SSL/TLS

26.10.2005

Institut für Mathematik, TU Berlin

Rosa Freund -- rosa@pool.math.tu-berlin.de

Übersicht

- Einführung
- **SSL vs. TLS**
- **SSL: Anwendung und PKI**
- SSL Protokoll: Record Protocol und Handshake Protocol
- Batching
- Referenzen

SSL vs. TLS

- SSL: Secure Sockets Layer
- TLS: Transport Layer Security
- TLS 1.0 basiert auf SSL 3.0 und wird von der IETF weiterentwickelt
- Essentiell dasselbe: Hier synonym benutzt.

Einführung

- SSL ist ein Netzwerkprotokoll
- Protokolle legen den Ablauf und das Format fest, nach dem zwei Stellen Daten miteinander austauschen

Praxis

- SSL als Protokoll kann theoretisch bei beliebigen Kommunikationsanwendungen als Verschlüsselungsverfahren eingesetzt werden
- Praxis: Fast ausschließlich Anwendung bei Verschlüsselung im Internet (Client/Server, https://)

Praxis

- Aus Performancegründen wird asymmetrische Verschlüsselung (meist RSA) lediglich benutzt, um geheimen Schlüssel für symmetrische Verschlüsselung zu übertragen.
- Die Daten werden dann ausschließlich symmetrisch verschlüsselt (je nach Server/Client z.B. RC4, Triple DES, AES)
- Es wird also PKI zur Authentifizierung des Servers benötigt

Praxis: PKI

- Certificate Authorities (CAs)
- Firmen beantragen SSL-Zertifikat für ihre Domain mit Certificate Signing Request (CSR). Dies muß auf dem Server, auf dem die Domain liegt, generiert werden und beinhaltet u.A. :
- öffentlichen Schlüssel
- Domainnamen, Firmenname, Firmensitz
- CAs überprüfen u.A. Domaininhaber und Handelsregister und vergeben dann Zertifikate

Certificate Authorities

- Einige große CAs sind z.B. GeoTrust, Verisign, Thawte
- Kosten: ab ca. 200 Dollar/Jahr
- Nichtkommerzielle CA: CACert, steckt noch in den Kinderschuhen
- In gängige Browser sind jeweils ca. 20 Root Certificates eingebaut (allerdings nicht dieselben!)
- Kriterien, welcher Browser welche RCs integriert, sind wenig transparent

Praxis: Authentifizierung

- Server sendet ein Zertifikat an Client. Dies besteht aus Informationen (u.A. Firmenname, öffentlicher Schlüssel), die von einer CA signiert sind.
- Client prüft Signatur der Informationen mit in Browser eingebauten Root Certificates.

Praxis: Schlüsseltausch

- Client verwendet diesen öffentlichen Schlüssels des Servers, verschlüsselt zufällige Bytes damit und schickt den Ciphertext an den Server
- Die korrekte Entschlüsselung dieser Bytes durch den Server ist Grundlage für die weitere Kommunikation
- Die Authentifizierung und der Schlüsseltausch geschieht im vierstufigen SSL-Handshake

