15-827 Security and Cryptography
Lecture 7

Introduction to Electronic Commerce

Jeannette M. Wing

Electronic Commerce: Now and the Future

- 1998: > $6 billion now
  - Prediction: Rise by factor of two each year for next four years.
- Already: 25-40 million people have bought/sold something on the Web.
  - Prediction: 1 billion by 2010.
- Security challenges:
  - Ensure transaction parties are identifiable and have appropriate authority while preserving privacy and resistance to intrusion.
  - Resolve tension between
    - being able to track every move (time and location) on the Web and
    - being able to hide one's identity in ways that are impossible in the physical world.
  - Support new ways of commerce (e.g., involving multiple parties for comparison shopping, competitive bidding, alternatives, bundling, etc.)

Stu Feldman, IBM

Director of Institute for Advanced Commerce

Outline

- Electronic Commerce Overview
  - Properties of money
  - Features of electronic payment protocols
- Account-based Protocols
  - iKP, SET
  - First Virtual
  - Netbill
- Token-based Protocols
  - Digital cash
  - Brand's protocol
  - Millicent

Money

- Different payment instruments
  - **Cash, credit-cards**, checks, cashier's checks, Green Stamps, coupons, traveler's checks, frequent flier miles, …, vodka, cigarettes, wampum, beaver pelts.

- "Money Past," (Ch. 26), *Digital Cash*, Pater Wayner, AP Professional, 1997.

Properties of Money/Cash

- Scarcity
  - The amount available.
- Ability to make more
  - How easily can it be prepared? Forged?
- Transferability
  - Ownership. Passing possession or rights to someone else
  - Physical vs. electronic.
- Durability
  - Good currency should not lose value over time.
- Recognizability
  - Verifiable digital signatures are easy to recognize. The process of recognition should not destroy money.
- Measurable value (intrinsic vs. extrinsic)
  - Measure should be universal: Weigh gold, diamonds examined by experts.
- Uniform size values
  - Penny, dime, dollar
- Variable values (scalability)

More Properties of Money

- Divisibility
  - Convenient for owner to split a $10 bill into two $5s. But not by ripping it in half!
- Universality
  - Accepted as medians of exchange.
- Portable
  - Need cash to pay taxi.
- Anonymity
  - Full/partial/conditional. Serial numbers on cash.
- (User) Untraceability
  - Privacy of user is protected. No one can trace users to their purchases.
- Unlinkability
  - Different coins for same purchase should not be linked.
- Utility
- Unstealable

More Properties of Money

- Liability
  - Who is responsible if its stolen? AMEX checks cannot be used till signed; thus, replaced if stolen.
- Copiable (!)
  - Back up an electronic wallet so if destroyed it can be restored.
- Cash-property
  - Cash is currency that is (almost) always accepted. What money is may depend on context.
- Expiry
  - Does it have an expiration date?
- Revocability
  - Cannot be done with paper money.

Features of Electronic Payment Protocols

- Transaction cost
  - Micro vs. macro
- Security
  - Authentication
    - Parties involved in a transaction are who they say they are.
  - Message integrity
  - No double spending
    - Customer cannot copy coins or spend one multiple times. Applicable to token-based.
  - Nonrepudiation
    - Parties cannot deny their part in a transaction. Applicable to account-based.
  - Payment traceability
    - Delivery of goods linked to payment for the goods
    - Real world: shipment tracking numbers, credit card transactions, …
  - Certified delivery
    - Proof that goods purchased were delivered. What you got is what you asked for.

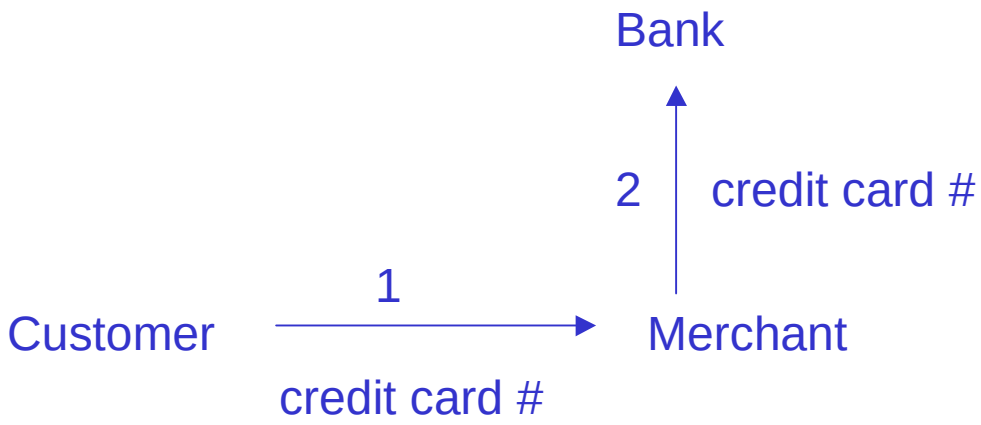
Features of Electronic Payment Protocols

- Atomicity [Tygar]
  - Goods atomicity
    - Customer gets goods and merchant gets money.
  - Money atomicity (Conservation of money)
    - Money cannot be spontaneously generated or destroyed.
- Privacy
  - Payment confidentiality
    - Observers monitoring network traffic cannot discover details of transaction.
  - Anonymity
    - Only a customer's pseudonym is known to merchant.
  - Customer untraceability
    - System cannot trace payments back to payer.
- On-line vs. off-line verification
- Software-only vs. tamperproof hardware
- *Other token-based system features*
    - Divisibility, change-return, transferability, purse-to-purse transfer, ...

