CS 161: E-commerce

October 24, 2005

Stages in E-commerce purchase

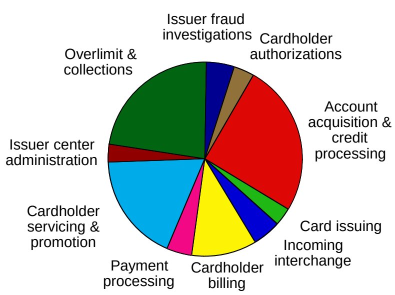
Stages in e-commerce purchase

- **Advertising**
- **Solicitation**
- **Negotiation**
- **Purchase**
- **Payment**
- **Delivery**
- **Ordering/support**

Credit cards as an enabler

- **Standard purchase model reveals credit information**
- **Overhead costs can be high for microtransactions**
- **Acquiring Bank vs. Consumer Bank**
- **Payment processors**

Why is a credit card transaction 50¢?



Information goods

- **Consider the purchase of an information good or service:**
  - **Library information**
  - **Search services**
  - **Software**
  - **Video clips**
- **These transactions may be large value or microtransactions**
- **In either case, atomicity is crucial**

Payment methods: Atomicity

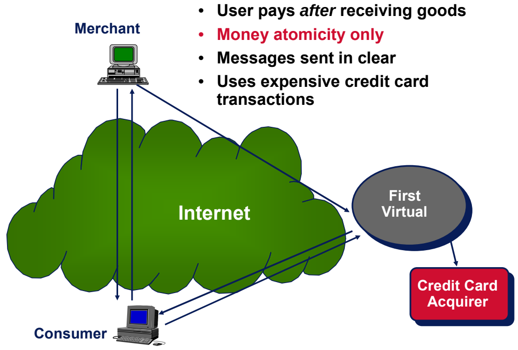
What Is atomicity?

- I won't try to give a formal definition
- 3 types of atomicity:

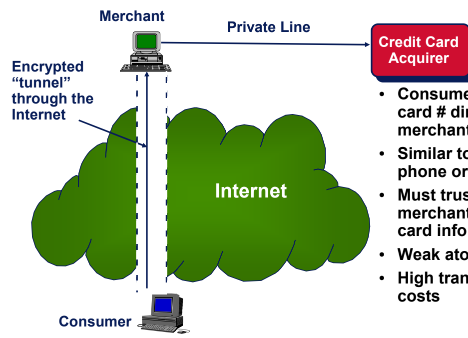
- **Money atomicity**
  - All money transfers complete with non-ambiguous results
  - Money is neither destroyed nor created
- **Goods atomicity**
  - One receives goods if and only if one pays
  - Example: Cash On Delivery parcels
- **Certified delivery**
  - Both buyer and seller can prove the delivered content
  - If you get bogus goods, you can prove it

First Virtual

- User pays *after* receiving goods
- Money atomicity only
- Messages sent in clear
- Uses expensive credit card transactions



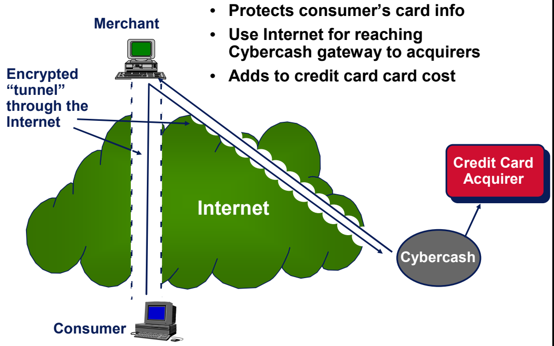
Netscape/SSL model



- Consumer sends card # direct to merchant
- Similar to today's phone order
- Must trust merchant with card info
- Weak atomicity
- High transaction costs

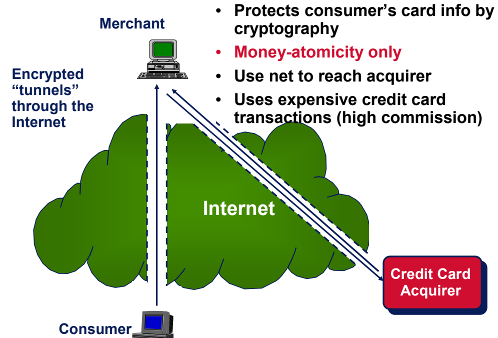
Third party intermediary model (Cybercash)

