Vorlesung (WS 2014/15)

Sicherheit: Fragen und Lösungsansätze

Dr. Thomas P. Ruhroth

TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

“Sicherheitsprotokolle“

**[mit freundlicher Genehmigung basierend
auf einem Foliensatz von
Prof. Dr. Claudia Eckert (TU München)]**

Literatur:

Claudia Eckert: IT-Sicherheit: Konzept - Verfahren - Protokolle, 7., überarb. und erw. Aufl., Oldenbourg, 2012.

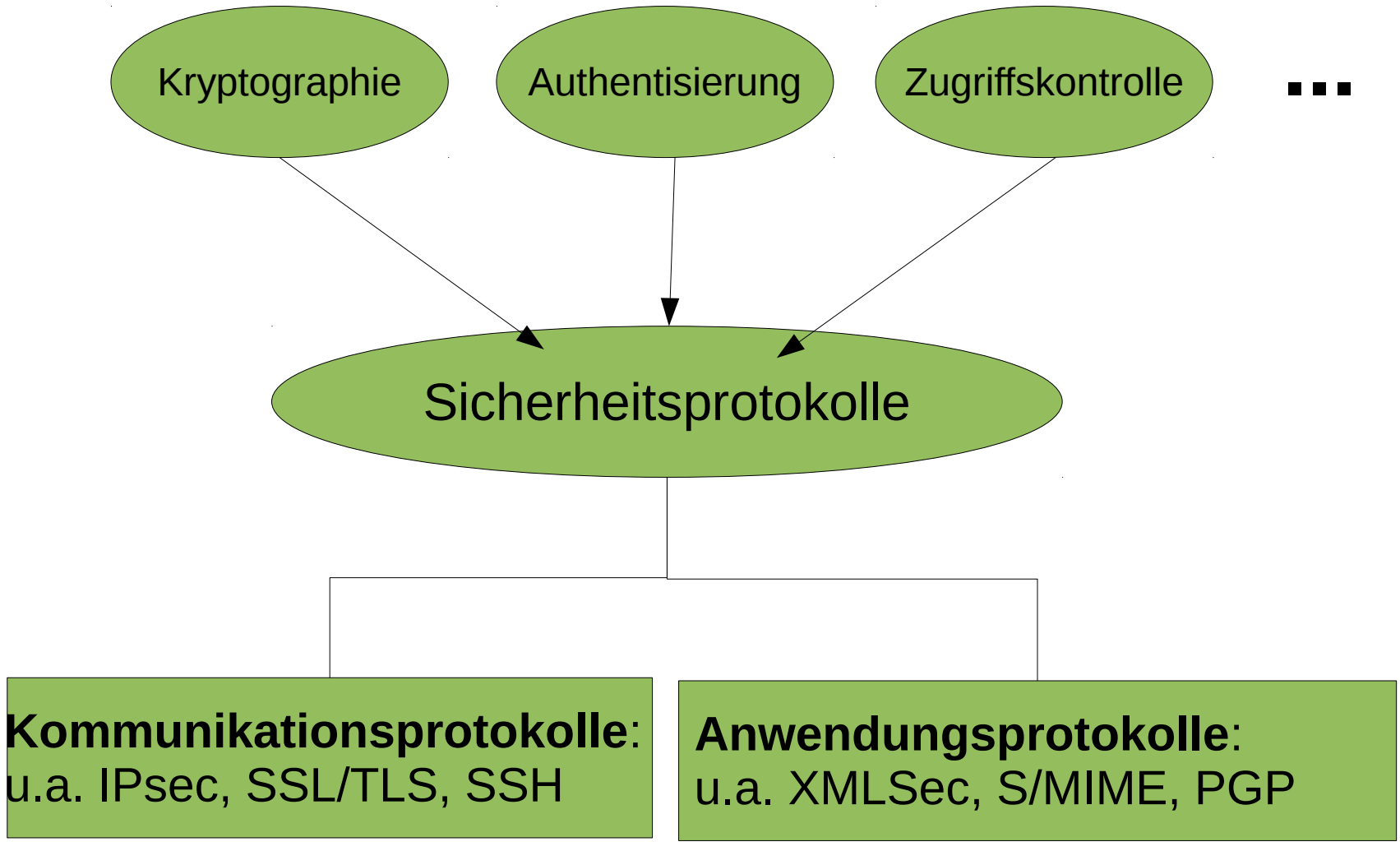
E-Book: http://www.ub.tu-dortmund.de/katalog/titel/1362263

Agenda

- Sicherheitsprotokolle
  - IPsec
  - SSL

- Verstehen wie aus den verschiedenen Sicherheits-Bausteinen Protokolle gebaut werden.
- Kennen der Beispiele IPsec und Ssl

Überblick


Kryptographie
Authentisierung
Zugriffskontrolle
...
Sicherheitsprotokolle
Kommunikationsprotokolle:
u.a. IPsec, SSL/TLS, SSH
Anwendungsprotokolle:
u.a. XMLSec, S/MIME, PGP

