Electronic payment systems

- overview of basic concepts
- credit-card based systems (MOTO, SSL, SET)
- electronic cash systems (DigiCash)
- micropayment schemes (PayWord, probabilistic schemes)

Overview of basic concepts

- brief history of money
- traditional forms of payment
  - cash
  - payment through bank
  - payment cards
- electronic payment systems
  - security requirements
  - basic classification
A brief history of money

- **barter**
  - most primitive form of payment
  - still used in primitive economies or under exceptional conditions
  - problem: “double coincidence of wants”
    - you want to change food for a bicycle
    - you need to find someone who is hungry AND has a spare bicycle

- **commodity money**
  - physical commodities which have recognized value
    - e.g., salt, gold, corn, …
  - desirable properties
    - portability
    - divisibility
  $\Rightarrow$ gold and silver coins became the most commonly used

A brief history of money (cont’d)

- **commodity standard**
  - ~ 19th century
  - use of tokens (e.g., paper notes) which are backed by deposits of gold and silver held by the note issuer
  - more comfortable and more SECURE!

- **fiat money**
  - tokens have value by virtue of the fact that a government declares it to be so AND this assertion is widely accepted
  - this works only if
    - the economy is stable
    - the government is trusted

- **electronic money**
  - ~ end of 20th century
  - paper tokens and metal coins are replaced by electronic representations of money
  - made possible by progress in computing and networking technology
Cash

- most commonly used form of payment today
  - ~80% of all transactions
  - average transaction value is low

- advantages of cash
  - easy to transport and transfer
  - no transaction costs (no third party is involved directly)
  - no audit trail is left behind (that's why criminals like it)

- disadvantages of cash
  - in fact, cash is not free
    - banknotes and coins need to be printed and minted
    - old bank notes and coins need to be replaced
    - this cost is ultimately borne by the tax payers
  - needs extra physical security when
    - transported in large quantities (e.g., from the mint to banks)
    - stored in large quantities (e.g., in banks)
  - vaults must be built and heavy insurances must be paid
  - risk of forgery

Payment through banks

- if both parties have accounts in a bank, then it is unnecessary for one party to withdraw cash in order to make a payment to the other party who will just deposit it again in the bank
Payment by check

- advantages
  - no need for bank at the time of payment

- disadvantages
  - returned items
    - if funds are not available on the payer’s bank account, then the check is returned to the payee’s bank
    - if the payee has already been credited, then the bank loses money
    - otherwise the payee suffers
    - problem: no verification of solvency of the payer at the time of payment
  - processing paper checks is very expensive and time consuming
    - checks must be physically transferred between banks
    - authenticity of each individual check must be verified

- still popular in some countries
  - e.g., in the US, ~80% of non-cash payment transactions are check payments with an average value of ~1000$
Payments cards - brief history

- 1915: first card was issued in the US ("shoppers plates")
- 1950: Diners Club card (used for travel and entertainment)
- 1958: American Express card was born
- ...: many card companies have started up and failed
- today: two major card companies dominate the world
  - VISA International
  - MasterCard

Overview of basic concepts

Payment by card

0. issue card
1. present card
2a. authorization*
2b. authorization
3. prepare voucher
4. sign voucher
5. send vouchers
6. clearing
7. monthly statement

* authorization is optional, depends on policy
Payment cards – pros and cons

- **advantages**
  - flexibility of cash and checks (assuming infrastructure is in place)
  - security of checks (no need to carry cash in pocket)
  - solvency of the customer can be verified before payment is accepted

- **disadvantages**
  - needs infrastructure to be deployed at merchants
    - e.g., card reader, network connection, etc.
  - transaction cost
    - covered by merchants
    - paying with cards is not worth for very low value transactions (below 2$)

Payment card types

- **debit card**
  - the customer must have a bank account associated with the card
  - transaction is processed in real time: the customer's account is debited and the merchant's account is credited immediately

- **charge card**
  - the customer doesn't need to pay immediately but only at the end of the monthly period
  - if she has a bank account, it is debited automatically
  - otherwise, she needs to transfer money directly to the card association

- **credit card**
  - the customer doesn't need to pay immediately, not even at the end of the monthly period
  - the bank doesn't count interest until the end of the monthly period
Main security requirements for e-payment

- **authorization**
  - a payment must always be authorized by the payer
  - needs payer authentication (physical, PIN, or digital signature)
  - a payment may also need to be authorized by the bank

- **data confidentiality and authenticity**
  - transaction data should be authentic
  - external parties should not have access to data
  - some data need to be hidden even from participants of the transaction
    - the merchant does not need to know customer account information
    - the bank doesn’t need to know what the customer bought