- Protects consumer's card info
- Use Internet for reaching Cybercash gateway to acquirers
- Adds to credit card card cost

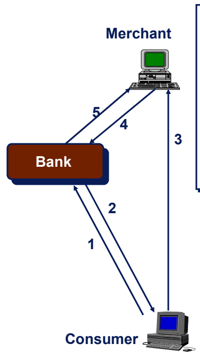


Mastercard/Visa SET

- Protects consumer's card info by cryptography
- Money-atomicity only
- Use net to reach acquirer
- Uses expensive credit card transactions (high commission)



Digicash



1 Consumer asks bank for anonymous digicash
2 Bank sends anonymous digicash bits to consumer
3 Consumer sends digicash to merchant in payment
4 Merchant checks that digicash has not been double spent
5 Bank verifies that digicash is valid

Problems

–No atomicity
–Anonymity restricted in US
–Interrupt transaction: ambiguous state
–Detecting double spending is expensive

NetBill goals

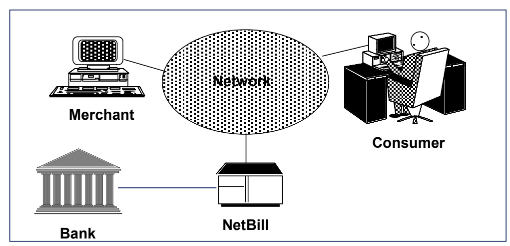
- Real service
- Highly atomic transactions
- Micro-transactions
- Full security and privacy

NetBill features

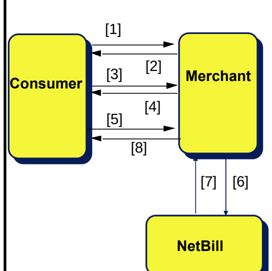
- Focus on info goods/services (journal articles)
- Microtransaction (10¢ purchase: 1¢ overhead)
- Variable pricing
- Fully integrated access control
- DES/RSA/DSA combo for best performance
- Electronic statements & account creation
- Certified delivery: proof of purchase/content

NetBill model

- An electronic credit card to enable network based commerce
- Provides billing services on behalf of network attached merchants.



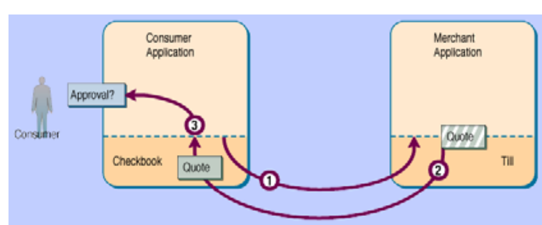
NetBill protocol



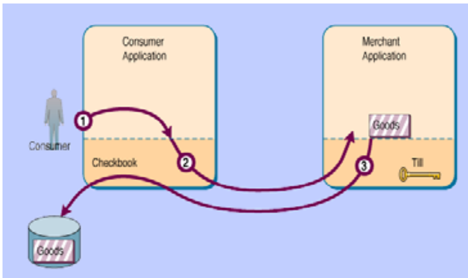
(All messages are encrypted with shared key S)

1 Buyer requests price
2 Seller makes offer
3 Buyer accepts offer
4 Goods delivered encrypted with K
5 Buyer signs EPO (electronic purchase order) <price,crypto-checksum,timeout>
6 Seller countersigns EPO, and signs K
7 NetBill checks account, timeout; stores K & crypto-checksum; transfers price money; sends signed receipt including K
8 K received; goods decrypted

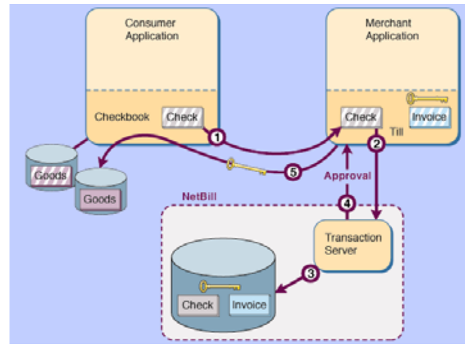
NetBill protocol - low level



NetBill protocol - low level



NetBill protocol - low level



Why atomic?

