Overview: Read Chapters

1. Introduction and Overview
- ECommerce: Definition, Benefits, Statistics, ...

2. EC from a Business Perspective
- A Taxonomy of EC Business Models
- Trends & Directions

3. Enabling Web- and Software Technologies from the perspective of an EC enterprise
- Client / Server Architectures and the Internet
- Web Protocol: HTTP
- Multi-Tier Web Architectures
- Three-Tier Web Architecture
- Platform Choices and Connectivity Options
- Mapping (Inter-)Organizational Structures to Enterprise Networks

4. B2C, B2E Systems: Concepts and Architectures
- Business-to-Consumer Systems
- Business-to-Employee: Enterprise Information and Knowledge Management

5. Concepts and Technologies for B2C Transaction

5.1 Making Contact on the Web

5.2 Negotiating with a Consumer

5.3 Electronic Fulfillment & Payment

5.3.1 Secure Communication, Security and Trust

5.3.2 Encryption: Standards, Authentication: Digital Signatures, Certification Authorities

5.3.2 Electronic Payment Models, Standards and Systems

5.4 Supporting and Improving Customer Feedback

Security and Trust (1)

Security and **trust** are important factors for ECommerce:

- In commerce the *security* of money and goods is most important.
- Buyers will only trade with sellers whom they *trust*.

Security can be achieved technically (e.g., by encryption, authentication, access control to resources).

Trust cannot be achieved technically, can be gained by

- being certified by a trusted organization (e.g., Trust-E)
- number of customers (increases confidence)
- word of mouth (tell your friend)

Security (1)

First step: Secure communication:

Business partners send authentication information, digital cash, contracts over the Internet. For this secure communication is needed. Secure transmission of web documents is achieved, e.g., via Secure HTTP (see chapter 3).

But this is not enough for **secure commerce.** Questions concerning secure commerce:

- Customer view:
  - Ensure that the merchant is who he claims to be (merchant authentication)
  - Ensure merchant is not selling customer profile information (see profiling standards)
- Merchant view: Problems with “interrupted” business processes: Customer denies having
  - placed an order
  - signed a digital contract (**repudiation**).

Security (2)

Problems and proposed solutions (details on following slides):

PROBLEM	SOLUTION
Eavesdropping: Third parties (man in the middle) read exchanged documents (orders)	Privacy (Encryption)
Modification of documents (e.g., orders)	Integrity (Digital Signatures), Privacy (Encryption)
Erasure of documents	Backups, Multiple Transmissions, Mirrors
Spoofing (simulate an identity)	Certificates, Certification Authorities, Public Key Infrastructure (PKI)
Repudiation	Non-repudiation (Sign contract with digital signature)

Trade off between effort and risk (when to apply security measures).

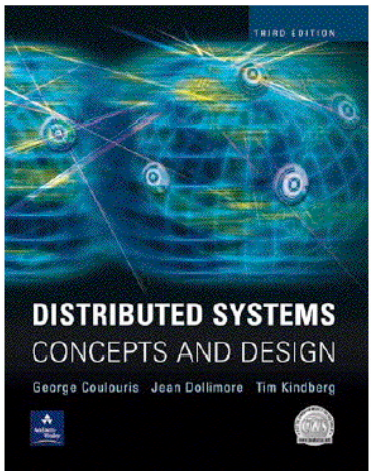
Example: Securing documents against eavesdropping / modification:

- Irrelevant for information browsing
- Relevant when credit-card numbers, amounts of money are sent across the Internet

Cryptography in E-Commerce

Ralf Möller, TUHH

Based on:



Introduction

Cryptography: encode message data so that it can only be understood by intended recipient.

Romans used it in military communication

Given knowledge of encryption algorithm, brute force attempt: try every possible decoding until valid message is produced.

Computers are good at this!

Modern schemes must be computationally hard to solve to remain secure.

Cryptographic Terminology

Plain text: the message before encoding.

Cipher text: the message after encoding.

Key: information needed to convert from plain text to cipher text (or vice-versa).

Function: the encryption or decryption algorithm used, in conjunction with key, to encode or decode message.

Key distribution service: trusted service which hands out keys.

Encryption

Encrypting data prevents unauthorised access and modification to the data (i.e. prevents eavesdropping and tampering).

If encrypted data can only be decrypted with a matching key, this can be used to prove sender's identity (i.e prevents masquerading).

Likewise, it can be used to ensure that only intended recipients can use the data.

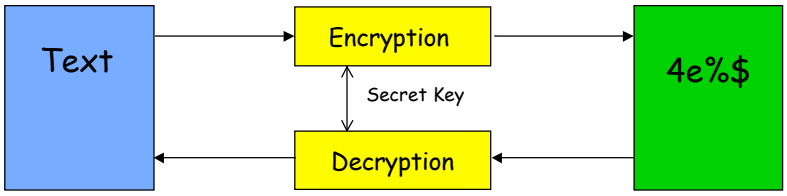
Two main ways: secret key & public key.

