digunakan dalam IPSec, sedangkan SSL hanya mendukung Digital Signature. Penggunaan Secret Key 2048 bit secara random merupakan metode autentikasi yang sangat baik. Pada poin ini IPSec memiliki keunggulan yang lebih.

4.2 Metode Autentikasi

IPSec hanya memiliki satu metode autentikasi sedangkan SSL memiliki tiga metode autentikasi seperti yang terlihat dalam tabel berikut.

Tabel 1. Metode autentikasi IPSec

Metode autentikasi	**Algoritma autentikasi**
Mutual authentication	*PSK*
	RSA/DSA Digital Signature
	RSA Public Key
	KINK

Tabel 2. Metode autentikasi SSL

Metode autentikasi	**Algtoritma autentikasi**
Autentikasi server	RSA (Challenge/Response)
	DSA Digital Signature
Autentikasi client	RSA/DSA Signature
Anonymous	Tidak ada

4.3 MAC

MAC atau *Message Authentication Code* merupakan teknik yang digunakan untuk autentikasi pesan yang dipertukarkan setelah koneksi terbentuk. Baik IPSec maupun SSL memerlukan implementasi HMAC-SHA-1 dan HMAC-MD5. HMAC merupakan fungsi hash yang menggunakan *secret key* untuk menghasilkan *message digest*.

Tabel 3. Tipe algoritma HMAC

Protokol	Algtoritma MAC	Panjang nilai hash
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 Mode Koneksi

IPSec memiliki dua mode koneksi yaitu:

- Mode Tunnel
  Koneksi ini terjadi antara *gateway* dengan *gateway*, *gateway* dengan *host* dan *host* dengan *host*. Penggunaan mode tunnel memerlukan header IP baru bagi paket IP asli.
- Mode Transport
  Koneksi ini terjadi antara *host* dengan *host*. Data yang dipertukarkan di antara dua entitas tersebut dienkripsi.

Keuntungan dari mode tunnel adalah tidak terdapat *overhead* pada setiap saluran. Kerugiannya hanya terjadi pada saat kunci yang digunakan diketahui oleh pihak ketiga. Sedangkan SSL merupakan tipe satu koneksi per sesi sehingga jika terdapat banyak sesi maka akan terdapat banyak koneksi pula. Hal ini tentu saja akan memperbesar *overhead*.

4.5 Remote Access

IPSec mengalami masalah pada saat digunakan dalam *remote access* menggunakan autentikasi PSK di dalam *Main Mode*. Identitas dalam PSK hanya mengacu pada alamat IP yang digunakan. Penggunaa alamat IP yang tidak statis akan membuat *shared key* tidak dapat ditemukan dan akhirnya koneksi tidak dapat dibentuk. Untuk menghidari terjadinya hal ini maka cara berikut dapat digunakan:

1. alamat IP dari *remote host* diatur pada nilai 0.0.0.0 dan menggunakan satu *shared key* untuk akses *remote host.*
2. Menggunakan *aggressive mode* dimana tipe identitas tidak terbatas pada alamat IP dan identitas ini dikirim ke *responder* pada saat awal negosiasi.
3. Melalukan adaptasi skema autentikasi user seperti PIC [12] atau XAUTH[13].

Penggunaan satu kunci dapat menimbulkan masalah dalam jaringan pada saat kunci tersebut hilang atau dicuri. Perubahan kunci dari waktu ke waktu juga tidak praktis karena semua *client* harus mengatur ulang sistemnya. Dalam *aggressive mode,* identitas client dikirim dalam bentuk teks biasa sehingga hal ini dapat menimbulkan resiko keamanan. XAUTH dapat digunakan sebagai protokol autentikasi user untuk IPSec dimana autentikasi user dilakukan setelah Phase 1 selesai dan sebelum Phase 2 dimulai.

Autentikasi SSL dalam lapisan transport (TCP) didasarkan pada pertukaran RSA atau DSA Digital Signature selama proses autentikasi server/client sehingga *remote access* dapat dilakukan tanpa perlu modifikasi.

4.6 Layer Transport

Negosiasi IPSec Phase 1 dilakukan menggunakan protokol UDP port 500 sehingga Retransmit Timer harus disiapkan dan digunakan dalam koneksi UDP ini. Sedangkan SSL Handshake dilakukan menggunakan TCP dan port yang digunakan dapat diubah.

