Peppercoin Micropayments

Ronald L. Rivest
MIT CSAIL

(joint work with Prof. Silvio Micali)

Outline

- Micropayment examples
- Challenges
- Aggregation methods
- The "Peppercoin" method
  (In England a *peppercorn* is smallest amount that can be paid in a contract)

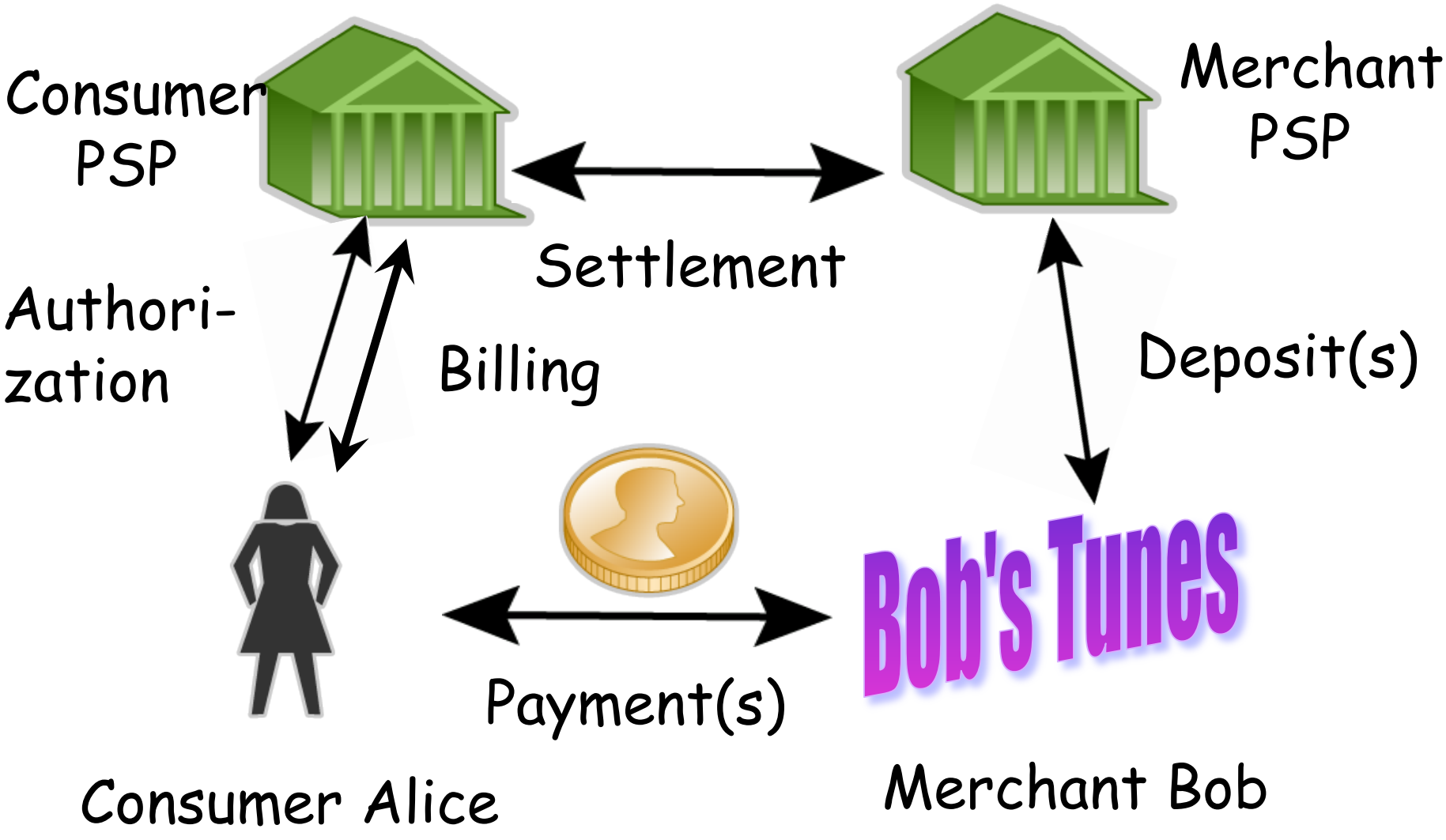
What is a "micropayment"?

