

**Risks**

To privacy:

1. The merchant learns your credit card number.
2. Merchant websites frequently hacked and CC numbers stolen.
3. The bank learns that you made a transaction with merchant X for amount Y.

To security:

1. Merchant charges a larger amount than agreed by client.

**Slide 3**

**Proposals to improve security of Credit Card payments**

1. iKP protocols (IBM)
2. SEPP (Mastercard)
3. STT (Visa and Microsoft)
4. SET (consortium, Mastercard, Visa, Netscape, IBM, MS) — combines ideas from iKP, SEPP, STT, — first version 1997, yet to be broadly adopted

**Electronic Payment Protocols**

**Slide 1**

**Credit Card Payments on the Web**

Insecure mechanism: credit card numbers sent as part of http request

“Secure” mechanism:

1. customer's browser opens SSL connection to merchant website, credit card number and payment amount sent encrypted
2. merchant website opens secure connection to customer's bank, sends credit card number and payment amount encrypted
3. (often) merchant website stores credit card number to expedite further transactions

### Parties to the Payment protocols

1. Customer C
  2. Merchant M
  3. Payment Gateway P,  
– a proxy for Merchant's bank & Customer's bank
- Objective:
1. Customer C wishes to obtain goods/service G from merchant M
  2. Merchant M wishes to receive payment from payment gateway P
  3. Payment gateway P charges payment to customers account

### Slide 5

#### Customer Requirements

1. Proof of Transaction Authorization by Payment Gateway  
The customer must have a proof that the payment gateway authorized the transaction.
2. Receipt from Merchant  
The customer must have a proof that the merchant who has made the offer has received payment and promised to deliver the goods/service
3. Confidentiality  
The customer's account information (*including credit card number*) should not be known to the merchant

### Slide 7

#### Merchant Requirements

1. Proof of Transaction Authorization by Payment Gateway
  2. Proof of transaction authorization by customer
- Payment Gateway Requirements:**
1. Proof of transaction authorization by customer
  2. Proof of transaction authorization by merchant

#### SET protocol

- Very complicated - over 400 pages of specification!
1. Cardholder Registration - cardholder registers signature key and a PIN-like secret with Certificate Authority
  2. Merchant Registration - merchant registers signature and encryption keys
  3. Purchase request - cardholder places order with merchant
  4. Payment Authorization - merchant verifies cardholder details with payment gateway, which authorizes transaction
  5. Payment capture - transfer of funds to merchant

Step 3: Customer sends Merchant agreed price, plus encrypted information (\*) for forwarding to payment gateway

C → M:

$$S^{Ks^c_1}(\text{TID}), \{ \text{TID}, \text{Price}_{CM}, K_{em} \}_{K_{ep}}$$

(\*)  $S^{Ks^c_1}(\{ \text{TID}, \text{Price}_{CP}, \text{Cardno}, \text{PIN} \}_{K_{ep}})$

where

1.  $K_{em} (K_{ep})$  is merchant (resp. payment gateway) public encryption key
2.  $\text{Price}_{CM} (\text{Price}_{CP})$  is price customer tells Merchant (resp. payment gateway)

Slide 11

Step 4: Merchant forwards (\*) to payment gateway, requests payment

M → P:

$$(*) S^{Ks^c_1}(\{ \text{TID}, \text{Price}_{CP}, \text{Cardno}, \text{PIN} \}_{K_{ep}}), S^{Ks^m_1}(\text{TID}), \{ \text{TID}, \text{Price}_{MP}, \text{MAC} \}_{K_{ep}}$$

where

1.  $K_{ep}$  is payment gateway public encryption key
2.  $\text{Price}_{MP}$  is price merchant asks to be charged to client account
3.  $\text{MAC}$  is merchant account number, for deposit

P verifies PIN, checks  $\text{Price}_{CP} = \text{Price}_{MP}$

We present a highly simplified *approximation* to Payment Authorization phase. (Based on the attempt to model SET by Lu and Smolka)

Slide 9

(after customer's search, negotiation of price)

Step 1: Customer Initiates payment phase

C → M: Initiate

Step 2: Merchant provides a transaction identifying number TID

M → C:  $S^{Ks^{-1}_1}(\text{TID})$

( $Ks^{-1}_1$  is merchant's private signature key)

# Digital Cash

Slide 15

- General structure of digital cash protocols:
1. customer withdraws money from her account, receives digital cash
  2. customer transfers digital cash to merchant in exchange for goods
  3. merchant deposits digital cash in his account

Step 5: Payment gateway confirms transaction result

$$P \rightarrow M: S^{K_{s^{-1}M}}(\text{TID}, \text{Result})$$

Step 6: Merchant forwards result to customer

$$M \rightarrow C: S^{K_{s^{-1}C}}(\text{TID}, \text{Result})$$

Slide 13

**Risks remaining with SET**

Your bank *still* learns that you made a transaction with merchant X for amount Y, and can build up a profile of you.

