# Seminar Kryptographie und Datensicherheit

## Andere Protokolle für digitale Unterschriften

Markus Gusowski, Andrea Hentschke

Wintersemester 2006/2007

# Gliederung

# Provably Secure Signature Schemes

- beweisbar sicher
- oft als theoretisches Modell

- One-Time-Signature
  - sicher, wenn nur eine Nachricht signiert

# Lamport Signature Scheme

- One-Time-Signature
- nutzt Einwegfunktion
- einfach, aber unpraktikabel

# Lamport Signature Scheme

Kryptosystem:

- $\mathcal{P} = \{0,1\}^k$, $k \in \mathbb{N}^+$ (Nachrichten)
- Einwegfunktion: $f : Y \rightarrow Z$
- $\mathcal{A} = Y^k$ (Signatur)
- $y_{i,j} \in Y$ zufällig gewählt, $1 \leq i \leq k$, $j = 0,1$
- $z_{i,j} = f(y_{i,j})$, $1 \leq i \leq k$, $j = 0,1$
- $K = (y_{i,j}, z_{i,j} : 1 \leq i \leq k, j = 0,1)$; $y_{i,j}$ geheim, $z_{i,j}$ öffentlich
- $sig_K(x_1, ..., x_k) = (y_{1,x_1}, ..., y_{k,x_k})$
- $ver_K((x_1, ..., x_k), (a_1, ..., a_k)) = true \Leftrightarrow f(a_i) = z_{i,x_i}$, $1 \leq i \leq k$

# Lamport Signature Scheme - Beispiel

Einwegfunktion $f(x) = \alpha^x \mod p$
konkret: $f(x) = 3^x \mod 7879$

Nachricht: $x = (1, 1, 0)$

geheimer Schlüssel: (zufällig)

| $y_{1,x_j}$ | $y_{2,x_j}$ | $y_{3,x_j}$ |
|---|---|---|
| $y_{1,0} = 5831$ | $y_{2,0} = 803$ | $y_{3,0} = 4285$ |
| $y_{1,1} = 735$ | $y_{2,1} = 2467$ | $y_{3,1} = 6449$ |

öffentliche Schlüssel: (aus y berechnet)

| $z_{1,x_j}$ | $z_{2,x_j}$ | $z_{3,x_j}$ |
|---|---|---|
| $z_{1,0} = 2009$ | $z_{2,0} = 4672$ | $z_{3,0} = 268$ |
| $z_{1,1} = 3810$ | $z_{2,1} = 4721$ | $z_{3,1} = 5731$ |

# Lamport Signature Scheme - Beispiel

Signatur für $x$: $sig_K(1,1,0) = (y_{1,1}, y_{2,1}, y_{3,0}) = (735, 2467, 4285)$

Verifikation:

$3^{735} \mod 7879 = 3810$
$3^{2467} \mod 7879 = 4721$
$3^{4285} \mod 7879 = 268$

# Lamport - Sicherheit

- Sicherheit von One-Time Eigenschaft abhängig
  - Nachricht 1: $(1, 0, 1)$ mit Signatur $(y_{1,1}, y_{2,0}, y_{3,1})$
  - Nachricht 2: $(0, 1, 0)$ mit Signatur $(y_{1,0}, y_{2,1}, y_{3,0})$
  - hier: geheimer Schlüssel nach Verifikation vollständig bekannt
  - $\Rightarrow$ für alle anderen Nachrichten kann Signatur gefälscht werden
- Beweis der Sicherheit (Widerspruchsbeweis)
  - Bedingungen: key-only attack (vgl. one-time Eigenschaft)
  - Annahme: $f$ ist bijektive Einwegfunktion, Elemente des public key disjunkt

# Lamport - Sicherheitsbeweis

- Angenommen, Angreifer kann existenzielle Fälschung produzieren
  - $\rightarrow$ LAMPORT-FORGE($\mathcal{Z}$) mit $\mathcal{Z}$ public key
- Konstruiere Algorithmus, der Urbild $y$ eines $z \in Z$ berechnet ($f : Y \rightarrow Z$):

