Secure Electronic Commerce

Michael Waidner

IBM Zurich Research Laboratory
CH8803 Rüschlikon, Switzerland
Tel +41 1 724 8220 / Fax +41 1 710 3608
<wmi@zurich.ibm.com> / <http://www.zurich.ibm.com>

European Research Seminar in Advanced Distributed Systems
Zinal, March 17-21, 1997

IBM

Outline

Outline

- **Introduction**
  - Secure Electronic Commerce, Multi-party Security
- **Secure Devices**
  - Smartcards, Electronic Wallets
- **Cryptography**
  - Types of crypto-systems, Discrete Logarithm, Diffie-Hellman, Schnorr Signatures
- **Public-Key Infrastructure**
  - Third Party Services, Certification, Signatures
- **Secure End-to-End Transport**
  - Link and End-to-End Security, Anonymity
- **Notary Services**
  - Logical Time Stamps, Fair Exchange

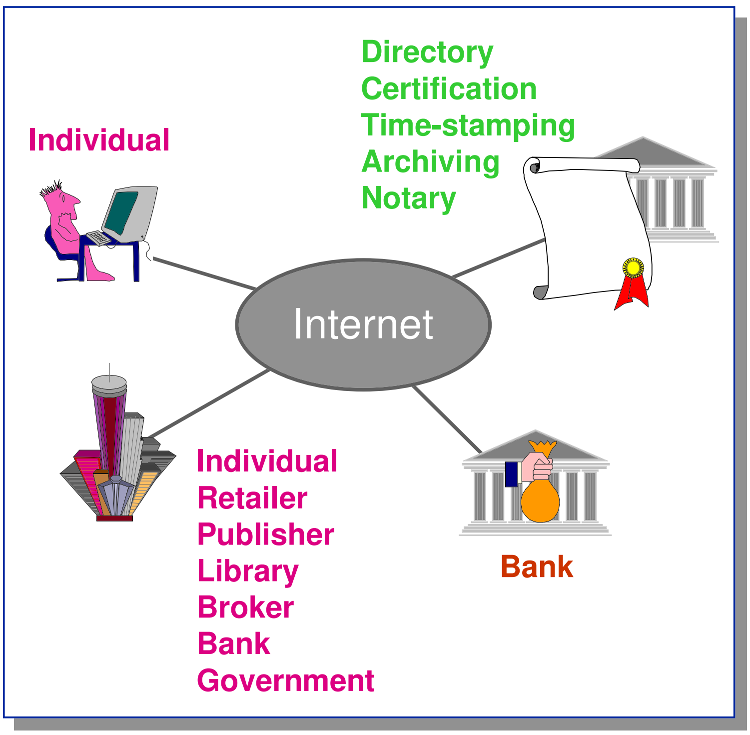
IBM

Outline

- **Payment Systems**
  - Requirements
  - Simulating the Paper World: SET
  - Smartcard-based Electronic Purses
  - Untraceable Cash: ecash, Blinded Schnorr Signatures, CAFE
  - Micropayments: CAFE, Billing Server, MiniPay
- **Secure Transfer of Digital Originals**
  - Superdistribution: Cryptolopes
  - Fingerprinting
- **Frameworks for Secure Electronic Commerce**
  - SEMPER
- **Summary and Outlook**

© IBM 2/15/97 <wmi@zurich>

Electronic Marketplace “Internet”



- **Secure Email**
- **Secure Document Exchange**
- **Mail Order Retailing**
- **Electronic Publishing**
- **Ticketing**
- **Subscriptions**
- **Information Brokerage**
- **Contract Signing**
- **Secure Auctioning**
- **...**

Introduction

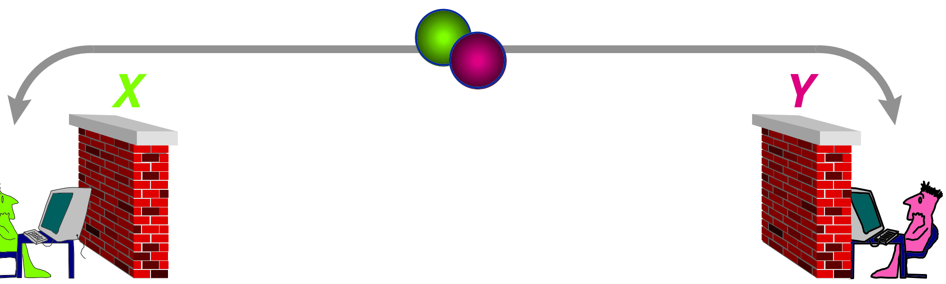
Secure Electronic Commerce

- **More than Secure Communication**
  - **Multi-party problems:** payments, notarized contract signing, auctioning, copy protection, ...
  - **Multi-party security:** limited trust in others
  - **Processes:** Systems must be securely linked, e.g., *contract* with *payment* with *delivery*
- **Technical Security + Legal Certainty**
  - **Non-repudiation:** Dispute handling; Digital signatures; Registration & certification; Secure hardware; Unified User Interfaces; ...
  - **Privacy protection:** Confidentiality, Anonymity
  - **Applicable law in cross-border commerce**

IBM

Introduction

Abstract View

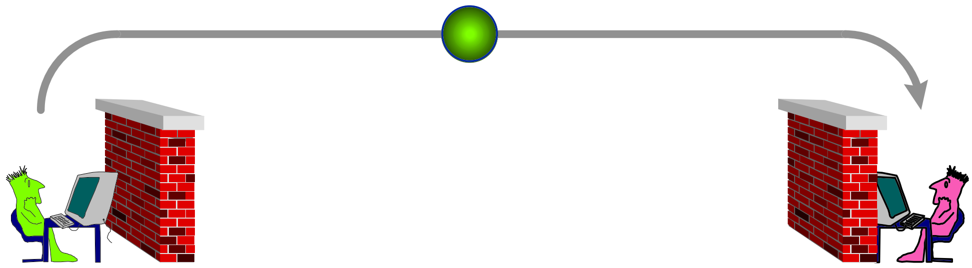


- **Secure Transfer / Secure Fair Exchange**
  - from ***Pseudonym X*** to ***Pseudonym Y***
  - of **"business documents,"** which can be
    - **information** (documents, goods, ...)
    - **money**
    - copy-protected **original**

    by means of some **protocol**
  - at a **certain time**

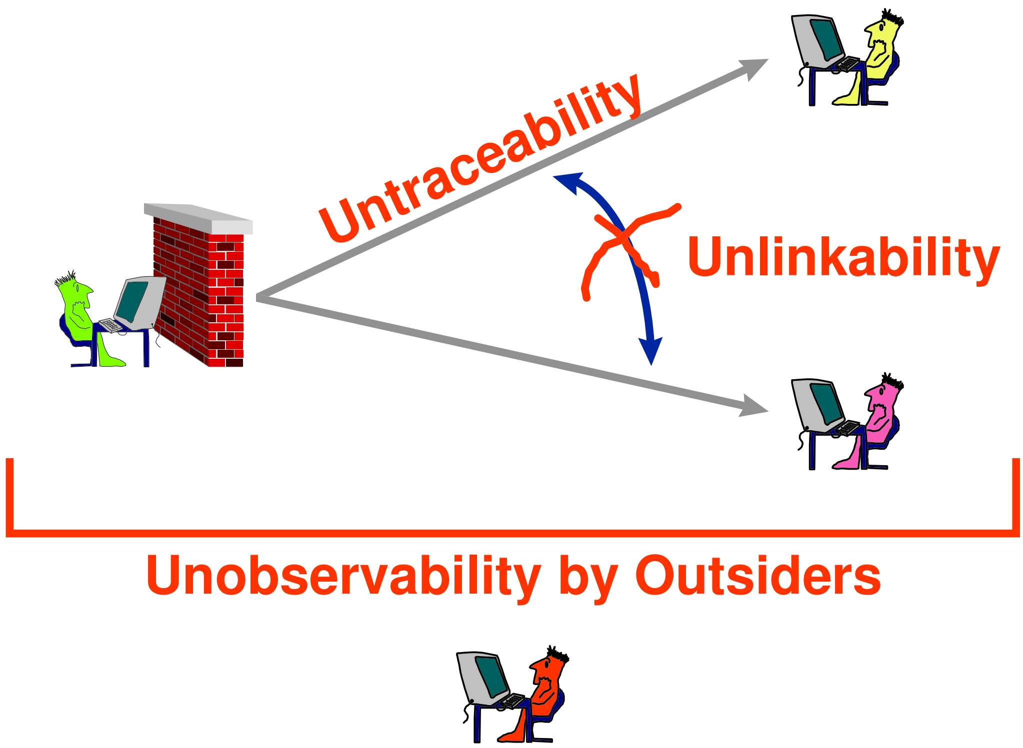
IBM

Security Properties



- **Availability:** ***guaranteed progress***
- **Integrity:** ***what happens is correct***
- **Confidentiality:**
  - Message confidentiality
  - Anonymity