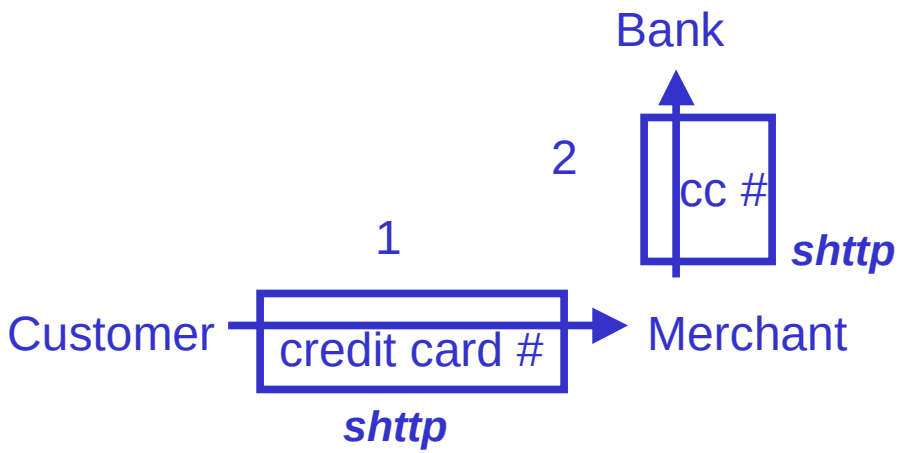
Classification of Some Real Schemes

Token-based	Electronic cash		*DigiCash, Millicent*
	Electronic purse (smart-card)		CAFÉ, MONDEX
Account-based (Notational)	EPOs over net		*NetBill*, NetCheque
	Credit card billing over net	Encrypted credit cards	Cybercash, *SET (iKP)*
		Third-party authorization	*FirstVirtual*
	Smart-card based		FSTC e-check

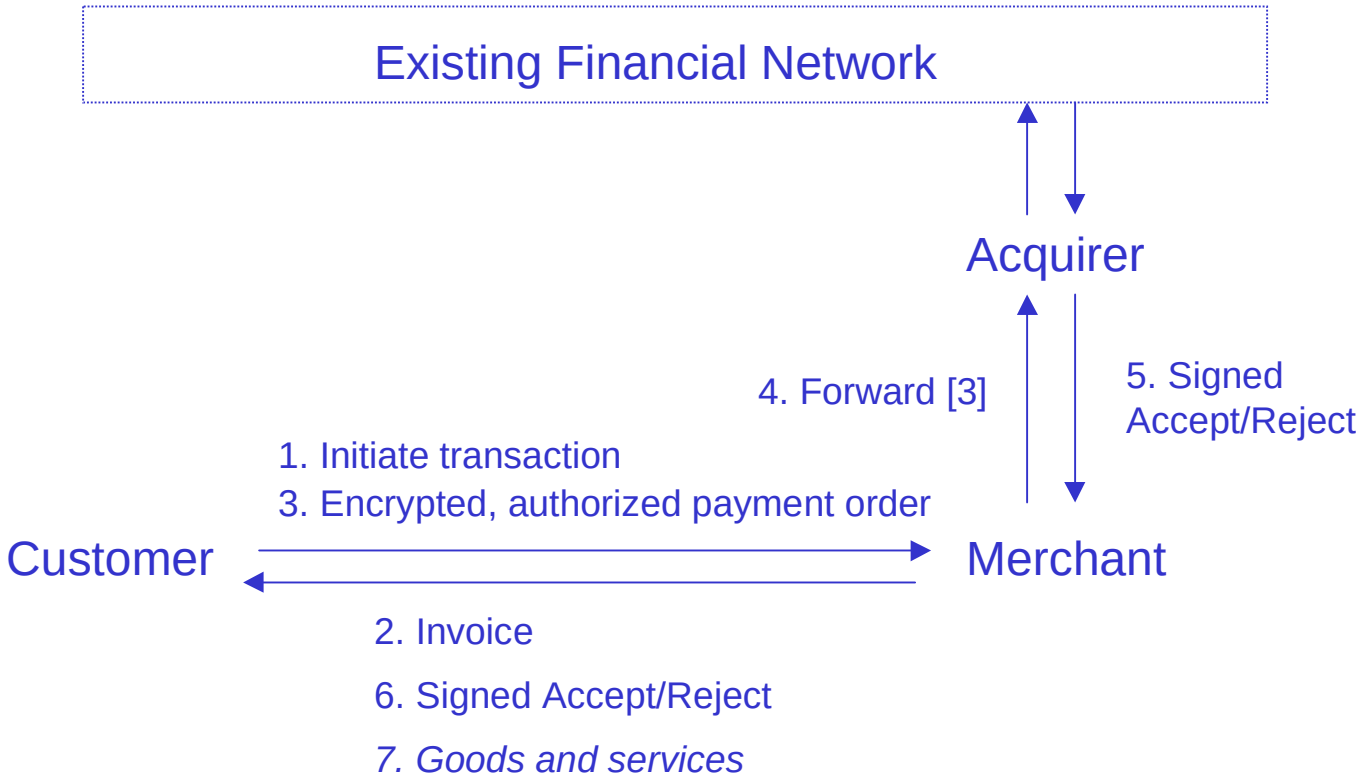
Credit-card payment protocol: Attempt #1



Credit-card payment protocol: Attempt #2



Credit-card payment protocol: iKP (i = 1, 2, or 3)



iKP Setup

H strong collision resistant one-way hash function

E_i encryption using public key of party i

S_i signing using private key of party i

$Cert_i$ certification of the public key of party i

Offer = <offer description, price, date, merchant ID>

Order = <order description, price, date, merchant ID, delivery address>

Slip = <price, date, merchant ID, credit card #, expiration date, PIN, H(Order)>

Auth = <accept/reject, H(price, date, merchant ID), H(Order)>

1KP

- Only the Acquirer (A) has a public key pair.