In the real world, paying in cash prevents this problem.

*Digital Cash* attempts to reproduce this privacy property in the digital world.

**How to verify authenticity of a coin**

Idea: get the bank to sign that the coin is authentic

coin is message

"This is a coin of value \$1 with number 345789234 issued by bank B, signed B"

Problem: bank can link coin number to the customer it was issued to!

Slide 19

Real world solution: (blind signatures)

Customer puts paper with coin number and value + carbon paper in a sealed envelope

Bank signs the envelope (pressing hard), returns to customer

customer opens envelope, removes paper, now containing a carbon copy of bank's signature

Two types of protocols:

- Online:** merchant verifies digital cash with bank at time of transaction
- Offline:** merchant verifies digital cash with bank some time after the transaction (e.g. deposits all cash at end of day)

Slide 17

**Risks to be avoided**

- Forgery of digital cash
  - use authenticity mechanisms to check that coin has been issued by bank if purports to come from
- Multiple spending
  - bank maintains record of coin number once used
  - (a) online protocols: check against multiple spending at time of transaction
  - (b) offline protocols: use mechanism to detect cheater identity in case of multiple spending

Problem: How to prevent the customer putting a piece of paper in the envelope that says "the bank will pay the customer \$1M"?

Solution 1: use a different key for each different coin value e.g. \$1

Solution 2: (cut and choose)

1. Customer sends bank 100 identical sealed envelopes

2. Bank randomly chooses 99/100, asks customer to open them,

verifies that the writing on the paper says what the customer

claimed

3. Bank signs the remaining envelope

If customer cheats, banks chance of catching them out is 99/100.

### Slide 21

#### Blind digital signatures

Let

$K^{-1} = (e, n)$  be Bob's RSA private signature key

$K = (d, n)$  be the corresponding (public) signature verification key

To get Bob to blindly sign a message  $M$ :

1. Alice generates a random number  $k$  relatively prime to  $n$

2. Alice computes  $k^d \cdot M$ , gets Bob to sign this

The signed message =

$(k^d \cdot M)^e \cdot M^e \pmod n = k \cdot M^e \pmod n$ , which after dividing by  $k$  is the same as  $M$  signed using  $K^{-1}$ !

#### Putting it together: An Online digital cash protocol

##### Withdrawal:

1. Alice creates (n copies of, for cut and choose) a coin of value  $V$   
(NB: Alice, not the bank randomly chooses the coin number!)  
and blinds it

2. Alice sends the blinded coin to the bank

3. (for cut and choose: Alice and bank run the cheating detection protocol)

4. Bank signs the blinded coin and debits Alice's account to value  $V$

5. Bank sends signed coin to Alice

6. Alice unblinds the coin

### Slide 23

#### Payment/Deposit:

1. Alice gives Bob the coin

2. Bob contacts Bank and sends the coin

3. Bank verifies signature on the coin

4. Bank verifies that coin has not already been spent

5. Bank enters coin in spent-coin database

6. Bank credits Bob's account and informs Bob

7. Bob gives Alice the goods.

$(g, p, q$  are public, only Alice knows  $m, b$ )

Alice tells Bob: I know the secret for  $(b, m)$ !

Bob: prove it!

Alice generates a random point  $(x, y)$  on the line

Bob verifies that

$$g^y = m^x \cdot b \pmod{p}$$

Bob's knowing one point on the line is not enough to know the line.

Suppose Alice and Bob play this game twice.

Then Bob knows two points on the line, hence can compute  $m$  and  $b$ !

Slide 27

### Applying Schnorr to avoid double spending

Basic idea: when the bank issues a coin to Alice, it encodes Alice's name in the coin using the Schnorr construction.

If Alice double spends, here name will be revealed.

### How to make this into an offline protocol?

Basic idea: include identifying information hidden in the coin, that is usually invisible, but that will be revealed if Alice (or Bob) tries to double-spend.

Slide 25

### Schnorr Proof of possession protocol

Alice would like to convince Bob that she knows a secret  $M$ , without revealing it.

$p, q$  large primes, with  $q$  a factor of  $p - 1$

$g$  a generator mod  $p$ , i.e. satisfying  $g^q = 1 \pmod{p}$

We can describe a line  $\{(x, y) \mid y = mx + b \pmod{q}\}$  by the quantities  $m$  and  $b$

The secret is encoded as such a pair  $(m, b)$

We hide  $m$  and  $b$  as  $b' = g^b \pmod{p}$  and  $m' = g^m \pmod{p}$

### Digital Cash in Practice?

1. Invented, patented by Chaum, mid-late 1980's
2. Chaum started a company Digicash
3. Minimal bank interest (competition with SET)
4. Digicash went into Chapter 11 bankruptcy, acquired by EPay, acquired by Infospace, patents sold to First Data (and on it goes....)
5. Not much in use, generally considered too complex.

Slide 29