Secret Keys

One key is used to both encrypt and decrypt data

Encryption and decryption functions are often chosen to be the same

Security should not be compromised by making function well-known as security comes from secret keys



Using Secret Keys

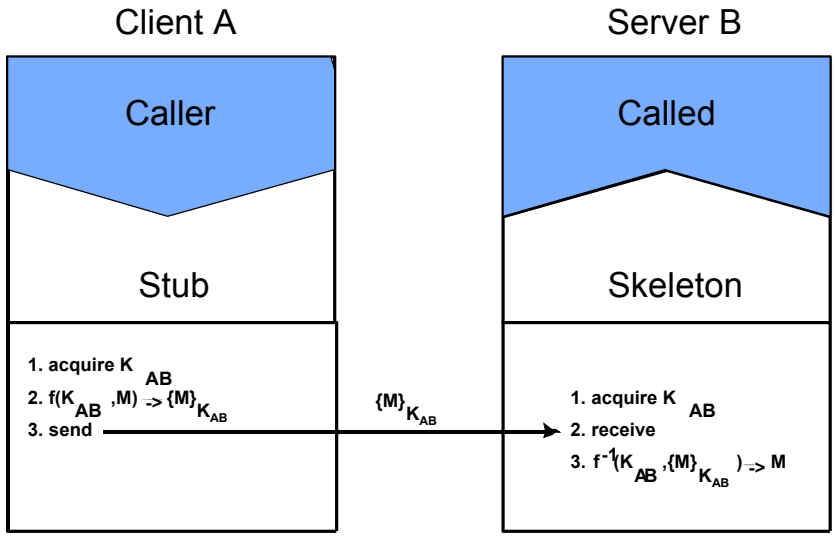
Sender and recipient exchange keys through some secure, trusted, non-network based means

Sender encodes message using function and sends, knowing that only the holder of key (the intended recipient) can use it

Recipient decodes message and knows that only sender could have generated it

Message can be captured but is of no use

Using Secret Keys in Object Request



Brute force approaches for determining K_{AB}

Given Message M‘ = $\{M\}K_{AB}$

1. For all k

   If useful(M'_k) then return k

2. For all k

   For all M

   If M_k= M‘ then return k

Public Keys

Diffie and Hellman 1976

Gives 'one-way' security.

Two keys generated, one used with decryption algorithm (private key) and one with encryption algorithm (public key).

Generation of private key, given public key, is computationally hard.

Do not need secure key transmission mechanism for key distribution.

Using Public Keys

Recipient generates key pair.

Public key is published by trusted service.

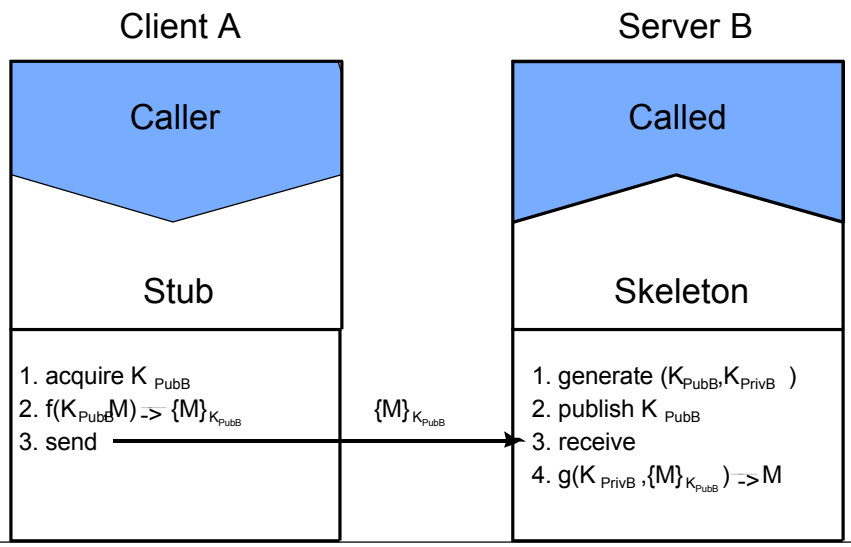
Sender gets public key, and uses this to encode message.

Receiver decodes message.

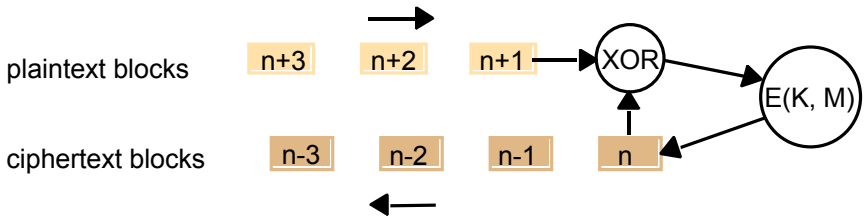
Replies can be encoded using sender's public key from the trusted distribution service.

Message can be captured but is of no use.

Using Public Keys in Object Request

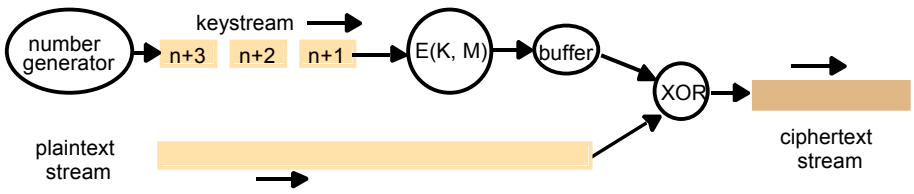


Cipher Block Chaining



Initialization vector required (e.g., timestamp)

Stream Ciphers



Number generator:
E.g. random number with seed value on which both parties agree

Cryptography: Secret Keys

Main Ideas

- Confusion (XOR, circular shifing, ...)
- Diffusion (transposition of plaintext block portions)

Cyptographic Algorithms

- DES (Data Encryption Standard, 1981, 56bit)
  - Encryption and decryption function identical
- TEA (Tiny Encryption Algorithm, 128bit)
  - Wheeler and Needham 94
- IDEA (Intern. Data Encryption Algorithm, 128bit)
- Blowfish