Schutzziele

- Authentifikation:
  - **Unterschiede**: Device oder einzelner Nutzer, wechselseitig, …
- Vertraulichkeit, Integrität:
  - **Unterschiede**: Schlüssel pro Anwendung oder generisch für Verbindung, Nachrichten- oder Transaktionsintegrität
- Verfügbarkeit wird idR **nicht** adressiert
- Verbindlichkeit nur in sehr speziellen Anwendungsprotokollen

Kommunikationsprotokolle

Kommunikationsprotokolle:

- Initiale Protokollschritte: vorab Aufbau eines sicheren ‚Kanals‘
- Kanalaufbau erfordert Handshake:
  - Protokollabwicklung zwischen Subjekten (idR Geräten)
  - Aushandeln einer gemeinsamen Security-Policy für den Kommunikationskanal:
  - wie vereinbart man Policies?
  - was ist festzulegen?
- **Kommunikationskanal:**
- Vereinbarte Schlüssel gelten für mehrere Protokoll-nachrichten zwischen den Kommunikationspartnern
- Schutz der Daten ‚endet‘ am Kanalende: dort erfolgt Entschlüsselung und ggf. Authentizitätsprüfung

Anwendungsprotokolle

Anwendungsprotokolle:

- Ziel idR: One-step-Processing, d.h.
  - individuelle Verschlüsselung, ggf. Signaturerstellung etc.
  - Daten der Anwendungsebene enthalten alle benötigten Informationen zur Absicherung
    - Welche Informationen sind erforderlich?
    - Probleme?
- Entkopplung des Empfangs von Daten und der Durchführung von kryptographischen Operationen und Überprüfungen
  - Konsequenzen?
  - mögliche Probleme?

OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendungen (7)	Anwendungen	HTTP, UDS, FTP, SMTP, POP, Telnet, OPC UA
Darstellung (6)		
Sitzung (5)		
Transport (4)	Transport	TCP, UDP, SCTP
Vermittlung (3)	Internet	IP (IPv4, IPv6), ICMP (über IP)
Sicherung (2)	Netzzugang	Ethernet, Token Bus, Token Ring, FDDI, IPoAC
Bitübertragung (1)		

SSL

Secure Socket Layer

Secure Socket Layer (SSL), RFC 6101

Background

- 1995 als SSLv2 durch Netscape in Navigator 1.1 eingeführt
- 1996 Überarbeitung von SSLv2 zu SSLv3 (Netscape)
  - ‚Quasi-Standard' für sichere TCP-Kommunikation
- 1999 Einführung von TLS (Transport Layer Security) v1.0
  - IETF Industrie-Standard, eng an SSLv3 angelehnt, aber
  - inkompatibel zu SSLv3 (u.a. HMAC statt MAC, unterschiedliche Berechnung der Schlüssel)
  - Open-Source-Implementation unter http://www.openssl.org
- RFC 5246 (TLS 1.2)
- RFC 2818 (HTTPS, SSL-basiertes HTTP)

SSL/TLS: Konzepte und Schutzziele

Basis: SSL/TLS setzt auf Schicht 4, der Transportschicht, auf Client-Server-Kommunikationsparadigma

Session:

- Aushandlung von zu verwendenden Krypto-Algorithmen und von einem gemeinsamen Master-Secret (MS),
- einer Session können mehrere Verbindungen zugeordnet sein,
- alle Verbindungen einer Session basieren auf dem MS.

Verbindung: entspricht einer TCP-Verbindung

- individuelle Schlüssel pro Verbindung, abgeleitet aus MS
- Verwendung der Algorithmen der zugeordneten Session
- Verbindungsaufbau: vorhandene Session, oder neue nutzen