- A payment in the range 0.1¢ to $10.
- A payment small enough that processing it is relatively costly. (Processing one credit-card payment costs about 25¢ …)
- *Processing cost* is the key issue for micropayment methods.

Lydians invented coins 640 B.C.

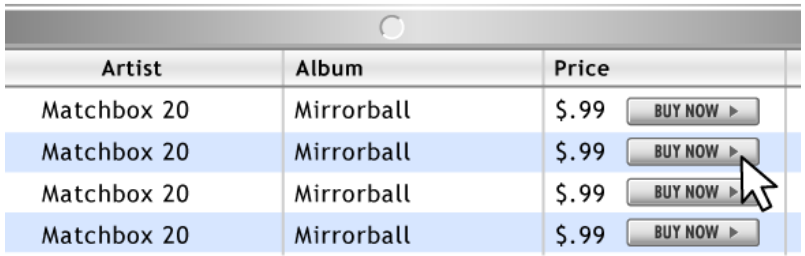
- Before 640 B.C.: ***gold bars, barter***
  small purchases difficult.
- After 640 B.C.: ***coins***
  small purchases easy.
- Before 2003: ***credit cards***
  small on-line purchases difficult.
- After 2003: …

Generic Payment Framework



How we'll make small payments

- Web download
  - Music (even streaming)



- Mobile phone
  - Map
  - Ringtones

- Physical POS
  - Vending machine

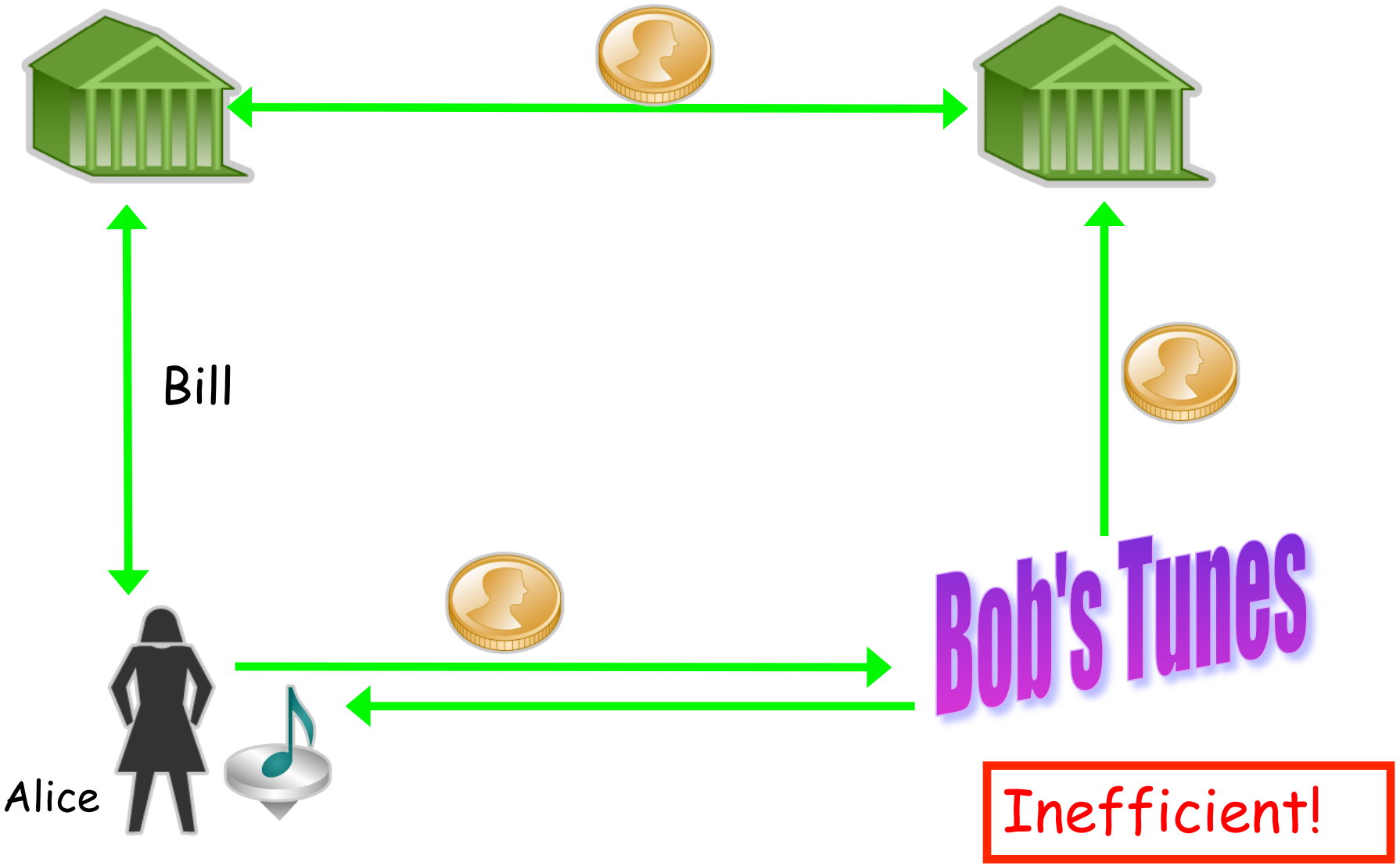
Challenges:

- Ease-of-use
- Low-Cost
- Extending existing payment framework
- Security
- … (many other issues, too)

Aggregation

- To reduce cost, ***micropayments*** must be aggregated into fewer ***macropayments***.
- Possible levels of aggregation:
  - **None**: Every payment deposited with PSP
  - **Merchant-level**: A consumer's payments are aggregated by merchant
  - **MicroPSP**: Monopoly service that disintermediates existing payment services; doesn't scale well
  - **Universal**: Payments aggregated across all users and merchants, even those supported by different cooperating PSPs

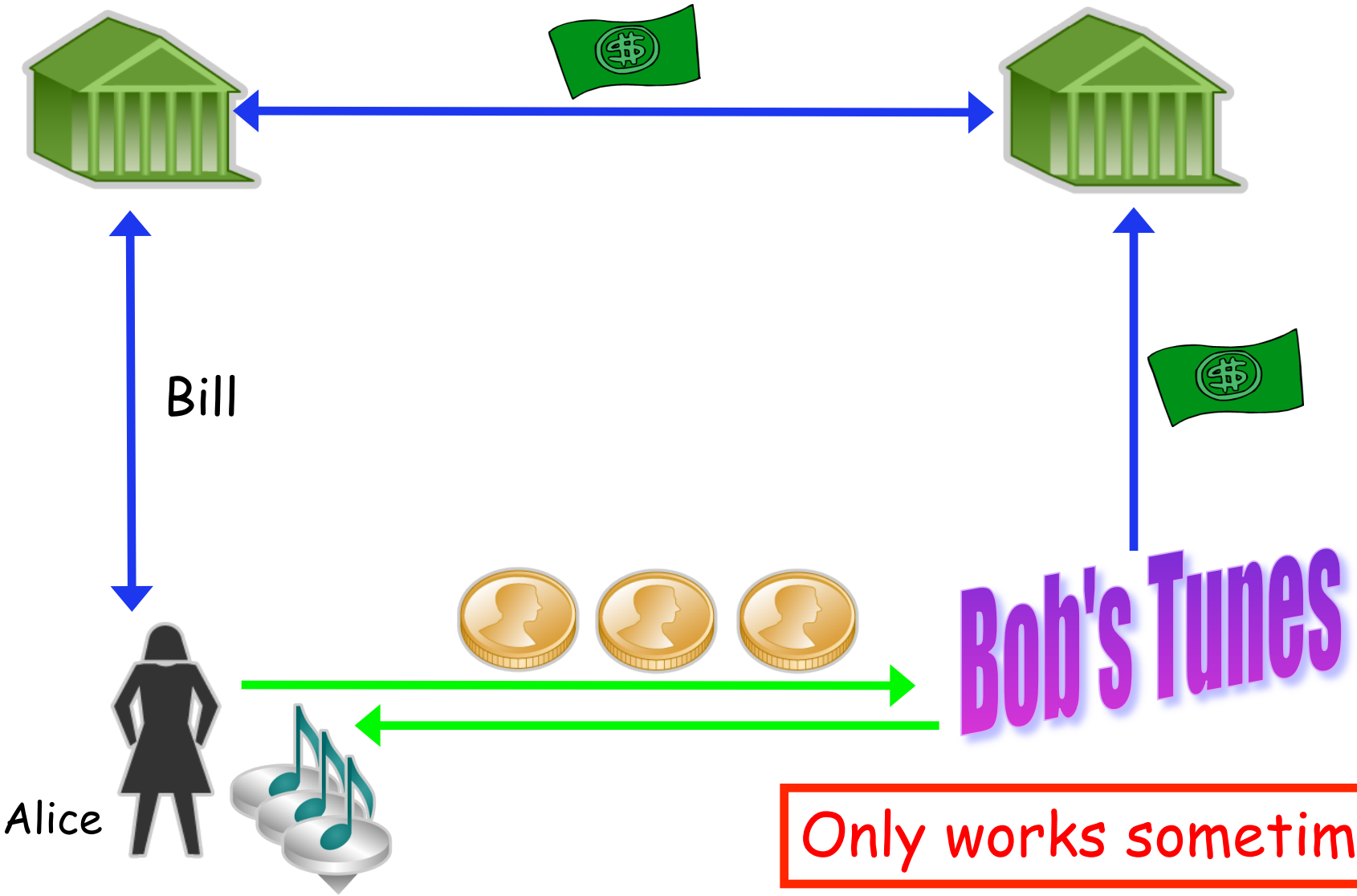
No Aggregation



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)
- No aggregation: every coin spent is returned to the PSP.

