Bahan kuliah II4031 Kriptografi dan Koding

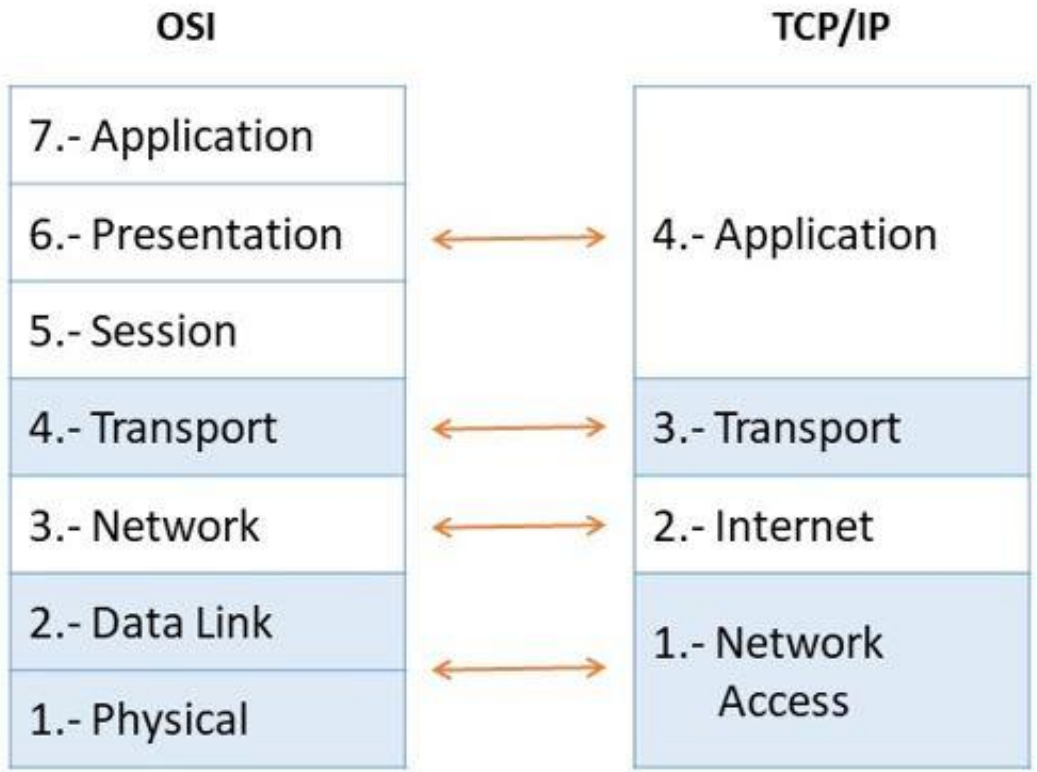
Secure Socket Layer (SSL)