Anonymity



Introduction

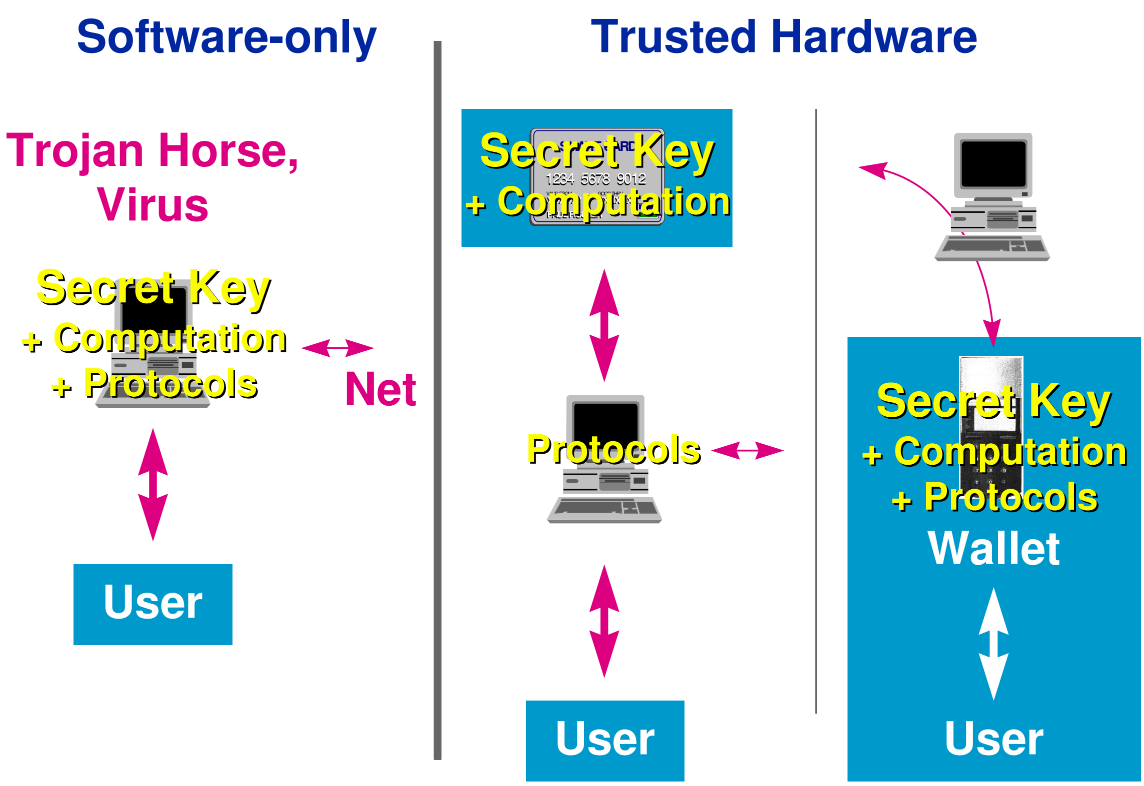
Multi-Party Security

- **Multiple users with individual requirements**
- **Formulate explicit trust requirements**
  - Who must be trusted?
  - What is the worst case behaviour of an attacker?
  - User must never depend on opponent!
- **Minimize trust requirements**
  - Aim at accountable, verifiable third parties
  - User must be able to choose among entities
  - Distribute trust among many entities
- **Exception handling and dispute resolution**
  - Explicit rules, covering all cases

IBM

Secure Devices

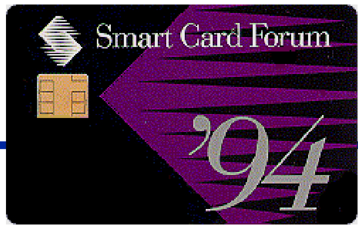
Need for Secure User Devices



IBM

Secure Devices

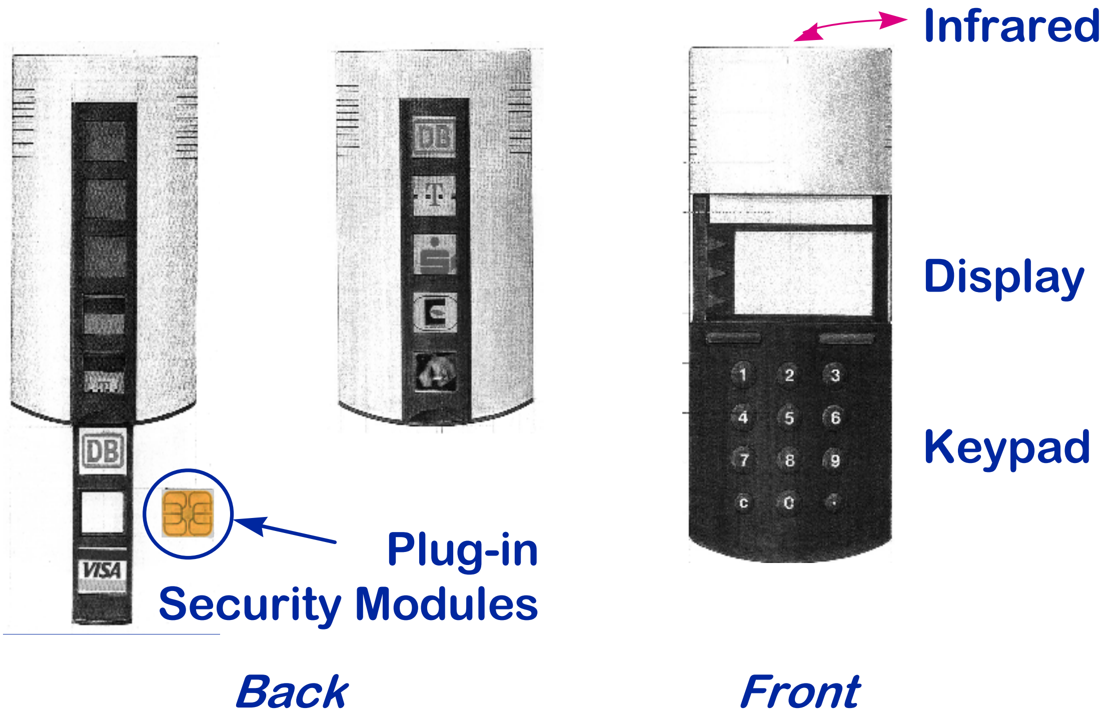
Smartcards



- **Very small one-chip computer**
  - 8-bit Microcomputer, < 5MHz
  - < 512 RAM, 8K EEPROM, 16K ROM
  - ~ 9.6 Kbps IO
  - 0.3 s for 512 RSA signature with co-processor
  - $1 - $10
- **Tamper resistance**
  - **never tamper proof**
  - insider attacks: always possible
  - reading ROM easy
  - reading EEPROM relatively difficult, not impossible

Secure Devices

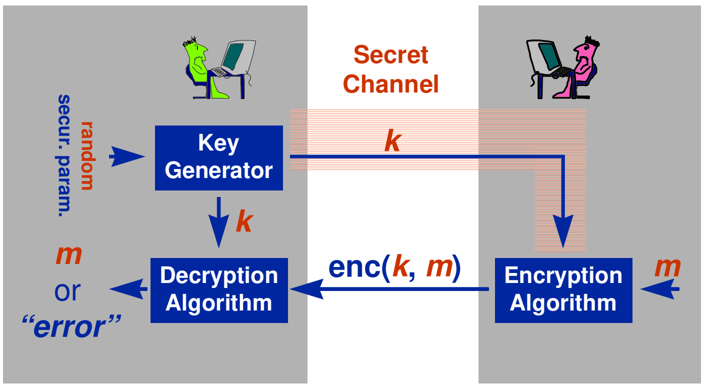
Electronic Wallet



Cryptography

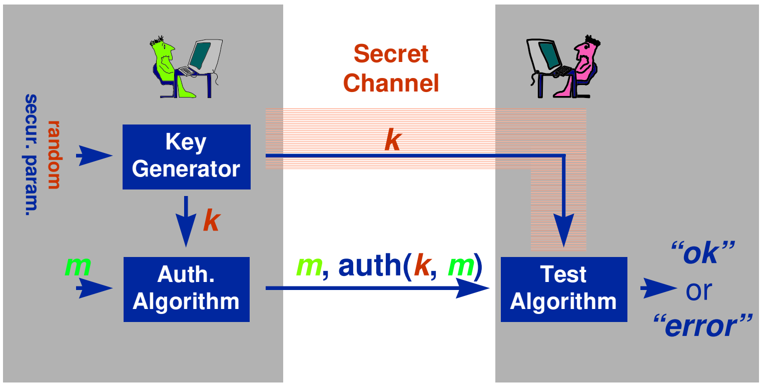
- **Concelation**
  - **Steganography:**
    - *Hide message, e.g., in a picture*
  - **Encryption:**
    - *encr_algorithm: (plaintext, key) $\rightarrow$ ciphertext*
- **Authentication**
  - **Identification, Entity Authentication:**
    - *Who is on the other end of this line, currently?*
  - **Message Authentication:**
    - *Who created this message?*
  - **Digital Signature:**
    - *Convince a third party about who created this message.*

Shared-key Encryption Scheme



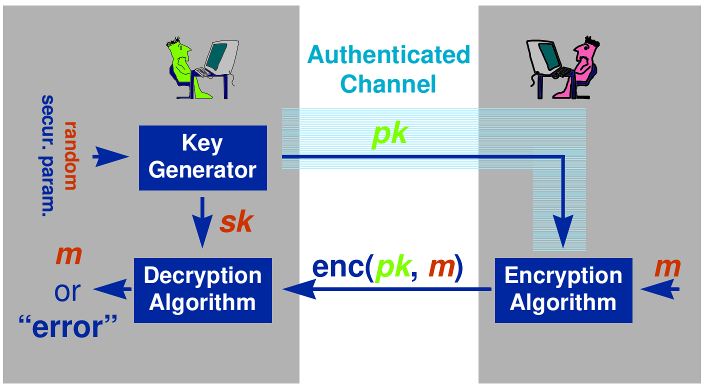
- **One-time pad, DES, IDEA, LFSR ciphers**
- **Fast (MB-GB/s), but typically not verified**
- **Key sizes of 56 - 128 bit**
- **Key distribution: *secret channel* needs key distribution center or public-key scheme**

