Pengamanan Aplikasi Web Menggunakan Protokol Secure Socket Layer

Farah Rosaria[1], Haris Yuniarsa[2] dan Fahmi [3]

Departemen Teknik Informatika
Institut Teknologi Bandung
Jalan Ganesha 10 Bandung 40132

E-mail : if11069@students.if.itb.ac.id[1], if12024@students.if.itb.ac.id[2], if12047@students.if.itb.ac.id[3]

Abstrak

Semakin majunya teknologi menuntut perkembangan di semua bidang. Salah satunya adalah teknologi internet, pemanfaatannya untuk mempermudah komunikasi tidak terelakkan. Saat ini, komunikasi dan transaksi antar dua pihak atau lebih yang saling berjauhan banyak dibantu oleh teknologi internet, terutama web. Pesan-pesan yang saling dipertukarkan dalam komunikasi tersebut seringkali mengandung informasi yang penting dan rahasia. Oleh karena itu, dibutuhkan suatu mekanisme untuk menjaga kerahasiaan dan keamanan data selama transaksi berlangsung. Salah satu cara yang dapat dilakukan adalah dengan menggunakan protokol komunikasi yang aman. Dalam makalah ini akan dibahas tentang protokol Secure Socket Layer atau disingkat SSL, merupakan protokol yang mempunyai lapisan tersendiri dan terpisah dari lapisan protokol lain. Penggunaan SSL cukup luas karena tidak hanya mendukung keamanan pada aplikasi Hypertext Transfer Protocol (HTTP), tetapi juga aplikasi internet lain seperti Net News Transfer Protocol (NNTP) dan File Transfer Protocol (FTP) walaupun pembangunan SSL lebih banyak berorientasi pada keamanan web.

***Kata kunci**: SSL, Secure Socket Layer, protokol, kriptografi, kunci privat, kunci publik, web, transaksi, komunikasi*

1. Pendahuluan

Saat ini, penggunaan web semakin berkembang dengan pesat. Transaksi web memungkinkan orang yang saling berjauhan untuk berinteraksi satu sama lain. Sebagai contoh, seorang pengguna yang berada di Berlin, Jerman dapat melakukan transaksi online melalui website ke San Jose, California.

Dalam transaksi jarah jauh tersebut tentunya terjadi pengiriman data berisi informasi penting dari pengguna, seperti nomor kartu kredit. Data tersebut akan menjalani jalur yang kompleks melalui banyak negara, jaringan dan fasilitas-fasilitas yang berbeda dimana terdapat kemungkinan tidak adanya aturan atau hukum yang menjamin kerahasiaan informasi yang dibawa.

Pengguna dan website itu sendiri tidak mempunyai kontrol terhadap jalur pesan atau memeriksa siapa saja yang melihat isi pesan sepanjang rute perjalanannya. Penyadap bukan satu-satunya ancaman terhadap pengguna web. Secara teoritis mungkin dilakukan pengalihan transmisi pesan ke website palsu. Website palsu ini dapat memberikan informasi yang salah,

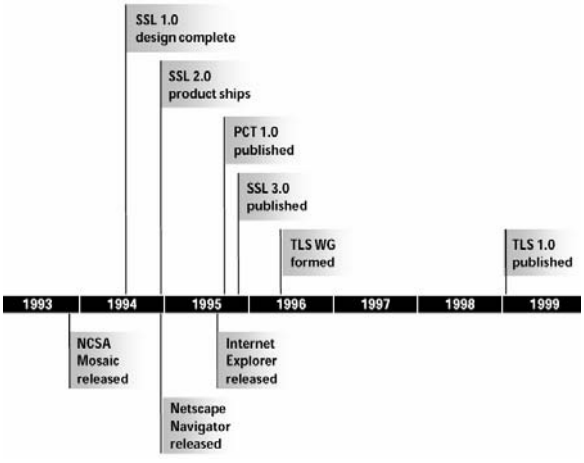
mengumpulkan data seperti nomor kartu kredit atau melakukan kejahatan-kejahatan lainnya.

Oleh karena itu, internet membutuhkan cara agar pengguna dapat memeriksa kebenaran identitas website, sebaliknya website juga melakukan verifikasi identitas pengguna. Hal yang tidak kalah penting pada lapisan pengguna website adalah integritas pesan. Salah satu cara untuk mewujudkan hal tersebut adalah dengan menggunakan *socket secure layer* (SSL).