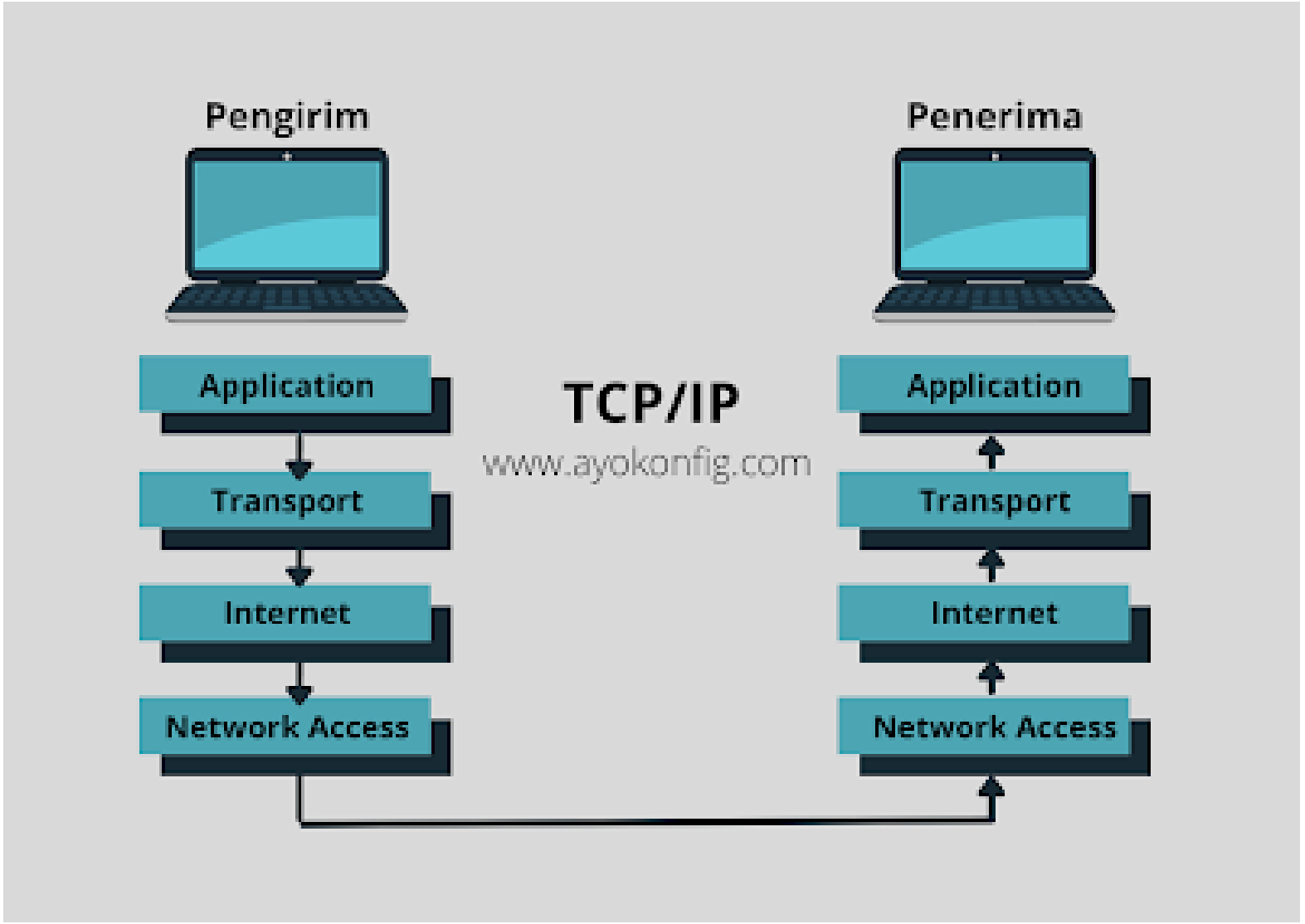
Oleh: Rinaldi Munir

Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
ITB - 2024

TCP/IP

- *Transmission Control Protocol/Internet Protocol* (TCP/IP) adalah standard protokol yang digunakan untuk menghubungkan komputer dengan jaringan sehingga membentuk jaringan yang lebih besar (WAN atau Internet).





Sumber gambar: https://www.ayokonfig.com/2021/12/pengertian-tcpip-beserta-fungsi.html

Keamanan Web

- *Secure Socket Layer* (*SSL*) adalah protokol yang digunakan untuk *browsing web* secara aman. *SSL* bertindak sebagai protokol yang mengamankan komunikasi antara *client* dan *server*.

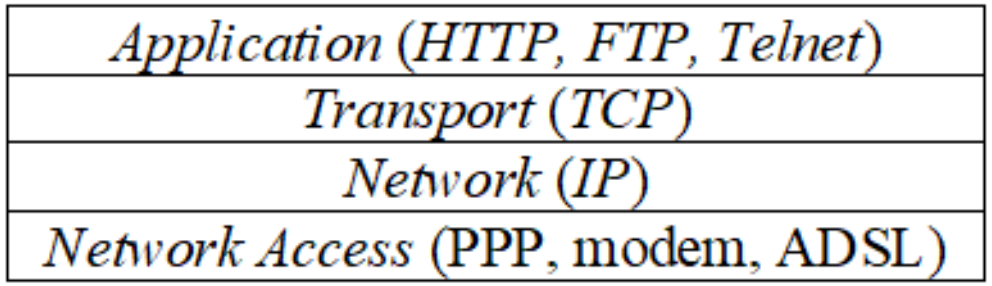
- *SSL* beroperasi antara lapisan *Application* dan lapisan *Transport. SSL* seolah-olah berlaku sebagai lapisan baru (*security layer*) antara kedua lapisan tersebut.

Application (*HTTP, FTP, Telnet*)
Security (*SSL*)
Transport (*TCP*)
Network (*IP*)
Network Access (PPP, modem, ADSL)

Gambar Lapisan (dan protokol) untuk *browsing* dengan *SSL*

- *SSL* dikembangkan oleh *Netscape Communations* pada tahun 1994.
- Ada beberapa versi *SSL*, versi 2 dan versi 3, tetapi versi 3 paling banyak digunakan saat ini.
- SSL didefinisikan di dalam RFC2246: http://www.ietf.org/rfc/rfc2246.txt
- Implementasi *open-source* SSL tersedia di: http://www.openssl.org/

Cara kerja TCP/IP



- Kebanyakan transmisi pesan di Internet dikirim sebagai kumpulan potongan pesan yang disebut **paket**.
- *IP* bertanggung jawab untuk merutekan paket (lintasan yang dilalui oleh paket).
- Pada sisi penerima, *TCP* memastikan bahwa suatu paket sudah sampai, menyusunnya sesuai nomor urut, dan menentukan apakah paket tiba tanpa mengalami perubahan.
- Jika paket mengalami perubahan atau ada data yang hilang, *TCP* meminta pengiriman ulang.