1. C -> M: CID
2. M -> C: Offer, $Cert_A$
3. C -> M: E_A(Slip), Order
4. Calculates H(Order)

   M -> A: E_A(Slip), H(Order)
5. Verifies steps, forwards data to payment system.

   A -> M: S_A(Auth, E_A(Slip))
6. M -> C: S_A(Auth, E_A(Slip))
7. M -> C: service, goods

1KP Notes

- Acquirer sees only hash of Order and not the Order description
  => provides some privacy to Customer
- Both Customer and Acquirer have no assurance of identity of Merchant.
- Customer is only identified by the correctness of credit card # and PIN.
- Easy to implement since only the Acquirer needs to have a public key.

2KP

- 2KP rectifies problem of assuring the identity of the Merchant.

  => Merchant also has public key pair.

  => Merchant sends both Customer and Acquirer a certificate, signs message 4, and appends contents of message 6

1. C -> M: CID
2. M -> C: Offer, $Cert_A$, **$Cert_M$**
3. C -> M: E_A(Slip), Order
4. Calculates H(Order)

   M -> A: **S_M(**E_A(Slip), H(Order)**)**, **$Cert_M$**
5. Verifies steps, forwards data to payment system.

   A -> M: S_A(Auth, E_A(Slip))
6. M -> C: **S_M(E_A(Slip), H(Order)),** S_A(Auth, E_A(Slip))
7. M -> C: service, goods

.

2KP Notes

- Perhaps would be preferable to have Merchant sign Offer in Message 2.
  - Would add assurance to Customer that Offer is valid.
- Could add an extra signature by having Merchant sign message 6 rather than append message 4 to it.

3KP

- Customer also has public key pair.

  Modify protocol to have Customer send certificate to Merchant (which is forwarded to Acquirer) and sign message 3 (which now contains a hash of Order).

1. C -> M: CID, **$Cert_C$**
2. M -> C: Offer, $Cert_A$, $Cert_M$
3. C -> M: **$Cert_C$, $S_C($**E_A(Slip), **$H($**Order**$)$**)
4. Calculates H(Order)

   M -> A: S_M(**$S_C($**E_A(Slip), H(Order)**$)$**), **$Cert_M$, $Cert_C$**
5. Verifies steps, forwards data to payment system.

   A -> M: S_A(Auth, E_A(Slip))
6. M -> C: S_M(**$S_C($**E_A(Slip), H(Order)**$)$**), S_A(Auth, E_A(Slip))
7. M -> C: service, goods

SET

- 800+ page standard developed by VISA and MasterCard with participation from Microsoft, IBM, Netscape, SAIC, GTE, RSA, Terisa Systems, and Verisign.
- Very complicated.
  - The chances of it not having any serious flaws are virtually zero.
- Merchant problem: Since merchants like to organize their DBs sorted on credit card #, they have requested this info. Hence SET provides a mode where merchants get credit card #'s back from Acquirer!
- Uses "expensive" credit card transactions. Cannot support microtransactions adequately.
- Status
  - First live SET transaction performed in Denmark Dec. 31, 1996 as an IBM pilot scheme.
  - Still in testing mode with first applications expected by end of 1998.
  - Expectation is that it will supplant other schemes and become a standard for all network transactions involving credit cards in the near future.

More on SET

- Relies on strong chain of *certificate authorities* (CAs)
  - RootCA, trees of brand CAs (VISA, MasterCard, Discover, AMEX), cardholder CAs, merchant CAs, geopolitical Cas, …
  - Certificate revocation based on same idea as “bad card number” books, expiration dates, and thumbprints (hash of a list of revoked certificates).
  - Certificate is a binding between cardholder and digital signature (like your plastic card).

- Go to www.visa.com for 800+ page *Programmer’s Handbook.*

Certificate Authorities

- Third parties responsible for establishing and vouching for the authenticity of public keys.
  - Binding public keys to distinguished names through signed certificates. In practice, X.509 certificates are used.
  - Managing certificate serial numbers.
  - Certificate revocation.