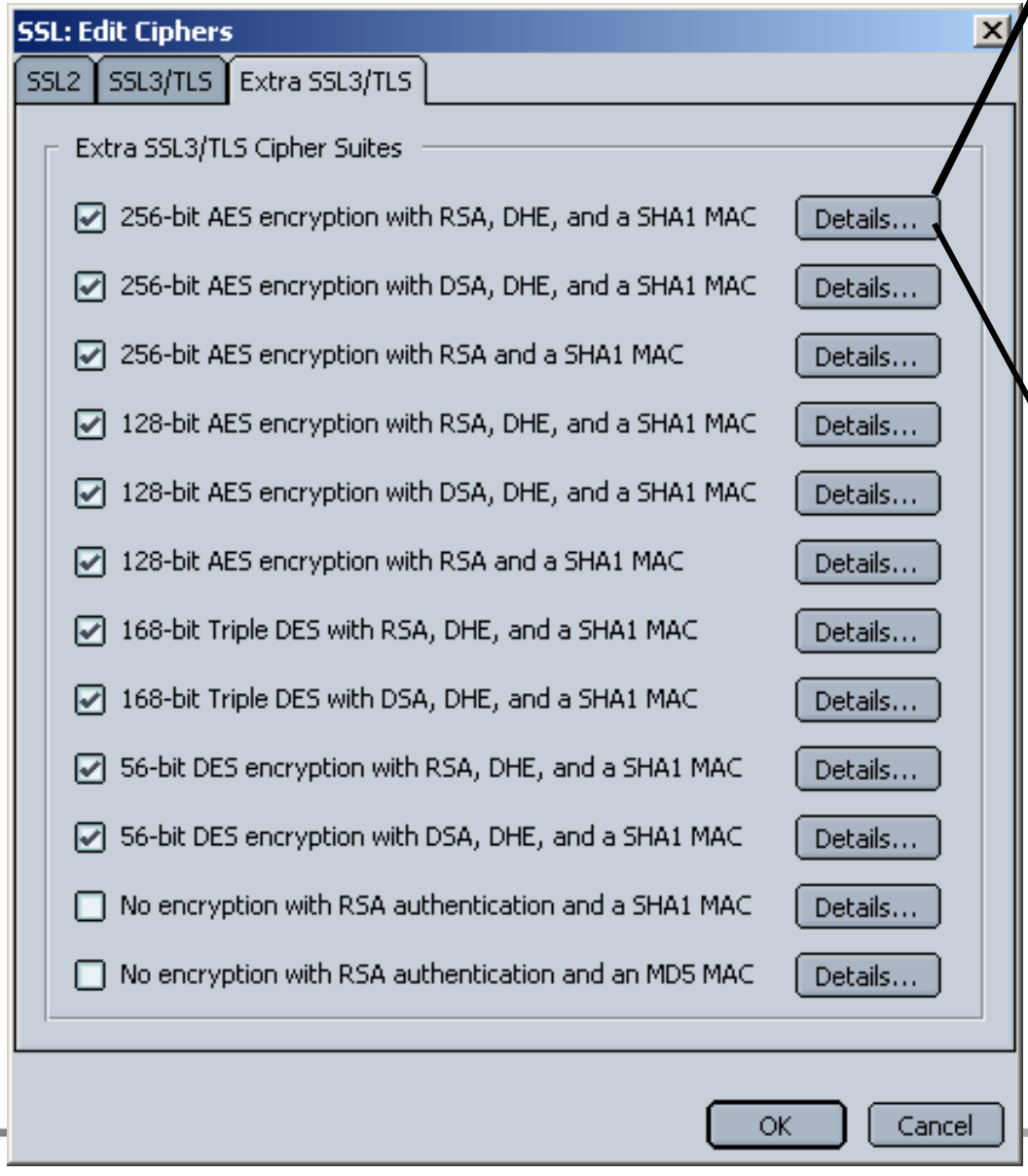
Schutzziele

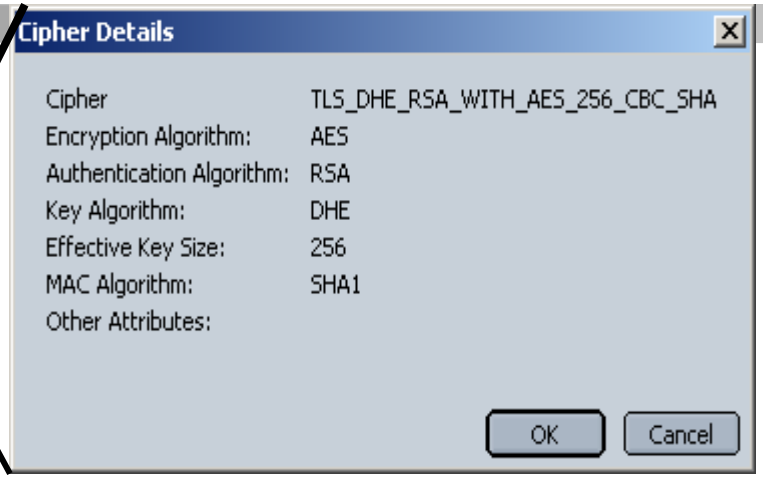
Schutzziele von SSL/TLS

Sichere Client-Server-Kommunikation

- Authentifikation:
  - wechselseitig, einseitig (meist nur Server), anonym
  - Mechanismen: X509-Zertifikate, MACs (MD5, SHA-1)
- Integrität von Nachrichten
  - mit HMAC/MAC-Verfahren: SHA-1, MD5
- Vertrauliche Datenübertragung
  - wählbare **Cipher-Suite,** (siehe nächste Folie)
  - symmetrische Verfahren
- Schlüsselaustausch:
  - Public-Key-Verfahren: RSA, DH