2. Sejarah

2.1 Perkembangan SSL

Pada awal perkembangan web, masalah keamanan telah menjadi perhatian khusus. National Center for Supercomputing Application (NCSA) merilis browser web Mosaic 1.0 yang cukup terkenal pada November 1993. Delapan bulan kemudian, Netscape Communications melengkapi rancangan SSL versi 1.0 dan diikuti versi-versi selanjutnya dari SSL. Perkembangan SSL dapat dilihat pada Gambar 1



Gambar 1 Perkembangan SSL

SSL dibangun sedemikian sehingga dapat digunakan untuk hampir semua browser dan server. SSL bertugas memberikan kerahasiaan, otentikasi dan integritas pesan secara aman ke pengguna web. Adanya SSL dapat diamati dari URL ”https” yang merupakan *SSL-secured URL* atau dapat diamati dari ikon gembok kecil pada bagian kanan bawah tampilan windows browser.

Meskipun demikian, untuk mendukung browsing web secara aman, server web tidak cukup hanya dengan menggunakan SSL. Browser web juga harus memperoleh sertifikat kunci publik dari organisasi yang dipercaya. Untuk pengguna pada internet publik, organisasi itu biasanya berkuasa atas sertifikat publik. Beberapa pemilik sertifikat yang terkemuka adalah AT&T Certificate Services, GTE CyberTrust, KeyWitness International, Microsoft, Thawte Consulting dan VeriSign.

2.2 Pendekatan Protokol Terpisah

Arsitektur pada internet berupa lapisan-lapisan protokol yang masing-masing membangun layanan-layanan dibawahnya, termasuk layanan keamanan. Perancang SSL memilih untuk membuat lapisan protokol yang seluruhnya baru secara terpisah. Alternatif yang lain antara lain dengan memasukkan layanan keamanan pada protokol aplikasi atau menambahkan pada protokol jaringan inti. Perbandingan untuk tiap pendekatan keamanan jaringan dapat dilihat pada Tabel 1

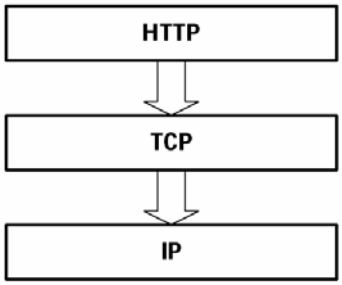
Tabel 1 Perbandingan Arsitektur Jaringan

Arsitektur Protokol	Contoh	Keuntungan				
		A	B	C	D	E
Lapisan Protokol Terpisah	SSL	√	√			√
Lapisan Aplikasi	S-HTTP	√		√		√

Terintegrasi dengan Protokol Inti	IPSEC	√	√		√	
Protokol Paralel	Kerberos		√			√

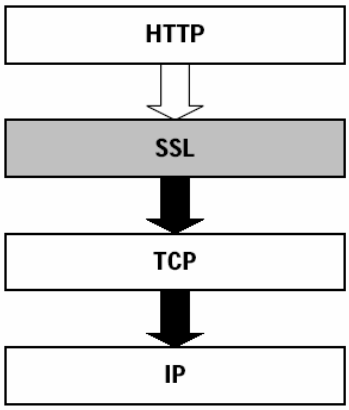
Keterangan:
A. Keamanan Penuh (Full Security)
B. Multi Aplikasi (Multiple Applications)
C. Layanan yang Disesuaikan (Tailored Services)
D. Transparan terhadap Aplikasi (Transparent to Application)
E. Mudah di-*deploy* (Easy to Deploy)

Karena perancang SSL memutuskan untuk membuat protokol terpisah untuk keamanan, maka arsitektur protokol internet harus ditambahi dengan sebuah lapisan SSL. Protokol kunci untuk komunikasi web terdiri atas Internet Protocol (IP), Transmission Control Protocol (TCP) dan Hypertext Transfer Protocol (HTTP). IP bertanggung jawab pada routing pesan melalui jaringan dari sumber ke tujuan. TCP membangun layanan IP untuk menjamin bahwa komunikasi yang terjadi adalah komunikasi yang handal. HTTP mengetahui detil interaksi antara server web dan browser web. Arsitektur protokol kunci tersebut dapat dilihat pada Gambar 2.



Gambar 2 Arsitektur Protokol Komunikasi Web tanpa SSL

Pada arsitektur baru, SSL bertindak sebagai protokol keamanan terpisah, berada di antara aplikasi HTTP dan TCP yang menghendaki perubahan sesedikit mungkin dari protokol di atas dan di bawahnya. Arsitektur protokol setelah ditambahkan SSL dapat dilihat pada Gambar 3. Antarmuka aplikasi HTTP dengan SSL hampir sama dengan TCP tanpa pengamanan. SSL hanya seperti aplikasi lain yang menggunakan layanan TCP. Sebagai tambahan untuk meminta perubahan minimal dari implementasi yang telah ada, pendekatan ini memiliki keuntungan lain, yaitu memungkinkan SSL mendukung aplikasi selain HTTP. SSL juga dapat digunakan untuk pengamanan NTTP dan FTP.