**LAMPORT-PREIMAGE($z$)**

wähle zufällig $i_0 \in \{1, ..., k\}$ und $j_0 \in \{0, 1\}$
wähle zufällig public key $\mathcal{Z} = (z_{i,j} | i \in \{1, ..., k\}; j \in \{0, 1\})$ mit $z_{i_0,j_0} = z$
$((x_1, ..., x_k), (a_1, ..., a_k)) \leftarrow$ LAMPORT-FORGE($\mathcal{Z}$)
if $x_{i_0} = j_0$
    then return $(a_{i_0})$
    else *fail*

# Lamport - Sicherheitsbeweis

- Wahrscheinlichkeit, dass LAMPORT-PREIMAGE ein Urbild findet: 1/2
- d.h. LAMPORT-PREIMAGE existiert
- $\longrightarrow$ Widerspruch mit Einwegeigenschaft von $f$
- $\Rightarrow$ Es kann LAMPORT-FORGE nicht geben

- Schema unpraktikabel
- z.B. $f$: Exponentiation in Restklassen, sicher mit 1024 bit
- $\Rightarrow$ Signaturlänge $= 1024 \cdot$ Nachrichtenlänge

# Full Domain Hash

- man will auch Nachrichten beliebiger Länge effizient signieren
- Lösung: Hash-Then-Sign
  - Verwendung einer kryptographischen Hashfunktion $H : \{0,1\}^* \rightarrow \{0,1\}^L$
  - statt der Nachricht unterschreibt man dann tatsächlich den Hashwert
- Full Domain Hash (Bellare & Rogaway): nutzen einer Trapdoor-Einwegpermutation um ein sicheres Signaturschema im Random Oracle Model zu konstruieren
  - Wertebereich des Random Oracle ist gleich dem Definitionsbereich der Trapdoor-Einwegpermutation

# Full Domain Hash

Kryptosystem:

- $\mathcal{F}$ Familie von Trapdoor-Einwegpermutationen, so dass $f : \{0,1\}^k \to \{0,1\}^k, \forall f \in \mathcal{F},\ k \in \mathbb{N}^+$
- zufällige Hashfunktion $G : \{0,1\}^* \to \{0,1\}^k$
- $\mathcal{P} = \{0,1\}^*$ und $\mathcal{A} = \{0,1\}^k$
- $K = \{(f, f^{-1}, G) : f \in \mathcal{F}\}$; $f^{-1}$ geheim, $(f, G)$ öffentlich
- $sig_K(x) = f^{-1}(G(x))$, $x \in \{0,1\}^*$
- $ver_K(x, y) = true \Leftrightarrow f(y) = G(x)$, $y = (y_1, ..., y_k) \in \{0,1\}^k$

# Full Domain Hash - Sicherheit

notwendige Sicherheitsbedingung:

kollisionsresistente Hashfunktion, d.h. es ist unmöglich zwei Inputs $M \neq M'$ zu finden mit $H(M) = H(M')$

$\Rightarrow$ FDH ist sicher im Zufallsorakel-Modell, wenn $f$ nicht invertiert werden kann

# Full Domain Hash

RSA basierte Implementierung:

- $f^{-1}$: RSA Signierungsfunktion, z.B. decryption
- $f$: RSA Verifizierungsfunktion, z.B. encryption
- $k = 1024$
- statt Orakel $G$ SHA-1 (liefert 160 bit message digest)
- 160 bit Hashwert muss zu 1024 bit aufgefüllt werden

# Undeniable Signatures – Motivation

- Alice signiert Angebot, schickt es an Bob's Firma
- Alice's Konkurrent Klaus bekommt (irgendwie) die Nachricht
- Klaus verifiziert Signatur, unterbietet daraufhin das Angebot
- $\longrightarrow$ Alice will verhindern, dass Klaus die Signatur verifizieren kann
- Lösung: Alice muss beim verifizieren "helfen"
  - Challenge-and-response Protokoll
- Nachteil: Alice könnte (korrekte) Verifizierung verweigern
- hier: Alice kann Angebot abstreiten
- Lösung: Alice kann beweisen, falls Fälschung vorliegt
  - Disavowal-Protokoll
  - "Undeniable" (unabstreitbar), da Alice Signatur nicht abstreiten kann
  - Bei herkömmlichen Signaturen "integriert"