Cipher-Suites

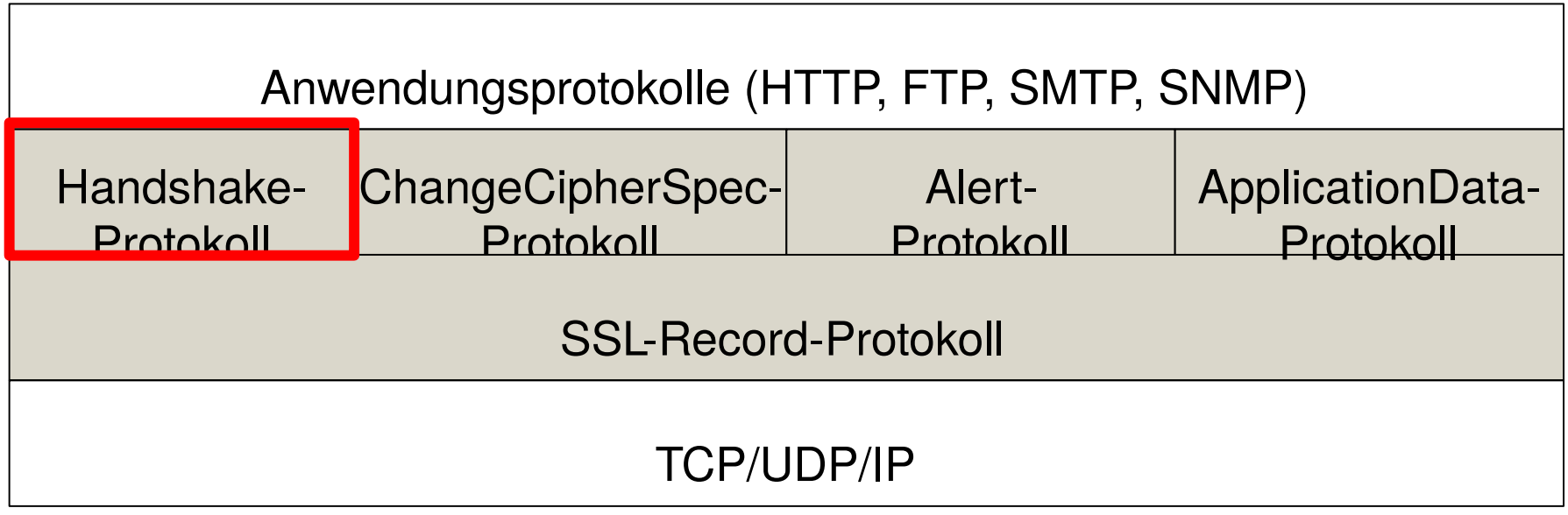




Beim SSL-Verbindungsaufbau wird die zu verwendene Suite zwischen Client und Server im SSL-Handshake ausgehandelt.

SSL-Protokolle

SSL-Protokolle



ChangeCipherSpec-Protokoll: 1Byte

- Zeigt die Verwendung der vereinbarten Verfahren an

Alert-Protokoll: dient zur Fehlerbehandlung, (2 Byte)

- Erstes Byte: Schwere des Fehlers (1=Warnung; 2=Abbruch)
- Zweites Byte: Fehlertyp (z.B. MAC-Prüfung schlägt fehl)

SSL/TLS-Handshake-Protokoll

Aufgaben:

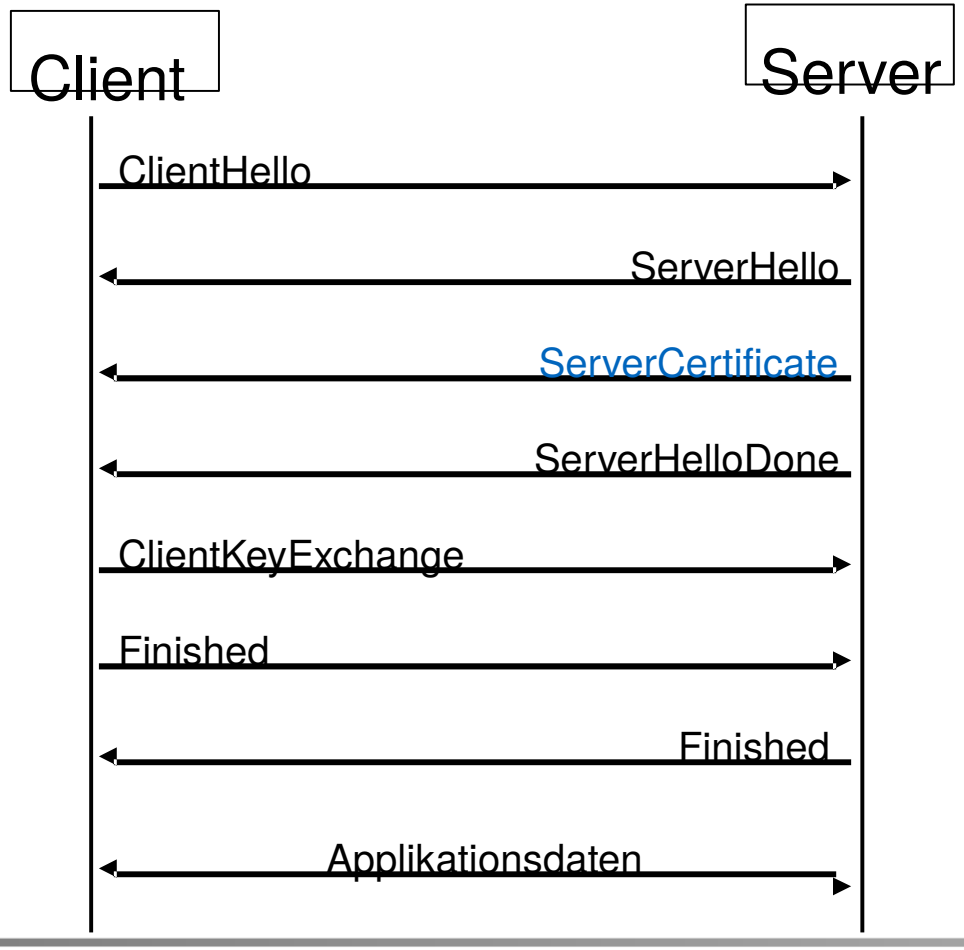
- Festlegen der zu verwendenden Krypto-Verfahren (Suite)
- Austausch von geheimer Basis-Information
- unidirektionale Verbindung: mit unterschiedlichen Schlüsseln
- Authentifikation mittels X509.v3-Zertifikaten
- Verwaltung von Sitzungsinformationen: Client/Server kann mehrere offene Sitzungen haben