TEA encryption function

```
void encrypt(unsigned long k[], unsigned long text[]) {
    unsigned long y = text[0], z = text[1];                          1
    unsigned long delta = 0x9e3779b9, sum = 0; int n;                2
    for (n= 0; n < 32; n++) {                                        3
        sum += delta;                                                4
        y += ((z << 4) + k[0]) ^ (z+sum) ^ ((z >> 5) + k[1]);        5
        z += ((y << 4) + k[2]) ^ (y+sum) ^ ((y >> 5) + k[3]);        6
    }
    text[0] = y;  text[1] = z;                                       7
}
```

TEA decryption function

```
void decrypt(unsigned long k[], unsigned long text[]) {
    unsigned long y = text[0], z = text[1];
    unsigned long delta = 0x9e3779b9, sum = delta << 5;  int n;
    for (n= 0; n < 32; n++) {
        z -= ((y << 4) + k[2]) ^ (y + sum) ^ ((y >> 5) + k[3]);
        y -= ((z << 4) + k[0]) ^ (z + sum) ^ ((z >> 5) + k[1]);
        sum -= delta;
    }
    text[0] = y; text[1] = z;
}
```


TEA in use

```
void tea(char mode, FILE *infile, FILE *outfile, unsigned long k[]) {
/* mode is 'e' for encrypt, 'd' for decrypt, k[] is the key.*/
    char ch, Text[8]; int i;
    while(!feof(infile)) {
        i = fread(Text, 1, 8, infile);          /* read 8 bytes from infile into Text */
        if (i <= 0) break;
        while (i < 8) { Text[i++] = ' ';}    /* pad last block with spaces */
        switch (mode) {
        case 'e':
            encrypt(k, (unsigned long*) Text); break;
        case 'd':
            decrypt(k, (unsigned long*) Text); break;
        }
        fwrite(Text, 1, 8, outfile);            /* write 8 bytes from Text to outfile */
    }
}
```

Cryptography: Public Keys

$D(K_d(E(K_e,M)) = M$

Decryption key K_d must be a secret

Encryption key K_e is public

RSA Encryption - 1

To find a key pair *e*, *d*:

1. Choose two large prime numbers, *P* and *Q* (each greater than 10100), and form:
   $N = P \times Q$
   $Z = (P-1) \times (Q-1)$
2. For *d* choose any number that is relatively prime with *Z* (that is, such that *d* has no common factors with *Z*).
   We illustrate the computations involved using small integer values for *P* and *Q:*
   $P = 13$, $Q = 17 \rightarrow N = 221$, $Z = 192$
   $d = 5$
3. To find *e* solve the equation:
   $e \times d = 1 \bmod Z$

That is, $e \times d$ is the smallest element divisible by *d* in the series $Z+1, 2Z+1, 3Z+1, \ldots$.

$e \times d = 1 \bmod 192 = 1, 193, 385, \ldots$
385 is divisible by *d*
e = 385/5 = 77

RSA Encryption - 2

To encrypt text using the RSA method, the plaintext is divided into equal blocks of length k bits where $2^k < N$ (that is, such that the numerical value of a block is always less than N; in practical applications, k is usually in the range 512 to 1024).

k = 7, since $2^7 = 128$

The function for encrypting a single block of plaintext M is:

$E'(e,N,M) = M^e \bmod N$

for a message M, the ciphertext is $M^{77} \bmod 221$

The function for decrypting a block of encrypted text c to produce the original plaintext block is:

$D'(d,N,c) = c^d \bmod N$

Rivest, Shamir and Adelman proved that E' and D' are mutual inverses (that is, $E'(D'(x)) = D'(E'(x)) = x$) for all values of P in the range $0 \le P \le N$.

The two parameters e,N can be regarded as a key for the encryption function, and similarly d,N represent a key for the decryption function.

So we can write K_e = <e,N> and K_d = <d,N>, and we get the encryption function:

$E(K_e, M) = \{M\}_K$ (the notation here indicating that the encrypted message can be decrypted only by the holder of the private key K_d) and $D(K_d, \{M\}_K) = M$.

Blowfish

Symmetric block cipher encryption/decryption algorithm

Can be used as a drop-in replacement for DES or IDEA.

Takes a variable-length key, from 32 bits to 448 bits, ...

... making it ideal for both domestic and exportable use.

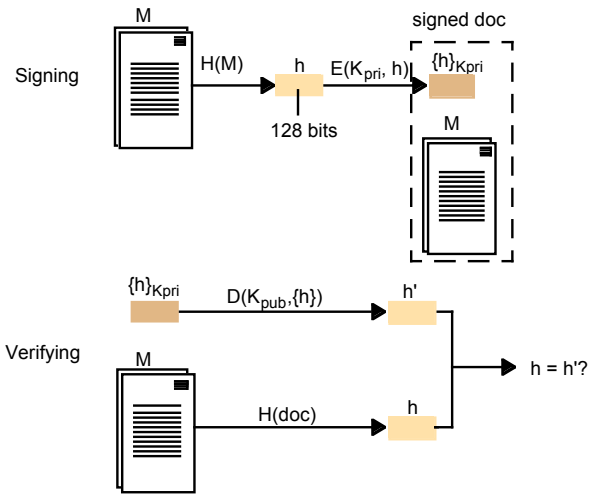
Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. Blowfish is unpatented and license-free, and is available free for all uses.