Shared-key Authentication Scheme



- Message Authentication Codes
- Specific constructions, or based on block ciphers or keyed hash functions
- *Limitation:* Third party cannot check authenticity

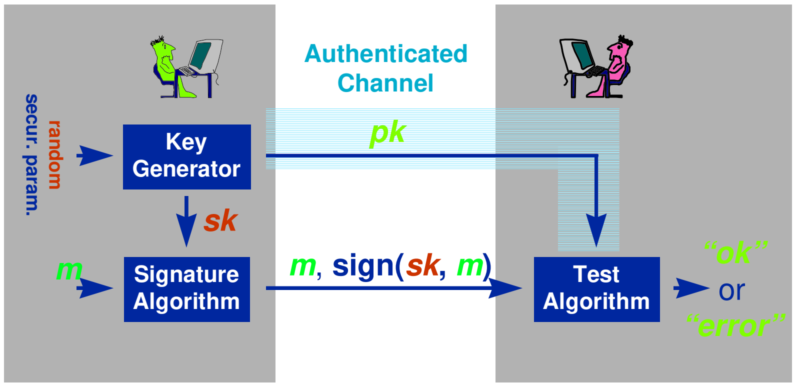
Public-key Encryption Scheme



- RSA, Diffie-Hellman / El Gamal
- ~ 10-times slower than symmetric schemes
- “Key size” of 512 - 2048 bit for RSA
- Typically used to exchange a shared key for a symmetric scheme

Cryptography

Digital Signature Scheme



- **RSA, El Gamal, Schnorr / DSS, GMR**
- **Asymmetry allows third party to check authenticity**

IBM

Cryptography

Discrete Logarithm

- Cyclic Group

$$G = \langle g \rangle = \{g^1, \ldots, g^q\}$$

- For each $a \in G$ there is a unique $i \in \{1, \ldots, q\}$ such that

$$a = g^i$$

  called ***Discrete Logarithm*** (***DLog***) of a (base g)

- It is common *belief* that there is no efficient algorithm that computes the DLog in groups of large prime order.
- Other popular assumption: ***Factoring in N***

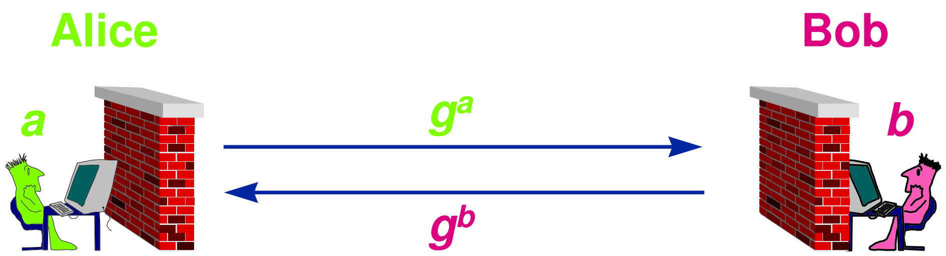
IBM

Abstract DLog Assumption

- **Generator of a Family of Groups: *Gen***
  - input: $\mathbf{1}^k$
  - output: cyclic group G with generator g, and algorithms polynomial in $\mathbf{1}^k$ for computing group operation and picking a random element
- **DLog Assumption for this Family**
  - For each polynomial time algorithm $\boldsymbol{A}$, constant $\boldsymbol{c}$, and sufficiently large $\boldsymbol{k}$:

$$\mathrm{Prob}(\boldsymbol{g}^{\boldsymbol{i}} = \boldsymbol{a} \;/\; (\boldsymbol{G}, \boldsymbol{g}, \ldots) \leftarrow \boldsymbol{Gen}(1^k);\; \boldsymbol{a} \in \boldsymbol{G} \text{ randomly chosen};\; \boldsymbol{i} \leftarrow \boldsymbol{A}(\boldsymbol{G}, \boldsymbol{g}, 1^k, a)) < 1/\boldsymbol{k}^c$$

Diffie-Hellman Key Exchange



Both can compute shared secret

$$g^{ab} = (g^a)^b = (g^b)^a$$

- **Diffie-Hellman Assumption**
- **NB: Diffie-Hellman is *not* authenticated**
- **El Gamal Encryption for $pk = g^b$**
  - pick a, g^a, $k := (g^b)^a$, use k to encrypt m
  - ciphertext $:= (g^a, k^*m)$ or $:= (g^a, \mathrm{DES}_k(m))$

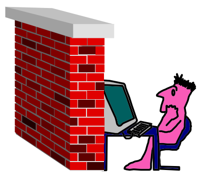
Schnorr Signatures

- **Specific group**
  - $p = 2q + 1$ $\quad p, q$ prime
    $\Rightarrow$ $\boldsymbol{Z_p^*}$ is group with p-1 = 2q elements
  - pick subgroup with q elements,
    $\boldsymbol{G} = <\boldsymbol{g}>$

- **Collision resistant hash function *H***
  - Infeasible to find $\boldsymbol{x}$ such that $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{H}(\mathbf{y})$ for randomly chosen but unknown $\boldsymbol{y}$
  - Variable length input, fixed length output
  - Infeasible to find $\boldsymbol{x} \neq \boldsymbol{y}$ such that $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{y})$
  - $\boldsymbol{H}(\boldsymbol{x})$ simulates a "random oracle"

Cryptography

Schnorr Identification Scheme

Signer Alice		**Verifier Bob**
		
$(sk, pk) = (x,\ h = g^x)$		**Knows** $pk = h$
Random key pair $(\hat{x}, \hat{h} = g^{\hat{x}})$	$\xrightarrow{\hat{h}}$	**Random challenge** c
	$\xleftarrow{c}$	
Compute $r = \hat{x} + xc$	$\xrightarrow{r}$	**Check** $g^r = \hat{h}h^c$

Schnorr Identification Scheme

- **Does not reveal information about *x***
  - not proven to be zero-knowledge
  - known: $g^{\hat{x}}$ and $r = \hat{x} + xc$ for random $\hat{x}$

- **Is a proof of knowledge:**
  - assume Alice can convince Bob
  - $\Rightarrow$ Alice knows an $\hat{h}$ such that she can answer at least two challenges, c_0, c_1

$$\begin{array}{l} g^{r_0} = \hat{h} g^{x c_0} \\ g^{r_1} = \hat{h} g^{x c_1} \end{array} \Rightarrow g^{(r_0 - r_1)} = g^{x(c_0 - c_1)} \Rightarrow x = \frac{r_0 - r_1}{c_0 - c_1}$$

Schnorr Signature Scheme

Signer Alice

$(sk, pk) = (x, \ h = g^x)$

Random key pair

$(\hat{x}, \hat{h} = g^{\hat{x}})$

Compute challenge

$c = H(m, h, \hat{h})$

Compute response

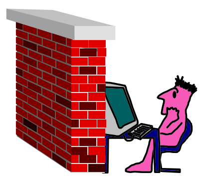
$r = \hat{x} + xc$

Shamir's *H*()-Trick

Signature

$m, \hat{h}, r$ $\longrightarrow$

Verifier Bob



Knows $pk = h$

Compute

$c = H(m, h, \hat{h})$

Check $g^r = \hat{h} h^c$

Cryptography

IBM

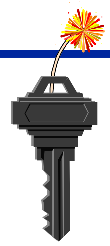
Cryptographic Strength

- **Information theoretic security**
  - only few systems: one-time pad, authentication codes, (verifiable) secret sharing
  - usually not relevant for practical life
- **Complexity theoretic security**
  - Violating a certain property is *polynomially equivalent* to violating a certain *assumption* about the *asymptotic complexity* of, e.g., *DLog* or *Factoring in N*
  - Concrete complexity, concrete security parameters?
- **Practical security**
  - Not broken so far ...