Verschiedene Modi:

- Server authentifiziert, Client anonym (u.a. bei Web-Portalen)
- Server und Client authentifiziert (über Zertifikate)
- Server und Client anonym

SSL/TLS-Handshake-Protokoll

Modus: Server authentifiziert, Client anonym



ClientHello:
- Protokollversion
- Randomzahl R_c
- bekannte Verschlüsselungsmethoden
- bekannte Komprimiermethoden

ServerHello:
- Protokollversion
- Randomzahl R_s
- gewählte Verschlüsselungsmethoden
- gewählte Komprimiermethoden

ServerCertificate:
- Liste der Zertifikate der Zertifizierungskette
- Server-Public-Key, beim RSA-Verfahren

ClientKeyExchange:
- Pre-master Secret mit Server-Public-Key verschlüsselt

Finished (Server bzw. Clientkennung):
- Berechnen des Master-Secrets aus Pre-master und den Random-Werten R_c, R_s
- MD5-Hash-Wert mit Master-Secret über die bisher ausgetauschten Nachrichten
- SHA-Hash-Wert mit Master-Secret über die bisher ausgetauschten Nachrichten

SSL-Record-Protokoll

Aufteilung der Daten in Fragmente und Kompression

- MAC-Berechnung u. Verschlüsseln der Daten
- **Schlüsselerzeugung**
- **Schlüsselberechnung** aus *Pre*, R_c , R_s
  - Berechnen des Master-Secrets durch Client u. Server
  - master_secret = MD5(*Pre* | SHA('A' | *Pre* | R_c | R_s)) |
    MD5(*Pre* | SHA('BB' | *Pre* | R_c | R_s)) |
    MD5(*Pre* | SHA('CCC' | *Pre* | R_c | R_s))
- **Master-Secret:** 3 · 128 Bit = 384 Bit = **48 Byte**
- Aus Master-Secret werden geheime Session- und der MAC-Schlüssel durch mehrmalige Anwendung von MD5, SHA-1 auf das Master-Secret abgeleitet

Ergänzende Bemerkungen

Ergänzende Bemerkungen:

SSL unterstützt verschiedene Schlüsselaustauschverfahren

- RSA ist eine Möglichkeit
- Alternative: Nutzung des DH-Verfahrens:
  - idR in Kombination mit DSS, wg. Man-in-the-Middle
  - Server generiert temporären DH-Public_Key: K_{DH}
  - Server signiert K_{DH} mit seinem DSS-Signaturschlüssel
  - Server erhält die DH-Parameter des Clients
  - Server und Client berechnen den gemeinsamen DH-Schlüssel K_{SC}
  - der Schlüssel K_{SC} ist dann das Pre-Master-Secret
- Anonymes DH: nur Austausch von DH-Parametern, aber
  - anfällig gegen Man-in-the-Middle

Client-Authentifikation

Client-Authentifikation: optional

- Server initiiert die Client-Authentifizierung, d.h. er entscheidet, ob Client-Authentifizierung notwendig ist
- Server sendet Certificate_Request-Nachricht
- Client signiert die Daten aus dem Request: RSA(Request_Data, $Private\_Key_{Client}$)
- Client sendet Client-Zertifikat und signierten Request
- Server prüft die Signatur

Session: das Handshake-Protokoll unterstützt:

- die erneute Verwendung von Session-IDs, z.B. Web-Zugriffe
- das Aushandeln neuer Ciphersuites während einer Session
- Re-Keying einer Verbindung: Handshake mit frischen Nonces

