Einführung in Zahlungssysteme mit Doublespender-Identification

Eric Auer und Andreas Franke

6. Juli 2001

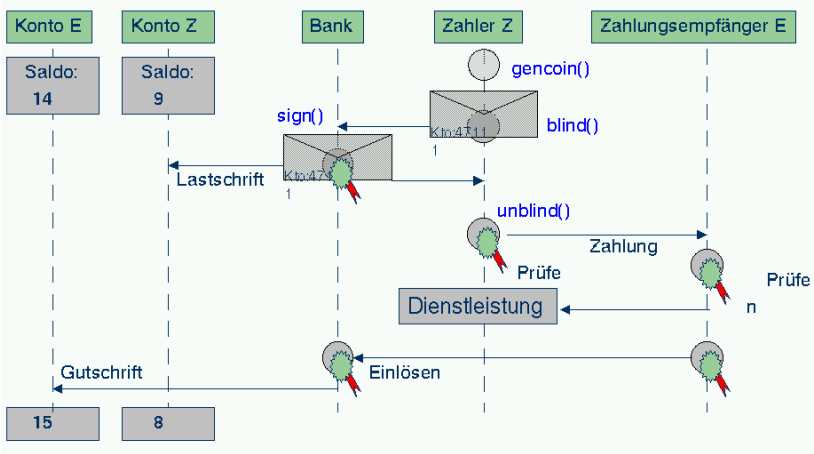
Gliederung

- Motivation: anonymous offline Probleme
- Wiederholung:
  Blinde Signaturen und Anonyme Münzen
- Doublespending zerstört Anonymität
- Challenge - Response Systeme
- Schnorr Identifikation
  - Sicherheitsaspekte
- Schnorr Signaturen
- Praktisch: viel komplexere Systeme
- Ausblick: verbleibende Probleme

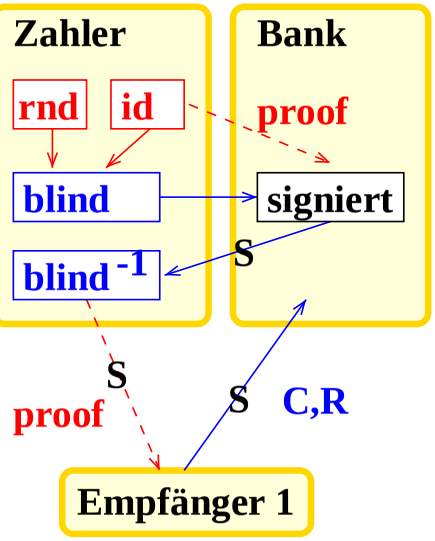
Motivation

- Offlinesysteme: Bezahler hat eMünzen
- eGeldbeutel im Zweifel nicht sicher vor Bezahler
  ⇒ Anonyme Münzen könnten dupliziert werden
- Lösung *Bedingte* Anonymität: geht verloren, wenn eine Münze mehrfach ausgegeben wird!
- Wie geht das?
- ⇒ dieser Vortrag! In etwa:
  OTP plus Challenge verhüllen Id,
  2 Challenges zu 1 OTP nicht mehr

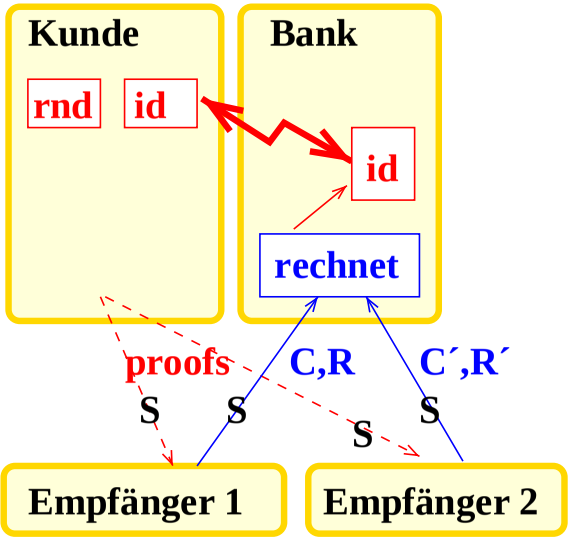
Wiederholung blinde Signaturen



Wiederholung eMünzen



Doublespending enttarnt Betrüger



Zu den Challenges

- Szenario: Zahler + zwei Empfänger gegen Bank
  - gleiche Challenge von beiden Empfängern: $C' = C$
  - Bank kann nicht doppelt gutschreiben!
  - Soll einfach der erste das Geld bekommen?
    Nein, sonst:
    - Zahler + Empfänger 2 betrügen Empfänger 1:
    - C wiederverwenden, und schneller zur Bank
- Lösung:
  - Challenges enthalten Id des Zahlungsempfängers
  - Gutschriften nur auf dessen Konto