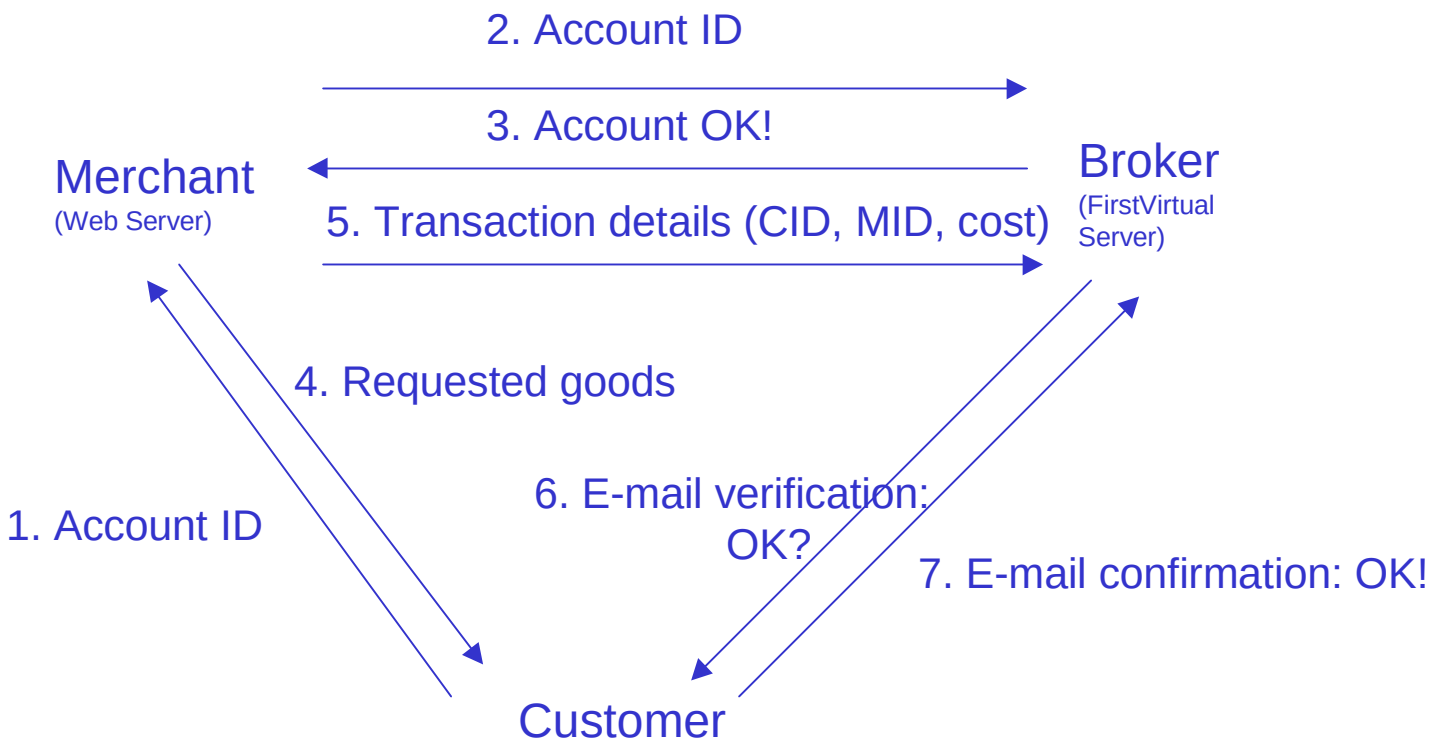
FirstVirtual

- Uses No Cryptography
  - Identification info travels in the clear.
  - Confirmation of purchases is done through ***e-mail***.
- Setup
  - Customer and Merchant establish accounts with sole Broker. Each gets account identification.
- Customers get goods before bank transmits approval