- Terlihat bahwa *TCP/IP* tidak memiliki pengamanan komunikasi yang bagus. Pesan ditransmisikan dalam bentuk plainteks.

- *TCP/IP* juga tidak dapat mengetahui jika pesan diubah oleh pihak ketiga (*man-in-the-middle attack*).

- *SSL* membangun hubungan (*connection*) yang aman antara pengirim dan penerima, sehingga pengiriman pesan antara dua entitas dapat dijamin keamanannya.

Application (*HTTP, FTP, Telnet*)
Security (*SSL*)
Transport (*TCP*)
Network (*IP*)
Network Access (PPP, modem, ADSL)

- Perlu dicatat bahwa *SSL* adalah protokol *client-server*, yang dalam hal ini *web browser* adalah *client* dan *website* adalah *server*.

- *Client* yang memulai komunikasi, sedangkan *server* memberi respon terhadap permintaan *client*.

- Protokol *SSL* tidak bekerja kalau tidak diaktifkan terlebih dahulu (biasanya dengan meng-klik tombol yang disediakan di dalam *web server*)

Komponen SSL

SSL disusun oleh dua sub-protocol (*layer*):

1. *SSL handshaking*, yaitu sub-protokol untuk membangun koneksi (kanal) yang aman untuk berkomunikasi,

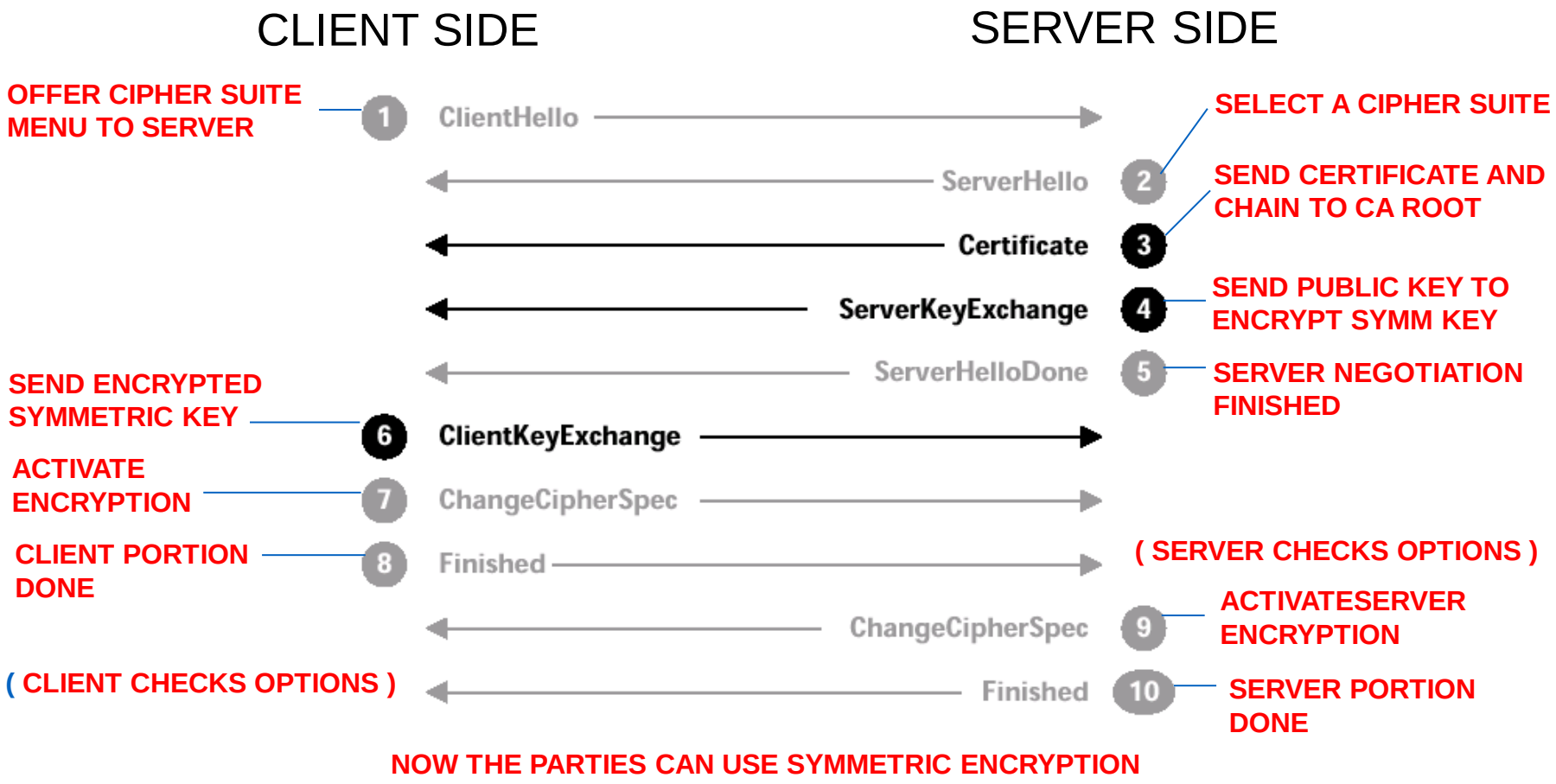
2. *SSL record*, yaitu sub-protokol yang menggunakan kanal yang sudah aman. *SSL Record* membungkus seluruh data yang dikirim selama koneksi.