Cryptography

IBM

Export Restrictions



- **USA et al. consider crypto as**

  military technology
- **Authentication tools: *free***
- **Confidentiality tools:**
  - 40bit symmetric systems: *free* (but insecure)
  - 56bit DES, 512bit RSA for key exchange: *free if manufacturer supports key recovery in 2 years*
  - beyond that: *banking applications only*
- **Discussions are going on**
  - serious economic problem for US industry
  - key recovery initiative supported by major companies (e.g., IBM)
  - ... but also criticized by other major companies

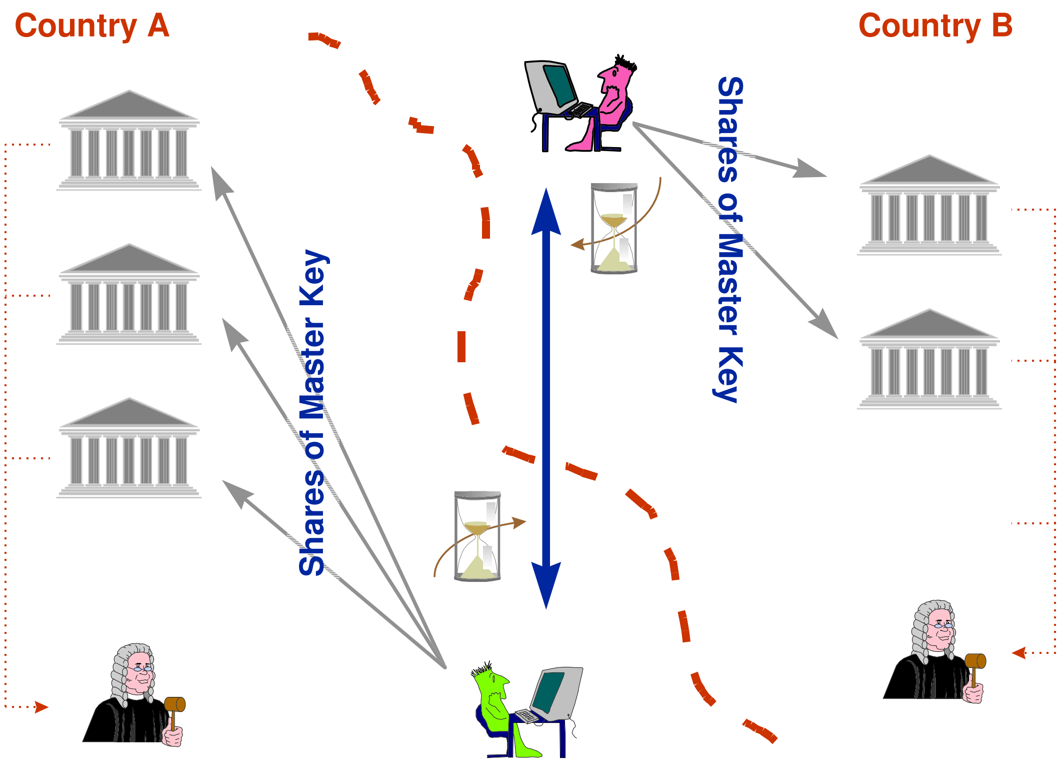
Cryptography

Application Restrictions

- **France et al. prohibit cryptography internally**
- **Why?**
  - Criminal investigations
- **Technical “compromise”**
  - Key escrow, key recovery
- **Obvious limitations and risks**
  - Strong cryptography is available *everywhere*
  - Steganography *hides* presence of ciphers, i.e., regulations are not enforceable
  - Political systems might change: who would like to live in a totalitarian system with key escrow?

Cryptography

Principle of Key Recovery

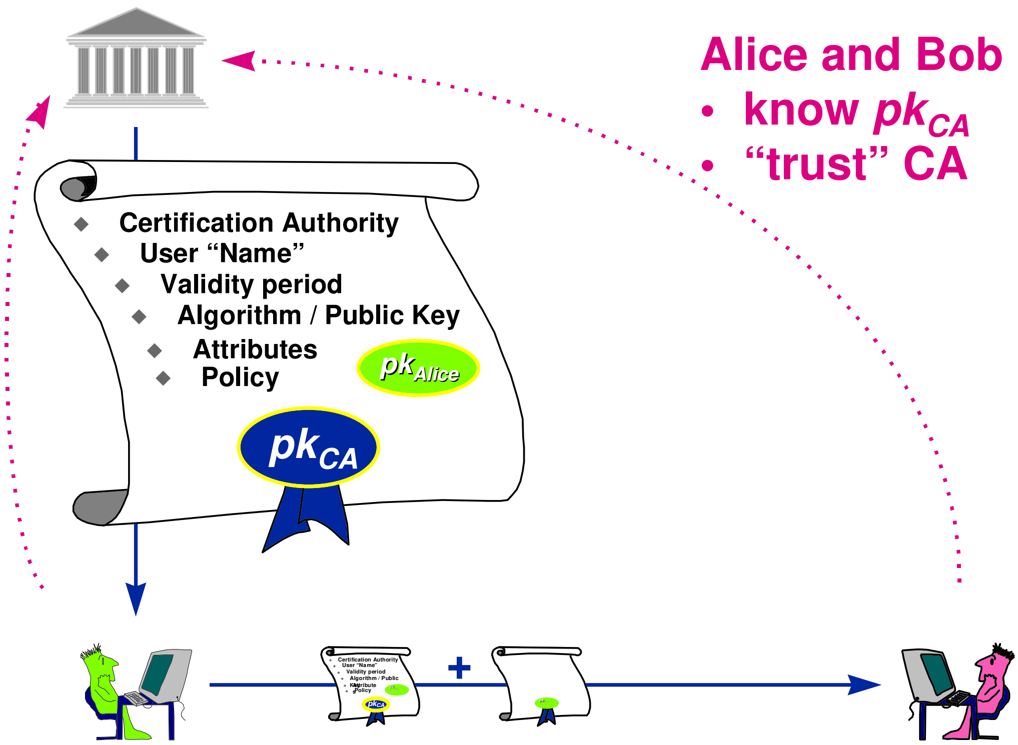


Third Party Services

- **Public Key Infrastructure**
  - Registration / Certification / Directory Services
- **General Certification Services**
  - Credentials (Attributes), Document Templates, Software
- **Notary Services**
  - Key Escrowing / Key Recovery
  - Time Stamping of Documents
  - Trusted Intermediary, e.g., for “fair exchanges”
  - Dependable Archiving of Documents
- **Arbitration Services**
  - Courts, arbiters

Public-Key Infrastructure

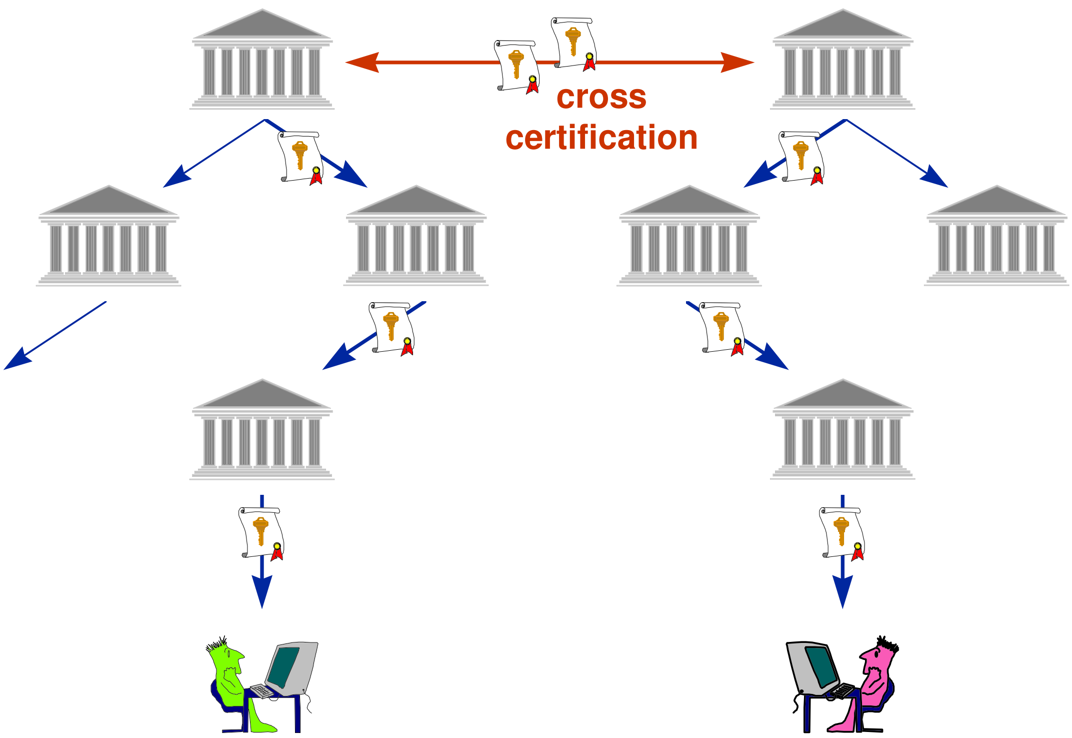
Public-key Certification



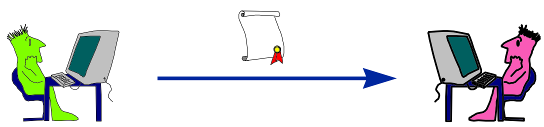
Technical Problems and Solutions

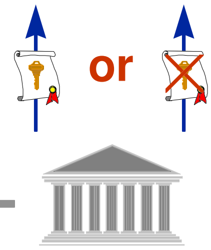
- **Semantics of "Name"**
  - Personal Identifier
  - Pseudonym, *maybe* linked to another pseudonym, *maybe* identifying a person
  - Public key
- **Scalability**
  - Hierarchical Certification, Cross Certification
- **Certificate Revocation**
  - Certificate Revocation Lists (CRL)
  - Revocation Certificates
  - No revocation, but short periods of validity
- **Root Key Compromise**
  - Short periods of validity
  - Backup key pre-distributed

Certification Hierarchies



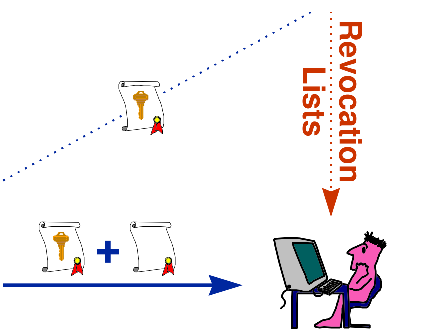
Certificate Revocation

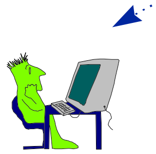




Pull Model with
Revocation Certificates

Push Model with
Certificate Revocation Lists





Legal Problems

- **Legal Semantics of a Digital Signature?**
  - Process of ordinary signing ...
    - attracts signer's attention
    - completes a document
    - declares signer's will
    - identifies the signer (not always required)
  - Formal requirements
    - *none*, court is free to accept or reject
    - signed by hand
    - signed by witnesses
    - notarized (i.e., signed, stamped and recorded)
    - notarized with specific requirements regarding proofs of identity

Implications for System Design

- Certificate per se *cannot* bind a user
- Ideal registration:
  - CA and user sign a contract
    - Certification policy
    - Accepted liabilities
    - Contents of certificate

Did supposed signer accept liability for such signatures?