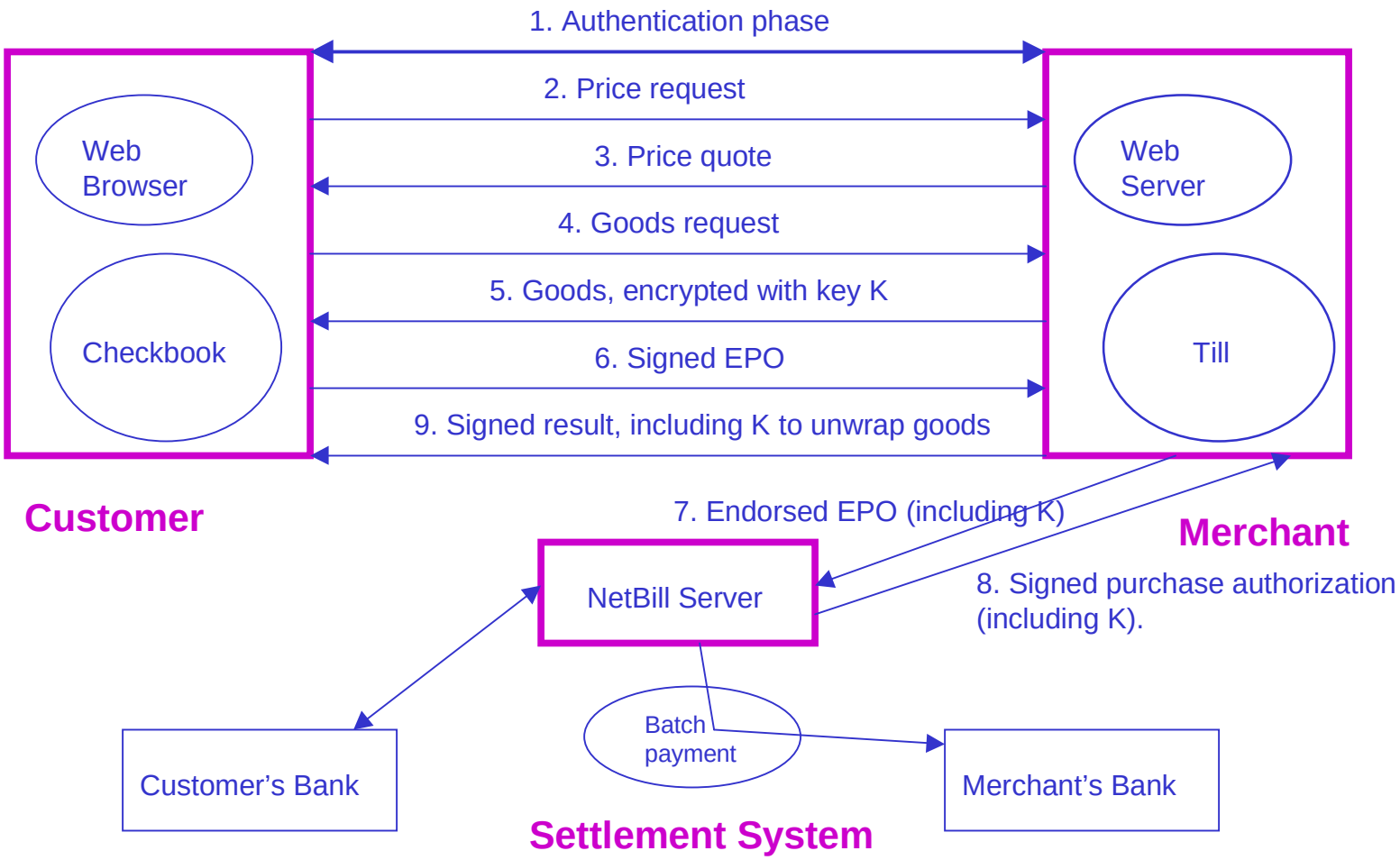
  => customers can cheat

  => just close accounts for the customers who do that too often.
- Status
  - First product launched in October 1994. It claims more than 180,000 customer accounts and 2,650 merchants in 166 countries.
- www.fv.com

FirstVirtual



NetBill


1. Authentication phase
2. Price request
3. Price quote
4. Goods request
5. Goods, encrypted with key K
6. Signed EPO
9. Signed result, including K to unwrap goods
Web Browser
Checkbook
Web Server
Till
Customer
Merchant
7. Endorsed EPO (including K)
8. Signed purchase authorization (including K).
NetBill Server
Batch payment
Customer's Bank
Merchant's Bank
Settlement System

NetBill Features

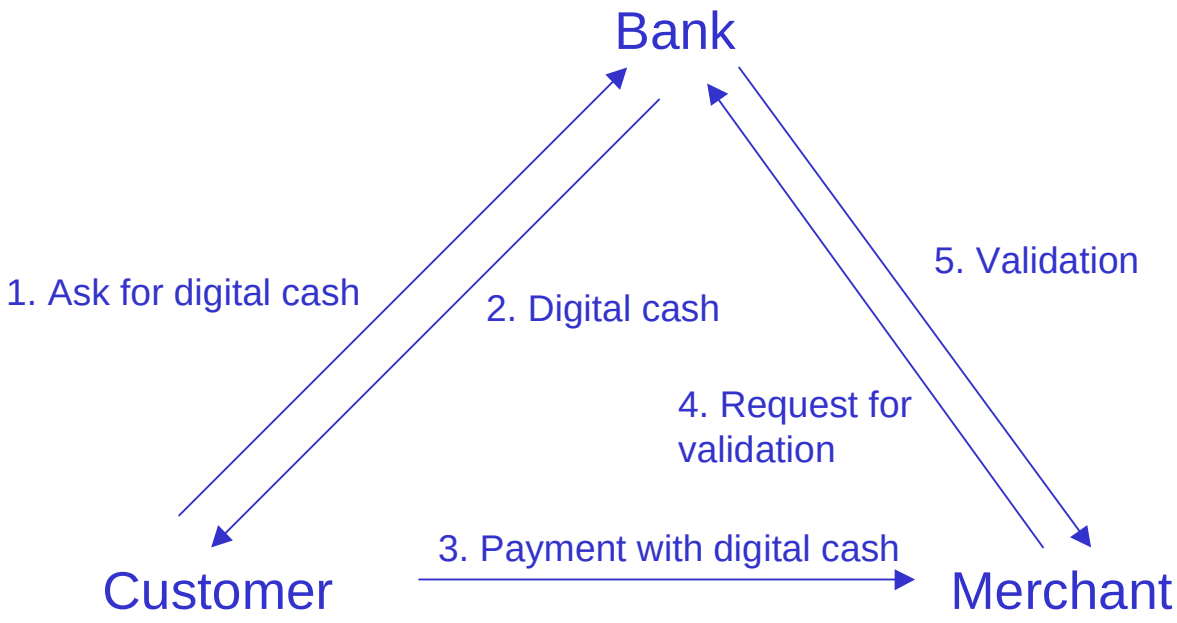
- Transaction cost
  - Medium due to requirement for extensive network traffic during transaction.
- Authentication and payment confidentiality
  - Parties identify themselves. Current scheme uses RSA and Kerberos.
- Message integrity
  - All network communication between Customer and Merchant is encrypted to protect against adversaries. More secure than FirstVirtual, but is it worth it?
- Nonrepudiation
  - Many ways to settle disputes.
- Anonymity
  - NetBill server knows both parties' identity and transaction amounts. Jean Camp's thesis adds anonymity to NetBill protocol.
- Atomicity
  - Money and goods atomic. Database transactions run at NetBill Server atomicity.
- Certified delivery
  - Guaranteed because of steps 5-9.
- Status
  - Alpha trial on campus. Uses fake money. Unclear future.

Digital Cash

- Token money
  - Dollar bills, coins, postage stamps, ...

Token-Based Approach: DigiCash

- Tokens, e.g., bills, coins, stamps



Problems with DigiCash

- Not atomic
  - What happens if the transaction is interrupted?
    - Suppose Message 3 is lost: Customer wants to spend money but Merchant did not get money. He might try to return money back to Bank but could be accused of double spending.
- Detecting double spending is expensive.