# Undeniable Signatures

- 3 Komponenten
  - Signierungsalgorithmus
  - Verifikationsprotokoll
  - Disavowal-Protokoll
- erstmals 1989 - Chaum, van Antwerpen

# Chaum-van Antwerpen

- Basiert auf multiplikativer Gruppe über $\mathbb{Z}_p$
- $p = 2q + 1$ Primzahl, so dass: $q$ Primzahl und diskreter Logarithmus in $\mathbb{Z}_p$ schwer lösbar
- Öffentlich: $p$, $\alpha$, $\beta$
  - $\alpha \in \mathbb{Z}_p^*$ mit Ordnung $q$
  - $\beta = \alpha^a \pmod p$
- Geheim: $a$ mit $1 \leq a \leq q - 1$
- $G$ sei multiplikative Untergruppe von $\mathbb{Z}_p^*$, der Ordnung $q$ (d.h. $|G| = q$, $\alpha$ ist generierendes Element)
- $\mathcal{P}$ (Nachrichten) $= \mathcal{A}$ (Signaturen) $= G$
- $\mathcal{K}$ (Schlüssel) $= \{(p, \alpha, a, \beta) | \beta \equiv \alpha^a \pmod p\}$

# Chaum-van Antwerpen

- Signierung:
  - Schlüssel $K = (p, \alpha, a, \beta)$, Nachricht $x \in G$
  - Signatur $y = sig_K(x) = x^a \bmod p$

- Verifikation:
  1. Bob wählt zufällig $e_1, e_2 \in \mathbb{Z}_p^*$
  2. Bob berechnet $c = y^{e_1}\beta^{e_2} \bmod p$ und schickt es an Alice ("challenge")
  3. Alice berechnet $d = c^{a^{-1} \bmod q} \bmod p$ und schickt es zurück ("response")
  4. Bob akzeptiert die Gültigkeit g.d.w. $d \equiv x^{e_1}\alpha^{e_2} \pmod p$

# Chaum-van Antwerpen

- $p = 2q + 1$, denn: Berechnung Inverse in $G$, da $\mathcal{P} = \mathcal{A} = G \Rightarrow G$ möglichst groß

- Zu zeigen: Signatur gültig $\Rightarrow$ Bob akzeptiert die Signatur
  - Annahme: Alice und Bob ehrlich
  - $c^{a^{-1} \bmod q} \equiv x^{e_1}\alpha^{e_2} \pmod p$
  - $(y^{e_1}\beta^{e_2})^{a^{-1} \bmod q} \equiv x^{e_1}\alpha^{e_2} \pmod p$

# Chaum-van Antwerpen – Sicherheit

- Zu zeigen: Signatur ungültig $\Rightarrow$ Bob lehnt die Signatur ab
  - (d.h., Alice kann Bob nicht dazu bringen, eine gefälschte Signatur zu akzeptieren)
  - Annahme: nur Bob ehrlich
  - viele ($q$) mögliche $c$ zu jeder Signatur (vgl. $e_1, e_2$ zufällig)
  - $\Rightarrow$ Wahrscheinlichkeit das Alice Bob täuschen kann: $1/q$
  - unconditional security

# Chaum-van Antwerpen – Disavowal

1. Bob wählt zufällig $e_1, e_2 \in \mathbb{Z}_p^*$
2. Bob berechnet $c = y^{e_1}\beta^{e_2} \mod p$ und schickt es an Alice ("challenge")
3. Alice berechnet $d = c^{a^{-1} \mod q} \mod p$ und schickt es zurück ("response")
4. Bob überprüft, dass $d \not\equiv x^{e_1}\alpha^{e_2} \pmod p$
5. Bob wählt zufällig $f_1, f_2 \in \mathbb{Z}_p^*$
6. Bob berechnet $C = y^{f_1}\beta^{f_2} \mod p$ und schickt es an Alice ("challenge")
7. Alice berechnet $D = C^{a^{-1} \mod q} \mod p$ und schickt es zurück ("response")
8. Bob überprüft, dass $D \not\equiv x^{f_1}\alpha^{f_2} \pmod p$
9. Bob sieht $y$ als Fälschung an, g.d.w. $(d\alpha^{-e_2})^{f_1} \equiv (D\alpha^{-f_2})^{e_1} \pmod p$