SSL hanya bekerja menggunakan TCP karena dalam UDP dapat terjadi lost data yang tidak dapat dipulihkan. IPSec menangani masalah ini dengan menambahkan header TCP baru pada paket aslinya sehingga aplikasi yang berbasis UDP ataupun TCP tetap dapat menggunakan IPSec.

Jika IPSec digunakan di belakang firewall maka semua port IPSec harus disediakan secara permanen di dalam firewall. Sedangkan penggunaan SSL dibelakang firewall tidak menjadi masalah mengingat SSL bekerja antar entitas akhir.

Tabel 4. Penggunaan port dalam IPSec dan SSL

Protokol	**Mode**	**Port**
IPSec	Server	ESP 50/TCP
		AH 51/TCP
	Client	ESP 50/TCP
		AH 51/TCP
SSL	Server	HTTPS 443/TCP
	Client	Bebas

4.7 Urutan Operasi Kriptografi

IPSec melakukan enkripsi data terlebih dahulu lalu membuat MAC untuk data terenkripsi tersebut. Jika data baru dimasukkan di tengah – tengah transaksi maka IPSec akan melakukan verifikasi MAC sebelum melakukan proses dekripsi [1].

Gambar 6. IPSec

Sedangkan SSL melalukan hal yang sebaliknya. Dalam SSL, MAC dari teks dibuat terlebih dahulu kemudia teks dienkripsi.

Gambar 7. SSL

4.8 Interoperabilitas

IPSec kurang terintegrasi secara baik beberapa vendor IPSec [14]. Beberapa kasus integrasi memerlukan modifikasi. Sedangkan SSL dapat bekerja dan terintegrasi dengan baik di berbagai aplikasi.

4.9 Overhead Size

Salah satu kelemahan IPSec adalah header tambahan pada paket asli yang cukup signifikan. SSL memiliki overhead yang lebih rendah seperti yang terlihat pada Tabel 5.

Tabel 5. Ukuran overhead

Protokol	**Mode**	**Ukuran (Byte)**
IPSec Tunnel Mode	ESP	32
	ESP dan AH	44
IPSec Transport Mode	ESP	36
	ESP dan AH	48
SSL	HMAC-MD5	21
	HMAC-SHA-1	25

4.10 Penggunaan Layer

IPSec bekerja pada layer IP sehingga dapat bekerja dengan layer diatasnya secara mudah dan baik. Sedangkan SSL bekerja pada layer Aplikasi dan hal ini membuat sebuah masalah pada beberapa aplikasi. Stunnel merupakan sebuah solusi bagi aplikasi berbasis TCP untuk bekerja sama dengan aplikasi SSL.

Karena IPSec bekerja di layer IP maka satu tunnel dapat digunakan oleh banyak user. Hal ini berbeda dengan SSL dimana satu user menggunakan satu koneksi dan tiap koneksi menggunakan kunci enkripsi yang berbeda. Jika dilihat dari sudut pandang ini maka IPSec memiliki overhead konseksi yang lebih rendah jika dibandingkan dengan SSL. Namun SSL memiliki keunggulan yaitu jika keamanan satu koneksi dapat dibongkar maka koneksi yang lain tidak akan terpengaruh.

4.11 Waktu Proses Handshake

Waktu untuk membentuk sesi merupakan elemen penting lain yang menjadi referensi kinerja. Tabel 6 menunjukkan waktu yang untuk membentuk sebuah sesi menggunakan IPSec.

Tabel 6. Waktu pembentukan sesi IPSec

Mode	**Waktu**
Main Mode (PSK)	97 mili detik
Aggressive Mode (PSK)	56 mili detik
Main Mode (RSA)	170 mili detik

Tabel 7 menunjukkan waktu yang diperlukan untuk membentuk sebuah sesi menggunakan SSL. Hasil tersebut didasarkan pada penggunaan kunci RSA 2048 bit dan DH 768 bit.

Tabel 7. Waktu pembentukan sesi SSL

Mode	**Waktu**
Autentikasi server	41,7 mili detik
Autentikasi client	74,8 mili detik
Autentikasi server (Diffie-Helman)	66,1mili detik
Autentikasi client (Diffie-Helman)	118,6 mili detik

4.12 Session Resumption dan Rekeying

Sebuah sesi komunikasi dalam SSL maupun IPSec yang terputus dapat dilanjutkan kembali menggunakan teknik *Session Resumption and Rekeying*. Setiap layer yang terlibat dalam komunikasi menggunakan SSL atau IPSec dapat mempengaruhi penyambungan sesi. Pada teknik SSL, penggunaan saluran tergantung pada aplikasi yang membutuhkan layanan SSL. Jika aplikasi sudah selesai digunakan maka saluran yang menggunakan SSL akan dihilangkan.