Merchant-Level Aggregation

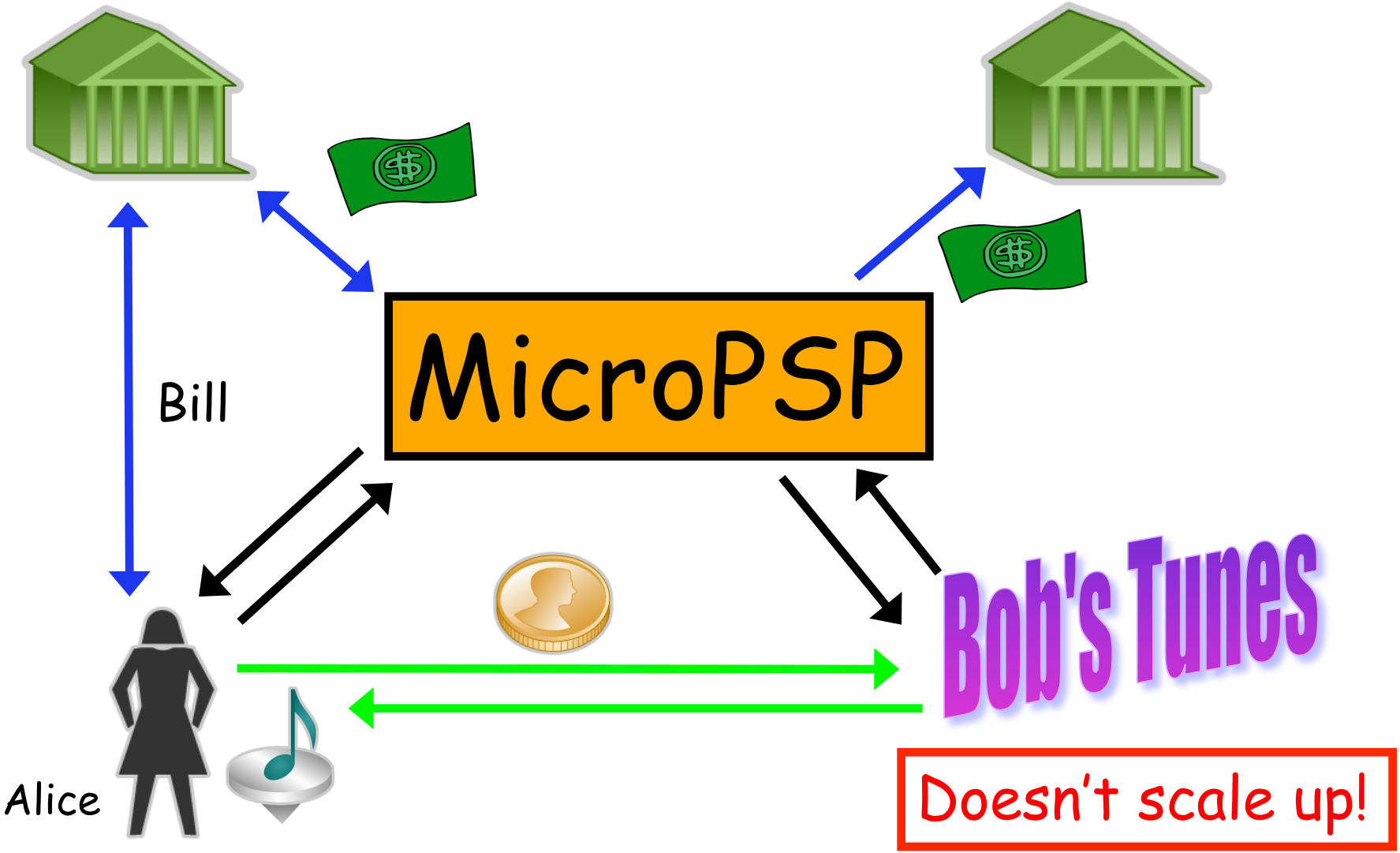


Previous Work: PayWord

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

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

MicroPSP Aggregation



Universal Aggregation

- ***Universal aggregation*** dramatically reduces processing cost, independent of spending patterns.
- *Also called* ***many/many/many aggregation****:* Aggregates payments from
  - *Many* consumers
  - *Many* merchants
  - *Many* PSP's

  in any combination. No need to aggregate sales per consumer.

Universal Aggregation Idea

- Would merchant prefer:

  (a) twenty *50 cent payments*, or

  (b) *$0 for 19 payments, and $10 for one?*

  No difference to merchant, on average

Universal Aggregation Idea

- Would merchant prefer:

  (a) twenty *50 cent payments*, or

  (b) *$0 for 19 payments, and $10 for one?*

  No difference to merchant, on average.

What if processing costs 20 cents per payment?

(a) nets only 30 cents per payment

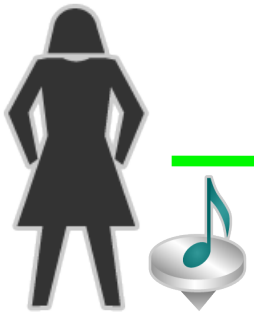
(b) nets 49 cents net per payment!

Merchant strongly prefers (b) !