DigiCash: Attempt #1

1. C -> B: I'd like $100.
2. B -> C: $\{\text{I am a \$100 bill}\}_{SK_B}$
3. C -> M: $\{\text{I am a \$100 bill}\}_{SK_B}$
4. M -> B: $\{\text{I am a \$100 bill}\}_{SK_B}$
   Bank credits Merchant's account.
5. B -> M: OK

Problems

- Not anonymous: Bank can link Customer to transaction of that $100 payment for that Merchant (and maybe even item purchased).
- Copyable (double spending)

Attempt #2: Anonymity

Solve anonymity problem: use blind signatures.

- <$100> denotes a blinded $100 bill.

1. C -> B: Please sign this enclosed $100, <$100>
   B blindly signs <$100>.
2. B -> C: $<\$100>_{SK_B}$

Problem: Customer can cheat Bank and Merchant

1. C -> B: Please sign this enclosed $100, <$1>

...

Attempt #3: Fix Dollar Amount

Solve that problem: use cut-and-choose

1. C -> B: I would like \$100, <\$100>, <\$100>, …, <\$100>

B opens 999 and checks that they all say \$100. B signs the 1000th.

2. B -> C: $<\$100>_{SK_B}$

Attempt # 4: No Double Spending

Problem still: Double spending. $<\$100>_{SK_B}$ is easy to copy.

Solve this by adding a serial number to bill.

1. C -> B: <$100, #5703>, …

   B records in a database the serial numbers for all bills signed.

   When M deposits $<\$100, \#5703>_{SK_B}$ B checks database to make sure it hasn't been spent already.

Still some problems

- Database has to keep growing forever.
  - Solve this by having expiration dates on bills
    - Yikes! The value of cash is supposed to never “expire.”
- Two customers could coincidentally choose the same serial numbers for the same amount of money.
  - Solve this by requiring that the serial number includes a hash of the customer’s unique id or by requiring that serial numbers all be chosen from a sufficiently large space.
- Bank doesn’t know who cheated, Customer or Merchant?

Attempt # 5: Identification of Cheater

Solution: Use Random Identification Strings (RIS) that are guaranteed to be different for every payment such that the spender can be identified by the RIS.

- If B receives two identical bills with different RIS values, then C cheated.
- If B receives two identical bills with the same RIS values, then M cheated.

Problem: How can you create a bill that only the customer can write on inside the signature?

Solution: Use cryptography, e.g., use secret splitting and bit commitment scheme.

Attempt # 5, cont'd

- Let M = (I'm \$100, #5703, (y_{1L}, y_{1R}), (y_{2L}, y_{2R}), …, (y_{nL}, y_{nR}))
  - $y_{iL} = H(x_{iL})$, $y_{iR} = H(x_{iR})$, and H is a one-way hash function

- RIS = (**oneof**$\{x_{1L}, x_{1R}\}$, **oneof**$\{x_{2L}, x_{2R}\}$, …, **oneof**$\{x_{nL}, x_{nR}\}$).

- To trace Customer, we need a property such that the pattern of x_{iL}s and x_{iR}s reveals the Customer's identity.
  - E.g., let x_{iL} XOR x_{iR} = Customer Name.

How to Detect the Identity of the Cheater

- Then if Customer double spends and uses two different RISs, her name is revealed.
- Bank needs to make sure Customer behaves and to keep the Customer's anonymity without associating the Customer with the serial numbers.

Putting it all together

1. C -> B: I want \$100, <M>, …, <M>

   B opens all but one, signs the remaining one. (Cut-and-choose)

   B debits C's account by \$100.

2. B -> C: $< (\$100, \#5703, (y_{1L}, y_{1R}), (y_{2L}, y_{2R}), \ldots, (y_{nL}, y_{nR}))>_{SK_B}$

3. C -> M: $< (\$100, \#5703, (y_{1L}, y_{1R}), (y_{2L}, y_{2R}), \ldots, (y_{nL}, y_{nR}))>_{SK_B}$

   M verifies B's signature.

   M asks C to reveal either L or R of each y-pair.

4. M -> C: $(b_1, \ldots, b_n)$

   C computes RIS

5. C -> M: RIS

6. M -> B: RIS, $< (\$100, \#5703, (y_{1L}, y_{1R}), (y_{2L}, y_{2R}), \ldots, (y_{nL}, y_{nR}))>_{SK_B}$

   B checks #5703 in database.

   - If not there, B credits M's account by \$100. B records RIS and bill info.
   - If there, B checks RIS. If the same, M copied bill (double spent). If different, C copied.

Check Properties

- Can Customer cheat?
- Can Merchant cheat?
- Can Customer and Merchant collude?
  - No, Bank accepts only once because of unique serial number.
- Can Bank and Merchant collude to figure out Customer info?
  - No, blinding keeps Customer unlinked.
- What about Eve?
  - Yes, she can eavesdrop between C and M and deposit money before M. B will accept it and even worse, when M goes to deposit money, he'll be accused of cheating!
  - Or, she can spend the money before C and then C will be identified as cheater.
  - No way to prevent this because of anonymity of cash.