Probleme, Grenzen von SSL

keine Verbindlichkeit: keine Signaturen

- Probleme beim Zusammenwirken mit Firewalls sind möglich
- Vielzahl von vorkonfigurierten CAs: Vertrauen?
  - Werden Zertifikate geprüft? Warnungen ignoriert?
  - Stimmt Name im Zertifikat mit URL überein?
- Sichere Speicherung des Master-Secrets, der Session-Keys?
- Hacker-Angriffe auf CAs (u.a. DigiNotar 2011): Haftung für CA-Betreiber? (EU-Regelung in Vorbereitung)
- Häufig fehlende Zertifikatprüfung bei non-Browse-Anwendungen (Cloud-Storage, PayPal Payment etc) durch fehlerhafte Nutzung von SSL-Libraries: anfällig gegen MitM Angriffe

IPsec

6.2 IPsec

IPsec (IETF-Standard)

Standardprotokoll für Schicht 3

6.2.1 Ipsec-Grundlagen

- **Sicherheitsarchitektur** für Internetprotokolle, seit 1995 verschiedene RFCs: 4302 (AH), 4303 (ESP), 4306 (IKE) …
- optionaler Einsatz im IPv4 und verpflichtend für Ipv6
- 2 Modi: Transport und Tunnel-Modus (insbes. bei VPN)

Ziel: Gewährleistung von Schutzzielen auf IP-Ebene

- Authentizität des Datenursprungs
- Vertrauliche Datenübertragung (Payload)
- Integrität u. Schutz vor Replay-Attacken
- Schlüsselmanagement: Erneuerung, Austausch

IPsec in a Nutshell

IPsec in a Nutshell

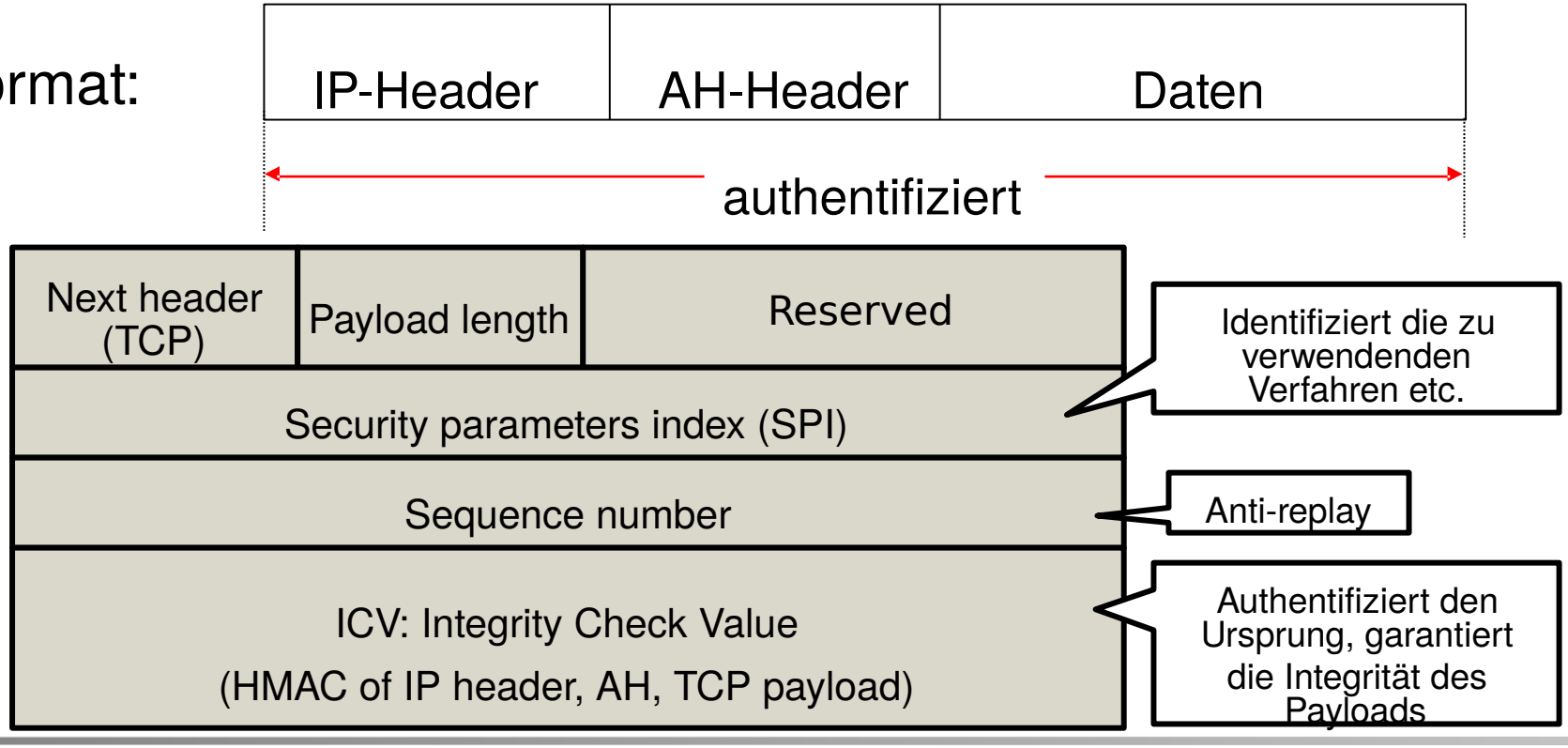
- **Protokolle:**
  - AH und ESP: Integrität, Vertraulichkeit angewandt auf einzelne IP-Pakete
  - IKE: Aushandeln der Verfahren und Schlüssel
- **Regelwerk (Policy):** muss konfiguriert werden
  - welche Pakete, von wem, zu wem, mit welchen Verfahren
  - Security-Policy-Database (SPD) (pro Ipsec-Rechner)
- **Speicherung der Sicherheitsparameter:**
  - Security-Association (SA): Verfahren, Schlüssel, ….
  - Verfahren, Schlüssel pro ‚Verbindung‘ : Zustand in IP!
  - Inbound, Outbound-Datenbanken