Gambar 3 Arsitektur Protokol Komunikasi dengan SSL

3. Socket Secure Layer

SSL mempunyai beberapa pilihan dan variasi. Protokol ini terdiri atas sekumpulan pesan dan aturan tentang kapan waktu untuk saling berkirim pesan dan kapan tidak.

3.1 Peran pada SSL

SSL mendefinisikan dua peran berbeda untuk komunikasi, yaitu client dan server. Client adalah sistem yang menginisiasi

komunikasi. Sedangkan server adalah sistem yang merespon request dari client. Dalam SSL, browser web adalah client dan website adalah server.

Client dan server mempunyai perbedaan utama pada aksi yang dilakukan ketika melakukan negosiasi tentang parameter keamanan. Client bertugas untuk mengajukan opsi SSL yang akan digunakan pada saat pertukaran pesan, dan server menentukan opsi mana yang akan digunakan.

3.2 Pesan SSL

Client dan server berkomunikasi dengan bertukar pesan. Secara teknis, SSL mendefinisikan beberapa level pesan yang dapat dilihat pada Tabel 2

Tabel 2 Beberapa Level Pesan pada SSL

Message	Description
Alert	Memberitahu pihak lain dalam komunikasi tentang kemungkinan adanya penerobosan pada keamanan atau kegagalan komunikasi
ApplicationData	Informasi aktual bahwa kedua pihak saling bertukar pesan yang dienkripsi, diotentikasi dan/atau diverifikasi oleh SSL
Certificate	Pesan yang membawa sertifikat kunci publik pengirim
CertificateRequest	Request dari server kepada client untuk memberikan sertifikat kunci publik
CertificateVerify	Pesan dari client berisi verifikasi bahwa client mengetahui kunci privat yang bersesuaian dengan sertifikat kunci publik client
ChangeCipherSpec	Tanda untuk mulai munggunakan layanan keamanan yang telah disetujui sebelumnya, seperti enkripsi
ClientHello	Pesan dari client yang memberi informasi layanan keamanan yang diinginkan dan dapat didukung oleh client
ClientKeyExchange	Pesan dari client yang membawa kunci kriptografi untuk komunikasi
Finished	Indikasi bahwa semua negosiasi awal sudah selesai dan komunikasi yang aman telah dibangun
HelloRequest	Pesan dari server kepada client untuk memulai atau restart proses negosiasi SSL
ServerHello	Pesan dari server yang menunjukkan layanan keamanan yang akan digunakan untuk komunikasi
ServerHelloDone	Indikasi dari server bahwa server telah melengkapi semua request ke client yang diperlukan untuk membangun komunikasi
ServerKeyExchage	Pesan dari server yang membawa kunci kriptografi untuk komunikasi

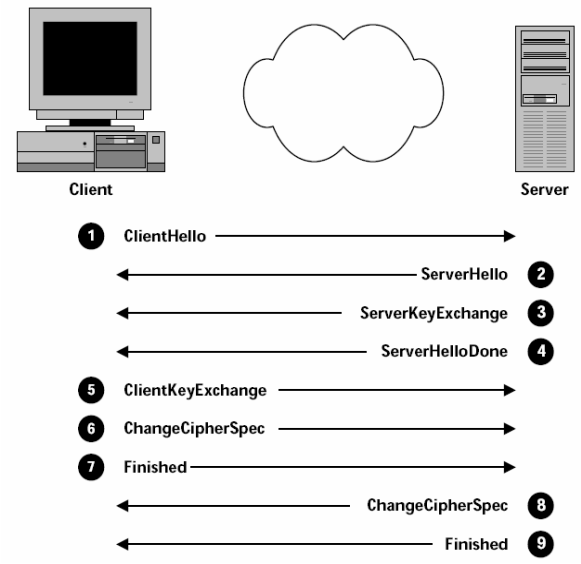
3.3 Membangun Komunikasi yang Dienkripsi

Membangun channel untuk komunikasi terenkripsi adalah fungsi dasar dari client dan server SSL. Pertukaran pesan pada operasi ini dapat dilihat pada Gambar 4 sedangkan langkah-langkah pembangunan channel dapat dilihat pada Tabel 3