Authentication: Digital Signatures (1)

Definition

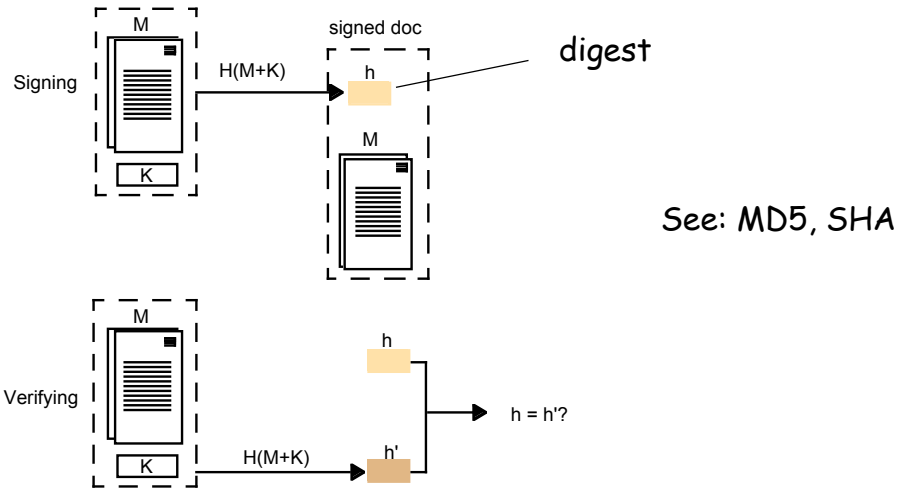
A **digital signature** is applied to a digital document. It states that the document (as an entirety) has been signed by the signer. Others can verify this.

Digital Signatures with Public Keys



Digital Signatures with Secret Keys

MACs: Message Authentication Codes



Performance of encryption and secure digest algorithms

	Key size/hash size (bits)	*Extrapolated speed (kbytes/sec.)*	*PRB optimized (kbytes/s)*
TEA	128	700	-
DES	56	350	7746
Triple-DES	112	120	2842
IDEA	128	700	4469
RSA	512	7	-
RSA	2048	1	-
MD5	128	1740	62425
SHA	160	750	25162

Authentication: How to distribute keys?

Key distribution by non-networked means is impractical for distributed object systems

Key distribution is a problem for both secret and public keys

- Secret keys: Obvious
- Public keys: How do we know the principal that gives us a public key is who we assume the principal is?

Use trusted key distribution service and secure key distribution protocol!

Needham/Schroeder Protocol

Provides a secure way for pairs of components to obtain keys to be used during communication

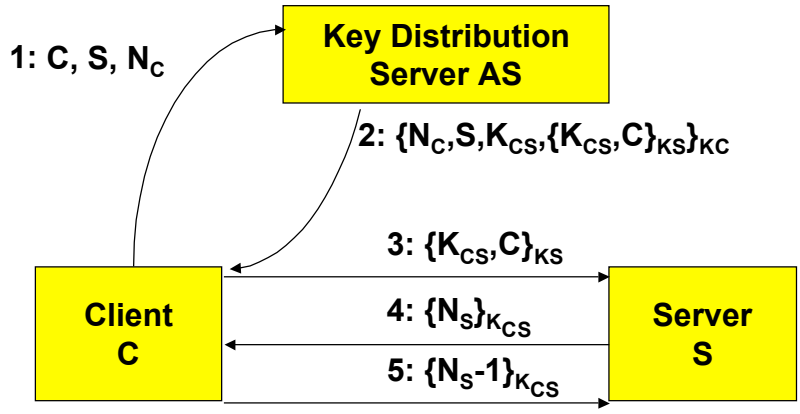
Based on an authentication server:

- maintains a name and a secret key for each component
- can generate keys for peer-to-peer communications

Secret keys are used for communication with authentication server

Needham/Schroeder Protocol

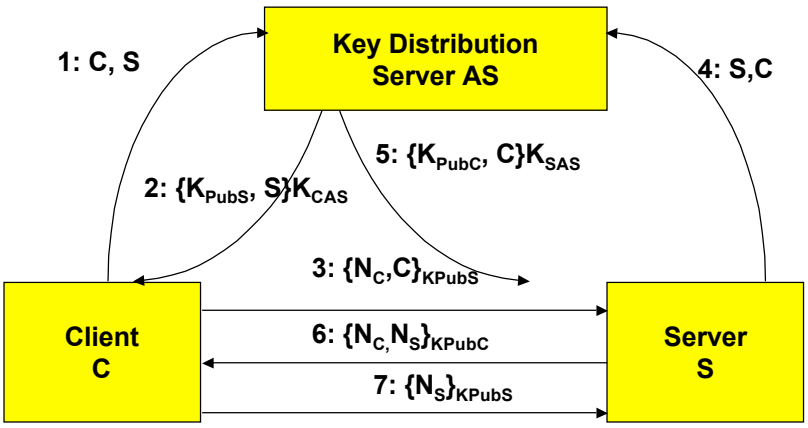
For Secret Keys:



Electronic Commerce (WS-02/03) 5-33

Needham/Schroeder Protocol

For Public Keys:



Electronic Commerce (WS-02/03) 5-34

Secure Socket Layer (SSL)

Secure Transport between Browser and Web-Server

Also used for object-oriented middleware

Based on RSA public key technology

Client generates secret session key

Client uses public key of server to encrypt session key and transmit it to the server

Session key is used to encrypt any communication between client and server

Authentication: Blind Signature (1)

Blind signatures are a way of signing electronic data that can be **authenticated without revealing** some **important aspect(s) of the information owner / information creator.**

Two Roles:

Information owner / information creator is one role.
Signer / authenticator is the other role.

Examples:

- Electronic voting: the identity of the person who voted (owner).
- Digital cash: the identity of the person who creates cash units (creator).

