VIII. Key Management

oder Wie benutzt man Public-Key-Kryptographie?

VIII.1. PKI (Public-Key-Infrastruktur)

VIII.1.1. „Definition“

Ein digitales Zertifikat ordnet einem öffentlichen Schlüssel eindeutig einen Inhaber zu.
Eine Public-Key-Infrastruktur ist die Gesamtheit der organisatorischen Maßnahmen, die für eine vertrauenswürdige Ausgabe und Verifikation von Zertifikaten notwendig ist.

Aufgaben einer PKI sind z.B. die Ausstellung, Verteilung, Prüfung und der Widerruf digitaler Zertifikate. (Impliziert dies, dass der Inhaber seinen eigenen secret key kennt?)

VIII.2. Beispiel: X.509-Zertifikat

(die folgende Liste ist nicht vollständig)

1. Versionsnummer $\in \{1, 2, 3\}$, je nachdem, welche erweiterten Elemente vorhanden sind
2. Seriennummer, wird von Certification Authority (CA) gewählt und sollte pro CA eindeutig sein
3. Signaturalgorithmus und Parameter
4. Distinguished Name (DN) des Ausstellers
5. Gültigkeitsdauer
6. Distinguished Name des Inhabers
7. Public-Key-Informationen: Algorithmus, Parameter, public key
8. Erweiterungen
9. Signatur auf 1.-8.

VIII.3. Certificate Revocation

mittels Certificate Revocation Lists (CRL) (Listen einer CA, die widerrufene Zertifikate enthalten)

Diese enthalten ein Ausstellungsdatum und das Datum der nächsten geplanten CRL. Zertifikate bleiben auf der CRL, bis sie abgelaufen sind. (Genauer: Online Certificate Status Protocols)

VIII.4. Web of Trust

$\widehat{=}$ sozialem Netzwerk von „Mini-CAs“.

- Key-Signing-Partys zum Signieren
- Jeder kann selbst festlegen, welchen anderen Usern er vertraut.

Im Einzelfall unsicherer (schlecht geschützte CAs), aber Angriffe skalieren nicht so.

VIII.5. TLS (Transport Layer Security)

- Standard im Internet
- Nachfolger von SSL-Protokollen

VIII.5.1. Ablauf

FIXME: Bild TLS, S. 31
Jeder kann eine key-renegotiation beantragen und das Protokoll startet dann neu.

VIII.5.2. Besonderheiten

- Schutz gegen Downgrade (kein Versionswechsel während Protokollablauf)
- die Nachricht „Finish“ enthält einen Hash über alle bisherigen Protokollnachrichten der Session
- die verwendete pseudozufällige Funktion teilt den Input in zwei Hälften, hasht mit MD5 und SHA1 und XORs die Ergebnisse $\rightarrow$ das bleibt sicher, auch wenn beispielsweise MD5 Schwächen zeigt („Robust Combiner“)

VIII.6. Key Renegotiation Attack

auf TLS

VIII.6.1. Ziel

Client und Server sollen sich in unterschiedlichen Anwendungskontexten befinden, was der Angreifer ausnutzen kann (z.B. Client will Passwort senden, Server will nächste Nachricht an Adresse XY weiterleiten)

VIII.6.2. Ablauf

FIXME: Bild Key Renegotiation Attack, S. 32