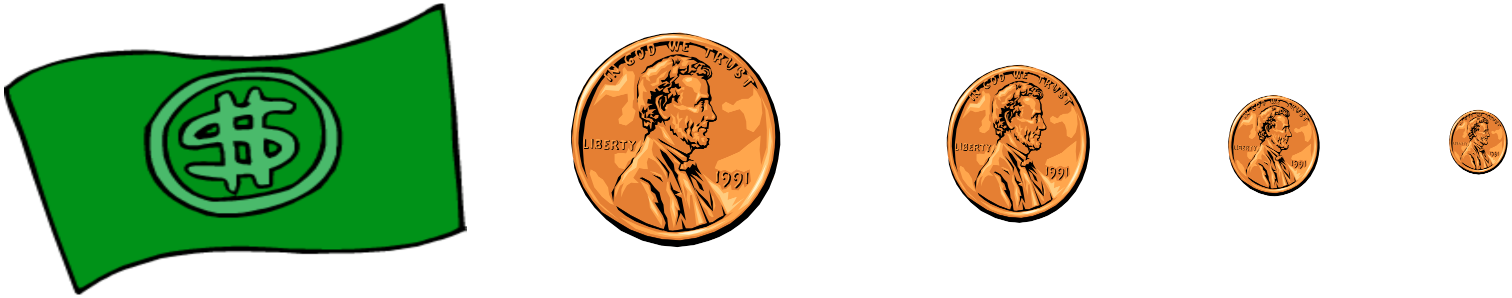


Micropayments Revisited

Ronald L. Rivest
(with Silvio Micali)
MIT Laboratory for Computer Science

RSA Conference 2002

Outline

- The need for micropayments
- Dimensions in micropayment approaches
- Previous work
- The “Peppercorn“ proposal

What is a “micropayment”?

- A payment small enough that processing it is relatively costly. Note: processing one credit-card payment costs about 25¢
- A payment in the range 0.1¢ to $10.
- *Processing cost* is the key issue for micropayment schemes. (There are of course other issues common to all payment schemes…)

The need for small payments

- "Pay-per-click" purchases on Web:
  - Streaming music and video
  - Information services

- Mobile commerce ($20G by 2005)
  - Geographically based info services
  - Gaming
  - Small "real world" purchases

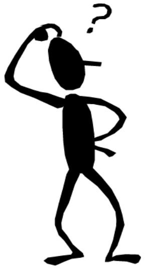
- Infrastructure accounting:
  - Paying for bandwidth

Payment schemes

- Dominant today:
  - Credit cards
  - Subscriptions
  - Advertisements

FOR SALE

- Other possibilities:
  - Electronic checks
  - Anonymous digital cash
  - *Micropayments*