Sebuah sesi SSL dapat disambung kembali jika sesi tersebut belum mencapai waktu kadaluarsa. Jika hal ini dilakukan maka client dan server harus mempertukarkan identitas sesi (*session ID)* dari sesi yang akan disambung kembali tersebut. *Session ID* tersebut akan digunakan untuk mengenali *pre-master-key* yang akan digunakan untuk menghasilkan *session key* baru. Tabel 8 menunjukkan waktu yang diperlukan untuk melanjutkan kembali sebuah sesi SSL.

Tabel 8. Waktu yang diperlukan untuk melanjutkan kembali sebuah sesi SSL

Mode	Waktu
Autentikasi server	1,3 mili detik
Autentikasi client	
Autentikasi server (Diffie-Helman)	
Autentikasi client (Diffie-Helman)	

Session resumption dalam IPSec memiliki konsep yang berbeda yang disebut sebagai *rekeying.* Teknik rekeying ini menjadi lebih rumit karena IPSec tidak terikat pada sebuah aplikasi dan memiliki dua fase yang berbeda. Dua konsep yang digunakan dalam teknik rekeying adalah:

- *Continous channel*
  Jika ISAKMP SA mencapai wakti kadaluarsa-nya, maka IPSec SA harus segera dihapus. Hal ini dikarenakan bahwa ISAKMP SA bertanggung jawab terhadap pertukaran pesan informal seperti notifikasi penghapusan dan *Dead Peer Detection* [6].
- *Dangling SA*
  Walaupun ISAKMP SA telah kadaluarsa, IPSec SA tetap valid hingga mencapai waktu validitas karena ISAKMP SA telah menyelesaikan proses autentikasi-nya.

Sehingga proses rekeying ini tergantung pada status ISAKMP SA. Jika ISAKMP SA kadaluarsa sebelum IPSec SA maka:

- dalam Continous Channel IPSec dihapus dan negosiasi Phase 1 dan Phase 2 dijalankan
- dalam Dangling SA, pada saat IPSec SA perlu proses rekeying maka negosiasi Phase 1 dan Phase 2 dijalankan

Jika IPSec SA kadaluarsa sebelum ISAKMP dan perlu proses rekeying dalam Continous Channel atau Dangling SA maka hanya Phase 2 yang dijalankan. Tabel 9 menunjukkan waktu yang diperlukan dalam proses rekeying.

Tabel 9. Waktu yang diperlukan dalam proses rekeying

Mode	Waktu
Main mode (PSK)	26 mili detik
Aggressive mode (PSK)	
Main mode (PSK)	

4.13 NAT Traversal

Seperti disebutkan di dalam paragraf di atas bahwa SSL tidak terikat pada port tertentu sehingga keberadaan NAT diantara client dan server tidak akan mempengaruhi jalannya komunikas data. Sedangkan client IPSec terikat pada port tertentu, sehingga NAT dan NAPT dapat menimbulkan masalah pada penggunaan IPSec. Namun solusi untuk masalaha ini telah diatasi oleh penelitian [17].

4.14 Algoritma Kompresi

Kompresi pada IPSec menggunakan protokol yang disebut IPComp. Namun kompresi tidak banyak digunakan pada teknik SSL. Hanya OpenSSL yang dapat melakukan kompresi.

4.15 Kinerja

Kinerja kedua sistem diamati pada percobaan menggunakan dua mesin dengan spesifikasi sebagai berikut:

- sistem operasi menggunakan Ubuntu 9.10
- prosesor Intel Core 2 Duo RAM 1G
- NIC 100 Mbps dan 1000 Mbps
- FreeSWAN
- Stunnel

- Ethereal
- Iperf

4.15.1 IPSec ESP-SHA-1

Tabel berikut menunjukkan nilai throughput pada jaringan 1000Mbps. Konsumsi CPU bervariasi antara 94% sampai 97%. Algoritma 3DES merupakan algoritma yang paling banyak menggunakan CPU.