Brand's Protocol: Background math

- Define q to be a prime such that we can also define p to be prime and p = 2q + 1.
  - E.g., q = 5, p = 11. Note that Z_{11}^* = {2, 4, 8, 5, 10, 9, 7, 3, 6, 1}
- All math from here on is mod p.
- Find a subgroup of size q by choosing any element of Z_p^* (except 1) and use it as a generator, where g is an element of G_q, such as 4.
  - E.g., G_q = {4, 5, 9, 3, 1} = $\{g^x \mid 0 \leq x \leq q-1\}$
- Solving $y = g^x$ for x given g and y is DLP, which remember is hard.

More Background

- Let g_1 and g_2 be distinct generators of G_q.
  - They should be randomly chosen so no known rel'n holds between them.
- y is an arbitrary number.
- User knows a representation of y if he knows x1 and x2 s.t.

$$y = g_1^{x1} g_2^{x2}$$

- There are multiple such representations.
- Since DLP is hard, finding two such reps for a given y is also hard.

The Setup: User and Bank Do This First

- Bank makes a secret x.
- Bank publicizes g, h = g^x, g_1, h1 = g_1^x, g_2, and h2 = g_2^x.

- User makes a secret u_1. (Bank does not know this.)
- User gives I = $g_1^{u_1}$ to Bank. This is User's account number.
- User computes and remembers

$$Z = (I\ g_2)^x = g_1^{u_1 x} g_2^x$$

The Coin (must make it anonymous)

- User picks s, x_1, x_2 at random. He computes

$$A = g_1^{u_1} g_2^{s}$$

$$B = g_1^{x_1} g_2^{x_2}$$

- I.e., he knows a representation for A and B.
- User constructs coin as the triple

$$(A, B, sig_B)$$

- Assume blind signature is possible.

Payment Protocol

1. U -> M: A, B, sig_B
2. M -> U: d = Hash(A, B, UniqShopID, time)
3. U -> M: $r_1 = d(u_1 s) + x_1$, $r_2 = ds + x_2$

   M checks that $g_1^{r_1} g_2^{r_2} = A^d B$. If ok, then

4. M -> B: (A, B, sig_B), d, time, (r_1, r_2)

Catching Double Spending: Just math

- Goal: Bank has two info packets. It needs to compute User's account number, I.

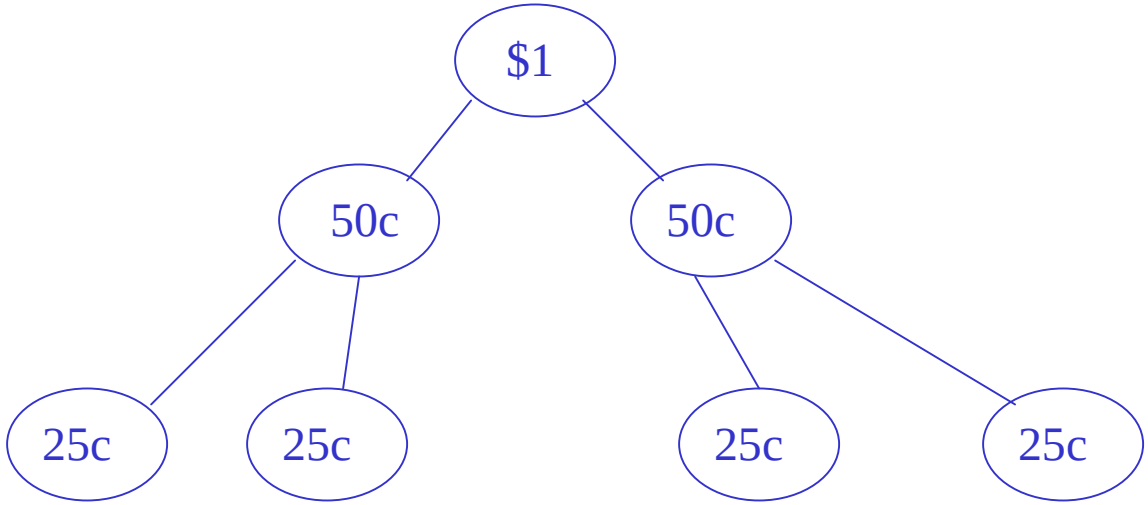
- Bank has (A, B, sig_B), d, time, (r_1, r_2), shop
  (A, B, sig_B), d', time', (r_1', r_2'), shop'
- Bank does two subtractions

$$\begin{array}{r} r_1 = d(u_1 s) + x_1 \\ -\, r_1' = d'(u_1 s) + x_1 \\ \hline r_1 - r_1' = u_1 s(d - d') \end{array} \qquad \begin{array}{r} r_2 = ds + x_2 \\ -\, r_2' = d's + x_2 \\ \hline r_2 - r_2' = s(d - d') \end{array}$$

- Then divides left by right to get u_1: $(r_1 - r_1')/(r_2 - r_2')$
- And then computes $I = g_1^{u_1}$ which is the user's account number.

Divisibility

- Problem: Coins cannot be subdivided.
- Solution (Okamoto and Ohta): Use trees to represent coins.
  - Buy whole tree. Spend pieces of it.