# Entrusted Undeniable Signatures

- Erstmals S. Park, K. Lee, D. Won; 1995
- "Lügendetektor-Problem":
  - Alice schickt Dokumente mit Beschuldigungen an Zeitung
  - Alice erlaubt nur Reporter Verifizierung
  - Alice's Vorgesetzter Bob vermutet, dass Alice Dokumente geschickt hat
  - Bob verlangt disavowal von Alice
  - Alice weigert sich $\Rightarrow$ Bob kann nicht wissen, ob Alice lügt
  - aber: Bob interpretiert dies als "schuldig", kündigt Alice
- Lösung:
  - Disavowal-Protokoll nur von Drittem, Gericht Carol ausführbar
  - Streit so "gerichtlich" schlichtbar

# Entrusted Undeniable Signatures

- Idee:
  - Alice verwendet zufälligen private key $r \cdot a$ (statt $a$) zur Signierung
  - Carol hat RSA private und public key
  - Alice legt sich auf $r$ fest (*commitment* $c$, Verschlüsselung mit Carol's public key)
  - $\rightarrow$ nur Carol kann Disavowal-Protokoll ausführen, da ihr private key benötigt wird, um $r$ zu berechnen

# Threshold Entrusted Undeniable Signatures

- Erstmals S. Kim, D. Won; 2004
- "Lügendetektor-Problem" Fortsetzung:
  - Bob besticht Carol (um Alice als "schuldig" zu entlarven)
- Lösung:
  - Nur (mind.) $t$ von $n$ "Geschworenen" können Disavowal-Protokoll ausführen
- Idee:
  - Geschworene haben alle public-private-key-Paare
  - Doppelte Exponentiation

# Fail-stop Signatures

- erhöhte Sicherheit gegen sehr mächtige Angreifer (Möglichkeit einer gefälschten Signatur)
- Alice kann mit hoher Wahrscheinlichkeit eine Fälschung nachweisen
- wenn eine Fälschung gefunden, wird das System angehalten

- hier: van Heyst and Pedersen Signature Scheme
  - One-Time Signature Scheme
  - Signierungs- und Verifikationsalgorithmus, sowie proof of forgery Algorithmus

# van Heyst and Pedersen Signature Scheme

- Primzahlen: $p$ und $q$, so dass $p = 2q + 1$
- diskrete Logarithmus Problem in $\mathbb{Z}_p$ schwer zu lösen
- $\alpha \in \mathbb{Z}_p^*$ Element der Ordnung $q$
- $1 \leq a_0 \leq q - 1$, $\beta = \alpha^{a_0} \mod p$
- $p, q, \alpha, \beta, a_0$ von zentraler Stelle gewählt
- $p, q, \alpha, \beta$ sind öffentlich; $a_0$ ist geheim (auch vor Alice)
- $\mathcal{P} = \mathbb{Z}_q$ und $\mathcal{A} = \mathbb{Z}_q \times \mathbb{Z}_q$
- $K = (\gamma_1, \gamma_2, a_1, a_2, b_1, b_2)$, wobei $a_1, a_2, b_1, b_2 \in \mathbb{Z}_q$,
  $\gamma_1 = \alpha^{a_1}\beta^{a_2} \mod p$ und $\gamma_2 = \alpha^{b_1}\beta^{b_2} \mod p$
- $(\gamma_1, \gamma_2)$ ist öffentlich, $(a_1, a_2, b_1, b_2)$ ist geheim
- $sig_K(x) = (y_1, y_2)$, $x \in \mathbb{Z}_q$,
  wobei $y_1 = a_1 + xb_1 \mod q$ und $y_2 = a_2 + xb_2 \mod q$
- $ver_K(x, y) = true \Leftrightarrow \gamma_1\gamma_2^x \equiv \alpha^{y_1}\beta^{y_2} \pmod p$,
  $y = (y_1, y_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$

# Fail-stop Signatures - Sicherheit

Definition Zwei Schlüssel $(\gamma_1, \gamma_2, a_1, a_2, b_1, b_2)$ und $(\gamma_1', \gamma_2', a_1', a_2', b_1', b_2')$ sind äquivalent, wenn $\gamma_1 \equiv \gamma_1'$ und $\gamma_2 \equiv \gamma_2'$.