Sub-protokol *handshaking*

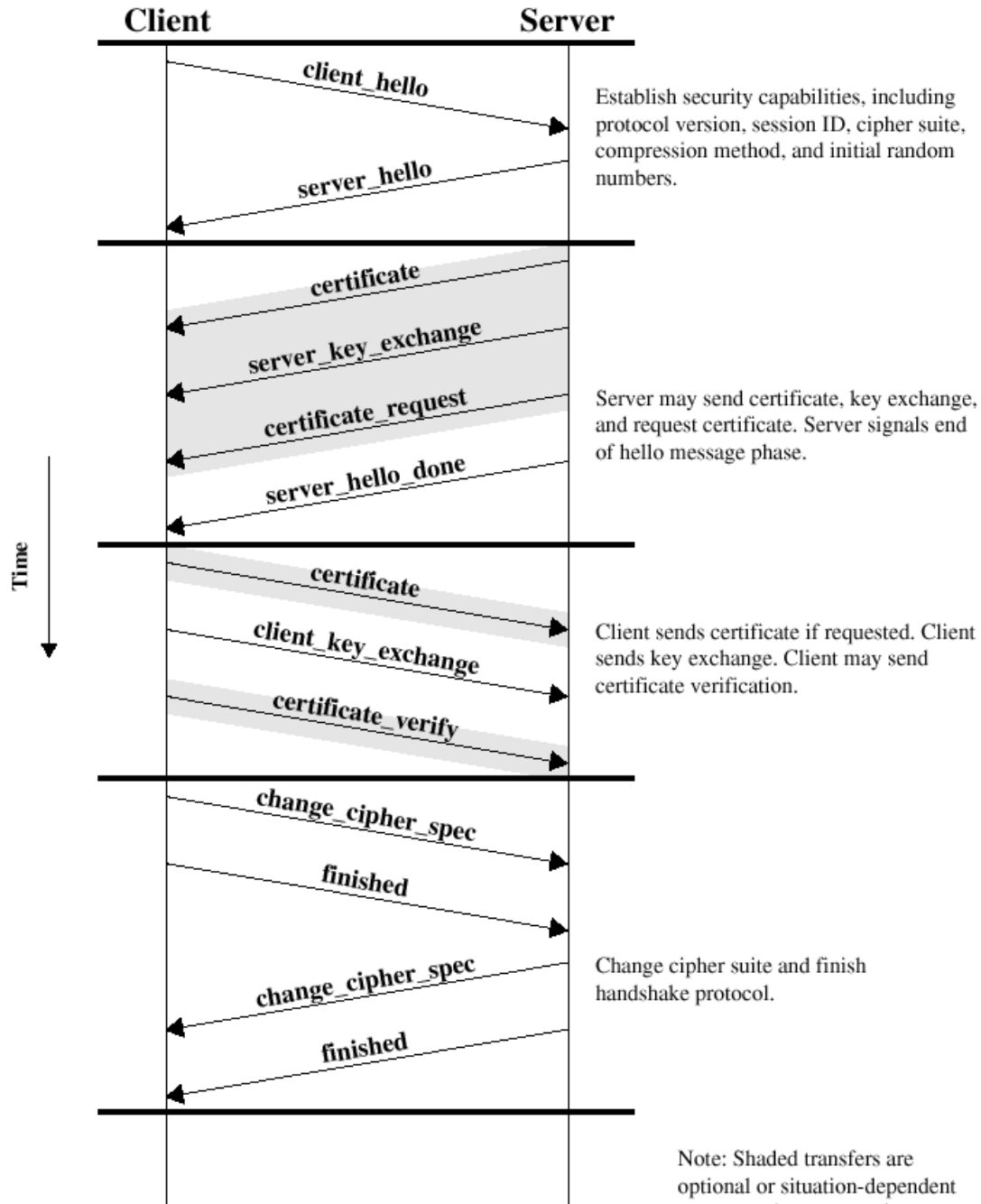
- Merupakan bagian yang paling kompleks di dalam SSL.
- Proses yang dilakukan di dalam sub-protokol handshaking:
  - *Say* ’helllo’
  - *Client* dan *server* melakukan otentikasi satu sama lain
  - Pertukaran kunci (untuk enkripsi dengan algoritma simetri)
  - Menegosiasikan algoritma enkripsi, *hash*, kompresi, dan MAC