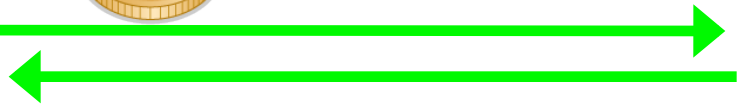
Peppercoin's Universal Aggregation

- One micropayment in 20 is "cryptographically selected" by merchant, and deposited for 20x its value, as a macropayment!
- Yet consumer pays *only* for what she has spent: each micropayment records cumulative amount she has spent at all merchants.

Peppercoin's Universal Aggregation



50 cents



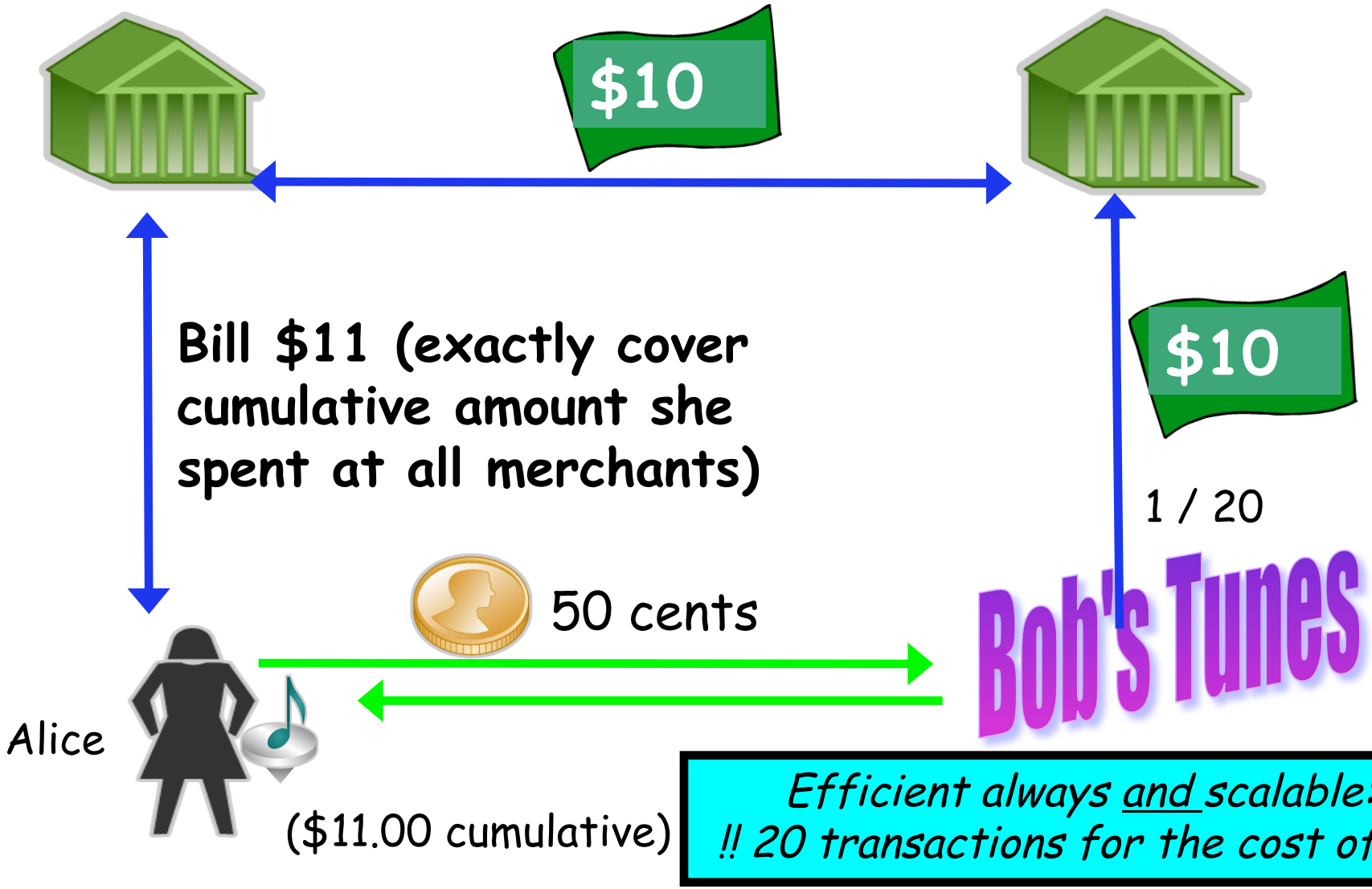
19 / 20

Log

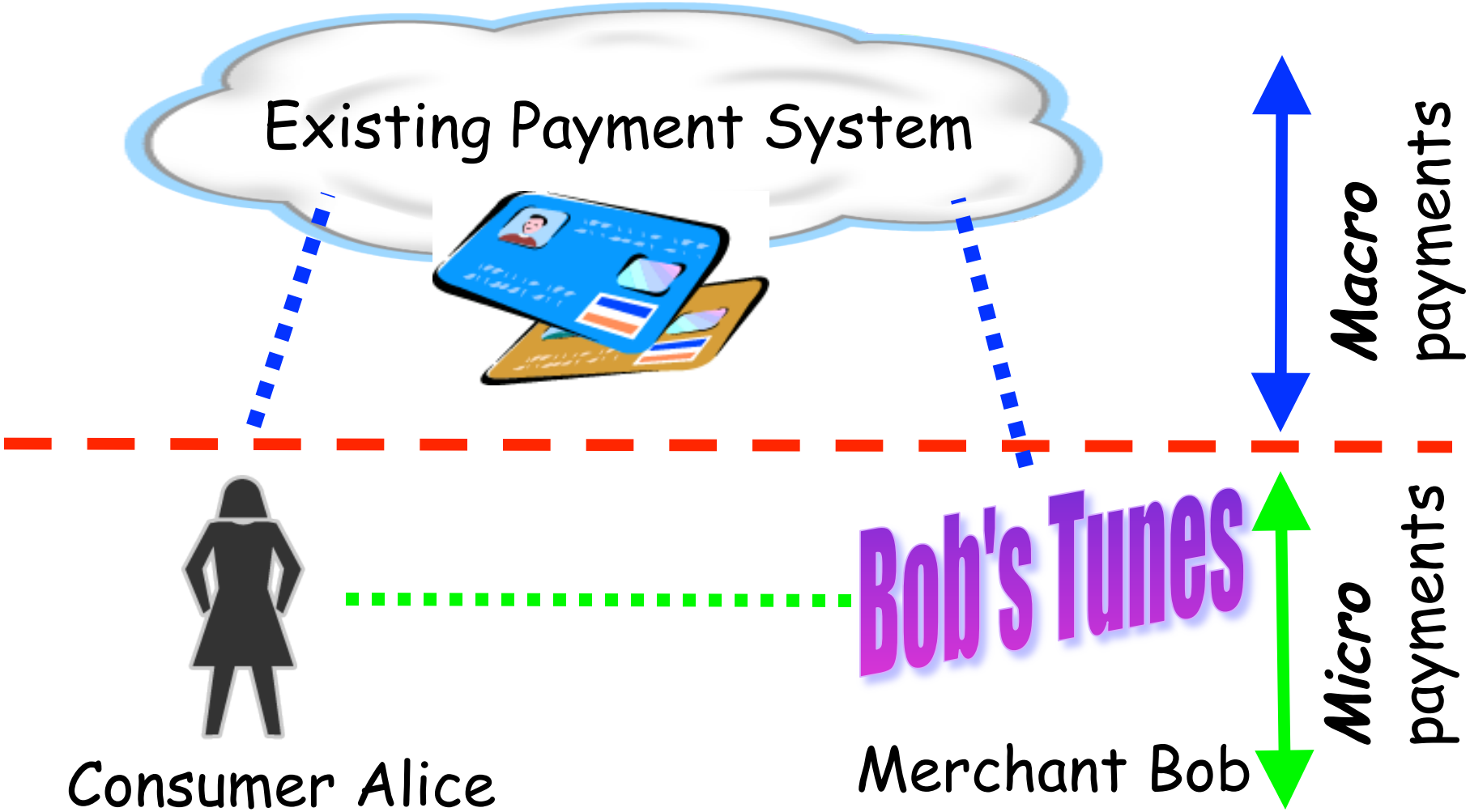
Alice ($8.50 cumulative)

