Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

08.06.2017

Überblick

Überblick

Erinnerung

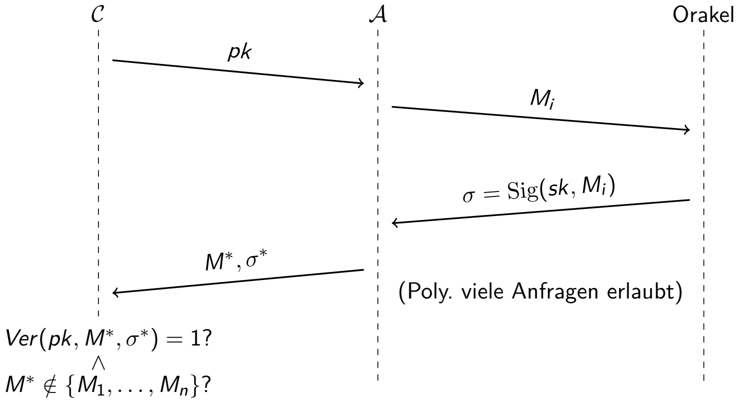
- Grundidee:

$$\texttt{Alice}_{pk} \xleftarrow{(M,\sigma)} \texttt{Bob}_{sk}$$

- Signieren: $\sigma \leftarrow \mathrm{Sig}(sk, M)$
- Verifizieren: $\mathrm{Ver}(pk, M, \sigma) \in \{0, 1\}$
- Standard-Sicherheitsbegriff: EUF-CMA

Erinnerung: EUF-CMA

- Herausforderer $\mathcal{C}$ führt $(pk, sk) \leftarrow \mathrm{Gen}(1^k)$ aus.
- $\mathcal{C}$ stellt $\mathrm{Sig}(sk, \cdot)$-Orakel für $\mathcal{A}$ bereit.



Überblick

RSA als Signaturschema

- Erinnerung RSA-Verschlüsselung:

$$pk = (N, e) \qquad sk = (N, d)$$
$$\mathrm{Enc}(pk, M) = M^e \bmod N$$
$$\mathrm{Dec}(sk, C) = C^d \bmod N$$

- (**Warnung/Erinnerung:** in dieser Form nicht sicher!)
- Betrachte RSA als Signaturschema:

$$\mathrm{Sig}(sk, M) = M^d \bmod N$$
$$\mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad M = \sigma^e \bmod N$$

Eigenschaften von RSA

- RSA als Signaturschema:

$$\mathrm{Sig}(sk, M) = M^d \bmod N$$
$$\mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad M = \sigma^e \bmod N$$

- **PKE→Sig-Konversion nicht allgemein**
  - Allgemeiner lassen sich Nachrichten nicht unbedingt „zuerst ent-, dann wieder verschlüsseln“ (Datentypproblem)
  - Außerdem muss/sollte Enc nicht deterministisch sein
- Zahlreiche Sicherheitsprobleme (nachfolgend)
- Aber: RSA-Signaturen können „repariert“ werden

Probleme von RSA-Signaturen

$$\mathrm{Sig}(sk, M) = M^d \bmod N$$
$$\mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad M = \sigma^e \bmod N$$

- Problem: unsinnige Nachrichten können signiert werden
  1. Wähle *zuerst* Signatur $\sigma \in \mathbb{Z}_N$ beliebig
  2. Setze dann $M := \sigma^e \bmod N$
  3. Damit ist σ gültige RSA-Signatur für M
- Bricht EUF-CMA-Sicherheit, (künstliche) problematische Anwendungen denkbar

Probleme von RSA-Signaturen

$$\mathrm{Sig}(sk, M) = M^d \bmod N$$
$$\mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad M = \sigma^e \bmod N$$

- Weiteres Problem: Homomorphie von RSA
  1. Angenommen, $\sigma_i = M_i^d \bmod N$ bekannt (für einige i)
  2. Setze dann $\sigma := \prod_i \sigma_i = \prod_i M_i^d = (\prod_i M_i)^d \bmod N$
  3. Damit ist σ gültige RSA-Signatur für $M := \prod_i M_i$
- Neue Signaturen lassen sich aus bekannten berechnen
- Bricht auch EUF-CMA-Sicherheit, (leicht weniger künstliche) problematische Anwendungen denkbar

Lösungsvorschläge?

- **Diskussion:** Wie könnte man RSA-Signaturen „reparieren“?