Trust Issues

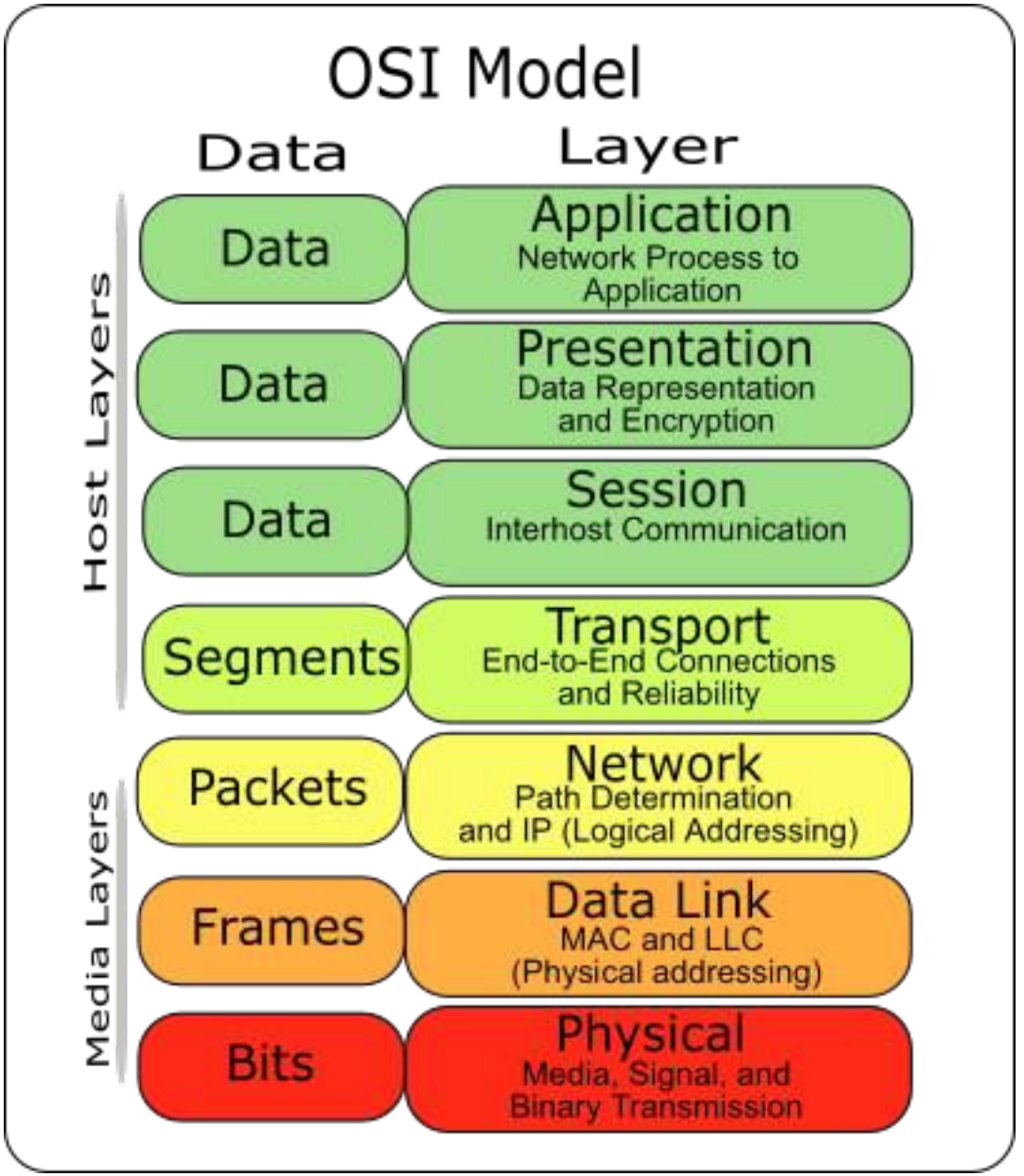
- Die Certificate Authorities (CAs) überprüfen die Identität hinreichend sorgfältig
- In meinem Browser sind korrekte Root Certificates der CAs eingebaut

Übersicht

- Einführung
- SSL vs. TLS
- SSL: Anwendung und PKI
- **SSL Protokoll: Record Protocol und Handshake Protocol**
- Batching
- Referenzen

ISO/OSI Schichtenmodell

- "Open Systems Interconnection Reference Model"
- Protokollfunktionen werden auf die Schichten verteilt
- Jede Schicht benutzt nur Funktionen der direkt unter ihr liegenden Schicht
- Von der International Organisation for Standardization (ISO) entwickelter Standard für Netzwerkkommunikation
- Ermöglicht Zusammenarbeit von Hard- und Software unterschiedlicher Hersteller



Quelle: Wikipedia

SSL Schichten

- SSL befindet sich auf dem Presentation Layer
- Besteht aus zwei Teilen:
- SSL Record Protocol zur Datenübertragung
- SSL Handshake Protocol liegt über dem Record Protocol und initiiert die Datenübertragung

