Micromint

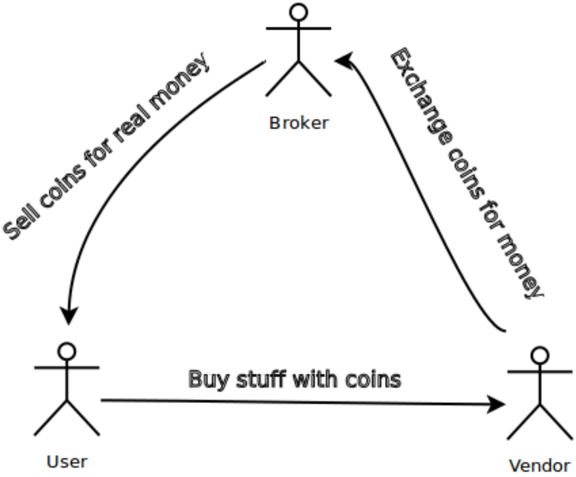
Quentin Delhaye

Université Libre de Bruxelles
INFO-F-514 Protocols, cryptanalysis and mathematical cryptology

March 19th 2014

1. Outline of the scheme
2. Basic Implementation
3. Security Concerns
4. Conclusion

- Off-line micropayement scheme.
- Rivest and Shamir in 1995.
- No public key operations.



Broker

Sell coins for real money

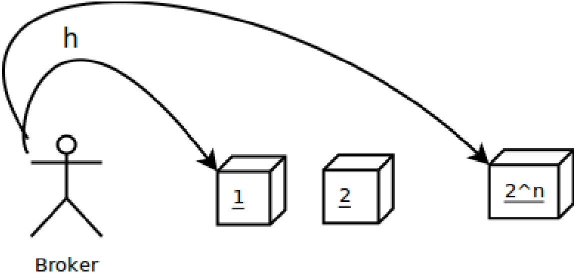
Exchange coins for money

Buy stuff with coins

User

Vendor

- K-way collision based coins.
- Input x on m bits, output y on n bits.
- $(x_1, x_2, \dots x_k)$ s.t. $\mathrm{h}(x_1) = \mathrm{h}(x_2) = \dots = \mathrm{h}(x_k) = \mathrm{y}$
- First collision needs $2^{n(k-1)/k}$ inputs.
- Examining c times as many values, $1 \leq c \leq 2^{n/k}$, gives c^k collisions.



- Ball x, bin of index y.
- Tossing $k2^n$ balls, each with $1/2$ chance to be part of a coin.
- Each bin with $\geq k$ balls can produce a coin.

- Storage cost is higher than computation cost.
- Reduce the amount of good balls by fixing the high order bits.
- $n = t + u$ and t is fixed to an arbitrary value z.
- The broker tosses $k2^n$ balls, remembers $k2^u$ and generates 2^{u-1} coins.

- User – Vendor
  - User buy stuff with his coins and Vendor verifies the validity of those by quickly computing the hashes.
- Vendor – Broker
  - Vendor returns the coins, Broker verifies their validity, that they have not been redeemed yet and that they have actually been minted by him.

3 Security Concerns

- Long-term Forging
- Theft of Coins
- Double Spending

Long-term Forging

- Problem:
  Attacker may spend months forging a huge amount of coins hoping to catch up with the broker.
- Solutions:
  - Validity period which is only disclosed at the beginning of the period.
  - Broker can cancel validity period at any time.
  - Hidden predicates.
  - Broker can generate coins for several months in advance.

Hidden predicates

The balls have to satisfy some hidden predicates.

$$\underbrace{x_0 x_1 x_2 \dots x_{n-1}}_{random} \underbrace{x_n \dots x_m}_{predicate}$$

The $m - n$ last bits determine the predicate to apply on those same bits.
The predicate should be hard, hidden and can be changed on a daily basis.

Preventive minting

Minting for the next eight months at the same time. Broker knows the validity for the upcomming months.
At the beginning of a new period, Broket should have all the coins for the month j, $\frac{7}{8}$ for the $j+1$, ..., $\frac{1}{8}$ for the j+7.
All the balls tossed can end up in any of the eight months bins.

Theft of Coins

- Problem:
  Theft coins could be sold to rogue users for them to use or used by the thief.
- Solutions:
  - Vendor-specific coins.
  - User-specific coins.
  - Generalization of the collision.

User-specific coins

- Additional condition h'($x_1, ..., x_k$) = h'(U), h' being a shorter hash function and U the identifier of a group.
- Trade-off between large groups (more potential rogue users for the thiefs) and small groups (large excess of coins needed to satisfy everyone needs).

Generalization of the collision

- A coin is now valid for U iff for $y_i = \mathrm{h}(x_i)$, $i = 1, ..., k-1$, we have $y_{i+1} - y_i = d_i (mod 2^u)$, and where $(d_1, ..., d_{k-1}) = \mathrm{h'}(\mathrm{U})$.
- Broker tosses balls in bins as previously, that part is not user-specific.

Generalization of the collision (cont'd)

When a user requires coins, Broker proceeds to some additional computations:

- Computes d_i's.
- Picks a random bin y_1 that will serve as the identifier of the coin.
- Computes y_i's.
- Takes the ball out of y_1 and a copy out of bins y_i, $i = 2, ..., k$.
- If one bin y_i is empty, Broker start again with a new y_1.

Double Spending

- Problem:
  Spending many times the same coin.
- Solutions:
  - Coins are tracable.
  - Each coin uniquely identified on the broker side.

Conclusion

Drawbacks:

- High investment cost.
- Continous upgrade.
- Small scale forgery id possible but negligeable.
- Not perfectly anonymous.

Advantages:

- Validity of coins easy to check.
- Off-line, the broker is not a bottleneck.

Questions.