Tabel 3 Langkah-Langkah Pembangunan Channel

Langkah	Aksi
1	Client mengirimkan pesan ClientHello untuk mengajukan opsi SSL
2	Server merespon dengan memilih opsi SSL menggunakan pesan ServerHello
3	Server mengirimkan informasi kunci publiknya dengan pesan ServerKeyExchange
4	Server mengakhiri bagian negosiasi dengan pesan ServerHelloDone
5	Clinet mengirimkan informasi kunci sesi yang dienkripsi dengan kunci publik server pada pesan ClientKeyExchange
6	Client mengirimkan pesan ChangeCipherSpec untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
7	Client mengirimkan pesan Finished sehingga server dapat mengecek opsi baru yang diaktifkan

8	Server mengirimkan pesan ChangeCipherSpec untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
9	Server mengirimkan pesan Finished sehingga client dapat mengecek opsi baru yang diaktifkan



Gambar 4 Pertukaran Pesan pada Pembangunan Komunikasi Terenkripsi

3.3.1 ClientHello

Pesan ClientHello memulai komunikasi antara dua pihak, client menggunaan pesan ini untuk meminta server memulai negosiasi layanan keamanan menggunakan komponen pesan ClientHello yang dapat dilihat pada Tabel 4

Tabel 4 Komponen Pesan ClientHello

Field	Kegunaan
Version	Mengidentifikasi versi tertinggi dari protokol SSL yang dapat didukung oleh client
RandomNumber	Bilangan random 32-byte yang digunakan untuk kalkulasi kriptografi
SessionID	Mengidentifikasi sesi khusus SSL
CipherSuites	Daftar parameter kriptografik yang dapat didukung oleh client
CompressionMethod	Mengidentifikasi metode kompresi data yang dapat didukung oleh client

3.3.2 ServerHello

Pesan ServerHello dikirim oleh server sebagai respon dari pesan ClientHello. Isi dari ServerHello hampir sama dengan ServerHello. Perbedaannya secara umum adalah pesan ClientHello digunakan client untuk membuat anjuran dan saran sedangkan pesan ServerHello digunakan server untuk membuat keputusan akhir. Komponen ServerHello dapat dilihat pada Tabel 5

Tabel 5 Komponen Pesan ServerHello

Field	Kegunaan
Version	Mengidentifikasi versi protokol SSL yang akan digunakan untuk komunikasi
RandomNumber	Bilangan random 32-byte yang digunakan untuk kalkulasi kriptografi
SessionID	Mengidentifikasi sesi khusus SSL
CipherSuite	Parameter kriptografik yang akan digunakan untuk komunikasi
CompressionMethod	Metode kompresi data yang akan digunakan untuk komunikasi

3.3.3 ServerKeyExchange

Pesan ini melengkapi field CipherSuite pada ServerHello. Pada waktu CipherSuite menunjukkan algoritma kriptografi dan ukuran kunci, pesan ServerKeyExchange berisi informasi kunci publik server dengan format kunci bergantung pada algoritma kunci publik yang digunakan. Pesan ini

ditransmisikan tanpa enkripsi sehingga hanya informasi kunci publik yang dapat disertakan. Client akan menggunakan kunci publik server untuk mengenkripsi kunci sesi yang nantinya akan digunakan untuk mengenkripsi data aplikasi untuk sesi tersebut.

3.3.4 ServerHelloDone

Pesan ini memberitahu client pada waktu server telah selesai dengan pesan negosiasi pertamanya. Walaupun pesan tidak mengandung informasi lain, namun sangat penting untuk client karena menandakan dapat berpindahnya client ke tahap selanjutnya dalam pembangunan komunikasi yang aman.

3.3.5 ClientKeyExchange

Pesan dari client untuk merespon negosiasi SSL awal yang dilakukan server. ClientKeyExchange memberikan informasi kunci client kepada server. Informasi kunci tersebut digunakan untuk algoritma enkripsi simetri yang akan digunakan pada sesi tersebut. Informasi ini dienkripsi menggunakan kunci publik server. Enkripsi ditujukan untuk melindungi informasi kunci yang berpindah melalui jaringan dan memungkinkan client memverifikasi bahwa kunci privat yang bersesuaian dengan kunci publik client adalah kepunyaan server. Sebaliknya, server tidak dapat mendekripsi pesan. Hal ini untuk melindungi dari penyerang yang menangkap pesan dari server asli dan berpura-pura sebagai server dengan meneruskan pesan ke client yang lain.

3.3.6 ChangeCipherSpec

Setelah client mengirimkan informasi kunci dengan pesan ClientKeyExchange, negosiasi awal SSL telah selesai. Hal ini menunjukkan bahwa kedua pihak dapat mulai menggunakan layanan keamanan. Protokol SSL mendefinisikan pesan ChangeCipherSpec sebagai identifikasi eksplisit bahwa layanan keamanan sudah dapat digunakan.