- **Money atomicity**
  - Accounts are held at a single server, and are modified with local atomic (ACID) transactions
- **Goods atomicity**
  - Customer receives decryption key for goods only if she pays
  - If customer pays, decryption key available from multiple sources (merchant and NetBill server)
  - Key can be delivered by alternative network (such as telephone) if necessary
- **Certified delivery**
  - If customer receives junk or bogus goods, can prove the contents to a judge
  - Crypto checksum of goods (signed by both customer and merchant) are stored at NetBill server
  - Signed copy of decryption key stored by all parties!

Role of Anonymity in EC

A puzzle

- **Suppose Berkeley grads want to find their average salary**
- **But, of course, no participant wants to reveal his/her salary**
- **How can we compute the average without giving away information about any participant's salary?**

Later, I will give several solutions to this puzzle

Why study anonymity?

- **Privacy concerns**
  - individual
  - corporate
  - national
- **Technology for collecting private statistics**
- **Understand theoretical limits, countermeasures**
- **Understanding semi-anonymity**
  - Allows government search in exceptional circumstances
- **Insights**
  - e-commerce
  - distributed protocols
  - cryptography
  - survivability

Anonymous computation

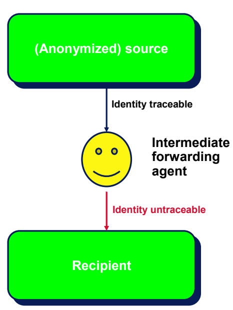
- There is extensive work on anonymous and secret communication (cryptography)
- But what if we want to compute a function of the secure values?
- In puzzle, we want to add “encrypted” values
- Examples:
  – Compute census statistics on usage or population
  – Make an anonymous purchase and then be able to prove that goods were delivered correctly
  – Anonymously auction goods — without revealing any bids (except the winning bid) or bidders

Is anonymous computation feasible?

- Good news:
  – In theory: any computation can be anonymized
- Bad news:
  – In general, constructions are complicated
  – Most constructions multiply number of messages by a factor of at least 1000 (and often, much higher, like 10^{20})
  – Usually, simple IP location tracing (traffic analysis) reveals identity of parties
  – Computation requires complex crypto operations.
  – Running times for “simple” anonymous computations are usually measured in days or years.
- So researchers have relied on partial solutions
  – Mixes, pseudonyms, escrow

Mixes

- Use intermediate forwarding agents
- Examples: onion routing, crowds, anonymizer.com, etc.
- Idea simultaneously thought of by several researchers
- Problems:
  – intermediary knows all
  – subject to traffic analysis and statistical analysis
  – can not link old messages to new messages



Pseudonymous identity

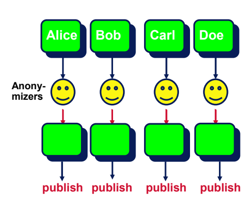
- Establish a consistent, but disguised identity
- Example: mail forwarders
- Can disguise basic facts about identity, but may be traceable from patterns of use
- Once identity is revealed, then all previous uses are traceable

Escrow

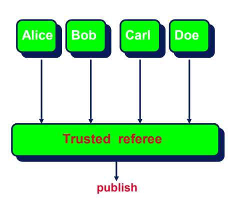
- Use pseudonym, but store real identity where law enforcement can find it.
  – Refinement: split identity into multiple parts
  – Store them in different locations
- Depends on procedural mechanisms (e.g. search warrants) for privacy
- Has drawbacks of pseudonym
- Government approach to cryptography

Unsatisfactory solutions to puzzle

- Mix approach:
  – Everyone sends salary anonymously to third parties who publish



- Escrow approach:
  – Everyone sends salary to trusted escrow agent



Fissionable data

- Idea:

1 fission data into different parts

each part is random, but combination is not random

2 perform operations on parts

3 recombine data

- Mathematics is based on theory of finite fields
- Anonymous addition & multiplication are fast
- My examples focus on addition (easy to show)