- User interface
  - Standard GUIs
  - Standard presentation SW
  - "Point of no return"

Was supposed signer aware of signed contents?

- Secure key handling and signature generation
  - secrets generated by user
  - secrets never leave trusted user device (e.g., electronic wallet)

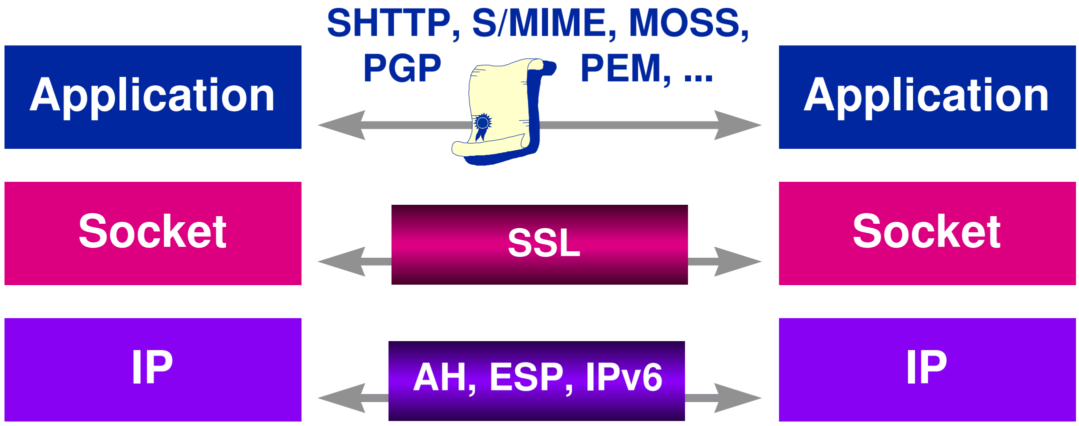
Was supposed signer the creator of a signature?

Verisign's Registration Classification

- **Class 1**
  - Unambiguous name, e-mail address
  - Verified in local repository
- **Class 2**
  - Real name, address, locale, e-mail
  - Personal information
  - Verified with 3rd party proofing source
- **Class 3**
  - Real name, address, locale, e-mail
  - Personal information
  - Verified with 3rd party proofing source & notarized docs
- **Class 4**
  - TBD

Secure End-to-End

Secure End-to-End Transport

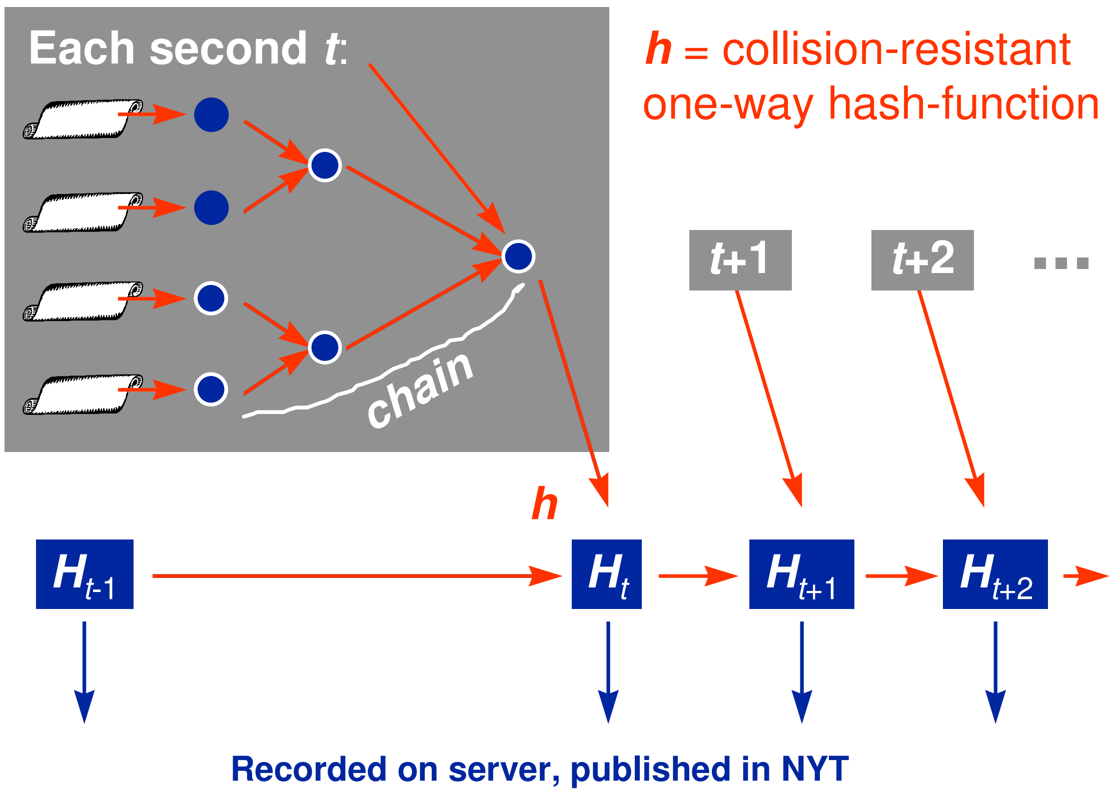


- **Cryptography**
  - Shared/Public Key Encryption (DES, RSA, etc.)
  - Stream/Document Authentication by MAC (keyed MD5, etc.) or Digital Signature (RSA, DSS, etc.)
  - Restrictions: National Security, Law Enforcement

IBM

Notary Services

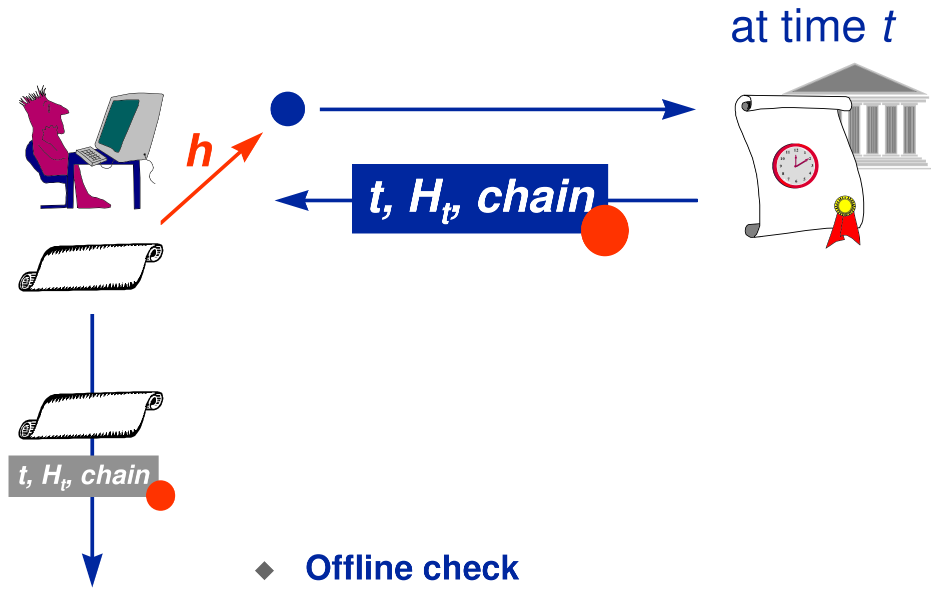
Logical Time Stamps

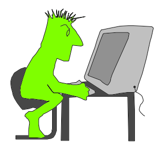


IBM

Notary Services

Logical Time Stamps





- Offline check
- Check H_t against copy at server, copy in NYT
- Check H_t against relation with other stamped documents

Notary Services

Logical Time Stamps

+ + +

- **Provides real time**
- **Minimal trust in time stamping server**
  - Server cannot change order
  - Network of servers can distribute trust
- **Only hash of documents are stamped**
- **Efficient**
  - log(#docs) h/s

– – –

- **Online service**
  - Server potential bottleneck
  - *but:* Network of time-stamping servers possible
- **“Eternity” problem**