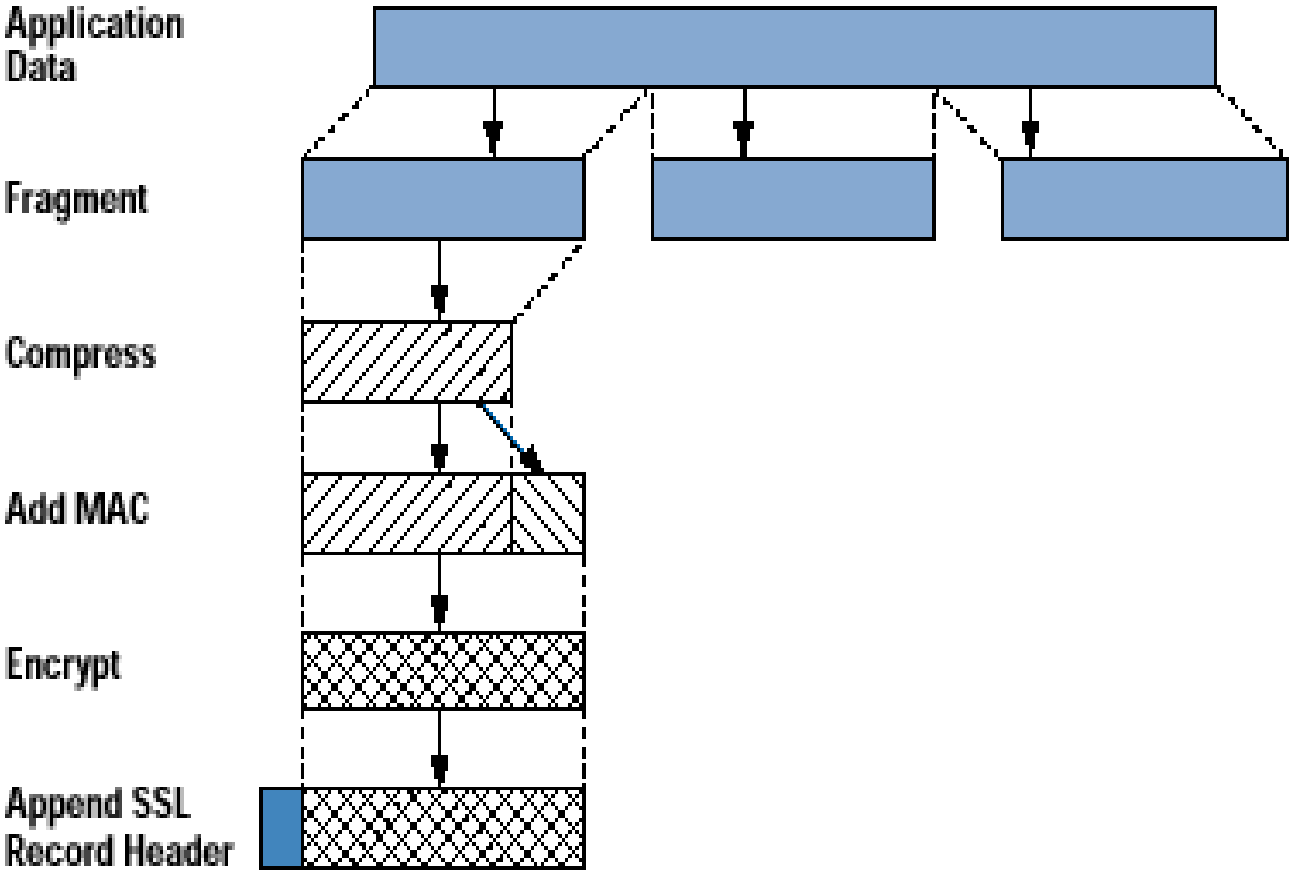
SSL States

- Während des Datenaustauschs befinden sich Client und Server stets in einem bestimmten Zustand (state)
- Es gibt ein “current state” und ein “pending state”
- Erhält Client oder Server eine ChangeCipherSpec – Message, wird der “pending state” als neuer “current state” übernommen (passiert nur einmal am Ende des Handshakes)

SSL Record Layer

- Während einer SSL-Session kommunizieren Server und Client über Records – Datenpakete, die einem bestimmten Format genügen müssen.
- Der Record Layer erhält Daten von darüberliegenden Schichten (z.B. Handshake Protocol). Diese werden fragmentiert, mit Headern versehen, ggf. verschlüsselt

SSL Record Layer



Quelle: http://www.cisco.com/en/US/about/ac123/ac147/archived_issues/ipj_1-1/ssl.html

SSL Record Layer

Auszüge:

```
struct {
       uint8 major, minor;
    } ProtocolVersion;

enum {
       change_cipher_spec(20), alert(21), handshake(22),
       application_data(23), (255)
   } ContentType;

struct {
      ContentType type;
      ProtocolVersion version;
      uint16 length;
      opaque fragment[SSLPlaintext.length];
   } SSLPlaintext;
```

Anwendung z.B.