6.2.2 IP-Protokollerweiterungen: AH und ESP

Authentication-Header-Protokoll (AH) RFC 4302

- Authentizität, Integrität des Datenursprung und Payloads
- Verhinderung von Replay-Attacken über Sequenznummern

Format:

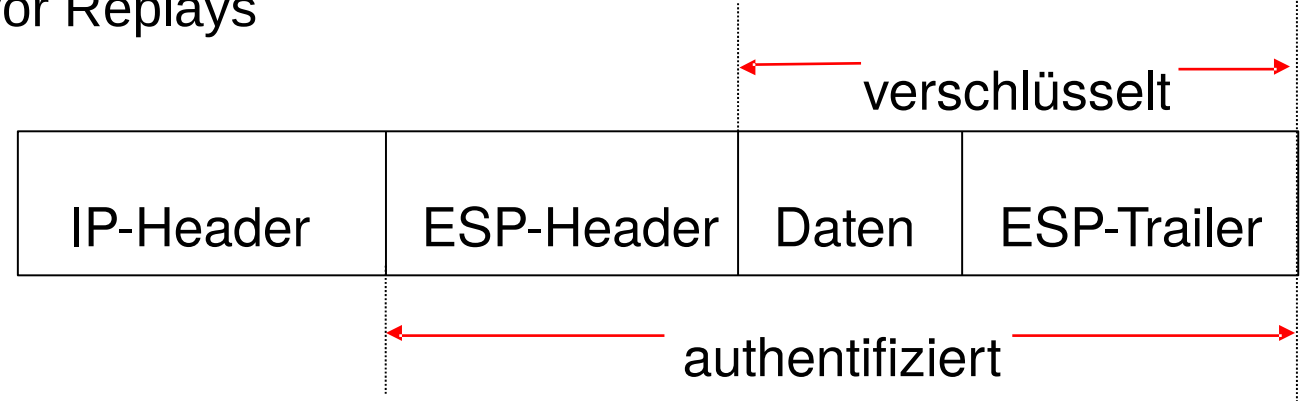


ESP

Encapsulating Security-Payload (ESP) RFC 4303

- Vertraulichkeit der Daten des IP-Datenpakets, Symmetrische Blockchiffre, auch NULL-Algorithmus zulässig
- Authentisierung des Payloads mittels HMAC
- (optional) Schutz vor Replays

Format:



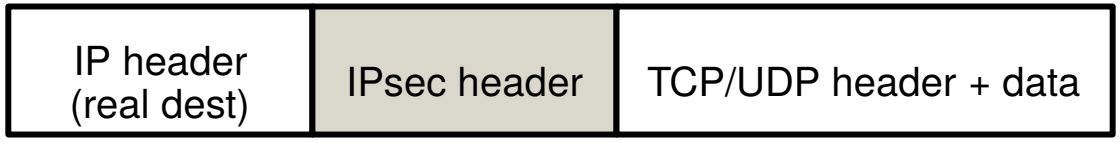
Durch die Nutzung von IPsec wird ein **IP-Paket verändert**:

- zusätzliche Header
- Verschlüsseln und/oder Hashen der Daten (Payload)

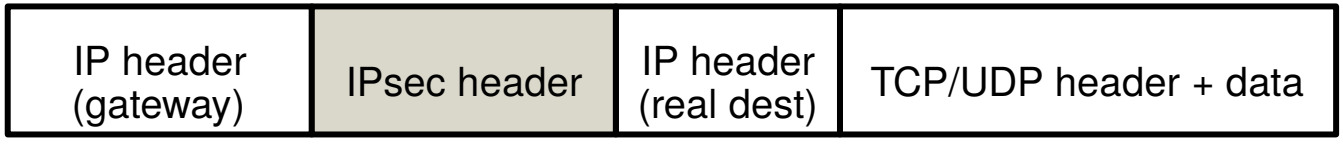
IPsec-Modi

IPsec-Modi: Transport und Tunnel-Modus

- Transport-Modus: Absichern des Payloads, **Konsequenz?**



- Tunnel-Modus: Einkapseln sowohl des IP-Headers als auch des Payloads in Ipsec-Paket



