

---

## Konzepte der Signatur

Nachrichten aus Nachrichtenraum:  $m \in M \subseteq A_1^*$ .

Signaturen aus Signaturenraum:  $\sigma \in S \subseteq A_2^*$ .

Schlüssel sind aus Schlüsselräumen:  $d \in K_1 \subseteq A_3^*$ ,  $e \in K_2 \subseteq A_4^*$ .

Signierungsverfahren  $\mathcal{S} : K_1 \times M \rightsquigarrow S$ .

Verifizierungsverfahren  $\mathcal{V} : K_2 \times M \times S \rightarrow \{0, 1\}$ .

Signatur von Nachricht  $m$  mit Schlüssel  $d$ :  $\sigma \leftarrow \mathcal{S}(d, m)$ .

Verifizierung von  $m, \sigma$  mit Schlüssel  $e$ :  $f \leftarrow \mathcal{V}(e, m, \sigma)$ .

$\mathcal{S}$  und  $\mathcal{V}$  sind effiziente Verfahren (z.B. Programme).

$\mathcal{S}$  und  $\mathcal{V}$  dürfen probabilistisch sein (dürfen den Zufall verwenden).

$\mathcal{S}$  kann mehrdeutig sein ( $\rightsquigarrow$ ).

---

1

19. Oktober 2006

---

## Symmetrisch

Symmetrisches Signatursystem (secret key, MAC):

- $d = e$  (oder  $d$  leicht aus  $e$  berechenbar).
- Sender und Empfänger haben  $e$  als gemeinsames Geheimnis.
- Gemeinhin als Message Authentication Code (MAC) bezeichnet, meistens deterministisch.

Erweiterte Nachrichten werden verschickt:  $(m, \sigma)$  wo  $\sigma \leftarrow \mathcal{S}(d, m)$ .

Integrität der erweiterten Nachrichten  $(m, \sigma)$  wird von  $\mathcal{V}$  mit  $\mathcal{S}(d, m) = \sigma$  überprüft.

Hashfunktion: MAC ohne geheimen Schlüssel.

---

2

19. Oktober 2006

---

## Asymmetrisch

(Asymmetrisches, public key) Signatursystem:

- $d$  geheim,  $e$  öffentlich.
- $d$  schwer oder gar nicht aus  $e$  berechenbar.
- Signierer besitzt  $e$  und  $d$ .
- Signierer verwendet  $d$  zum Signieren.
- Empfänger verwendet  $e$  zum Verifizieren.

---

3

19. Oktober 2006

---

## Angriffe und Sicherheitsmodelle

Das Prinzip von Kerckhoff soll gelten

(vielleicht ein Grund dagegen weniger hier).

Ziele des Angreifers:

- Existentielle Fälschung: Der Angreifer berechnet eine Signatur für eine Nachricht.
- Universelle Fälschung: Der Angreifer kann Signaturen für jede beliebige Nachricht berechnen.
- Total break: Der Angreifer berechnet den geheimen Schlüssel des Signierers.

---

4

19. Oktober 2006

---

## Angriffe und Sicherheitsmodelle

Informationen (Fähigkeiten) des Angreifers in aufsteigender Reihenfolge:

- Key-only Angriff: Der Angreifer kennt nur den öffentlichen Schlüssel des Signierers.
- Known-Signature Angriff: Der Angreifer erhält Nachrichten und die zugehörigen Signaturen.
- Chosen-Message Angriff: Der Angreifer kann sich die Nachrichten aussuchen und erhält die zugehörigen Signaturen.

Den letzte Variante gibt es auch in adaptiver Form.

Stärkstes Sicherheitsmodell: Sicherheit bezüglich existenzieller Fälschung unter adaptiven Chosen-Message Angriffen.

---

5

19. Oktober 2006

---

## Verwendung von Hashfunktionen

Hashfunktion  $H : M \rightarrow \{0,1\}^*$ .