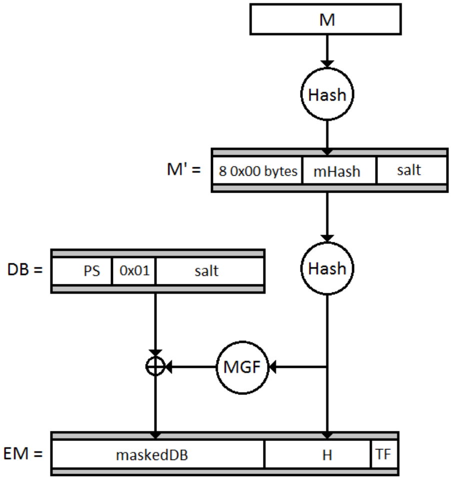
RSA-PSS

- (RSA-)PSS: „Probabilistic Signature Scheme“
- Idee von RSA-PSS: Vorverarbeitung (Padding) der Nachricht:

$$\mathrm{Sig}(sk, M) = (\mathrm{pad}(M))^d \bmod N$$
$$\mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad \sigma^e \bmod N \quad \text{gültiges } \mathrm{pad}(M)$$

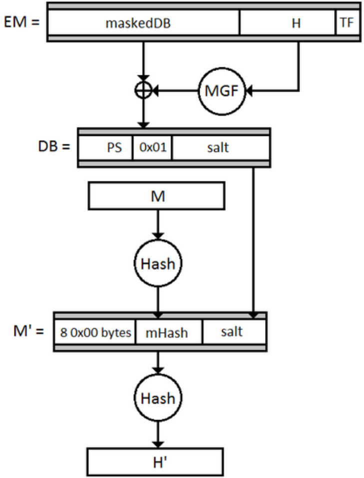
RSA-PSS

- Padding einer Nachricht: $\mathrm{pad}(M)$ (Quelle: Wikipedia)



RSA-PSS

- Verifikation einer gepaddeten Nachricht: (Quelle: Wikipedia)



Sicherheit von RSA-PSS

- RSA-PSS heuristisch (mit idealen H, MGF) EUF-CMA-sicher, sofern RSA-Funktion schwer zu invertieren
  - Jeder EUF-CMA-Angreifer *muss* RSA-Funktion invertieren
- RSA-PSS (wie RSA-OAEP) Teil von PKCS#1
- Bester bekannter Angriff: N faktorisieren (Zahlkörpersieb)
- Parameterwahl wie bei RSA-OAEP (somit $\log_2(N) \approx 2048$)
- Festes, kleines e (z.B. $e = 3$) möglich $\Rightarrow$ effiziente Verifikation

Überblick

Hin zu ElGamal-Signaturen

- Signaturverfahren über zyklischer Gruppe $\mathbb{G} = \langle g \rangle$
- Wie bei Verschlüsselung: $pk = (\mathbb{G}, g, g^x)$, $sk = (\mathbb{G}, g, x)$
- Erster Versuch:

$$\mathrm{Sig}(sk, M) = a \quad \text{mit} \quad a \cdot x = M \bmod |\mathbb{G}|$$
$$\mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad (g^x)^a = g^M$$

- **Frage:** warum problematisch?

ElGamal-Signaturen

- Erster Versuch: $pk = (\mathbb{G}, g, g^x)$, $sk = (\mathbb{G}, g, x)$, und

$$\mathrm{Sig}(sk, M) = a \quad \text{mit} \quad a \cdot x = M \bmod |\mathbb{G}|$$

- Zweiter Versuch (**ElGamal-Signaturen**): setze

$$\begin{gathered} a := g^e \quad \text{für zufälliges } e \\ b \quad \text{als Lösung von} \quad a \cdot x + e \cdot b = M \bmod |\mathbb{G}| \\ \mathrm{Sig}(sk, M) = (a, b) \\ \mathrm{Ver}(pk, M, \sigma) = 1 \quad :\Longleftrightarrow \quad (g^x)^{a} a^b = g^M \end{gathered}$$

- Achtung: $a = g^e$ wird sowohl als $\mathbb{G}$-Element als auch als Exponent interpretiert (zunächst nur für $\mathbb{G} = \mathbb{Z}_p^*$ gedacht)

Angriffe auf ElGamal-Signaturen

- Erinnerung: $\sigma = (a = g^e, b)$ mit

$$a \cdot x + e \cdot b = M \bmod |\mathbb{G}|$$

- **Nie** zweimal dasselbe e (für verschiedene M) verwenden:
  - Sonst: $(a = g^e, b, M)$ und $(a' = g^{e'} = a, b', M')$ bekannt mit

$$a \cdot x + e \cdot b = M \bmod |\mathbb{G}|$$
$$a \cdot x + e \cdot b' = M' \bmod |\mathbb{G}|$$

  - Daraus folgt $e = (M - M')/(b - b') \bmod |\mathbb{G}|$ und damit x
- Wird bei zufälliger Wahl von e vernachlässigbar oft passieren

Angriffe auf ElGamal-Signaturen

- Erinnerung: $\sigma = (a = g^e, b)$ mit

$$a \cdot x + e \cdot b = M \bmod |\mathbb{G}|$$

- ElGamal-Signaturen wie RSA nicht EUF-CMA-sicher:
  - $(a, b) = (g^x, -a)$ gültig für $M = a \cdot x + e \cdot b = 0 \bmod |\mathbb{G}|$
- Randomisierung $\Rightarrow$ Signaturen für unsinnige Nachrichten
  1. Wähle c zufällig
  2. Setze $a := g^c g^x = g^{c+x}$ und $b := -a \bmod |\mathbb{G}|$
  3. Damit ist (a, b) gültige Signatur für die Nachricht

$$M := a \cdot x + e \cdot b = a \cdot x - a(c + x) = -ac \bmod |\mathbb{G}|$$

- **Frage:** wie kann ElGamal „repariert“ werden?