Spesifikasi SSL menggambarkan proses dengan sangat tepat karena perpindahan ke komunikasi yang aman sangat penting. Diawali dengan identifikasi sekumpulan informasi berisi algoritma enkripsi simetri, algoritma integritas pesan dan kunci untuk kedua algoritma tersebut yang mendefinisikan layanan keamanan. Spesifikasi SSL juga memahami bahwa informasi tersebut dapat berbeda untuk arah komunikasi yang berbeda, satu set kunci akan mengamankan data yang dikirimkan dari client ke server dan satu set kunci yang lain akan mengamankan data yang dikirimkan dari server ke client.

SSL mendefinisikan *write state* dan *read state*, baik pada sistem client maupun sistem server. *Write state* mendefinisikan informasi keamanan untuk data yang dikirim oleh sistem, *read state* mendefinisikan informasi untuk data yang diterima oleh sistem.

Pesan ChangeCipherSpec bertindak sebagai isyarat untuk sistem mulai menggunakan informasi keamanan. Client dan server harus mengetahui informasi keamanan yang akan diaktifkan secara lengkap sebelum mengirimkan pesan ini. Segera setelah sistem mengirimkan pesan ini, maka sistem akan mengaktifkan *write state*. Demikian halnya jika sistem menerima pesan ini, maka sistem akan mengaktifkan *read state*.

Dalam proses ini, terjadi perubahan pada sisi client dan sisi server. *Write state* dan

read state dari tiap-tiap sistem dinyatakan dengan matriks yang terpisah. Dalam satu sistem terdapat kondisi *write* dan *read* yang jika salah satu aktif maka yang lain *pending*. Oleh karena itu, client dan server masing-masing mengelola 4 kondisi, yaitu *active write state*, *pending write state*, *active read state* dan *pending read state*. Sebuah *state* mempunyai elemen kunci yang terdiri atas algoritma enkripsi, algoritma integritas pesan (Message Authentication Code) dan kunci. *Active write state* pada satu sistem akan sama dengan *active read state* pada sistem yang lain. Pada waktu sistem mengirimkan pesan ChangeCipherSpec, sistem akan melakukan *update* pada *active state* dan sistem lainnya akan melakukan perubahan *active state* jika telah menerima pesan tersebut. Selama proses ini berlangsung, kedua sistem tidak dalam keadaan sinkron satu sama lain.

3.3.7 Finished

Sistem akan mengirimkan pesan Finished setelah mengirimkan pesan ChangeCipherSpec yang memungkinkan kemua sistem memverifikasi bahwa negosiasi telah berhasil dan keamanan belum disetujui. Dua aspek dari pesan Finished mempunyai kontribusi pada keamanan ini. Pesan Finished menunjuk pada *cipher suite* yang dinegosiasi sehingga pesan ini dienkripsi dan diotentikasi berdasar *suite* tersebut. Jika penerima tidak dapat melakukan dekripsi dan verifikasi pesan, maka telah jelas terjadi kesalahan pada negosiasi keamanan. Pesan Finished juga bertindak untuk melindungi keamanan negosiasi SSL. Tiap-tiap pesan ini berisi hash kriptografi dari informasi penting tentang negosiasi yang baru saja diselesaikan. Detil informasi yang dilindungi oleh hash tersebut adalah sebagai berikut:

- Informasi kunci
- Isi dari semua pesan *handshake* SSL sebelumnya yang dipertukarkan oleh sistem
- Sebuah nilai spesial yang memberi indikasi apakah pengirim pesan client atau server

3.4 Mengakhiri Komunikasi yang Aman

SSL mempunyai prosedur untuk mengakhiri komunikasi dengan tiap-tiap sistem saling mengirimkan ClosureAlert. Penutupan sesi secara eksplisit akan melindungi dari serangan pemepatan (*truncation attack*) dimana penyerang dapat menghentikan komunikasi sebelum waktunya. Pesan ClosureAlert menyatakan bahwa pesan yang diterima telah lengkap.