- $q^2$ Schlüssel in jeder Äquivalenzklasse

Lemma 1 Wenn $K$ und $K'$ äquivalent sind und $ver_K(x, y) = true$, dann gilt $ver_{K'}(x, y) = true$.

Lemma 2 Wenn $K$ ein Schlüssel ist und $y = sig_K(x)$, dann gibt es genau $q$ Schlüssel $K'$, die äquivalent zu $K$ sind, so dass $y = sig_{K'}(x)$.

Lemma 3 Wenn $K$ ein Schlüssel ist, $y = sig_K(x)$ und $ver_K(x', y') = true$, wobei $x' \neq x$, dann gibt es höchstens einen zu $K$ äquivalenten Schlüssel $K'$, so dass $y = sig_{K'}(x)$ und $y' = sig_{K'}(x')$.

# Fail-stop Signatures - Sicherheit

Folgerung:

- zu einer gültigen Signatur $y$ für $x$ gibt es $q$ mögliche Schlüssel, die $x$ ebenfalls mit $y$ signieren würden
- aber für jedes $x' \neq x$ liefern diese $q$ Schlüssel $q$ verschiedene Signaturen für $x'$

Theorem Wenn $sig_K(x) = y$ gegeben und $x' \neq x$, dann kann Oscar $sig_K(x')$ mit einer Wahrscheinlichkeit von $1/q$ berechnen.

$\Rightarrow$ unbedingte Sicherheit

# Fail-stop Signatures - Proof of forgery

- zu einer Signatur $y$ für Nachricht $x$ kann Oscar die Signatur $y'$ von Alice für $x'$ nicht berechnen
- vorstellbar: Oscar kann gefälschte Signatur $y'' \neq sig_K(x')$ berechnen, die verifiziert werden kann
- Alice kann mit Wahrscheinlichkeit $1 - 1/q$ beweisen, dass es eine Fälschung ist
- proof of forgery: $a_0 = log_\alpha \beta$ (nur der zentralen Stelle bekannt)
- Annahme, dass Alice den diskreten Logarithmus nicht berechen kann

# Fail-stop Signatures - Proof of forgery

Annahme: Paar $(x', y'')$, so dass $ver_K(x', y'') = true$ und $y'' \neq sig_K(x')$

$$\gamma_1 \gamma_2^{x'} \equiv \alpha^{y_1''} \beta^{y_2''} \pmod{p},\ y'' = (y_1'', y_2'')$$

Alice berechnet eigene Signatur für $x'$: $y' = (y_1', y_2')$

$$\gamma_1 \gamma_2^{x'} \equiv \alpha^{y_1'} \beta^{y_2'} \pmod{p}$$
$$\alpha^{y_1''} \beta^{y_2''} \equiv \alpha^{y_1'} \beta^{y_2'} \pmod{p}$$

$\beta = \alpha^{a_0} \mod p$ eingesetzt:

$$\alpha^{y_1'' + a_0 y_2''} \equiv \alpha^{y_1' + a_0 y_2'} \pmod{p}$$
$$y_1'' + a_0 y_2'' \equiv y_1' + a_0 y_2' \pmod{q}$$
$$y_1'' - y_1' \equiv a_0 (y_2' - y_2'') \pmod{q}$$