Fissionable solution to puzzle

- Fix a modulo n
- Each person S (T, U, …) picks k-1 random values

$S_1, S_2, \ldots S_{k-1}$

- Each person S picks a S_k such that

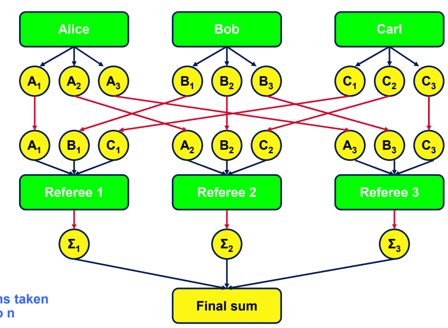
$S_1 + S_2 + \ldots + S_{k-1} + S_k$ = [Salary of S] (mod n)

- Each person S sends value S_i to referee I

(communications should be over a secure channel)

- Referee i sums $S_i + T_i + U_i + \ldots$
- The referees publish their results and we take sum

Fissionable solution to puzzle



All sums taken modulo n

Hierarchical approaches

- Because referees combine information locally, we can build hierarchies of referees
- This means that results can be combined at a communication point (such as an Internet router in the Active Network approach.)

Other forms

- We can also pick a random polynomial of degree q modulo p

$f(x) = x^q + a_{q-1}x^{q-1} + \ldots + a_1x + a_0 \pmod{p}$

(a_i are chosen randomly)

- Secret is $f(0) = a_0$
- Shares are f(1), f(2), . . .
- Note: q shares determine f(0) (Lagrangian interpolation)
- We can add and multiply values
- Fault-tolerant: we can use more than q shares for redundancy
- Super-fast!

Auction types

- Auctions
  - Allocate scarce resources
  - Proposed to ration Internet bandwidth
- Three types of auctions

English auction (price goes up)
- advantages: encourages "honest" bids
- disadvantages: slow
  not private

Sealed bid auction
- advantages: constant time
- disadvantages: does not encourage "honest" bids,
  auctioneer knows all

Dutch auction (price goes down)
- advantages: protects privacy
- disadvantages: slow
  does not encourage "honest" bids

Vickrey auction

- Vickrey gave a way to combine best features of English auctions and sealed-bid auction
- Second-price auction
  - Highest bidder wins
  - Price is the value of the second highest bid
  - Example: Alice is highest bidder for $100;
    Bob is second highest bidder for $80;
    Alice wins the bid, but pays only $80

Anonymous auctions

- Goal: combine best features of all three protocols
- Should run in a single round
- Should reveal only second highest bid
- Highest bidder can claim prize for second highest price
- No other information is revealed

Anonymous bids

- Each of n auctioneers gets a temporary ID
- Bid is bit vector of potential bids
- Non-zero entry represents bid

$5	$10	$15	$20	$25	$30	$35	$40	$45	$50	$55	$60	$65	$70	$75	$80
657	123	34	1	555	89	932	212	453	323	206	214	159	0	0	0

- This bidder is willing to bid up to $65
- We fission each element in the bid vector to protect individual bidders

Looking for the 2nd highest bid

- Each bid vector is fissioned
- We partition bidders $\log_2 n$ ways based on binary values of temporary IDs
  - low bit value 000/010/100/110 vs 001/011/101/111
  - 2nd bit value 000/001/100/101 vs 010/011/110/111
  - 3rd bit value 000/001/010/011 vs 100/101/110/111
- For each partition (element-by-element ops)
  - We anonymously add the vectors in blue and green partitions
  - We anonymously multiply blue sum with green sum
- We sum over all partitions
- The final vector has a non-zero entry exactly when at least 2 people bid that price

Anonymous auction

- The result is a bid vector; the highest non-zero entry is the second-highest bid
- All other entries are random, giving no information
- By using a technique called dynamic programming, we can dramatically reduce the number of operations
- Communications linear in the number of bids (as any auction must be!)

Anonymous auctions

- Goal: combine best features of all three protocols
  - Should run in a single round
  - Should reveal only winning bid
  - No other information is revealed
- Example:
  - In recent radio spectrum auctions, bidders signaled information by their bid
  - A bid of 2 million dollars and 37 cents = "we want to bid unopposed on lot 37"