- Subprotokol handshaking dilakukan sebelum data ditransmisikan antara client dan server

Sub-protokol *handshaking*



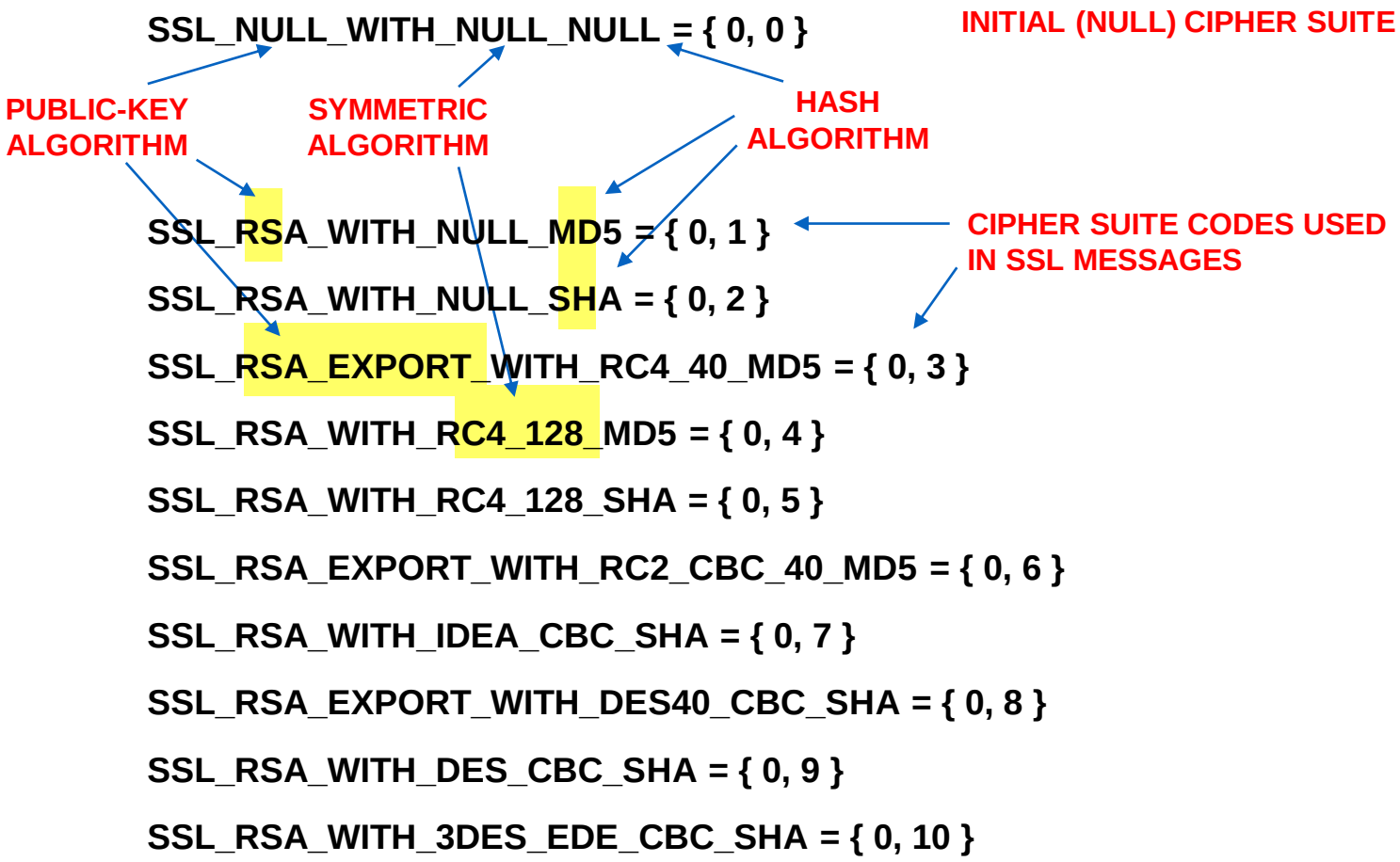
SOURCE: THOMAS, *SSL AND TLS ESSENTIALS*


Client
Server
client_hello
server_hello
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.
certificate
server_key_exchange
certificate_request
server_hello_done
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.
Time
certificate
client_key_exchange
certificate_verify
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.
change_cipher_spec
finished
change_cipher_spec
finished
Change cipher suite and finish handshake protocol.
Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Client Hello:

- Protocol version
  - SSLv3(major=3, minor=0)
  - TLS (major=3, minor=1)
- Random Number
  - 32 bytes
  - First 4 bytes, time of the day in seconds, other 28 bytes random
  - Prevents replay attack
- Session ID
  - 32 bytes – indicates the use of previous cryptographic material
- Compression algorithm