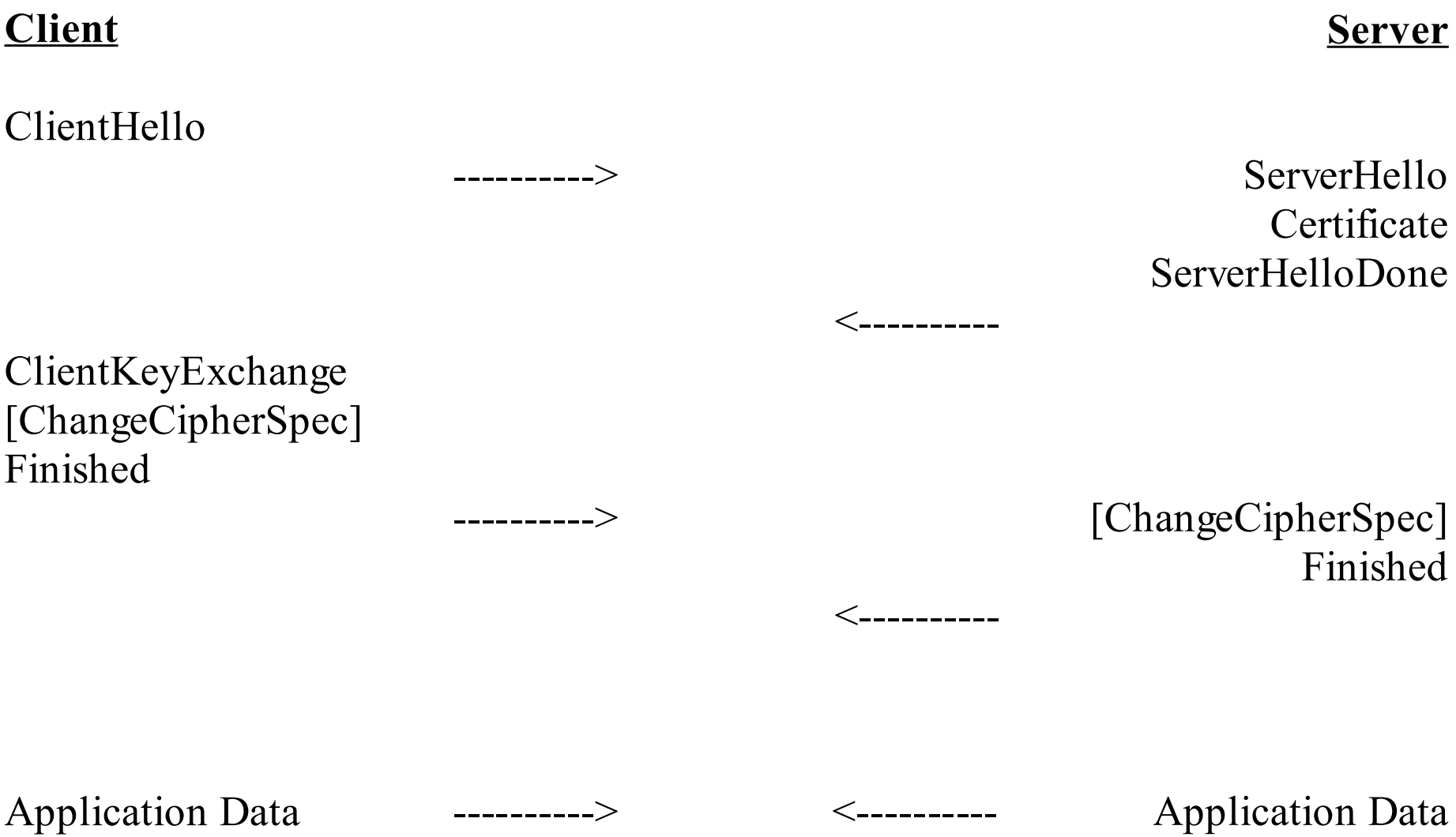
SSL 3.0: ProtocolVersion version = {3,0};