$y_2' \not\equiv y_2'' \pmod{q}$, da $y''$ gefälscht, folglich ex. $(y_2' - y_2'')^{-1} \pmod{q}$
und $a_0$ lässt sich leicht berechnen:

$$a_0 = log_\alpha \beta = (y_1'' - y_1')(y_2' - y_2'')^{-1} \pmod{q}$$

# Group Signatures

- Erstmals Chaum, van Heyst; 1991
- Motivation:
  - Firma mit vielen Rechnern und Netzwerkdruckern
  - Ein Drucker in jeder Abteilung
  - Nur Mitarbeiter einer Abteilung dürfen "Abteilungsdrucker" nutzen
  - $\rightarrow$ Drucker muss "Mitgliedschaft" des Nutzers sicherstellen
  - $\rightarrow$ Wahrung der Privatsphäre: Identität des Nutzers geheim
  - $\rightarrow$ aber: Bei übermäßiger Nutzung Offenlegung der Identität um Rechnung auszustellen
- Lösung:
  - Nur Gruppenmitglieder können Nachrichten signieren
  - Empfänger kann Signatur als gültige Signatur der Gruppe verifizieren ohne Mitglied zu kennen
  - Im Streitfall kann Signatur "geöffnet" werden (Identität des Signierers feststellbar)

# Ring Signatures

- Vereinfachung Group Signatures
- Möglichkeit eine Gruppe von möglichen Unterzeichnern zu spezifizieren ohne die Identität des eigentlichen Unterzeichners offen zu legen
- keine Manager, keine Koordination
- keine vorherbestimmten Gruppen
- einzige Annahme: jedes Mitglied hat einen öffentlichen Schlüssel eines Standard Signatur Schemas

- Unterzeichner wählt eine beliebige Gruppe mit möglichen Unterzeichnern
- Signatur berechnen mit eigenem geheimen Schlüssel und den anderen öffenlichen Schlüsseln
- kein Einverständnis der Anderen notwendig

# Noch mehr Protokolle für Digitale Signaturen

Provably Secure Signatures
Group Signatures
Short Group Signatures
Blind Signatures
Unanticipated Signatures
Undeniable Signatures
Convertible Undeniable Signatures
Designated Confirmer Signatures
Nominative Signatures
Convertible Nominative Signatures
Entrusted Undeniable Signatures
Threshold Entrusted Undeniable Sig.
Zero-Knowledge Undeniable Signatures
Zero-Knowledge Nominative Signatures

Ring Signatures
Deniable Ring Signatures
Short Ring Signatures
Threshold Ring Signatures
General Access Ring Signatures
Identity-based Ring Signatures
Identity-based Threshold Ring Sig.
Separable Ring Signatures
Linkable Ring Signatures
Verifiable Ring Signatures
Accountable Ring Signatures
Bilinear Ring Signatures
Bilinear Threshold Ring Signatures
Fail-stop Signatures

# Quellen


- Johannes Buchmann. Einführung in die Kryptographie. Springer Verlag 2003
- Douglas R. Stinson. Cryptography Theory and Practice Second Edition. Chapman & Hall/CRC 2002
- M. Bellare, P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, Seiten 62-73. ACM Press, 1993.
- M. Bellare, P. Rogaway. The exact security of digital signatures: how to sign with RSA and Rabin. *Lecture Notes in Computer Science*, 1070 (1996), 399-416. (Advances in Cryptology - EUROCRYPT '96.)
- D. Chaum, H. van Antwerpen. Undeniable signatures. *Lecture Notes in Computer Science*, 435 (1990), 212-216. (Advances in Cryptology - CRYPTO '89.)

# Quellen


- S. Kim, D. Won. Threshold Entrusted Undeniable Signature. Information Security and Cryptology - ICISC 2004, LNCS 3506, pp. 195-203. Springer-Verlag, 2005.
- E. van Heyst, T. P. Pedersen. How to make efficient fail-stop signatures. *Lecture Notes in Computer Science*, 658 (1993),366-377. (Advances in Cryptology - EUROCRYPT '92.)
- D. Chaum, E. van Heyst: Group Signatures. LNCS547, p.257-265, Springer-Verlag 1992, Berlin. (EUROCRYPT '91)
- R. L. Rivest, A. Shamir, Y. Tauman. How to Leak a Secret: Theory and Applications of Ring Signatures (www.mit.edu/∼tauman/ringsig-book.pdf)
- Stefan Lucks. Vorlesungsmaterial Kryptographie I (SS '05), Uni-Mannheim.