Blind signature properties:

- A blind signature is secure if it can be proved that the identity of the owner is never revealed. The unconditional intraceability of the owner must be guaranteed even in the case of collusion. This will ensure that the owner retains his/her anonymity. This is known as the *blindness property*.
- For a blind signature to be secure it must also be proven that the blind signature cannot be forged. [Marte03]

General Cash Properties

Properties of Cash:

- Validity:
  - valid cash units are authenticated by an authority (e.g., the ECB)
  - validity can be verified (to a variable degree, using tools) for any cash unit
- Anonymity and intraceability – except for when used in a crime, e.g. in case of ransom money, money laundry
  - process intraceability: A cash unit does not reveal the payment processes it has been used in
  - anonymity of the user: It does not reveal the spender's / receiver's identity.
- Transaction:
  - No / low transaction costs
  - Immediate transaction processing

These properties shall be provided for digital cash, too.

DigiCash Application of Blind Signatures (1)

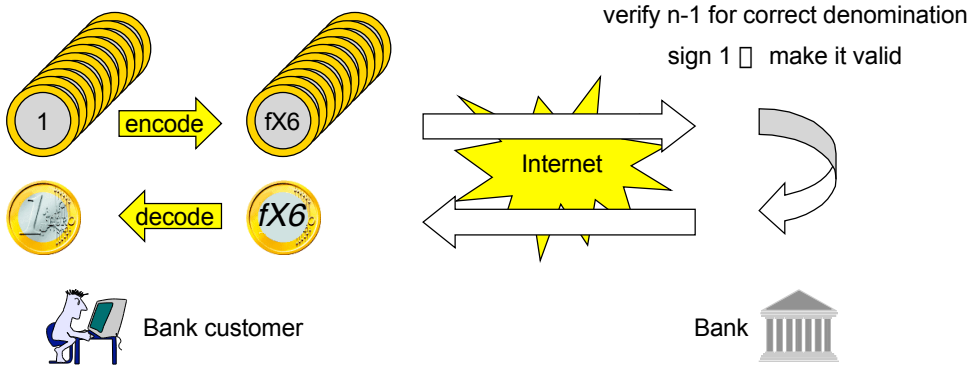
Application of blind signatures in DigiCash (ECash):

Blinding process example:

- The bank client (*information creator role*) wants to have 1 € as digital cash.
- He creates n (10.000.000) digital cash units of the value 1 € with random serial numbers. He encodes them and sends them to the bank. The bank (*authenticator role*) picks n-1 (9.999.999) at random, verifies the denomination and deletes them.
- One cash unit remains (Note: the bank does not know its serial number).
- Note: Checking the denomination reveals the serial number. It requires a decryption step by the user, thus the user knows which cash units are being verified.
- The probability is very low that the remaining cash unit is not a 1 €, but a 1.000.000.000 € cash unit (if the client tried to commit fraud).
- The bank signs this cash unit, and thus validates it to make it a *digital coin*. The coin is returned to the user who decodes it. The user has 1 € (very high probability) or 1.000.000.000 € (very low probability).

See figure on next slide.

DigiCash Application of Blind Signatures (2)



Authentication: Fair Blind Signature

Problem: The **intraceability** of anonymous electronic cash has problems: For example, criminals could obtain a ransom for a kidnapping or launder money without yielding a trace of identity.

Solution: **Unblinding the blind signature when needed**. Blind signatures that can reveal the identity of the holder of the signature are known as **fair blind signatures.**

Link-and-Recovery Fair Blind Signatures (obtain information about who spent the money)

- Involves three roles: cash user, signer (bank) and a third trusted entity (judge or certification authority).
- When needed, the cash signing protocol enables the trusted entity to reveal the sender of electronic cash (details omitted, see literature).

Trustee-Based Fair Blind Signatures (obtain information about user spending habits)

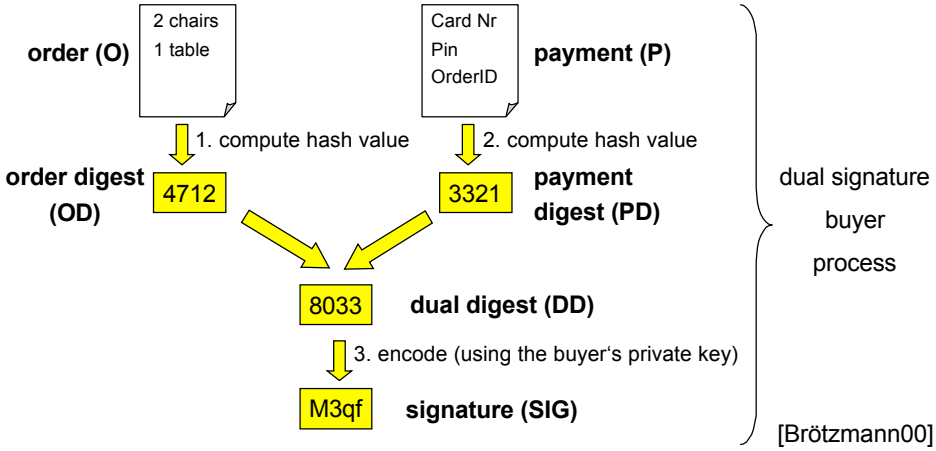
- The user provides *trustees* with information that allow the trustees to recognise the electronic notes of the user. Each of the trustees receives a part of this information that makes the electronic spending of the user traceable.
- When needed, the spending pattern of the user can be revealed by putting together the trustees' individual information. Problem: If trustees collude (get together), they can build up a user's spending profile.