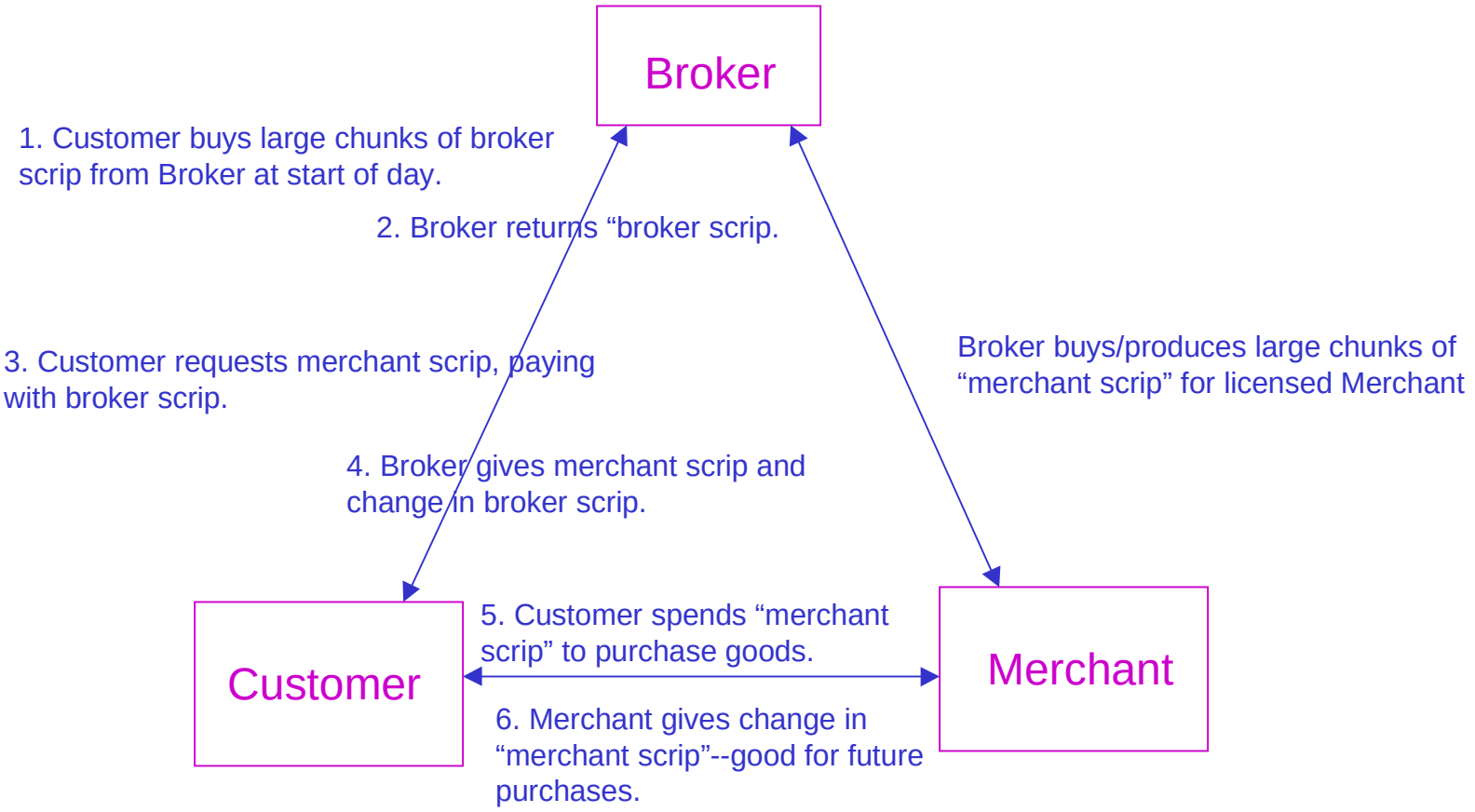
Using the Tree

- User sends signature on the root and the portion of the tree he wishes to spend.
  - When a node is used, all descendant nodes and all ancestor nodes of this node cannot be used.
  - No node can be used more than once.
- Why? To avoid double spending.
  - Implies need to use a cryptosystem that breaks user anonymity if user spends two coins that are ancestors of each other.
  - Weakness: user's spending can be linked since there is a way to tell which coins/nodes are ancestors of each other.
- Okamoto and Ohta's cryptosystem use Williams integers which are specific types of Blum integers…
- $1000 by the cent
  - 17-level tree, hence use on the average of 8 nodes in order to pay by the cent for each purchase (i.e., $334.36 payment)

Millicent (Manasse of SRC)

- Key idea is notion of *scrip*, digital cash for a specific vendor.
- On-line payment scheme for inexpensive (millicent), “casual” transactions.
  - Casual => people treat scrip the way they treat change in their pockets.

Millicent Model



Scrip Properties

- It has value at a specific merchant.
  - Text in scrip gives value and merchant identity.
- It can be spent only once.
  - Serial number to prevent double spending
- It is tamper resistant and hard to counterfeit.
  - Digital signature.
- It can be spent only by its rightful owner.
  - Customer signs each use of scrip with a secret associated with scrip.
- It can be efficiently produced and validated.
  - Signatures can be created and checked using fast one-way hash functions like MD5 or SHA.

Real Life

- Commercialized
  - Cybercash: www.cybercash.com
  - DigiCash: www.digicash.com
  - NetCheque: www.netcheque.com
  - FirstVirtual: www.fv.com
- In trial
  - SET
  - Millicent: http://www.millicent.digital.com/
  - CAFÉ
  - Mondex

Next lecture

- Topics: Electronic Voting, Electronic Auctions

- Readings
  - For this lecture
    - Required: CMU-CS-TR96-112 by Tygar; papers on iKP (Bellare et al.), Untraceable Electronic Cash (Chaum, Fiat, Naor), Untraceable Off-line Cash in Wallet with Observers (Brands), Universal Electronic Cash (Okamoto and Ohta), Millicent (Glassman et al.).
    - Edification: Section 13.4 (on public-key management).
  - For next lecture
    - Electronic voting, secure auction papers (to be handed out next time).