Client Hello - Cipher Suites

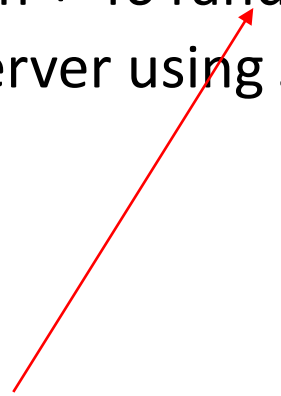


Server Hello:

- Version
- Random Number
  - Protects against handshake replay
- Session ID
  - Provided to the client for later resumption of the session
- Cipher suite
  - Usually picks client's best preference – No obligation
- Compression method

Client Key Exchange:

- Premaster secret
  - Created by client; used to “seed” calculation of encryption parameters
  - 2 bytes of SSL version + 46 random bytes
  - Sent encrypted to server using server’s public key



This is where the attack happened in SSLv2

- Master secret
  - Generated by both parties from premaster secret and random values generated by both client and server
- Key material
  - Generated from the master secret and shared random values
- Encryption keys
  - Extracted from the key material

- Sampai di sini, proses pembentukan kanal yang aman sudah selesai.

- Bila sub-protokol ini sudah terbentuk, maka *http://* pada *URL* berubah menjadi *https://* (*http secure*)

https://bankmandiri.co.id

ID | EN

Ketik untuk mencari

Log In

mandiri

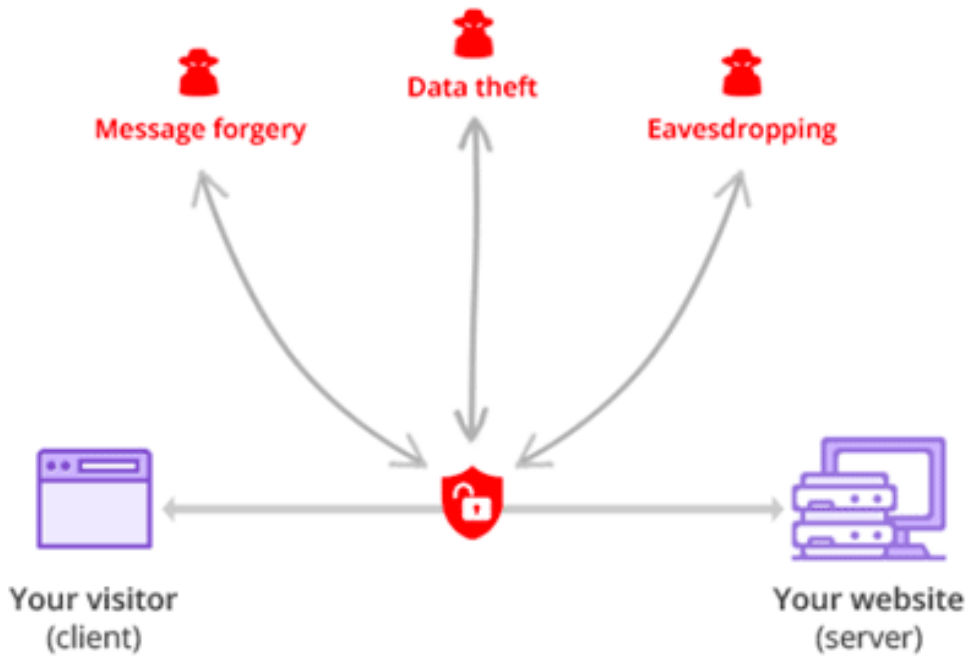
Perseorangan Bisnis Prioritas Tentang Mandiri Bantuan

Full Digital Banking World

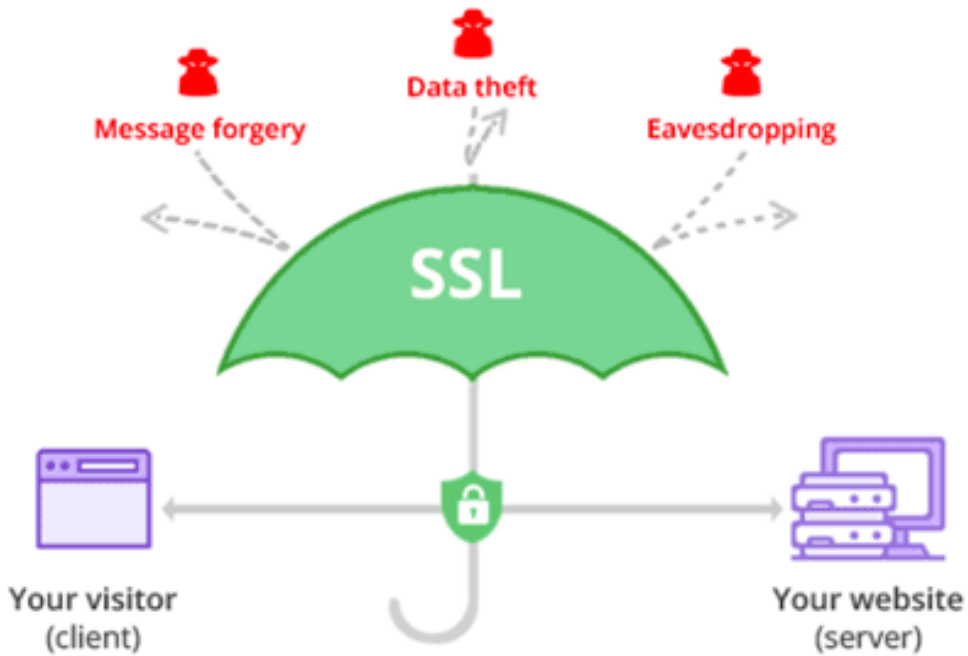
Financial Super App Livin' by Mandiri dan Wholesale Digital Super Platform Kopra by Mandiri