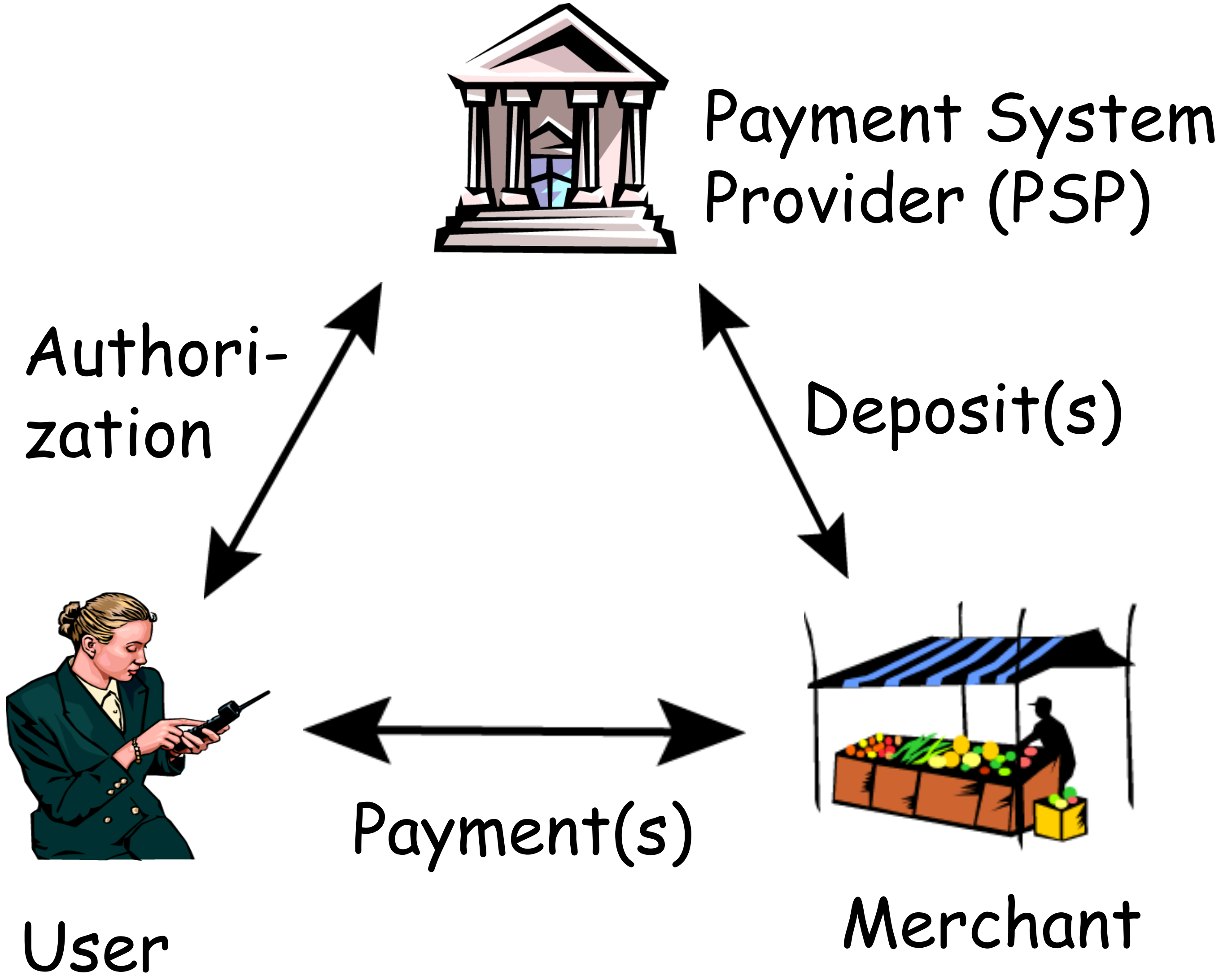
Why aren't micropayments already here?

- The market need is still nascent.
- Rolling out a new payment system requires the coordination of many players.
- Fundamentally: COST !
  Existing micropayment schemes are too costly to implement.

Payment scheme costs:

- Customer acquisition and support
- Disputes and chargebacks:
  - User says he didn't place order
  - User says goods were poor or missing
- Overspending (more than authorized, or more than user can afford)
- Communication, computation, equipment
- Fraud/Attacks on system

Payment Framework:



Payment System Provider (PSP)

Authorization

Deposit(s)

Payment(s)

User

Merchant

Dimensions to consider:

- Level and form of aggregation
- On-line PSP vs. off-line PSP
- Interactive vs. non-interactive
- Ability to handle disputes
- Ability to handle overspending
- Computation/communication cost
- Robustness against fraud

Level of Aggregation

- To reduce processing costs, many small micropayments should be aggregated into fewer macropayments.
- Possible levels of aggregation:
  - No aggregation: PSP sees every payment
  - Session-level aggregation: aggregate all payments in one user/merchant session
  - Global aggregation: Payments can be aggregated across users and merchants

Form of Aggregation

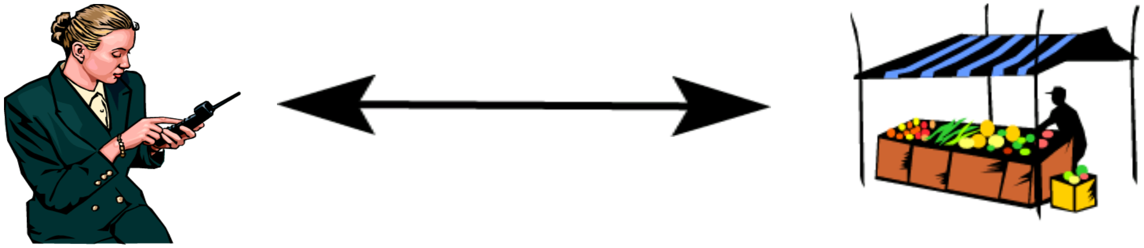
- Deterministic aggregation: Accounting is exact.
- Statistical aggregation: Value flow is accurately estimated (looks good for micropayments)
- Our Peppercorn proposal makes aggregation look deterministic/non-existent to user but statistical to merchant and bank.