3.5 Otentikasi Identitas Server

Otentikasi ditujukan untuk menghindari adanya pihak ketiga yang tidak seharusnya ikut serta dalam komunikasi. Langkah otentikasi dapat dilihat pada Tabel 6

Tabel 6 Langkah-langkah Otentikasi Server

Langkah	Aksi
1	Client mengirimkan pesan ClientHello untuk mengajukan opsi SSL
2	Server memberi respon dengan memilih opsi SSL melalui ServerHello
3	Server mengirimkan sertifikat kunci publik pada pesan Certificate
4	Server mengakhiri bagian negosiasi dengan pesan ServerHelloDone
5	Client mengirimkan informasi kunci sesi yang dienkripsi dengan kunci publik server melalui pesan ClientKeyExchange
6	Client mengirimkan pesan ChangeCipher-Spec untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
7	Client mengirimkan pesan Finished sehingga memungkinkan server mengecek opsi baru yang diaktifkan
8	Server mengirimkan pesan ChangeCipher-

	Spec untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
9	Server mengirimkan pesan Finished sehingga memungkinkan client mengecek opsi baru yang diaktifkan

3.5.1 Certificate

Server mengotentikasi identitasnya dengan mengirimkan pesan Certificate yang berisi rangkaian dimulai dengan sertifikat kunci publik server dan berakhir dengan sertifikat akar dari pemilik sertifikat.Client mengecek sertifikat yang diterima dengan melakukan verifikasi tanda tangan, masa berlaku dan status pembatalan pada sertifikat. Termasuk meyakinkan bahwa pemilik sertifikat adalah yang dipercaya oleh client. Biasanya hal ini dilakukan dengan mengetahui kunci publik dari pemilik sertifikat yang dipercaya tersebut.

Selain itu, client harus yakin bahwa sertifikat mengidentifikasi secara jelas teman komunikasi yang diinginkan untuk menghindari pihak yang menerima sertifikat dari client berperan sebagai orang lain. Pada web commerce, kunci pemecahan masalah ini terletak pada nama domain server. Sertifikat yang diakui menyertakan nama domain internet dari server web, kemudian browser web mengecek kesesuaian nama domain pada sertifikat yang diterima dengan nama domain yang ingin dikontak oleh pengguna.

3.5.2 ClientKeyExchange

Client mengenkripsi informasi pada ClientKeyExchange menggunakan kunci publik yang disediakan server pada pesan ServerKeyExchange. Langkah ini penting agar client dapat menjamin bahwa pihak yang berkomunikasi dengannya adalah benar-benar pemilik kunci privat server. Hanya sistem yang mempunyai kunci privat asli yang dapat mendekripsi pesan dan meneruskan komunikasi.

3.6 Memisahkan Enkripsi dari Otentikasi

Pada proses otentikasi yang dijelaskan pada bagian sebelumnya, kunci publik yang digunakan untuk verifikasi identitas server juga digunakan untuk mengenkripsi kunci pada pesan ClientKeyExchange. Hal ini seringkali tidak dapat dipenuhi. Oleh karena itu, dilakukan pemisahan antara enkripsi dan tanda tangan digital. Langkah pemisahan otentikasi server dan enkripsi dapat dilihat pada Tabel 7

Tabel 7 Langkah-langkah Pemisahan Otentikasi Server dan Enkripsi

Langkah	Aksi
1	Client mengirimkan pesan ClientHello untuk mengajukan opsi SSL
2	Server memberi respon dengan memilih opsi SSL melalui ServerHello
3	Server mengirimkan sertifikat kunci publik pada pesan Certificate
4	Server mengirimkan kunci publik yang harus digunakan oleh client untuk mengenkripsi kunci simetri pada ServerKeyExchange, kunci ini terdapat pada sertifikat server
5	Server mengakhiri bagian negosiasi dengan pesan ServerHelloDone
6	Client mengirimkan informasi kunci sesi pada pesan ClientKeyExchange (dienkripsi dengan kunci publik yang disediakan oleh server)
7	Client mengirimkan pesan ChangeCipher-Spec untuk mengaktifkan opsi yang dinegosiasi untuk semua pesan yang akan dikirimkan
8	Client mengirimkan pesan Finished sehingga memungkinkan server mengecek opsi baru yang diaktifkan
9	Server mengirimkan pesan ChangeCipher-Spec untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
10	Server mengirimkan pesan Finished yang

	memungkinkan client mengecek opsi baru yang diaktifkan

3.6.1 Certificate

Pesan Certificate ini mirip dengan uraian pada subbab 3.5, bedanya kunci publik pada sertifikat server hanya digunakan untuk verifikasi identitas server. Client mempunyai tugas untuk verifikasi tanda tangan, masa berlaku dan status pembatalan sertifikat serta harus mengecek bahwa pemilik sertifikat adalah pihak yang dipercaya dan sertifikat dikeluarkan kepada pihak yang diinginkan.