- **availability and reliability**
  - payment infrastructure should always be available
  - centralized systems should be designed with care
    - critical components need replication and higher level of protection

Main security requirements for e-payment (cont’d)

- **atomicity of transactions**
  - all or nothing principle: either the whole transaction is executed successfully or the state of the system doesn’t change
    - in practice, transactions can be interrupted (e.g., due to communication failure)
    - it must be possible to detect and recover from interruptions (e.g., to undo already executed steps)

- **privacy (anonymity and untraceability)**
  - customers should be able to control how their personal data is used by the other parties
  - sometimes, the best way to ensure that personal data will not be misused is to hide it
    - anonymity means that the customer hides her identity from the merchant
    - untraceability means that not even the bank can keep track of which transactions the customer is engaged in
Basic classification of e-payment systems

- pre-paid, pay-now, or pay-later
  - pre-paid: customer pays before the transaction (e.g., she buys electronic tokens, tickets, coins, ...)
  - pay-now: the customer's account is checked and debited at the same time when the transaction takes place
  - pay-later (credit-based): customer pays after the transaction

- on-line or off-line
  - on-line: a third party (the bank) is involved in the transaction (e.g., it checks solvency of the user, double spending of a coin, ...) in real-time
  - off-line: the bank is not involved in real-time in the transactions

Credit-card based systems

- motivation and concept:
  - credit cards are very popular today
  - use existing infrastructure deployed for handling credit-card payments as much as possible
  - enable secure transfer of credit-card numbers via the Internet

- examples:
  - MOTO (non-Internet based scheme)
  - First Virtual and CARI (non-cryptographic schemes)
  - SSL (general secure transport)
  - iKP (specific proposal from IBM)
  - SET (standard supported by industry including VISA, MasterCard, IBM, Microsoft, VeriSign, and many others)
MOTO – Mail Order / Telephone Order

- credit card number is sent via phone or post and then processed in the traditional way (using existing infrastructure)
- no cardholder signature!
  - user is allowed not to agree to the purchase (limited liability)
  - merchant must handle disputes (instead of banks)
  - some special rules and precautions are applied:
    - additional information is requested from user (e.g., name, address)
    - goods are delivered to the address associated with the card
- fraud is still possible (but hopefully the benefits outweigh the disadvantages)
- still very popular (in the US and Western Europe)

Credit-card payment with SSL

- the user visits the merchant’s web site and selects goods/services to buy
  - state information may be encoded in cookies or in specially constructed URLs
  - or state information may be stored at the merchant and referenced by cookies or specially constructed URLs
- the user fills out a form with his credit card details
- the form data is sent to the merchant’s server via an SSL connection
  - the merchant’s server is authenticated
  - transmitted data is encrypted
- the merchant checks the solvency of the user
- if satisfied, it ships the goods/services to the user
- clearing happens later using the existing infrastructure deployed for credit-card based payments
Pros and cons of SSL

- **advantages:**
  - SSL is already part of every browser and web server
    - no need to install any further software
    - users are used to it
    - this payment method can be used as of today

- **disadvantages:**
  - eavesdropping credit card numbers is not the only risk
  - another risk is that credit card numbers are stolen from the merchant's computer

SET - Secure Electronic Transactions

- a protocol designed to protect credit card transactions on the Internet
- initiated and promoted by MasterCard and Visa
  - MasterCard (and IBM) had SEPP (Secure E-Payment Protocol)
  - VISA (and Microsoft) had STT (Secure Transaction Technology)
  - the two proposals converged into SET
- many companies were involved in the development of the specifications (IBM, Microsoft, Netscape, RSA, VeriSign, ...)
- the SET specification is available on the web (Google)
- it consists of three books:
  1. Business Description
  2. Programmer’s Guide
  3. Formal Protocol Definition

(around 1000 pages all together)
**SET participants**

- **cardholder**
  - wants to buy something from a merchant on the Internet
  - authorized holder of payment card issued by an issuer (bank)
- **merchant**
  - sells goods/services via a Web site or by e-mail
  - has a relationship with an acquirer (bank)
- **issuer**
  - issues payment cards
  - responsible for the payment of the dept of the cardholders
- **acquirer**
  - maintains accounts for merchants
  - processes payment card authorizations and payments
  - transfers money to the merchant account, reimbursed by the issuer
- **payment gateway**
  - interface between the Internet and the existing credit-card payment network
- **CAs**

**Overview of operation**

Cardholder interacts with the merchant via the Internet, initiating an order and payment process. The transaction is processed through the payment gateway and the issuer, ensuring secure financial transactions. The diagram illustrates the flow of information and funds between participants.
**SET services**

- **cardholder account authentication**
  - merchant can verify that the client is a legitimate user of the card
  - based on X.509 certificates

- **merchant authentication**
  - client can authenticate the merchant and check if it is authorized to accept payment cards
  - based on X.509 certificates

- **confidentiality**
  - cardholder account and payment information (i.e., her credit card number) is protected while it travels across the network
  - credit card number is hidden from the merchant too!