Peppercoin's Universal Aggregation

Peppercoin's Universal Aggregation



Peppercoin Extends Existing Payment Systems to Micropayments

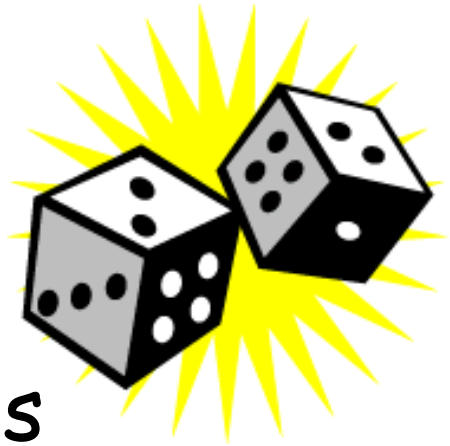


Dimensions to consider:

- Aggregation (*universal)*
- PSP on-line or off-line ? *(off-line)*
- Interactive vs. non-interactive *(non)*
  - (e.g. anti-spam payment in email)
- Computation Cost *(cheap)*
- User-fairness *(fair)*
- … (many other issues, too)

Previous Work: Lottery Tickets

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



"Lottery Tickets" Explained

- Assume micropayments are for ten cents.
- Merchant gives user y = hash(x) for random x.
- User writes check: "Pay Merchant $10 if two low-order digits of $hash^{-1}(y)$ are 75." (Signed by user, with cert from his PSP.)
- Merchant "wins" $10 with probability 1/100. Expected value of payment is 10 cents.
- Bank sees only 1 out of every 100 payments. (A plus for user privacy!)

Non-interactive

- Revised check:
  "Pay Merchant $10 if
  two low-order digits of
  the hash of Merchant's digital signature on this check are 75."
- Merchant's deterministic signature scheme unpredictable to user.
- Merchant can convince PSP to pay.

Computation Cost

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

Optimization for less Signing

- "Pay Merchant $10 if the two low-order digits of the hash of Merchant's digital signature on *the date of* this check are 75."
- Merchant only signs once a day.

Variable-sized payments

- To make micropayment of size m:
  - Chance of "winning" becomes
    m / M
    where M is the macropayment size.
- For example, a $1 micropayment converts to a $10 macropayment with probability 1/10.
- A one-penny micropayment converts to a $10 macropayment with probability 1/1000.

Is revenue variance an issue?

- **Theorem.** If Peppercoin reduces merchant fees by R percent of transaction value, then merchant will be ahead (with probability 999,999/1,000,000) after only
$$(5 / R)^2$$
macropayments have been received.
- **Example:** micro = 0.10, macro = $10, otherfee = 0.03, peppercoinfee = 0.01, R = 0.20, $(5/R)^2 = 625$ or $6250 total value.

Fraud models

- Security is challenging to achieve given that PSP has only partial information, parties may collude, and payment schedules are decoupled.
- For example, consumer and merchant may try to collude to defraud PSP's.
- One effective countermeasure is to make macropayment to merchant only from revenues from that specific consumer (perhaps deferring payment if necessary).

Conclusion

- Peppercoin micropayments are
  - Easy to use
  - Low-cost even for small payments
  - Flexible
    (interface with existing payment systems)
  - Secure
- www.peppercoin.com

Thanks!

(The End)