3.6.2 ServerKeyExchange

Pesan ServerKeyExchange mengikuti pesan Certificate, berisi kunci publik yang harus digunakan client untuk mengenkripsi informasi kunci sesi. Isi dari pesan ini mirip dengan subbab 3.3.3, bedanya informasi kunci diperoleh dengan menggunakan kunci publik pada sertifikat server. Langkah penting ini memberi client cara untuk memverifikasi bahwa server adalah benar-benar pemilik kunci privat yang bersesuaian dengan sertifikat kunci publik client.

3.6.3 ClientKeyExchange

Client menggunakan pesan ClientKey-Exchange untuk menyelesaikan proses negosiasi. Pesan ini berisi informasi kunci untuk algoritma enkripsi simetri yang digunakan. Informasi tersebut dienkripsi menggunakan kunci publik server dari pesan ServerKeyExchange, bukan dari Certificate.

3.7 Otentikasi Identitas Client

Proses otentikasi Client mempunyai mekanisme yang mirip dengan otentikasi server. Langkah yang dilalui dapat dilihat pada Tabel 8

Tabel 8 Langkah-langkah Otentikasi Client

Langkah	Aksi
1	Client mengirimkan pesan ClientHello untuk mengajukan opsi SSL
2	Server memberi respon dengan memilih opsi SSL melalui ServerHello
3	Server mengirimkan sertifikat kunci publik pada pesan Certificate
4	Server mengirimkan pesan CertificateRequest untuk menunjukkan bahwa server ingin mengotentikasi client
5	Server mengakhiri bagian negosiasi dengan pesan ServerHelloDone
6	Client mengirimkan sertifikat kunci publik pada pesan Certificate
7	Client mengirimkan informasi kunci sesi pada pesan ClientKeyExchange (dienkripsi dengan kunci publik server)
8	Client mengirimkan pesan Certificate-Verify yang menandai informasi penting tentang sesi menggunakan kunci privat client, server menggunakan kunci publik dari sertifikat client untuk memverifikasi identitas client
9	Client mengirimkan pesan ChangeCipher-Spec untuk mengaktifkan opsi yang dinegosiasikan untuk semua pesan yang akan dikirimkan
10	Client mengirimkan pesan Finished sehingga memungkinkan server mengecek opsi baru yang diaktifkan
11	Server mengirimkan pesan ChangeCipher-Spec untuk mengaktifkan opsi yang dinegosiasi untuk semua pesan yang akan dikirimkan
12	Server mengirimkan pesan Finished yang memungkinkan client mengecek opsi baru yang diaktifkan

3.7.1 CertificateRequest

Pada pertukaran pesan di SSL, server akan menentukan apakah otentikasi client diperlukan. Sedangkan client tidak mempunyai kendali dalam hal ini dan hanya mengikuti server. Server yang menginginkan otentikasi client akan mengirimkan pesan CertificateRequest sebagai bagian dalam negosiasi hello. Server mengirimkan CertificateRequest setelah pesan Certificate

dan juga mengikuti ServerKeyExchange yang dikirimkan server. Spesifikasi SSL melarang server mengirim CertificateRequest jika server tidak mengotentikasi dirinya. Aturan ini untuk menjamin client mengetahui identitas server sebelum membuka identitas dirinya. Komponen CertificateRequest dapat dilihat pada Tabel 9

Tabel 9 Komponen Pesan CertificateRequest

Field	Kegunaan
CertificateTypes	Daftar tipe sertifikat yang dapat diterima oleh server
DistinguishedNames	Daftar nama-nama berbeda dari pemilik sertifikat yang dapat diterima oleh server

Tipe sertifikat disusun berurut dari yang paling disukai dan dibedakan berdasar algoritma tanda tangan yang digunakan.

3.7.2 Certificate

Client memberi respon permintaan sertifikat dengan mengirimkan pesan Certificate segera setelah menerima ServerHelloDone. Format pesan Certificate client identik dengan Certificate server. Jika client tidak memiliki sertifikat yang memenuhi kriteria server atau bahkan tidak mempunyai sertifikat sama sekali, client memberi respon dengan NoCertificateAlert. Kemudian server dapat memilih untuk mengabaikan alert dan melanjutkan komunikasi atau menghentikan sesi.

3.7.3 CertificateVerify

Proses otentikasi client tidak dapat diselesaikan hanya dengan mengirim pesan Certificate. Client juga harus membuktikan kepemilikan kunci privat yang bersesuaian dengan kunci publik sertifikat menggunakan pesan CertificateVerify. Pesan ini berisi hash kriptografi dari informasi yang tersedia pada client dan server yang telah ditandatangani secara digital. Daftar informasi yang ditandai oleh client adalah sebagai berikut:

- Informasi kunci
- Isi dari semua pesan *SSL handshake* sebelumnya yang dipertukarkan oleh sistem

Server juga memiliki informasi ini dan akan menerima kunci publik client kemudian server akan memverifikasi tanda tangan dan mengetahui apakah client adalah benar-benar pemilik kunci privat yang sesuai.