On-line PSP vs. Off-line PSP

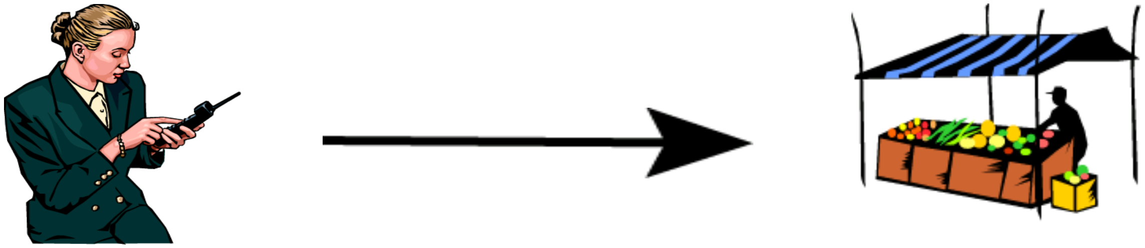
- On-line PSP:
  PSP authorizes each payment or each session.
- Off-line PSP:
  User and merchant can initiate session and transact without participation of PSP. (e.g. pay taxi)
- PSP should be off-line if scheme has global aggregation.
- If multiple PSP's involved, off-line is better.

Interactive vs. Non-interactive

- Interactive:
  Payment protocol is *two-way* dialogue

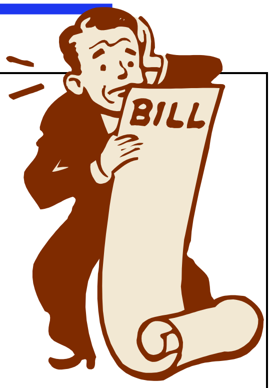


- Non-interactive:
  Payment protocol is *one-way*
  (e.g. anti-spam payment in email):



Ability to handle disputes

- User claims he didn't approve payment
  Solution: use digital signatures



- User claims goods are poor quality or were never sent.
  Solution: let user complain to merchant directly.

- A micropayment PSP can't afford to handle *any* such disputes!

Ability to handle overspending

- User may refuse to pay PSP for payments he has made.
  Solution: prepayment
- User may spend more than he was authorized to spend.
  Solution: penalties/deterrence

Computation Cost

- Digital signatures are still relatively “expensive” --- but _much_ *cheaper than they used to be!*
- Today, it seems reasonable to base a micropayment scheme on digital signatures. (E.g. Java card in cell phone)
- User and merchant are anyways involved with each transaction; digital signatures only add a few milliseconds.
- On-line/Off-line signature can also help.

Communication Cost

- Communication costs can be minimized by:
  - Keeping PSP off-line; both authorization and deposits are aggregated, so PSP only has overall view of value flow
  - Making payment protocol non-interactive (e.g. reduce number of round-trips needed when buying with pay-per-click using browser)

Robustness against Fraud

- Any party (user/merchant/PSP) may try to cheat another.
- Any two parties may try to cheat the third.

Previous Work: Digital Cash

- Example: Chaum's digital coins
- Emphasis on *anonymity:* Withdrawals use blind signatures
- Problem of double-spending handled by having doubler-spenders revealed (e.g. Brand's protocol)
- <u>No aggregation</u>: every coin spent is returned to the PSP.

Previous Work: PayWord

- Rivest and Shamir '96
- Emphasis on reducing public-key operations by using hash-chains instead:

  $x_0 \leftarrow x_1 \leftarrow x_2 \leftarrow x_3 \leftarrow \ldots \leftarrow x_n$
- User signs x_0 and releases next x_i for next payment
- Session-level aggregation only.