ContentType 23

SSL Handshake

- Jede SSL-Session beginnt mit einem Handshake
- Im Handshake findet die Authentifizierung statt, Protokollversion, Session ID, verwendete Algorithmen und Kompressionsmethode werden etabliert

Handshake: Übersicht



ClientHello

```
struct {
       ProtocolVersion client_version;
       Random random;
       SessionID session_id; //nur für resumen
       CipherSuite cipher_suites<0..2^16-1>;
       CompressionMethod compression_methods<0..2^8-1>;
     } ClientHello;
```

Format der Variablen ist vorgegeben, z.B.

```
CipherSuite SSL_RSA_WITH_NULL_MD5               = { 0x00,0x01 };
CipherSuite SSL_RSA_WITH_NULL_SHA               = { 0x00,0x02 };
CipherSuite SSL_RSA_EXPORT_WITH_RC4_40_MD5      = { 0x00,0x03 };
CipherSuite SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA    = { 0x00,0x0D };
```

ServerHello

```
struct {
          ProtocolVersion server_version;
          Random random;
          SessionID session_id;
          CipherSuite cipher_suite;
          CompressionMethod compression_method;
      } ServerHello;
```

Server Certificate

- Entspricht in der Regel X.509.v3

```
opaque ASN.1Cert<1..2^24-1>;
struct {
          ASN.1Cert certificate_list<1..2^24-1>;
      } Certificate;
```

ClientKeyExchange

```
struct {
          select (KeyExchangeAlgorithm) {
              case rsa: EncryptedPreMasterSecret;
              case diffie_hellman: ClientDiffieHellmanPublic;
              case fortezza_kea: FortezzaKeys;
          } exchange_keys;
      } ClientKeyExchange;


struct {
          public-key-encrypted PreMasterSecret pre_master_secret;
      } EncryptedPreMasterSecret;

struct {
          ProtocolVersion client_version;
          opaque random[46];
      } PreMasterSecret;
```

ChangeCipherSpec

- CipherSpec: Aktuell benutzte Verschlüsselungs-, Hash und Kompressionsmethoden

```
struct {
          enum { change_cipher_spec(1), (255) } type;
      } ChangeCipherSpec;
```

Master Secret

- Für die symmetrische Verschlüsselung ist ein geheimer Schlüssel notwendig: master_secret
- Dies kann nun von Client und Server berechnet werden:

```
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));
```

Finished

- Erste Nachricht, die mit den ausgetauschten Parametern verschlüsselt wird
- Nach Erhalt muß Korrektheit bestätigt werden

```
enum { client(0x434C4E54), server(0x53525652) } Sender;

struct {
      opaque md5_hash[16];
      opaque sha_hash[20];
     } Finished;

md5_hash MD5(master_secret + pad2 +
             MD5(handshake_messages + Sender +
             master_secret + pad1));
```

Also: SSL Zusammenfassung

- SSL wahrt:
- Authentizität durch Zertifikate und asymmetrische Verschlüsselung
- Integrität durch Hashes
- Datenschutz (privacy) durch symmetrische Verschlüsselung

Übersicht

- Einführung
- SSL vs. TLS
- SSL: Anwendung und PKI
- SSL Protokoll: Record Protocol und Handshake Protocol
- **Batching**
- **Referenzen**

Batching

Jetzt: “Improving SSL Handshake Performance via Batching”, Shacham und Boneh 2001)
(Basiert auf: “Batch RSA”, Fiat 1996)

Referenzen

SSL 3.0 “RFC”
http://wp.netscape.com/eng/ssl3/draft302.txt

Verisigns Informationen zu SSL (deutsch, sehr ausführlich und anwendungsbezogen)
http://www.verisign.de/products-services/security-services/ssl/index.html