[Marte03]

Authentication: Dual Signature (1)

Definition

A **dual signature** is used to verify whether data constitute different parts of a single logical unit. Example: Using a dual signature, one can verify whether ordering information (items, quantity) and payment information (card and pin number) belong to the same order.

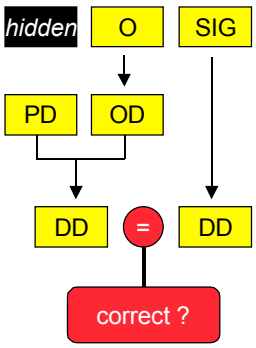


[Brötzmann00]

Authentication: Dual Signature (2)

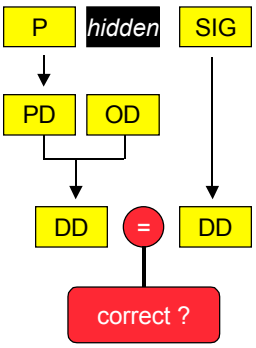
Seller Validation:

does not know payment information (P)



Bank Validation:

does not know order information (O)



[Brötzmann00]

Authentication: Certificates

Definition

A **certificate** is a verifiable statement made by a legal person / institution about circumstances (ex: driving license, master degree).

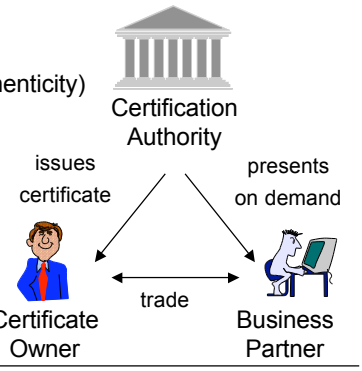
Digital Certificates are used in ECommerce for proving the affiliation of a *public key* to a *legal person*. They comply with the x.509 standard.

A certificate is used by legal persons in three roles:

- Certificate issuer (certification authority, CA)
- Certificate owner (legal person to be described)
- Certificate users (others verifying the owner's authenticity)

Certificates comprise the following information:

- Owner's name and public key
- Public hash function (algorithm) name
- Certificate serial number
- Start date and expiration date of certificate
- Certificate authority's name



Authentication: Certification Authority

Definition

Certification Authorities (CAs) ensure that a public key is affiliated to a legal person. CAs have different certification classes. Every class comprises costs, required documents for certification and liability of the CA.

Example: VeriSign (www.verisign.com) certification classes.

Class	Client (buyer) or server (seller)	Required information	Costs p.a.	CA Liabilty
1	Client	name, email	-	100 USD
2	Client	name, email, postal address, date of birth, SSN, employer	19,95 USD	5.000 USD
3	Client	...	290 USD first year, 75 USD following yrs	100.000 USD
4	Server	...	290 USD first year, 75 USD following yrs	100.000 USD

[Merz99]

Authentication: Public Key Infrastructure

Definition

A **Public Key Infrastructure** (PKI) is a system of Certificate Authorities (and other registration authorities) that verify and authenticate the validity of each party involved in an Internet transaction based on digital certificates. A PKI is also called a *trust hierarchy.* [WOp00].

PKIs are currently evolving and there is no single PKI nor even a single agreed-upon standard for setting up a PKI. Reliable PKIs are a necessary requirement for some forms of "pure" electronic commerce.

Two steps are necessary to build a successful PKI:

- The majority of legal persons (single persons, companies) register at a CA.
- The CAs must be certified.

Question

- Which organization certifies the CAs?

Answer

- CAs certify each other via cross-certification.

Electronic Payment (1)

Classification of payments by transaction volume [Merz99]:

- Zeropayments (0 €)
- Nanopayments (0.001 – 0.1 €)
- Micropayments (0.1 - 5 €)
- Medium Payments (5 – 1.000 €)
- Macropayments (> 1.000 €)

Different transaction volumes require different forms of payment. Example: Credit cards are not suitable for nano– and micropayments as transactions costs are higher than transaction volume.

Electronic Payment (2)

Mind the buyer‘s payment habits. Dominant forms of payment:

USA:

- Credit-cards
- Cheques

Germany:

- Credit-cards
- Direct debit (Lastschriftverfahren)
- Cash-on-delivery (Nachnahmesendung)

Asia

- Credit-cards

Implication: Global businesses must localize their forms of payment.

Payment Systems: Requirements

General Issues

- Security
- Scalability
- Reliability
- Usability

Electronic-Commerce related

- Must allow micropayments (low transaction costs)
- Payment channels (B2C, C2C, ...)
- Anonymity
- Immediate transactions (especially for soft goods, e.g. software licenses)

Note: Different delivery and payment models apply for hard good stores and soft good stores.