SELENGKAPNYA >

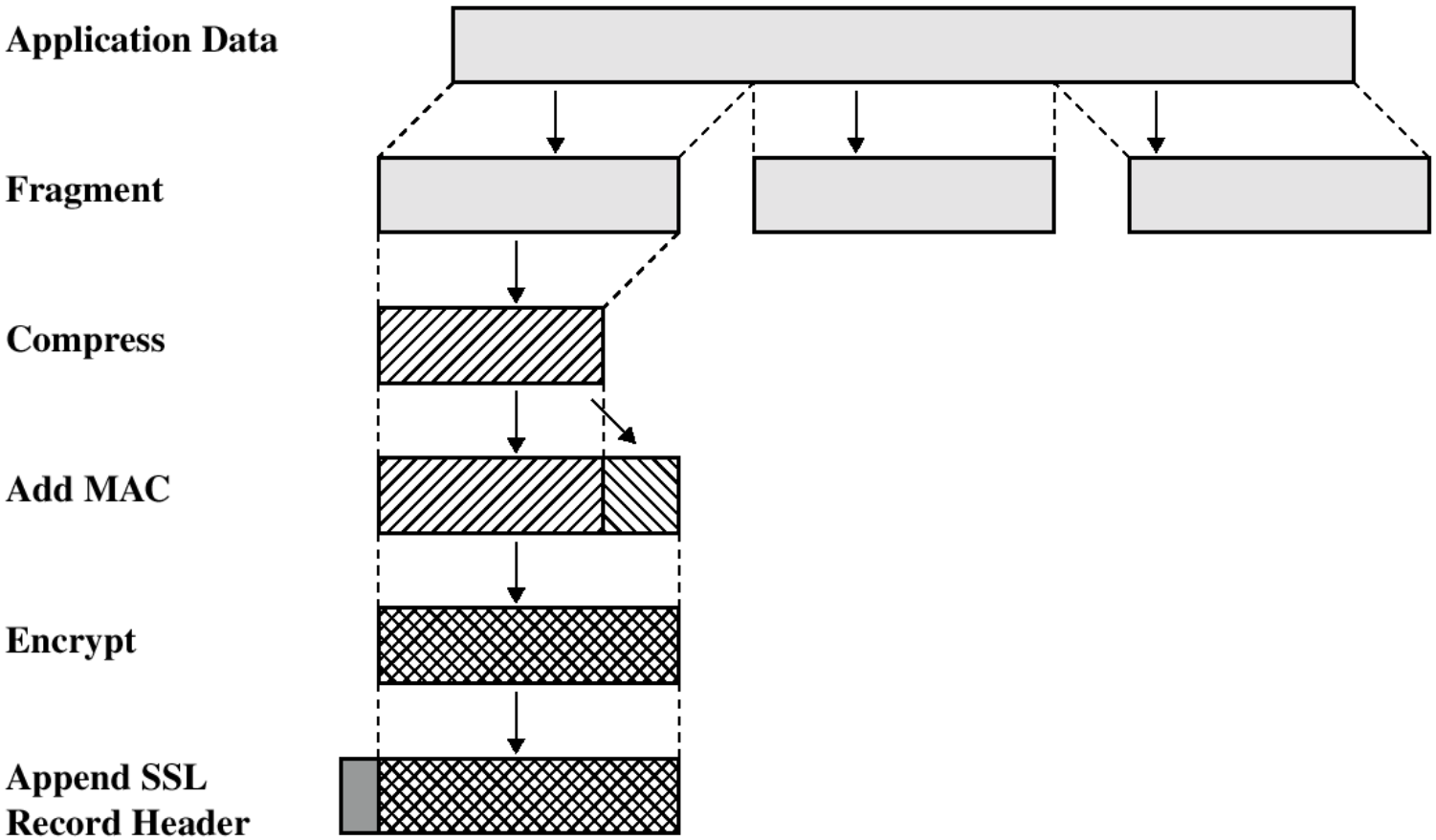
HTTP: No Encryption (no SSL)