Geschachtelte Tunnels sind möglich

Beispiel: Sichere Verbindung über eine Firewall:

- Äußerer Tunnel durch das Internet zum Gateway (Firewall)
- Zugriff auf Server hinter der Firewall:
  - innerer Tunnel, um die Verbindung von der Firewall zum Server abzusichern

6.2.3 Datenstrukturen und Datenbanken

Security-Association-Datenstruktur (SA)

- SAs werden in den SA-Datenbanken von A bzw. B verwaltet
- eine SA enthält alle benötigten Informationen für IPsec- Verbindung zwischen zwei Rechnern A und B
- SAs haben nur unidirektionale Gültigkeit
- für jedes Protokoll (AH/ESP) wird eine eigene SA benötigt
- SAs werden vorab idR über IKE ausgehandelt und erstellt
- Eine SA wird beim erstmaligen Verbindungsaufbau angelegt

SPI:

- jedes IPsec Paket enthält einen Index (SPI), der auf einen SA-Eintrag in der SA-DB des Empfängers verweist
- diese SA enthält die notwendigen Verarbeitungsinformationen

Eine SA enthält u.a. folgende Informationen:

- IP-Adresse des Empfängers
- AH-Informationen:
  - Algorithmus, Schlüssel, Schlüssellebenszeit
- ESP-Informationen:
  - Algorithmen, Schlüssel, Initialwerte, Lebenszeiten, …
- Lebenszeit der SA: Zeitintervall oder Bytecounter, nach dem die SA erneuert oder terminiert werden muss und Angabe, welcher dieser Aktionen auszuführen ist
- Sequenzzähler (ab IKEv2 64 Bit, vorher 32 Bit) AH bzw. ESP
- Modus: Transport oder Tunnel
- Anti-Replay-Window, um einkommende Replays zu erkennen
- Security-Level (z.B. für Multi-level sichere Systeme)

Security-Policy-Database

Security-Policy-Database: Pro IPsec-Rechner eine SPD:

- eine SPD legt Regeln für den Umgang mit IP-Paketen fest
  - individuelle Regeln für eingehende (inbound) und
  - für ausgehende Pakete (outbound)
- jede Regel wird über einen Selektor spezifiziert:
  - ein Selektor ist für ein IP-Paket anzuwenden, wenn die Einträge des IP Pakets mit den Selektorfeldern matchen
  - ist ein Selektor anwendbar (match), dann enthält die SPD die mit dem IP-Paket durchzuführende Aktion

From	To	Protocol	Port	Policy
1.1.1.1	2.2.2.2	TCP	80	Transport ESP with 3DES

Selektoren

Selektoren, die einen SPD-Eintrag bestimmen: u.a.

- die IP-Adresse bzw. Adressbereiche oder Wildcard des Empfängers bzw. des Senders
- Sender-, Empfänger-Ports bzw. Liste von Ports oder Wildcard
- Name: z.B. DNS-Name, X.500 Distinguished Name,
- Unterschied zwischen outbound/inbound Paketen:
  - bei ausgehenden Paketen werden beim Anwenden der Regel die erforderlichen SAs (AH, ESP) etabliert (IKE)
  - inbound Paketen: Paket verwerfen, falls keine SA vorhanden
- **Aktionen:** bypass: direktes Weiterleiten des Pakets
  - apply: IPsec muss angewandt werden, Verweis auf SA
  - discard: das Paket muss vernichtet werden

Ablauf

Ablauf beim Versand eines IPsec-Pakets von A nach B

- A sucht SA für Verbindung mit B in SA-Datenbank von A
- A verwendet die dort angegebenen Informationen: Verschlüsselung, Hashen, MAC berechnen, …
- Aus SA-Eintrag: SPI zum Empfangen des Pakets in SA_DB_B
- A trägt in IPsec-Header diesen SPI ein

SPD von A:

From	To	Protocol	Port	Policy
1.1.1.1	2.2.2.2	TCP	80	Transport ESP with 3DES

Ausgehende SA-DB von A:

From	To	Protocol	SPI	SA-Eintrag
1.1.1.1	2.2.2.2	ESP	10	3DES key

Fazit IPsec

Fazit IPsec:

- Konfigurieren von IPsec-Policies ist sehr komplex
  - fehleranfällig: viele Optionen, viele Freiheitsgrade
  - ggf. Nutzung schwacher Modi, unsichere Auswahl
- Konfigurierungsvarianten führen zu Interoperabilitätsproblemen
- IKE-Pakete werden über UDP übertragen: unzuverlässig und wird von einigen Firewalls blockiert (ggf. kein Aushandeln mögl.)
- Interworking von IPsec und Firewalls ist problematisch Aber: bei korrekter Nutzung hoher Sicherheitsgrad erreichbar

Nächste Woche

Weitere Sicherheitsprotokolle
Ipsec, PGP, DNSsec