[Brötzmann00]

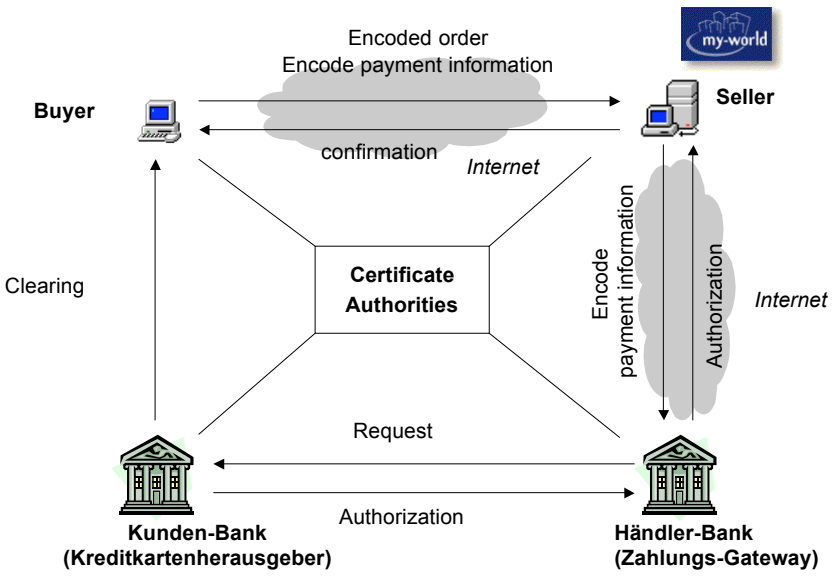
Payments: Macropayments

Transactions volume: > 1.000 €

Established business relationship between sellers and buyers. Business relationship is fixed by contracts. Therefore, payments are not as important as continuous business relationship.

Payments are not internet-based No internet based payment infrastructure.

Medium Payments: Secure Electronic Transactions: SET

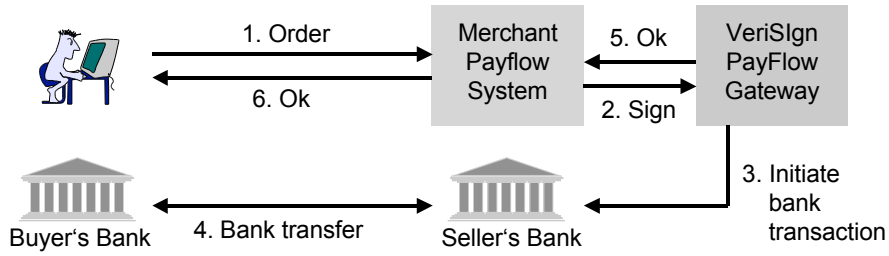


Medium Payments:: VeriSign PayFlow

Comprises different forms of payment (credit-card, direct debit, digital cash). Credit-card based transactions comply to the SET standard. Merchant installs the PayFlow software. Client data is sent encrypted to VeriSign for clearing.

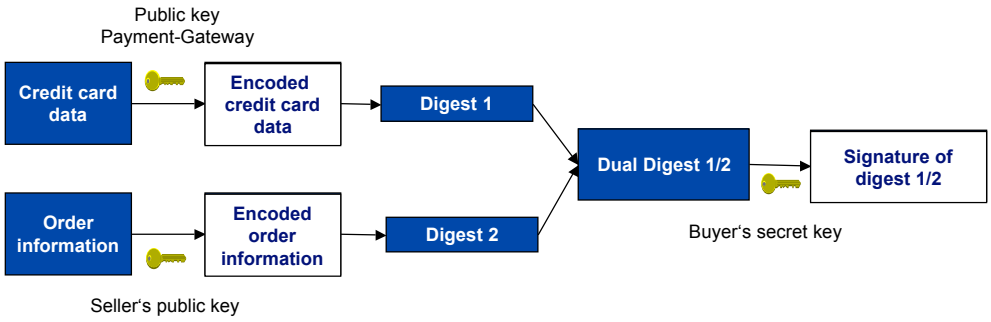
The buyer encrypts the credit-card information and sends it to the seller. The seller signs the information and sends it to CyberCash's payment gateway. The gateway verifies the buyer's credit-card number and initiates a bank transaction.

Formerly, the buyer had to install a "wallet" (software component) that encrypted the credit-card information before sending it to the merchant system. This has been abandoned. The credit-card information is now sent encrypted by standard SSL.



Electronic Commerce (WS-02/03) 5-51

Dual Signature and Encoding



Electronic Commerce (WS-02/03) 5-52

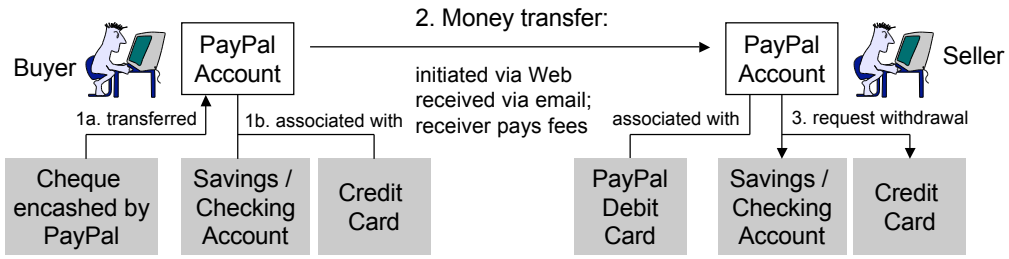
Medium Payments: PayPal

PayPal was introduced as a C2C email-based money transfer system, then extended to handle B2C financial transactions. Mainly used in US for online auction-related payments.

Buyer initiates a money transfer on PayPal's WebSite. PayPal debits his associated bank account / credit card, sends receiver an email notification and deposits on receiver's PayPal account.

Receiver always pays fees: 0,30 U$ + 2.2 % of payment amount.

Receiver can withdraw money directly via PayPal debit card (only B2C merchant, not in C2C model) or request money transfer to bank account / credit card.



Payments: Micropayments (1)

Transactions volume: 0.1 – 5 €

- Form of payment in traditional commerce: cash.
- Idea: Map **cash** to electronic commerce digital cash.
- Currently, there is no successful and widely adapted digital cash model.