3.8 Melanjutkan Sesi Sebelumnya

SSL mendefinisikan mekanisme ini untuk meminimalisir kalkulasi dan pesan dengan menggunakan kembali parameter SSL yang telah dinegosiasi sebelumnya. Mekanisme ini menghemat tenaga dan waktu karena mendirikan sesi SSL merupakan pekerjaan kompleks yang membutuhkan kalkulasi kriptografi yang rumit dan sejumlah besar pesan protokol. Dengan metode ini, pihak yang berkomunikasi tidak perlu memulai lagi negosiasi atau otentikasi kriptografi, cukup dengan melanjutkan langkah yang telah dilakukan sebelumnya. Langkah-langkah untuk mekanisme ini dapat dilihat pada Tabel 10

Tabel 10 Langkah-langkah untuk Melanjutkan Sesi

Langkah	Aksi
1	Client mengirimkan pesan ClientHello yang menetapkan ID sesi sebelumnya
2	Server memberi respon dengan ServerHello untuk menyetujui ID sesi
3	Server mengirimkan pesan ChangeCipher-Spec untuk mengaktifkan kembali opsi pengamanan sesi untuk pesan yang akan dikirim
4	Server mengirimkan pesan Finished yang memungkinkan client mengecek opsi baru yang diaktifkan kembali
5	Client mengirimklan pesan ChangeCipher-Spec untuk mengaktifkan kembali opsi

	yang dinegosiasi untuk semua pesan yang akan dikirimkan
6	Client mengirimkan pesan Finished yang memungkinkan server mengecek opsi baru yang diaktifkan kembali

4. Batasan Protokol

4.1 Batasan Utama

SSL berkonsentrasi pada pengamanan transaksi web. SSL membutuhkan protokol transport yang handal seperti TCP. Kebutuhan ini masuk akal dalam transaksi web karena Hypertext Transfer Protocol (HTTP) sendiri membutuhkan TCP. Oleh karena itu, SSL tidak dapat bekerja pada protokol transport connectionless seperti UDP.

Selain itu, SSL tidak dapat mendukung layanan keamanan *non-repudiation*, yaitu tanda tangan digital yang mencegah pihak pembuat dan penanda tangan melakukan penyangkalan pesan.

4.2 Batasan Kakas

SSL secara sederhana adalah protokol komunikasi. Implementasi SSL akan bersandar pada komponen lain untuk banyak fungsi, termasuk algoritma kriptografi. Algoritma-algoritma ini adalah kakas matematis yang secara aktual menjalankan task seperti enkripsi dan dekripsi. Tidak ada implementasi SSL yang lebih kuat daripada kakas kriptografi dimana SSL berada.

4.3 Batasan Lingkungan

Protokol jaringan secara sendiri hanya dapat menyediakan keamanan untuk informasi pada waktu informasi melintasi jaringan. Tidak ada protokol jaringan yang melindungi data sebelum data dikirim atau setelah tiba pada tujuan. Keamanan pada suatu jaringan komputer, baik internet publik atau fasilitas pribadi merupakan fungsi dari semua elemen yang membangun jaringan. Hal ini bergantung pada protokol keamanan jaringan, sistem komputer yang menggunakan protokol dan orang yang menggunakan komputer tersebut. Protokol SSL merupakan kakas keamanan yang kuat dan efektif, namun hanya merupakan kakas tunggal. Keamanan yang sesungguhnya membutuhkan banyak kakas seperti itu dan membutuhkan rencana yang menyeluruh untuk menggunakannya.

5. Kesimpulan

Makalah ini telah menguraikan secara singkat tentang protokol *Secure Socket Layer* untuk pengamanan web. Dimulai dari latar belakang munculnya teknologi SSL, pendekatan yang diambil secara arsitektural dan komponen serta operasionalnya. Pada bagian akhir disertakan dengan batasan-batasan yang dimiliki sebagai bahan pertimbangan untuk memutuskan protokol yang akan digunakan sehingga sesuai dengan kebutuhan sistem dan pengguna.

Daftar Pustaka

[1] Stephen A. Thomas, *SSL & TLS Essentials: Securing the Web*, Wiley Computer Publishing, John Wiley & Sons Inc., 2000.

[2] Rinaldi Munir, *Diktat Kuliah Kriptografi*, Teknik Informatika ITB, 2005