Notary Services

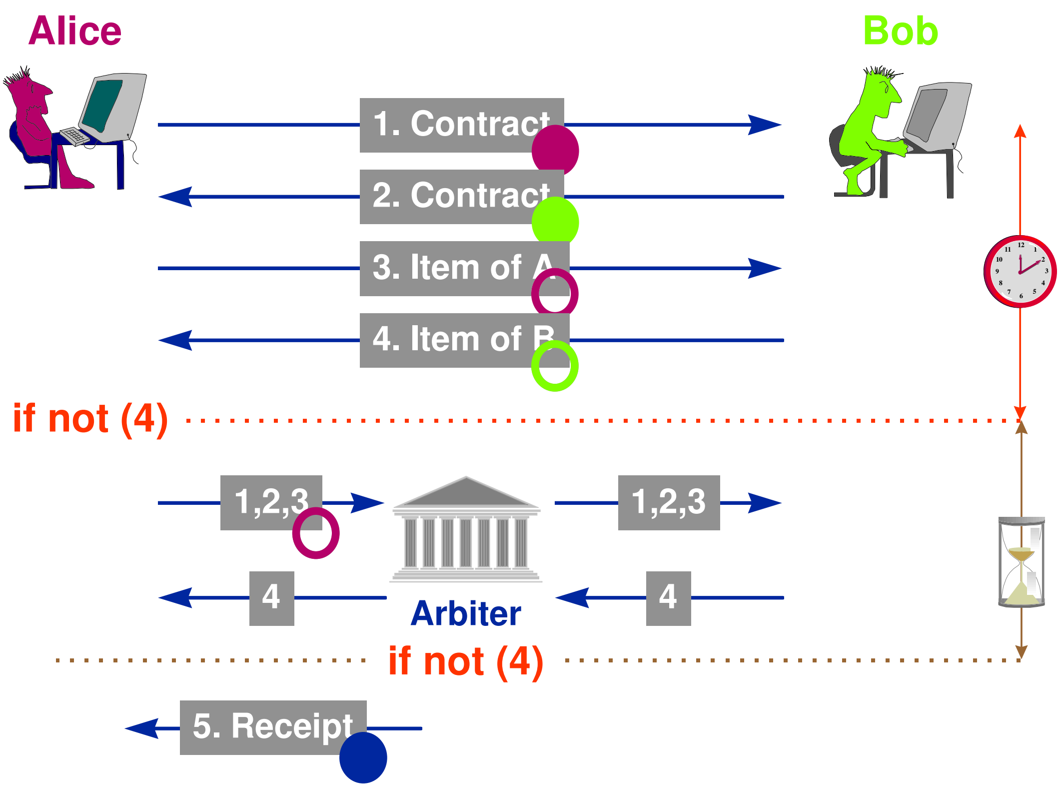
Fairness

- **Fairness**
  - Non-repudiation of
    - submission / delivery / receipt
  - Payment for
    - receipt, online goods
  - Contract signing
- **Objectives**
  - Minimize use of third party
  - Minimize use of expensive computations
  - Generic protocols that are independent of the items to be exchanged

© IBM 2/15/97 <wmi@zurich>

Notary Services

Optimistic 2-Party Fair Exchange



Optimistic 2-Party Fair Exchange

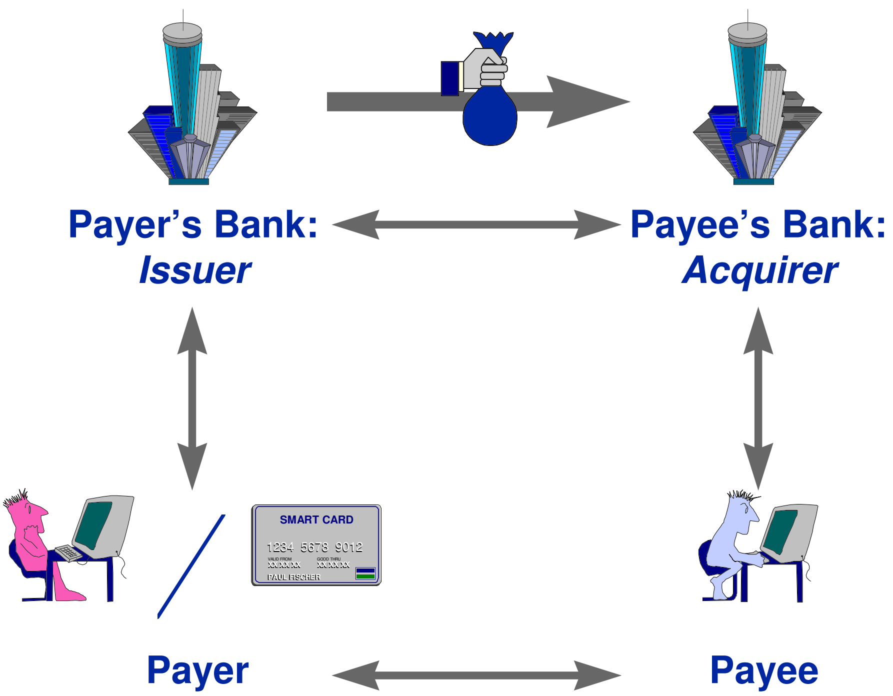
+ + +

- **Efficient in correct case**
  - early termination
  - 4 messages
  - 2 signatures
  - no 3rd party involved
- **Generic approach**
  - Non-repudiation of origin and receipt
  - Contract signing
  - Payment for receipt
  - Fair purchase

– – –

- **Weak fairness for Alice**
  - “Receipt by Arbiter” not always sufficient
- **Reliable communication with arbiter**
- **Late final termination**

What is a Payment System?



What will be Used in the Digital World?

- **The same types as in the paper world!**
  - Cash, for small and anonymous payments
  - Cheques, credit cards
  - Money transfer orders (push, pull)
  - Payment-like systems: vouchers, coupons
- **Same metaphor, i.e.,**
  - same "business model," same "look-and-feel"
  - at least as cost-effective
  - at least as secure, privacy protecting
  - ... but very different implementation

Payment Integrity

- **Integrity for *payer, payee, payment system***
  - **Nothing happens without authorization**
    - Nobody gives (sometimes also: receives) money without an explicit agreement, stating all necessary payment details
  - **Nothing happens without generating sufficient pieces of evidence**
    - If *X receives* money, it can *prove* this fact to "court"
    - If *X* has *not* given money, nobody can convince a "court" of the contrary
    - Sometimes: If *X* gives money, it gets a *receipt*
  - **The "rules" and technical procedures for "dispute handling" are part of the system!**

Payment Privacy

- **Privacy of payer and payee**
  - **Payment confidentiality:**
    - Payment details
      - payer, payee, account numbers, amounts, date and time, payment subject, etc.

    must not become known to outsiders
  - **Payment anonymity:**
    - **Payer anonymity:** Only a pseudonym is known to the payee, no identity (*cash*)
    - **Payer unlinkability:** Payee cannot link any two payments of the same payer (*cash*)
    - **Payer untraceability:** Payment system cannot trace payments back to the payer (*coins*)

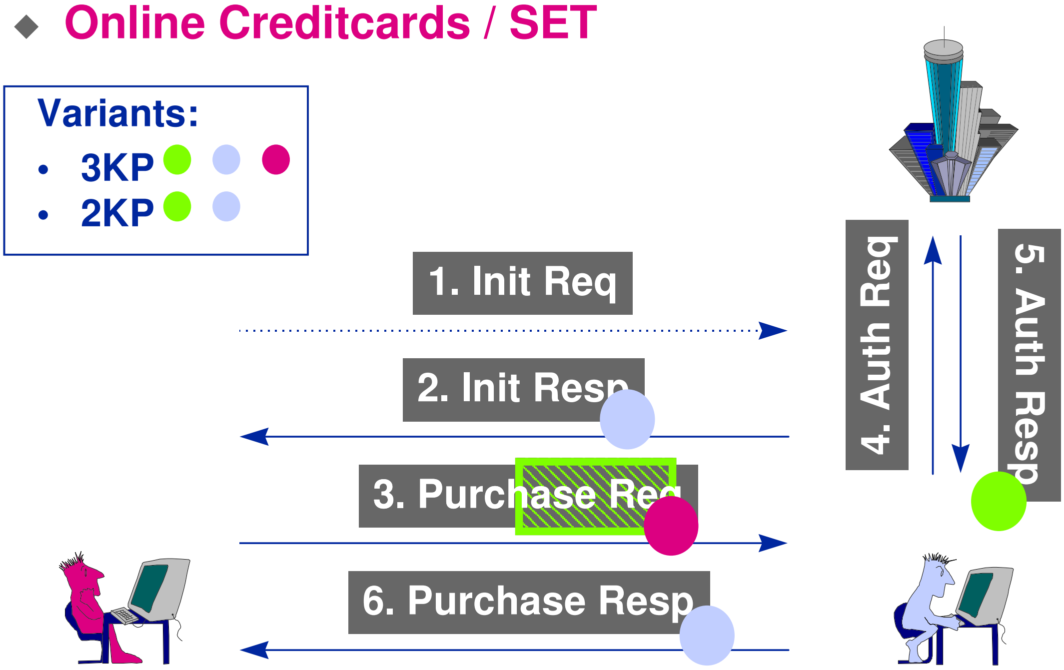
IBM

© IBM 2/15/97 <wmi@zurich>

"Money"