- **integrity**
  - messages cannot be altered in transit in an undetectable way
  - based on digital signatures

---

**Dual signature - basic concept**

- **goal:**
  - link two messages that are intended for two different recipients (e.g., order info and payment instructions in SET)
  - link may need to be proven in case of disputes
Dual signatures in SET

- goal:
  - same as in the basic case, but ...
  - the two messages have the same signature

Overview of message protection mechanisms

Cardholder (C)
- PReq
- OI
- signature
- dual signature
- digital envelop

Merchant (M)
- Auth.Req.
- PI
- K_{M}

Acquirer (via payment gtw)
- Cap.Req.
- Cap.Token
- K_{A}
- K_{M}

Cardholder (C)
- PRes
- PI
- K_{C}

Merchant (M)
- Auth.Res.
- Cap.Token
- K_{A}
- K_{M}

Acquirer (via payment gtw)
- Cap.Res.
- K_{A}
- K_{M}
Why did SET fail?

- less benefits than expected
  - merchants like to collect credit card numbers (they use it as indexes in marketing databases)
  - optionally, SET allows the merchant to get the credit card number from the acquirer → security improvements of SET are negated

- too high costs
  - SET requires a PKI

- no advantages for the customer!
  - the idea was that SET transactions would be handled as "cardholder present" transactions (due to the digital signature)
  - customers prefer MOTO-like systems where they can freely reverse a transaction if they are unhappy (not only in case of fraud) → customers were much worse off
  - SET requires the download and installation of a special software, and obtaining a public-key certificate

Electronic cash

- motivation and concept:
  - people like cash (75-95% of all transactions in the world are paid in cash)
  - design electronic payment systems that have cash-like characteristics
  - it is possible to ensure untraceability of transactions (an important property of real-world cash)

- examples:
  - DigiCash (on-line)
  - CAFE (off-line)
E-cash – a naïve approach

- electronic coins:
  \[(\text{serial\_number}, \text{value}, \text{Sig}_{\text{bank}}(\text{serial\_number}, \text{value}))\]

- problem 1: double spending
  - a solution to problem 1:
    - merchants deposit received coins before providing any service or goods
    - the bank maintains a database of spent serial numbers
    - only coins that have never been deposited before are accepted by the bank

- problem 2: ever increasing database at the bank
  - a solution to problem 2:
    - coins have an expiration time: \((\text{sn}, \text{val}, \text{exp}, \text{Sig}_{\text{bank}}(\text{sn}, \text{val}, \text{exp}))\)
    - bank needs to store deposited coins until their expiration time only

E-cash – a naïve approach (cont’d)

- problem 3: traceability

- a solution to problem 3: DigiCash
The main idea of DigiCash

- **blind RSA signatures**
  - the bank’s public RSA key is \((e, m)\), its private RSA key is \(d\)
  - user \(U\) generates a serial number \(s\), and a random number \(r\) (blinding factor)
  - \(U\) computes \(s \cdot r^e\) and sends it to her bank
  - the bank signs the blinded serial number by computing \((s \cdot r^e)^d = s^d \cdot r\)
  - when \(U\) receives the blindly signed serial number, it removes the blinding: \(s^d \cdot r \cdot r^{-1} = s^d\)
  - \(U\) obtained a digital signature of the bank on the serial number
  - the bank cannot link \(s^d \cdot r\) and \(s^d\) together (\(r\) is random)

- **notes**
  - the user must authenticate herself to the bank when withdrawing money, so that the bank can charge her account
  - the merchant must authenticate himself to the bank when depositing money, so that the bank can credit his account
  - messages should be encrypted in order to prevent theft of money

Further precautions

- **the serial number \(s\) shouldn’t be random**
  - a forger generates a random number \(c\) and computes \(s = c^e\)
  - since \(c = s^d\), \(c\) looks like a serial number signed by the bank (i.e., a coin)

- **two solutions:**
  1. \(s\) should have a well defined structure (e.g., its second half is a known function of its first half)
  2. \(s\) shouldn’t be a simple serial number, but \(s = h(\text{coin}_\text{data})\), where \(\text{coin}_\text{data} = \text{serial}_\text{number}, \text{value}, \text{expiration}_\text{time}, \ldots\)
Different denominations

