Financial Cryptography '2000

21-25 february 2000 - Anguilla

Self-Scrambling Anonymizers

David Pointcheval
Département d 'Informatique
ENS - CNRS

David.Pointcheval@ens.fr http://www.di.ens.fr/~pointche

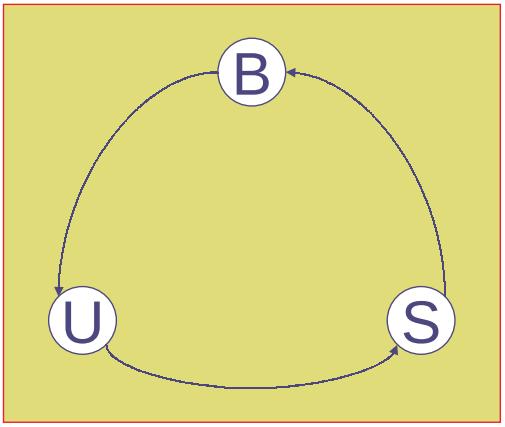
Overview

- Introduction to E-cash
- Weak/Strong Anonymity
- A New Scenario
- Self-Scrambling Anonymizer
- An Example: DL-based
- Security Analysis
- Conclusion

Introduction

E-cash usually involves 3 participants:

- the bank
- the user
- the shop

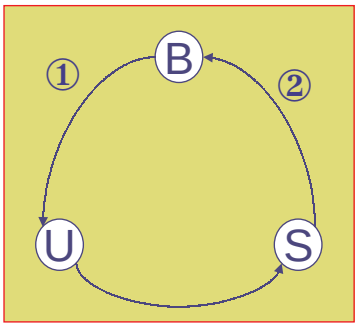


Classical Scenario

Use of e-coins:

- the coin is obtained from the bank
  $\Rightarrow$ withdrawal
- the user buys something with it
  $\Rightarrow$ spending
- the shop gives it back to the bank
  $\Rightarrow$ deposit

Anonymity



① B knows the coin it gives to U

② B sees the coin deposited by S

⇒ B learns the transaction U-S

Leakage of private data

❶ cannot be avoided

❷ usually avoided: blind signatures

Over-Spending

- Duplication of a coin:
  ⇒ possibility of spending it many times
- Two scenarios:
  - the bank is on-line during the spending
    → immediate detection
  - the bank is off-line
    → late detection

because of anonymity:
who is the bad guy?

Identity in the Coin

- Chaum-Fiat-Naor (1988):
  identity embedded in the coin such that
  - ID remains **concealed** after one use
  - ID is **revealed** after twice
- Still allows “perfect crime”:
  blackmailing without any risk!

  ⇒ **revokable anonymity**

Revokable Anonymity

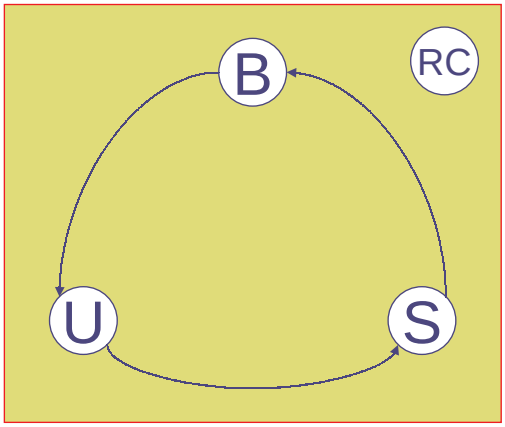
New participant: **Revocation Center**

→ can revoke anonymity

⇒ reveal the link between

- a coin and a user
- a transaction and a user

when the need arises



Strong Anonymity

Problem of hiding:

- the link transaction-user
  → **untraceability**
- the link transaction-transaction of one user → **unlinkability**

Strong notion:

any adversary cannot learn the link, but with negligible probability

Weak Anonymity

Weak notion:

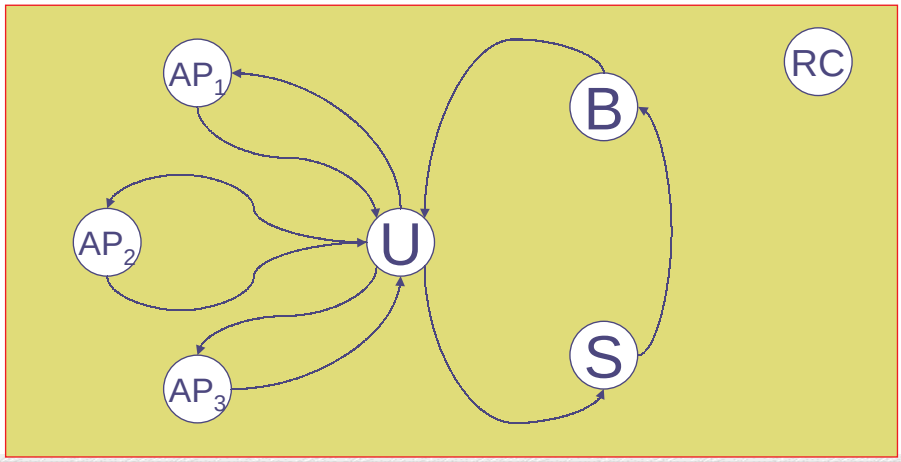
an adversary may know a link, however, he cannot prove it