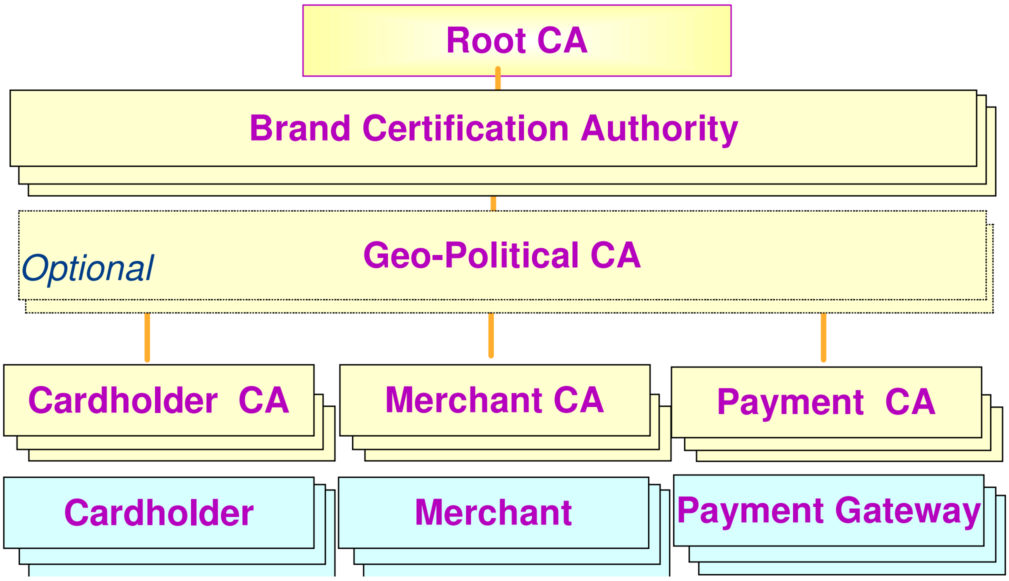
Basic Approach: Simulate Paper World

- **Online Creditcards / SET**



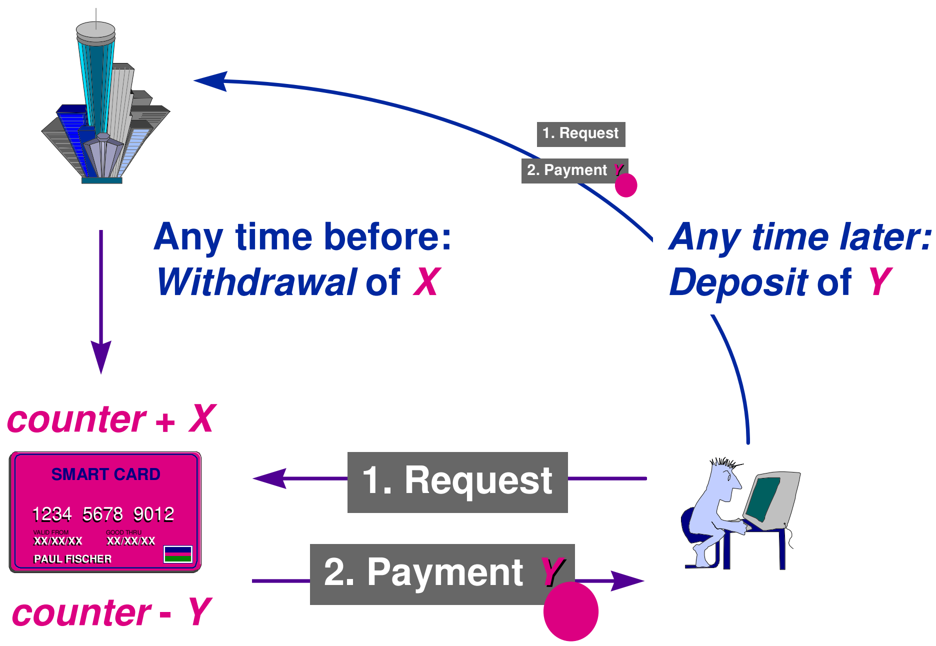
IBM

SET CA Hierarchy



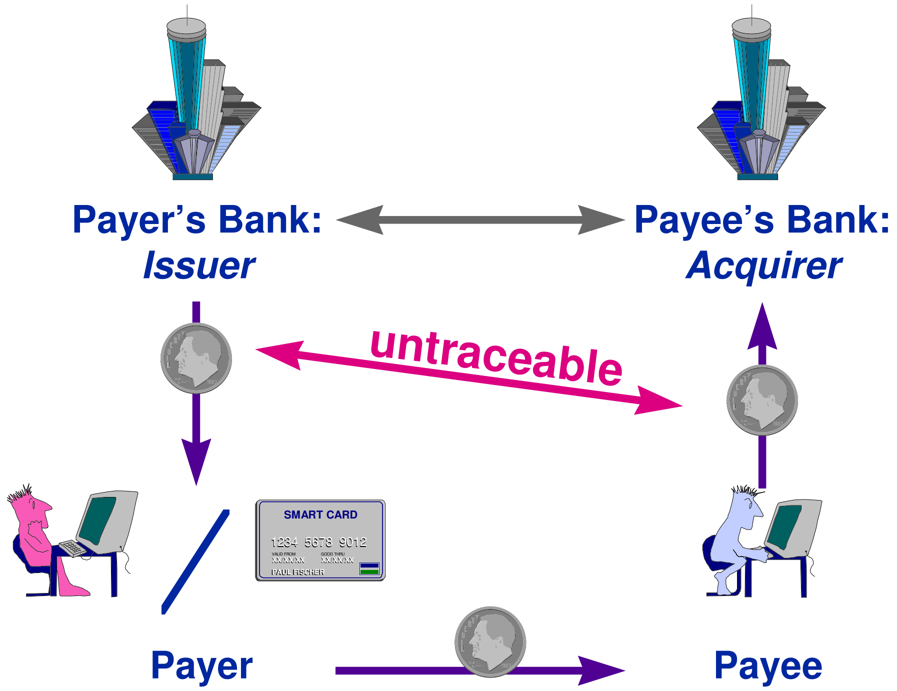
Double-spending Prevention

- **Signature-based Offline Purse** (e.g., CLIP)



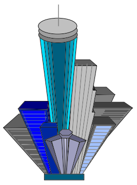
“Blind Signatures” for Unlinkability

- Online: ecash, offline: CAFE



ecash

- **Blinded RSA Signatures**
  - One-way function f
  - RSA pair (SK, PK) chosen by issuer, defines *one* denomination
- **Coins: ($coin$, $f(coin)^{SK}$)**
- **Withdrawal step**



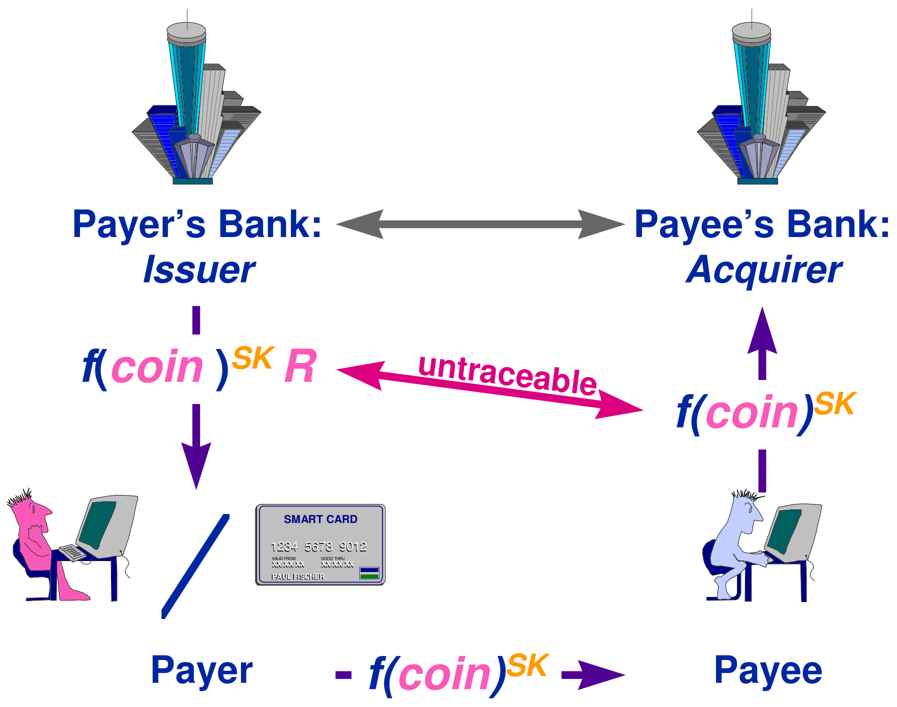
$f(coin)\, R^{PK}$ ←

→ $(f(coin)\, R^{PK})^{SK}$

$= (f(coin))^{SK}\, R \xrightarrow{\ast R^{-1}} f(coin)^{SK}$

Untraceable Cash

ecash



Untraceable Cash

Blinded Schnorr Signatures

$$(sk, pk) = (x, \quad h = g^x)$$

Message m $\quad \xrightarrow{(m,z)} \quad$ **Blinding factors**

Signature $z = m^x$

s, t random mod q, $s \neq 0$

$m' = m^s g^t$,

$z' = (m'^x =)\ z^s h^t$

Random versions $\quad \xrightarrow{(\hat{h},\hat{z})} \quad$ **Random values** u, v

$(\hat{x}, \quad \hat{h} = g^{\hat{x}}, \quad \hat{z} = m^{\hat{x}})$

$h'' = \hat{h}^u g^v, \quad z'' = (\hat{z}^s \hat{h}^t)^u m'^v$

$\xleftarrow{c}$ **Computes challenge**

$c' = H(m', m'', z', z'', h'')$

$c = c' u^{-1}$

Computes $\quad \xrightarrow{r} \quad$ **Checks**

$r = \hat{x} + xc$

$g^r = \hat{h} h^c, \quad m^r = \hat{z} z^c$

Computes $r' = ur + v$

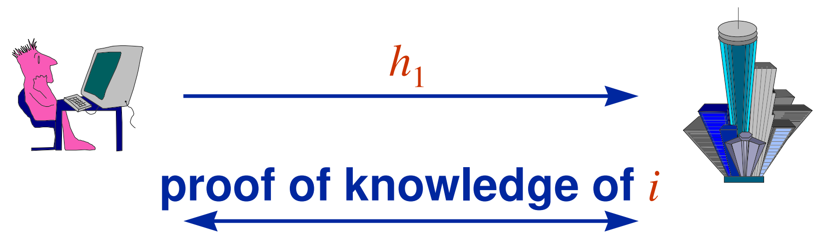
Blinded Schnorr Signatures

- **Signature:** $(r', m', m'', z', z'', h'')$
- **Verifier checks**
  - $h'' h^{c'} = (\hat{h}^u g^v) h^{cu} = (\hat{h} h^c)^u g^v = g^{ur+v} = g^{r'}$
  - $z'' z'^{c'} = (\hat{z}^s \hat{h}^t)^u m'^v (z^s h^t)^{cu} = (\hat{z} z^c)^{su} (\hat{h} h^c)^{tu} m'^v =$

$$m^{rsu} g^{rtu} m'^v = (m^s g^t)^{ru} m'^v = m'^{ru+v} = m'^{r'}$$

Brands' Restrictive Blinding

- **Let** g_1, g_2 **two new generators of** G
- **Payer chooses index** i **and sets** $h_1 := g_1{}^i$