- the bank signs the blinded coin → it does not know the value of the coin
  - how much should the user be charged?
- one can allow a single denomination only in the system, but that wouldn’t be practical
- in DigiCash, the bank uses different signing keys for different denominations

Micropayment schemes

- motivation and concept:
  - many transactions have a very low value (e.g., paying for one second of a phone call, for one article in a newspaper, for one song from a CD, for 10 minutes of a TV program, etc.)
  - transaction costs of credit-card, check, and cash based payments may be higher than the value of the transaction
  - need solutions optimized for very low value transactions (perhaps by sacrificing some security)
- examples:
  - PayWord
  - probabilistic micro-payment schemes
- the truth: micropayment schemes are not very successful so far
  - people are used to get these kind of things for free
  - if they have to pay, they prefer the subscription model
PayWord

- designed by Rivest and Shamir in 1996
- representative member of the big family of hash-chain based micropayment schemes
- check-like, credit based (pay later) system
  - payment tokens are redeemed off-line
- uses public key crypto, but very efficiently (in case of many consecutive payments to the same vendor)
  - the user signs a single message at the beginning
  - this authenticates all the micropayments to the same vendor that will follow

PayWord model

- players:
  - user (U)
  - vendor (V)
  - broker (B)
- phases:
  - registration (done only once)
  - payment
  - redemption

user

PayWord commitment

micropayment tokens

vendor

account information (e.g., credit-card number)

PayWord certificate

commitment + last received token

broker
Registration phase

- U provides B with
  - account information in a real bank (e.g., her credit card number)
  - shipping address
  - public key

- B issues a certificate for U

\[ \text{Cert}_U = \{ B, U, \text{addr}_U, K_U, \text{exp}, \text{more}_\text{info} \}_{K_B}^{-1} \]

\text{more}_\text{info}: \text{serial number, credit limit, contact information of B, broker terms and conditions, ...}

- the certificate is a statement by B to any vendor that B will redeem authentic paywords (micropayment tokens) produced by U turned in before the expiration date

Payment phase – generating the commitment

- when U is about to contact a new vendor, she computes a fresh payword chain

\[ w_n, w_{n-1} = h(w_n), w_{n-2} = h(w_{n-1}) = h^{(2)}(w_n), \ldots, w_0 = h^{(n)}(w_n) \]

where
- \( n \) is chosen by the user
- \( w_n \) is picked at random

- U computes a commitment

\[ M = \{ V, \text{Cert}_U, w_0, \text{date}, \text{more}_\text{info} \}_{K_U}^{-1} \]

- the commitment authorizes B to pay V for any of the paywords \( w_1, \ldots, w_n \) that V redeems with B before the given date

- paywords are vendor specific, they have no value to another vendor
Payment phase – sending micropayment tokens

- The i-th micropayment from U to V consists of the i-th payword and its index: \((w_i, i)\)
- When V receives \(w_i\), it can verify it by checking that it hashes into \(w_{i-1}\) (received earlier, or in the commitment in case of \(i = 1\))
- Since the hash function is one-way (preimage resistant) the next payment \(w_{i+1}\) cannot be computed from \(w_i\)
- V needs to store only the last received payword and its index
- Variable size payments can be supported by skipping the appropriate number of paywords
  - Let’s assume that the value of each payword is 1 cent
  - And the last payword that U sent is \((w_k, k)\)
  - If U wants to perform a payment of 10 cents, then she sends \((w_{k+10}, k+10)\)

Redemption phase

- At the end of each day, the vendor redeems the paywords for real money at the broker
- V sends B a redemption message that contains (for each user that contacted V) the commitment and the last received payword \(w_k\) with its index \(k\)
- B verifies the commitment and checks that iteratively hashing \(w_k\) \(k\) times results in \(w_0\)
- If satisfied, B pays V \(k\) units and charges the account of U with the same amount
Efficiency

- user U
  - needs to generate one signature per "session"
  - needs to perform as many hash computation as the number of paywords needed (pre-computation of hash chains is possible)
  - needs to store the hash chain and her current position in the chain (time-memory trade-off is possible)

- vendor V
  - needs to verify one signature per "session"
  - needs to perform one hash computation per micropayment received
  - needs to store only the last received payword with its index, and the commitment

- broker B
  - needs to verify signatures and compute lot of hashes but all these are done off-line