His knowledge is non-transferable

New Scenario

New participants: **Anonymity Providers**

→ help the user to get anonymous coins (still revocable by RC)



New Scenario

Usually: the bank "blindly" certifies a coin after an intricate proof of its validity (*i.e.* that revocability is possible by RC)

→ restrictive blind signatures

Here: the bank certifies $c=\mathbf{E}_{RC}(I_U ; r)$ after the view of both I_U and r

Coin = $(c, Cert_c)$

Advantages of $c = E_{RC}(I_U ; r)$

- **revocation**: very easy
  - just a decryption $I_U = D_{RC}(c)$
  - proof of it
- **ownership**

  = proof of knowledge of (sk_U, r)

  (sk_U, r) is the secret key related to c

Self-Scrambling Anonymity

But the bank will recognize c,… Anonymity?

- the user **"scrambles himself"**

  c into $c' = E_{RC}(I_U ; r')$

  $\Rightarrow$ c' unknown to the bank

  but c' is not certified!!
- the AP certifies c' when he knows that
  - c is valid: with $Cert_c$
    with a proof of ownership
  - $c' \sim c$: with a proof of equivalence

Proof of Equivalence

- to achieve, at least, weak anonymity this proof must be "non-transferable"
  $\Rightarrow$ *e.g. Zero-Knowledge Proof*
- to get evidences of over-spending (when a coin is used at least twice) this proof must be "non-repudiable"
  $\Rightarrow$ *e.g. Undeniable Proof*

An Example: DL-based

- Revocation Center: $\mathrm{pk}_{RC} = Y = g^{\mathrm{sk}_{RC}}$
- User: $\mathrm{pk}_U = I_U = g^{\mathrm{sk}_U}$
- **Coin**: El Gamal Encryption
  $$c = (a = g^r, b = Y^r I_U)$$
- **Ownership**: Okamoto's variant
  $\rightarrow$ knowledge of (r, sk_U) s.t. $b = Y^r g^{\mathrm{sk}_U}$

$u, v \in \mathbf{Z}_q$ and $t = Y^u g^v \bmod p$ $\xrightarrow{t}$

$\xleftarrow{e}$ $e \in \mathbf{Z}_{2^k}$

$\alpha = u - e \cdot r \bmod q$
$\beta = v - e \cdot \mathrm{sk}_U \bmod q$ $\xrightarrow{\alpha,\beta}$ $t \stackrel{?}{=} Y^\alpha g^\beta b^e \bmod p$

Self-Scrambling (1/2)

$c = (a = g^r, b = Y^r I_U)$ and $c' = (a' = g^{r'}, b' = Y^{r'} I_U)$
with $r' = r + \rho$

- Proof of equivalence of ciphertexts:
  $\log_g a'/a = \log_Y b'/b$
- Proof of ownership:
  signature of the message
  $m = (d=h^\rho$, AP, date, etc)
  with the secret (r, sk_U) related to $b = Y^r g^{sk_U}$
  $\Rightarrow$ the owner of c knows $\rho = \log_h d$

Self-Scrambling (2/2)

$c = (a = g^r, b = Y^r I_U)$ and $c' = (a' = g^{r'}, b' = Y^{r'} I_U)$
with $r' = r + \rho$

- Confirmation: proof of equality
  $\log_h d = \log_g a'/a = \log_Y b'/b$
  - *Interactively*:
    Zero-Knowledge proof
    which just convinces the AP
  - *Non-Interactively*:
    Designated-Verifier Signature

Anonymity

- None, if not required
  $\Rightarrow$ no extra cost
- Weak Anonymity:
  with at least one AP
  (under the DDH assumption)
- Strong Anonymity:
  with at least one honest AP

Security Analysis

- **Impersonation:** the secret sk_U is only used in ZK or NIZK proofs
  $\Rightarrow$ never leaked
  But required for any use of a coin
- **Revocation:** with the coin $c = (a,b)$
  $\Rightarrow I_U = b / a^{sk_{RC}}$
  with the proof of $\log_g Y = \log_a b/I_U$
  But under evidences of fraud…

Evidences

Two of some

- **spending:** signature with b,
  of some coin $c = (a,b)$, on a purchase
- **anonymizing:** signature with b,
  of some coin $c = (a,b)$, on
  $m = (d=h^{\rho}$, AP, date, etc)
  $\Rightarrow$ related coin $c' = (a',b')$ such that
  $\log_h d = \log_g a'/a = \log_Y b'/b$
  to be *blacklisted*

Fraud Detection

Counterfeit Money:

- **duplication** of a coin: over-spending
- **creation** of money by an AP

when a coin is used, the receiver

- the shop for a spending
- the AP for anonymizing

asks for its value to the certifier, the AP,
which is seen as a middleman

over-spent coin: asked ***many*** times

Conclusion

New tool for anonymity

- **efficiency**
  - no extra-cost, if no anonymity required
  - few exponentiations (~10) per anonymizing
- **security**
  - anonymity related to semantic security
    $\Rightarrow$ based on DDH
- **practicability: profitability**
  - AP gives c' of just 99.9% of the value of c