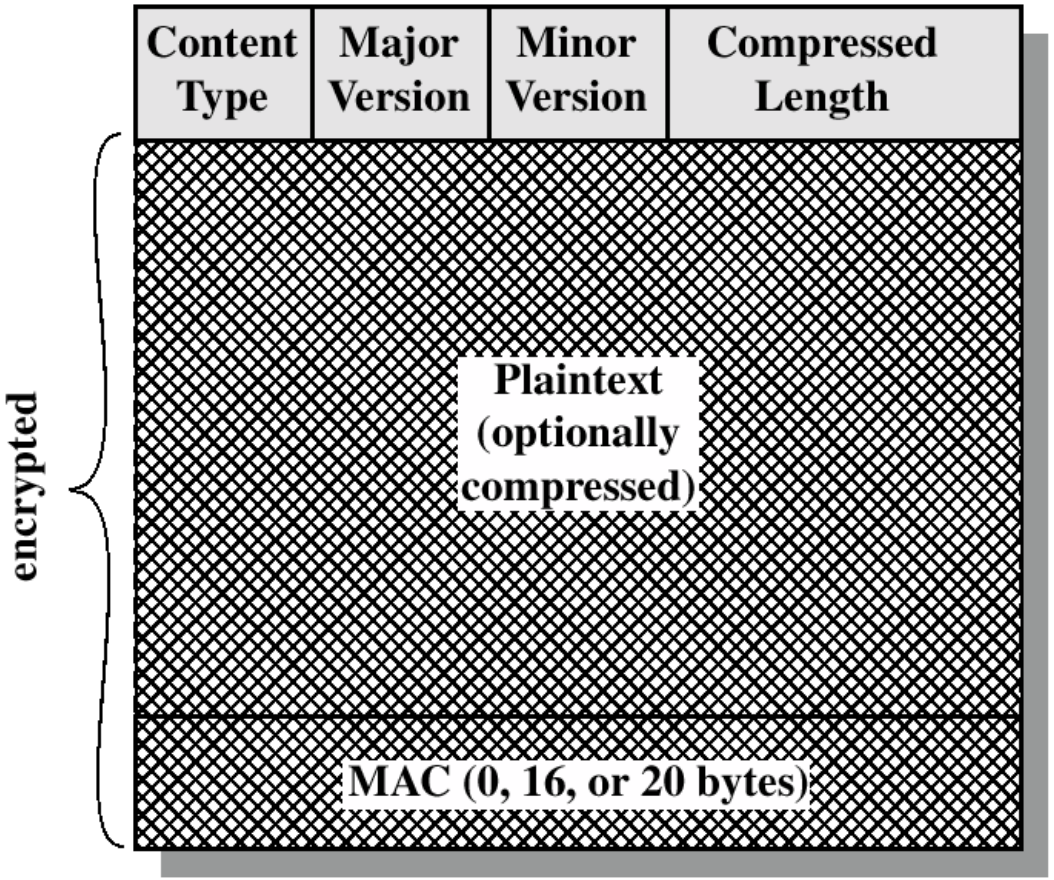
HTTPS: Secure Cheap SSL Connection



Sub-protokol *SSL record*



SSL Record Format



- Di tempat penerima, sub-protokol *SSL* Record melakukan proses berkebalikan: mendekripsi data yang diterima, mengotentikasinya (dengan *MAC*), men-dekompresinya, lalu merakitnya.

- Protokol SSL membuat komunikasi menjadi lebih lambat.

- Piranti keras, seperti kartu *peripheral component interconnect* (*PCI*) dapat dipasang ke dalam *web server* untuk memproses transaksi *SSL* lebih cepat sehingga mengurangi waktu pemrosesan

- Informasi lebih lanjut mengenai *SSL* dapat diperoleh dari tutorial *SSL* di `www.netscape.com/security/index.html`.

TLS (*Transport Layer Security*)

- Pada Tahun 1996, *Netscape Communications Corp*. mengajukan *SSL* ke *IETF* (*Internet Engineering Task Force*) untuk standardisasi.
- Hasilnya adalah *TLS* (*Transport Layer Security*). *TLS* dijelaskan di dalam *RFC* 2246
- Untuk informasi lebih lanjut perihal *TLS*, kunjungi situs *IETF* di *www.ietf.org/rfc/rfc22*.
- *TLS* dapat dianggap sebagai *SSL* versi 3.1, dan implementasi pertamanya adalah pada Tahun 1999

Transport Layer Security (TLS)

- The same record format as the SSL record format.
- Defined in RFC 2246.
- Similar to SSL v3.
- Differences in the:
  - version number
  - message authentication code
  - pseudorandom function
  - alert codes
  - cipher suites
  - client certificate types
  - certificate_verify and finished message
  - cryptographic computations
  - padding