Tabel 10. Jaringan 1000 Mbps (IPSec)

Algoritma	***Throughput* (Mbps)**	
	Tanpa kompresi	**Dengan kompresi**
Tanpa algoritma	427	N/A
DES	110	105
3DES	69,5	99,4
AES-128	156	104
BLOWFISH	123,5	105

Tabel 11. Jaringan 100 Mbps (IPSec)

Algoritma	***Throughput* (Mbps)**	
	Tanpa kompresi	**Dengan kompresi**
Tanpa algoritma	93,6	N/A
DES	89,3	104
3DES	70,7	101
AES-128	88,6	111

4.15.2 IPSec ESP-MD5

Tabel berikut menunjukkan throughput pada jaringan 1000 Mbps. Konsumsi CPU bervariasi antara 87% sampai dengan 93%.

Tabel 12. Jaringan 1000 Mbps (IPSec)

Algoritma	**Throughput (Mbps)**	
	Tanpa Kompresi	**Dengan Kompresi**
Tanpa algoritma	427	N/A
DES	137	113
3DES	75	107
AES-128	198	114
BLOWFISH	148	113

Tabel 13. Jaringan 100 Mbps (IPSec)

Algoritma	**Throughput (Mbps)**	
	Tanpa Kompresi	**Dengan Kompresi**
Tanpa algoritma	93,6	N/A
DES	89,8	122
3DES	78,8	115
AES-128	88,9	123

4.15.3 SSL

Tabel 14 merupakan hasil percobaan pengiriman data menggunakan SSL pada jaringan 1000 Mbps dengan konsumsi CPU antara 86% sampai dengan 93%.

Tabel 14. Jaringan 1000 Mbps (SSL)

Algoritma	Throughput (Mbps)
Tanpa algoritma	427
3DES-EDE-CBC-SHA	86
DES-CBC-SHA	152
RC4-128-SHA	219
RC4-128-MD5	246
EXP-RC2-CBC-MD5	216

4.15.4 Kecepatan Transfer Data 100MB

Tabel 15 menunjukkan hasil transfer data 100MB menggunakan berbagai macam algoritma. Hasil percobaan transfer data melalui jaringan 100Mbps memperlihatkan bahwa semua algoritma mengalami pununrunan kecepatan kecuali 3DES.

Tabel 15. Jaringan 100 Mbps (SSL)

Algoritma	Throughput
Tanpa algoritma	93,6 Mbps
3DES-EDE-CBC-SHA	75 Mbps
DES-CBC-SHA	90,3 Mbps
RC4-128-SHA	87,6 Mbps
RC4-128-MD5	90,2 Mbps
EXP-RC2-CBC-MD5	63,5 Mbps

5 Kesimpulan

Masing-masing protokol memiliki fitur unik. Pemilihan IPSec atau SSL tergantung pada kebutuhan keamanan data yang diperlukan. IPSec sangat cocok digunakan untuk komunikasi data antar gateway. SSL dapat bekerja dibelakang firewall dengan sangat baik jika dibandingkan dengan IPSec. Dalam implementasi IPSec, client perlu aplikasi IPSec khusus untuk remote access. Penggunaan kompresi data dalam jaringan dengan bandwidth rendah sangat menguntungkan dan hal ini terdapat dalam IPSec. IPSec memiliki kemampuan untuk memproteksi jaringan wireless.

6 Daftar Pustaka

Frankel, Sheila., "*Demystifying The IPSec Puzzle*", Artec House Publisher, 2001.

Rescorla, Eric., "*SSL and TLS, Designing and Building Secure Systems*", Addison-Wesley, Agustus 2001.

Atkinson., "*IP Encapsulation Security Payload (ESP)*", RFC 2406, November 1998.

M., Maughan, M., Schertler, "*Internet Security Association dan Key Management Protocol (ISAKMP)*", RFC 2408, November 1998.

Harkins, D., Carrel, D., "*The Internet Key Exchange (IKE)*", RFC 2409, November 1998.

Thomas, M., Vilhuber, J., "Kerberized Internet Negotiation of Keys (KINK)", Internet Draft., Januari 2003.

Madson, C., Glenn, R., "The Use of HMAC-SHA-96 within ESP and AH", RFC 2404, November 1998.

Sheffer, Y., Krawczyk, H., Aboba, Bernard., "PIC, A Pre-IKE Credential Provisioning Protocol", Internet Draft, Oktober 2002.

Kaufman, Charlie., "Internet Key Exchange (IKEv2) Protocol", Internet Draft, Maret 2004