Success factors for digital cash are:

- Offline usability (No bank needed / wanted for verification at every transaction)
- Anonymity (money spender stays anonymous, unless trying to double-spend digital cash)

Payments: Micropayments (2)

Current digital cash models:

Billing: Reduce costs by consolidating transaction volumes

- Phone + Code: Call a number. After a period of time a code is disclosed. Important: What percentage of the fee is taken by the telecommunication service provider (Germany: 50% by Deutsche Telekom!)
- Token-based and account-based billing systems (see following slides)

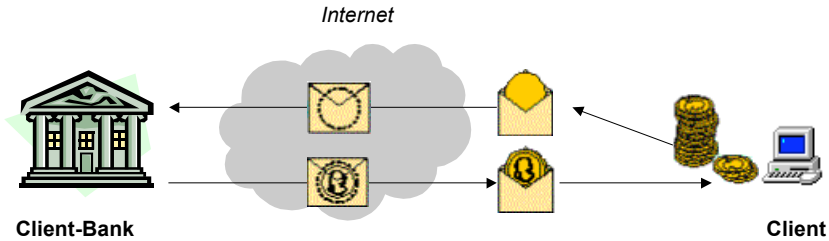
Money cards:

- White Cards (anonymous users), e.g., Mondex

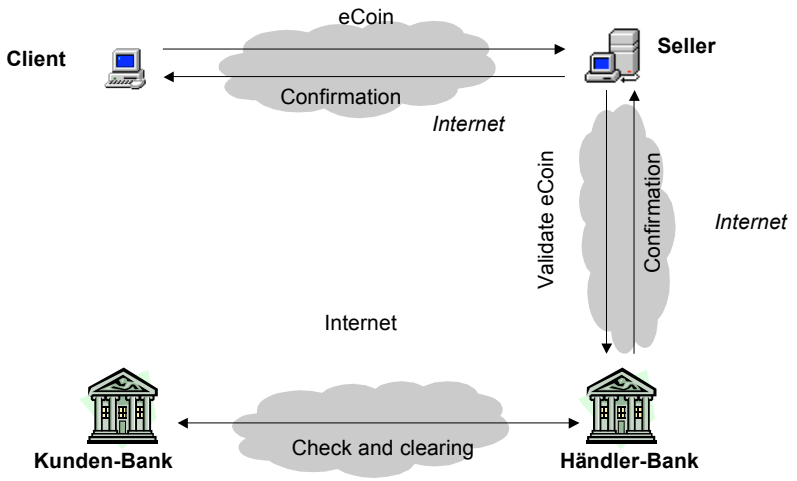
Digital cash (buyer is anonymous, double spending problem)

- eCash

Ecash with *blind signature*

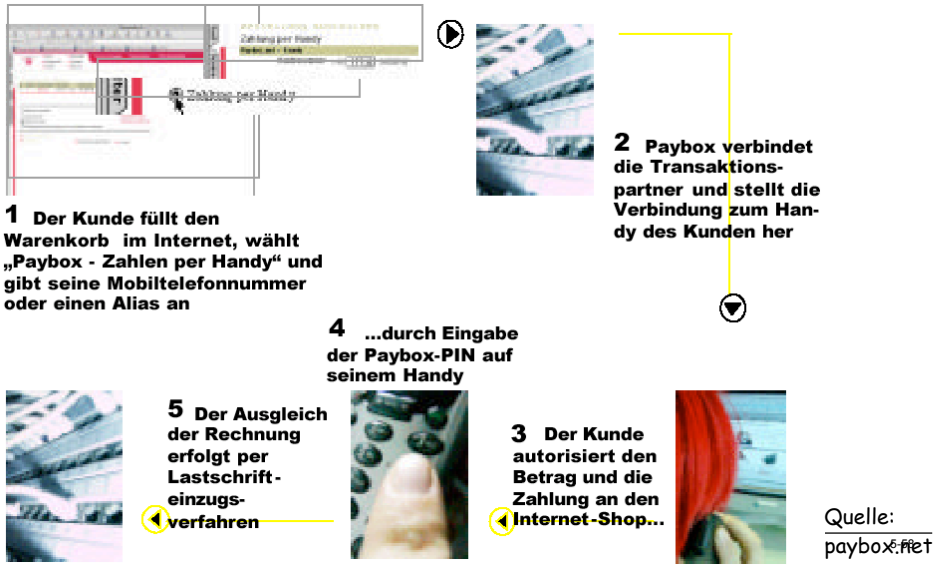


ecash



Electronic Commerce (WS-02/03) 5-57

New Developments: e.g. PayBox



Quelle: paybox.net

EI 5-58

Example Application: Pay the Taxi Driver

Quelle: paybox.net

New Developments: e.g. Web-Coupons

Web-Coupons are not linked to convertible currencies (U$, €) issued by countries but are an **artificial currency** created by company consortia to increase customer loyalty.

Earn coupons:

- Buy specific goods or services
- Provide information about yourself
- Provide information / services to others

Spend coupons:

- Discounts on sales and services of affiliates
- Privileged access to goods and services

Non-Internet Examples:

- Lufthansa Miles & More (many global partners)
- American Express Bonus "Miles" (many global partners)
- Payback Points (Germany only)

Internet-Only Examples:

- Former Web-Miles (dead)

Summary: Web-Coupons did not succeed (yet).

Payments: Summary

Summary

- Currently, no models for nanopayments exist.
- No successful model for micropayments, except for money cards.
- No standard for micropayments, standard models starting at medium payments only.
- Medium payments: Best supported model (online credit-card payments, email-based money transfer)
- Most payment systems vendors are gone bankrupt.
- Micropayment models can also be used for medium payments.