Probabilistic micropayment schemes

- motivation:
  - in traditional micropayment schemes, the vendor cannot aggregate micropayments of different users
  - if the user spent only a few cents, then the cost of redeeming the micropayment tokens may exceed the value of the payment
  - example: typical value of a payword is 1 cent, whereas processing a credit-card transaction costs about 25 cents

- main idea:
  - suppose that U wants to pay 1 cent to V
  - U sends to V a lottery ticket that is worth 10$ if it wins, and it wins with probability 0.001
  - the expected value of U's payment is exactly 1 cent
  - if V conducts business with many users, then he approximately earns the value of the services/goods provided
  - advantage: only winning lottery tickets are redeemed at the bank
    - number of vendor-bank transactions is greatly reduced
    - value of lottery tickets surely exceeds the transaction cost
**Micali-Rivest scheme**

- check based, the user simply signs the transaction
- notation
  - T – encoding of the transaction (IDs of user, merchant, bank, transaction time, value, etc.)
  - F – fixed public function that maps an arbitrary bit string to a number between 0 and 1
  - s – fixed selection rate of payable checks
- setup
  - everyone establishes his own public key and corresponding private key for a digital signature scheme
  - the merchants signature scheme must be deterministic
    - \( \text{Sig}_M(x) = \text{Sig}_M(x') \) if \( x = x' \)

**Micropayment schemes / Probabilistic schemes**

**Micali-Rivest scheme (cont’d)**

- payment
  - user U pays by sending \( C = (T, \text{Sig}_U(T)) \) to merchant \( M \)
  - \( M \) verifies if \( C \) is payable by checking if \( F(\text{Sig}_M(C)) < s \)
- selective deposit
  - \( M \) sends only payable checks to the bank for deposit
  - after verification, \( B \) credits \( M \)'s account with \( 1/s \) cents and debits \( U \)'s account with the same amount
Some properties of the Micali-Rivest scheme

- \( \text{Sig}_M(C) \) is unpredictable for both \( U \) and \( M \)
  - practically, \( F(\text{Sig}_M(C)) \) is a random number with close to uniform distribution over \([0, 1]\)
  - the probability that \( F(\text{Sig}_M(C)) < s \) is \( s \)
  - expected value of a check is 1 cent

- the bank essentially processes macropayments of value \( 1/s \)
  - e.g., if \( s = 1/1000 \), then the value is 10$

- potential "psychological" problem
  - possibility of user's excessive payments (in the short term)
  - e.g., it has a positive probability that the first 10 checks sent by the user are all payable
    - value of the goods/services received by the user is 10 cent
    - but her account is debited 100$
  - in the long run it will work, but users may not tolerate the risk of short term overpaying

Modified Micali-Rivest scheme

- notation and setup
  - same as for the basic Micali-Rivest scheme

- payment
  - \( U \) pays by sending \( C = (T, \text{Sig}_U(T)) \) to \( M \)
  - \( T \) contains a serial number \( SN \) (assigned sequentially to transactions by \( U \))
  - \( M \) verifies if \( C \) is payable by checking if \( F(\text{Sig}_M(C)) < s \)

- selective deposit
  - \( M \) sends only payable checks to the bank for deposit
  - \( \text{maxSN}_U \) denotes the highest serial number corresponding to \( U \) processed by \( B \) so far
  - if \( B \) receives a new payable check, then
    - \( B \) credits \( M \)'s account with \( 1/s \)
    - if \( SN > \text{maxSN}_U \), then it debits \( U \)'s account with \( \text{SN}-\text{maxSN}_U \) and sets \( \text{maxSN}_U \) to \( SN \)
Illustration of the modified MR scheme

Issued checks with serial numbers

<table>
<thead>
<tr>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

Note: total debit of the user is always less than or equal to the highest serial number signed by the user so far

Note: total debit of the user is always less than or equal to the highest serial number signed by the user so far

Some properties of the modified MR scheme

- Cheating is possible
  - The same serial number may be used with different merchants
  - If only one of the two checks is payable than the cheating will not be detected

- However, large scale cheating can be detected with statistical auditing
  - Example:
    - Assume the user uses every serial number twice
    - Number of payments made by the user is N
    - Highest serial number used is N/2, user is charged at most N/2 cents
    - Joint credit of the merchants is approximately N
    - This can be detected by the bank
    - In addition, the more the user cheats the higher the probability of two merchants depositing checks with the same serial number
Summary

- credit-card based
  - MOTO: non-cryptographic
  - SSL: most used today
  - SET: dual signature

- e-cash
  - DigiCash: untraceable, on-line

- micropayments
  - PayWord: hash chains
  - probabilistic: lottery tickets