Im allgemeinen wird nur ein Hashwert  $H(m)$  und nicht  $m$  selbst signiert.

- Effizienter, da  $H(m)$  viel kürzer als  $m$  ist.
- Beweisbare Sicherheit von in der Praxis relevanten Verfahren (allerdings im Zufallsorakelmodell, RO).

Offenbar muß  $H$  kollisionsfrei sein, man kann keine zwei Nachrichten  $m_1, m_2$  mit  $H(m_1) = H(m_2)$  berechnen.

---

6

19. Oktober 2006

---

## Brute-Force (exhaustive search) Angriff

Dieser Angriff kann bei Kryptosystemen mit komplexitätstheoretischer Sicherheit immer ausgeführt werden.

Für Verschlüsselungsverfahren zum Beispiel:

1. Alle Entschlüsselungsschlüssel ausprobieren.
2. Schauen, ob das Entschlüsselte Sinn macht (d.h. Ausnutzen von Redundanz in beispielsweise geschriebener Sprache etc.) oder ob der Schlüssel zu evtl. bekannten Klartext- und Chiffretextpaaren paßt.

Gegenmaßnahme im allgemeinen:

- Ausreichend großen Schlüsselraum haben, somit Aufwand für Angriff groß genug machen.

---

7

19. Oktober 2006

---

## Bemerkungen

Die meisten heute verwendeten Kryptosysteme sind **unsicher**,

- wenn ein Angreifer unbegrenzte Rechenleistung hat.

Darüberhinaus sind die meisten heute verwendeten Public-Key Verfahren bereits **unsicher**,

- wenn ein Angreifer auch nur gewisse Operationen besonders schnell ausführen kann (Quantencomputer).

Es gibt keine Kryptosysteme mit komplexitätstheoretischer Sicherheit, deren Sicherheit mathematisch bewiesen wurde (dies würde im Endeffekt  $\mathcal{P} \neq \mathcal{NP}$  implizieren).

---

8

19. Oktober 2006

---

## Vergleich Public-Key und Secret-Key Kryptographie

Public-Key:

- größere Funktionalität, z.B. Schlüsselaustausch, Signaturen, etc.
- basiert auf mathematischen Problemen (aus der Zahlentheorie).

Secret-Key:

- effizienter in Verschlüsselung (ebenso MAC und Hashfunktionen).

Hybridverfahren:

- Austausch geheimer, sogenannter Sitzungsschlüssel (Session Keys) mittels Public-Key Kryptographie.
- Danach Verwendung symmetrischer Verfahren.

---

## Protokolle

Sind definierte Abfolgen von Kommunikations- und Berechnungsschritten zwischen mindestens zwei Teilnehmern zum Lösen kryptographischer Aufgaben.

Beispiel: Challenge-Response Protokoll zur Identifikation.

- Alice besitzt öffentlichen Schlüssel  $e$  und geheimen Schlüssel  $d$  eines Signaturverfahrens.
- Bob soll überzeugt werden, daß Alice  $d$  kennt, ohne daß  $d$  an Bob geschickt wird.

1. Bob wählt zufällige Nachricht  $m$  und schickt sie an Alice.
2. Alice schickt Bob ihre Signatur der Nachricht  $m$  unter  $d$ .
3. Bob ist überzeugt, wenn die Signatur bzgl.  $e$  gültig ist.

---

## Protokolle

Eine Angreiferin Eve ohne Kenntnis von  $d$  müßte Signaturen von  $m$  fälschen, um Bob zu überzeugen.

Angreifer können aktiv oder passiv sein (substitute, replay, insertion attacks).

Sicherheitsbeweis eines Protokolls:

- Man zeigt, daß man mit Hilfe eines Angreifers zugrunde liegende kryptographische Primitive angreifen kann beziehungsweise daß man damit ein zugrundeliegendes, "schwieriges" mathematisches Berechnungsproblem lösen kann.