Idee für Responses (1)

- *One-time pad*
  1. response: $R = k$ (one-time pad)
  2. response: $R' = id \oplus k$
- gewünschte Eigenschaft ok:
  - R oder R' alleine verrät nichts über id
  - R und R' zusammen offenbaren id vollständig
- Problem:
  nur zwei challenges möglich

Idee für Responses (2)

- *Lineare Gleichung in zwei Variablen*: $c_1 id + c_2 k$

  $\Longrightarrow$ jetzt viele challenges c_1, c_2 möglich
- gewünschte Eigenschaft ok:
  - eine response: jede id möglich
  - zwei Gleichungen bestimmen beide Variablen eindeutig (falls linear unabhängig)
- Idee verwendet in Brands' system ($\rightsquigarrow$ Ausblick):

  zwei Gleichungen in vier Variablen pro Response

  $\Longrightarrow$ effiziente Verifikation

Schnorr Identifikation

$|G_q| = q, \qquad G_q < Z_p^*, \qquad < g > = G_q, \qquad x \in Z_q$

	Prover (Secret: x)	Common input: $q, p, g, h = g^x$	Verifier
1. Random variant	$w \in_R Z_q$ $a := g^w$	$\xrightarrow{a}$	
2. Challenge		$\xleftarrow{c}$	$c \in_R Z_q$
3. Response	$r := w + cx$	$\xrightarrow{r}$	Verify $g^r = ah^c$

Sicherheitsideen (1)

- *"Proof of knowledge"*: Prover besteht Verifikation mit signifikanter Wahrscheinlichkeit $\Longrightarrow$ er kennt x
- Beweisidee:
  - Ann.: Prover hat a gesendet,

    kann zwei challenges c_1 und c_2 beantworten
  - Seien r_1, r_2 seine responses
  - x und w eindeutig (diskr. Logs von h und a)
  - responses r_1 und r_2 werden nur akzeptiert, wenn $r_1 = w + c_1 x$ und $r_2 = w + c_2 x \pmod q$
  - auflösen: $(r_1 - r_2) = (c_1 - c_2)x \pmod q$
  - $\Longrightarrow$ Prover kennt x.

Sicherheitsideen (2)

- *"Kein nützliches Wissen nach außen"*:
- auch nach vielen Identifikationen:

  Angreifer erfährt nichts sinnvolles über x

  (x immer gleich, aber w jedesmal verschieden)
- Grundidee:

  für jedes c ist w ein one-time pad auf x

  (in der response r die der Angreifer bekommt)
- aber: kein wirklicher Beweis

Vom Ident.-Protokoll zur Signatur

- Gegeben:
  - 3-Schritt Identifikations-Protokoll
  - mit nur einer zufälligen Challenge

  $\Longrightarrow$ Signaturverfahren
- Trick: Signierer berechnet c selbst: $c = hash(m, a)$ (m: zu signierende Nachricht), wobei $hash$
  - zumindest einweg
  - hoffentlich pseudozufällig

    $\Longrightarrow$ nur dann so gut wie Identifikations-Protokoll
  - öffentlich (wg. Signaturtest)

  $\Longrightarrow$ Sicherheit nicht bewiesen!
- Signaturtest hier: $g^r = ah^c$ mit $c = hash(m, a)$

Weitere Verbesserungen

- Bisher: *Wissen* bewiesen („Habe Konto zu pk g^x")
- Chaum-Pedersen Signaturen: zusätzlich *Nachricht* m signiert
- Chaum-Pedersen Signaturen mit Blinding: Zahler erfährt nicht, *welches* m er *wann* als m' signiert hat.
- $\Rightarrow$ so nur wie RSA Signaturen verwendbar
- Trick: Id so in m codieren, dass noch in m' vorhanden
- Ergebnis (14.5.3.E): Anonym, offline, doublespending enttarnt Betrüger (Zahler hat 1 Münze, jede blinde Sig. durch Bank kostet, mit 1 Sig. doppelt zahlen gibt Münzklartext mit TypNo und Id)

Probleme / Ausblick

- Münzklartext enthält nur secret Id und secret TypNo (für alle Münzen eines Wertes gleich, beide von Zahler)
- Münze missbraucht: Id durch (nicht mehr) secret Id klar
- Aber unklar, wie *viele* Münzen dieses Benutzers wie *oft* missbraucht.
- Doppelzahlung bricht Verschlüsselung: Betrogene können mit gleicher Münze weiterbetrügen!
- $\Rightarrow$ Schadenserfassung kaum möglich