Previous Work: MicroMint

- Rivest and Shamir '96
- Eliminates public-key operations entirely; each digital coin is a four-way hash collision:

$$\begin{tikzcd} & & & y & & & \\ x_0 \arrow[urrr] & & x_1 \arrow[ur] & & x_2 \arrow[ul] & & x_3 \arrow[ulll] \end{tikzcd}$$

- <u>No aggregation</u>: each coin is returned to PSP.

Previous Work: Millicent

- Manasse et al. '95
- User buys merchant-specific *scrip* from PSP for each session.
- Requires PSP to be on-line for scrip purchase
- <u>Session-level aggregation</u> only

Previous Work: Lottery Tickets

- "Electronic Lottery Tickets as Micropayments" – Rivest '97 (similar to "Transactions using Bets" proposal by Wheeler '96; see also Lipton and Ostrovsky '98)
- Payments are *probabilistic*
- First schemes to provide <u>global aggregation:</u> payments aggregated across all user/merchant pairs.

“Lottery Tickets” Explained

- Merchant gives user hash value y = h(x)
- User writes Merchant check: “This check is worth $10 if three low-order digits of $h^{-1}(y)$ are 756.” (Signed by user, with certificate from PSP.)
- Merchant “wins” $10 with probability 1/1000. Expected value of payment is 1 cent.
- Bank sees only 1 out of every 1000 payments.

Our “Peppercorn“ Proposal

- Under English law, one peppercorn is the smallest amount that can be paid in consideration for value received.
- Peppercorn scheme is an improvement of basic lottery ticket scheme, making it:
  - Non-interactive
  - Fair to user: user never “overcharged”

Non-interactive payment

- Revised probabilistic payment: “This check is worth $10 if the three low-order digits of the hash of your digital signature on this check are 756.”
- Merchant’s deterministic signature scheme is unpredictable to user.
- Merchant can convince PSP to pay.

Non-interactive payment (cont)

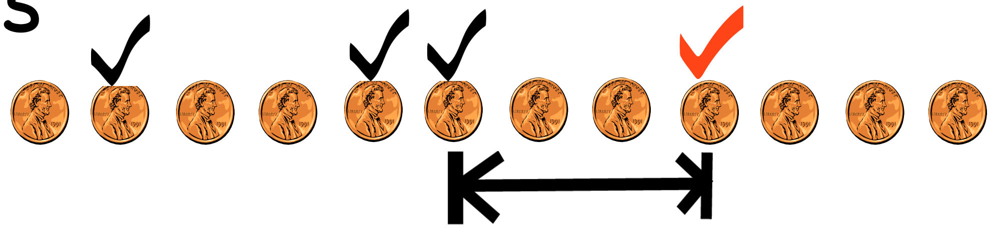
- Optimization:
  "This check is worth $10 if the
  three low-order digits of
  the hash of your digital signature
  on *the date of* this check are 756."
- Merchant's server only needs to apply signature function once a day.

User Fairness: No "Overcharging"

- With basic scheme, unlucky user might have to pay $20 for his first 2 cents of probabilistic payments!
- We say payment scheme is *user-fair* if user never need pay more than he would if all payments were non-probabilistic checks for exactly expected value (e.g. 1 cent)

Achieving User-Fairness

- Assume for the moment that all payments are for exactly one cent.
- Require user to sequence number his payments: 1, 2, …
- When merchant turns in winning payment with sequence number N PSP charges user N – (last N seen) cents



User charged three cents for ✓

User-Fairness (continued)

- Note that merchant is still paid $10 for each winning payment, while user is charged by difference between sequence numbers seen by PSP.
- Users severely penalized for using duplicate sequence numbers. If user's payments win too often, he is converted to basic probabilistic scheme. PSP can manage risk.

Conclusions

- Peppercorn micropayment scheme
  - Is *highly* scalable: bank can support *billions* of payments by processing only *millions* of transactions (1000x reduction)
  - Provides global aggregation
  - Supports off-line payments
  - Provides for non-interactive payments
  - Protects user from statistical variations
  - Uses digital signatures, but overhead for merchant and bank can be minimized

(The End)