Überblick

Hash-Then-Sign-Paradigma

Theorem (Sicherheit des Hash-Then-Sign-Paradigmas)

Sei $(\mathrm{Gen}, \mathrm{Sig}, \mathrm{Ver})$ *EUF-CMA-sicher und* H *eine kollisionsresistente Hashfunktion. Dann ist das durch*

$$\mathrm{Gen}'(1^k) = \mathrm{Gen}(1^k)$$
$$\mathrm{Sig}'(sk, M) = \mathrm{Sig}(sk, \mathrm{H}(M))$$
$$\mathrm{Ver}'(pk, M, \sigma) = \mathrm{Ver}(pk, \mathrm{H}(M), \sigma)$$

erklärte Signaturverfahren EUF-CMA-sicher.

- Beweis wie im MAC-Fall (Reduktion)
- **Vorteil:** Angriffe, die Signaturen für unsinnige Nachrichten liefern, müssen nicht mehr funktionieren

Überblick

Der Digital Signature Algorithm

- Erinnerung ElGamal: $\sigma = (a = g^e, b)$ mit

$$a \cdot x + e \cdot b = M \bmod |\mathbb{G}|$$

- Digital Signature Algorithm (DSA): $\sigma = (a = g^e, b)$ mit

$$a \cdot x + e \cdot b = \mathrm{H}(M) \bmod |\mathbb{G}|$$

  (hierbei H kollisionsresistente Hashfunktion)
- EUF-CMA-Sicherheit gegenwärtig unklar
- Vorgeschlagene Gruppen:
  - $\mathbb{G} \subset \mathbb{Z}_p^*$ oder $\mathbb{G} \subset \mathbb{F}_q^*$ (wobei $|\mathbb{G}| \approx 2^{2048}$)
  - Elliptische Kurven: $\mathbb{G} = \mathbf{E}(\mathbb{F}_q)$ (mit $|\mathbb{G}| \approx 2^{200}$)

Überblick

Zusammenfassung digitale Signaturen

- Digitale Signaturen sichern Integrität/Authentizität
  - Asymmetrisches Gegenstück zu MACs
- Sicherheitskriterium: EUF-CMA
- Wichtige Verfahren: RSA(-PSS), ElGamal/DSA
  - **Wichtig:** Verfahren nicht ungepadded/ungehasht benutzen!

Aktuelle Forschung digitale Signaturen

- Gitterbasierte Signaturen (asymptotische Effizienz)
  - NTRU(-Verschlüsselung, -Signaturen)
  - Delegierbare Signaturen
  - Problem: momentan noch sperrige Signatur-/Schlüsselgrößen
- Pairingbasierte Signaturen (effizient, platzsparend)
  - Automorphe Signaturen (Zero-Knowledge-kompatibel)
  - Delegierbare Signaturen

Überblick

Überblick

Motivation

- Ziel: gemeinsamen geheimen Schlüssel K aushandeln

$$\texttt{Alice} \quad \longleftrightarrow \quad \texttt{Bob}$$

- Kommunikationskanal unsicher, aber K soll geheim bleiben
- Verschiedene Szenarien denkbar:
  - Altes K schon vorhanden (frisches K gewünscht)
  - Secret-Key-Infrastruktur (mit Schlüsselzentrale)
    - `Alice` hat K_A, `Bob` hat K_B, Schlüsselzentrale kennt K_A und K_B
  - Public-Key-Infrastruktur
    - pk_A und pk_B öffentlich, `Alice` kennt sk_A, `Bob` kennt sk_B
  - `Alice` und `Bob` haben gemeinsames Passwort
  - `Alice` und `Bob` haben keine weiteren Informationen
    - Prinzipbedingt unsicher gegen aktive Angriffe

Überblick

Symmetrische Verfahren

- Szenario mit Schlüsselzentrum KC:

$$\begin{array}{ccc} \mathtt{Alice}_{K_A} & \longleftrightarrow & \mathtt{Bob}_{K_B} \\ \updownarrow & & \updownarrow \\ & \mathtt{Key\ Center\ (KC)}_{K_A,K_B} & \end{array}$$

- Alice kennt K_A, Bob kennt K_B, KC kennt K_A und K_B
- Kommunikation mit KC möglich, soll aber minimiert werden
- Werkzeug der Wahl: symmetrische Verschlüsselung $(\mathrm{Enc}, \mathrm{Dec})$
  - Grund: Effizienz, Lösungen allein mit $(\mathrm{Enc}, \mathrm{Dec})$ möglich