- **Message** $m = h_1 g_2 = g_1{}^i g_2$
- **In payments, payer has to know** $m' = g_1^{x_1} g_2^{x_2}$ **therefore blinding is restricted:**

$$m' = m^s g^t = g_1^{is} g_2^s g^t \quad \overset{?}{\Rightarrow} \quad t = 0$$

Untraceable Cash

IBM

Payment Protocol (1)

- **Withdrawal:**
  - Payer performs blinded signature protocol with
    $m = g_1^i g_2, t = 0,$ thus $m' = g_1^{is} g_2^s$
    $m'' = g_1^{y_1} g_2^{y_2}$ for randomly chosen y_1, y_2
- **Payment:**
  - Payer sends signature $(r', m', m'', z', z'', h'')$
  - Payee checks
  - Payer uses (m', m'') as public key for a fail-stop signature scheme, with secret key (is, s, y_1, y_2).
  - Payer signs message
    $C = H$(all details, including payee's name + date)

Untraceable Cash

IBM

Payment Protocol (2)

- $PK_{coin} = (m', m'') = (g_1^{is} g_2^s, g_1^{y_1} g_2^{y_2})$
  $SK_{coin} = (x_1, x_2, y_1, y_2) = (is, s, y_1, y_2)$
- **Signature:** $s_1 = x_1 C + y_1$
  $s_2 = x_2 C + y_2$
- **Verification:** $g_1^{s_1} g_2^{s_2} = g_1^{x_1 C + y_1} g_2^{x_2 C + y_2} = m'^C m''$
- **Doublespending with** $C, \tilde{C}$

$s_1 = x_1 C + y_1 \qquad s_2 = x_2 C + y_2$

$\tilde{s}_1 = x_1 \tilde{C} + y_1 \qquad \tilde{s}_2 = x_2 \tilde{C} + y_2$

Payment Protocol (3)

- **Doublespending with** $C, \tilde{C}$

$$s_1 = x_1 C + y_1 \qquad s_2 = x_2 C + y_2$$
$$\tilde{s}_1 = x_1 \tilde{C} + y_1 \qquad \tilde{s}_2 = x_2 \tilde{C} + y_2$$

$$\Rightarrow \quad x_1 = \frac{s_1 - \tilde{s}_1}{C - \tilde{C}}, \quad x_2 = \frac{s_2 - \tilde{s}_2}{C - \tilde{C}} \quad \Rightarrow \quad i = \frac{x_1}{x_2} = \frac{s_1 - \tilde{s}_1}{s_2 - \tilde{s}_2}$$

Identity of Doublespender

IBM

Payment Protocol (4)

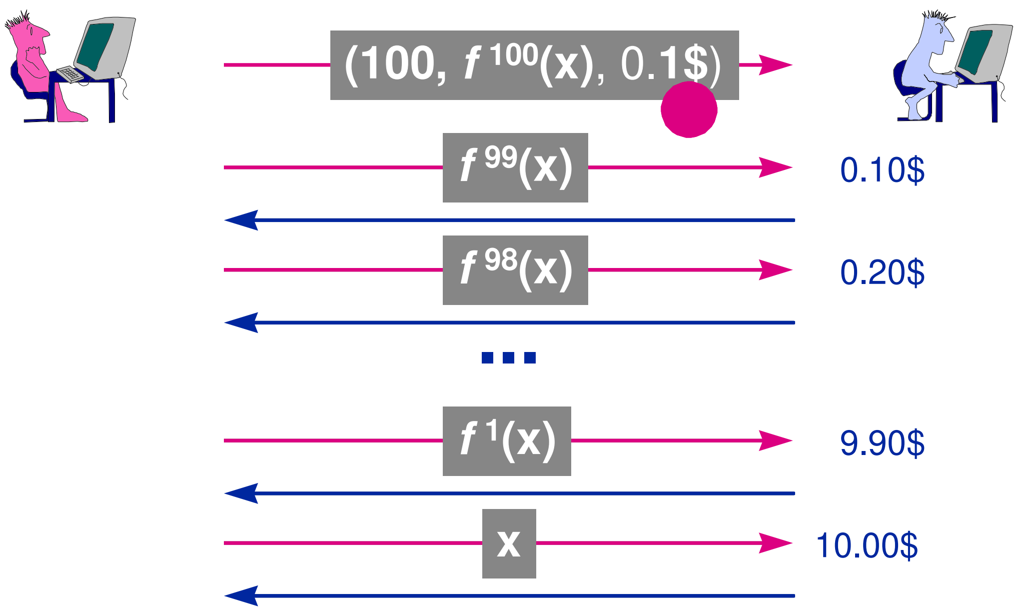
- **Applications**
  - **Coins (*Brands*):**
    - Each signature by bank has fixed denomination
    - Cash represented by these coins
  - **Cheques with Counter (*Brands*, *CAFE*):**
    - Value is determined during payment
    - Cash represented by tamper-resistant counters
- **Extensions**
  - **Loss tolerance / coin & cheque recovery**
  - **Improved divisibility**
  - **Improved storage: *n*-spendable coins / cheques**

IBM

Different Notions of Micropayments

- **Any combination of small amount (0.01 - 10 CHF)**
  - with many **payments in short time,** and/or
  - with payments to **many different payees**
  - e.g., pay-per-click, metered access
- **Problems**
  - policy for implicit user authorization
  - costs for clearing of small payments
  - high frequency / constant payee: CAFE
  - high frequency / different payees: NetBill
  - low frequency: MiniPay
- **Not clear yet**
  - economic need for high frequency / different payees

CAFE Phone Ticks



CAFE Phone Ticks

+ + +

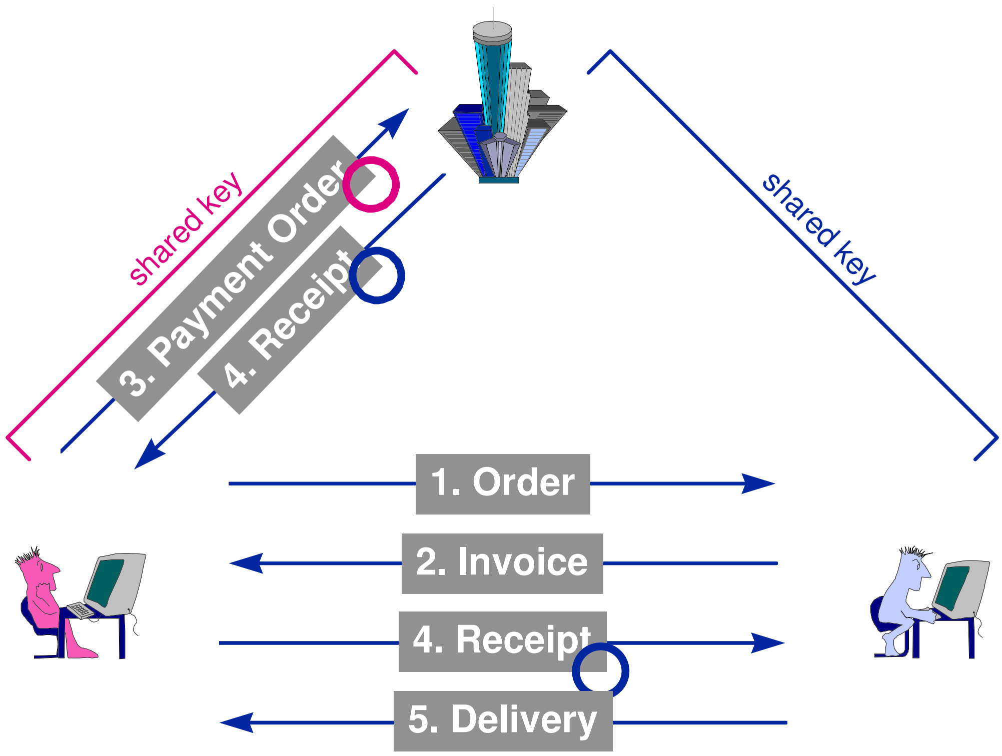
- **Efficient**
  - 1 hash per "tick"
- **Approximates fair exchange of tick / value**
- **Can be combined with any payment system**

– – –

- **Only for high frequency payments to *one* payee**

- **Reinvented by several people afterwards:**
  - Rivest/Shamir, Anderson, ...

Billing Server



Billing Server

+ + +	– – –

+ + +

- **Efficient**
  - MAC's are sufficient
- **Billing server can ensure fairness**
  - Keep money "on hold" until service delivered

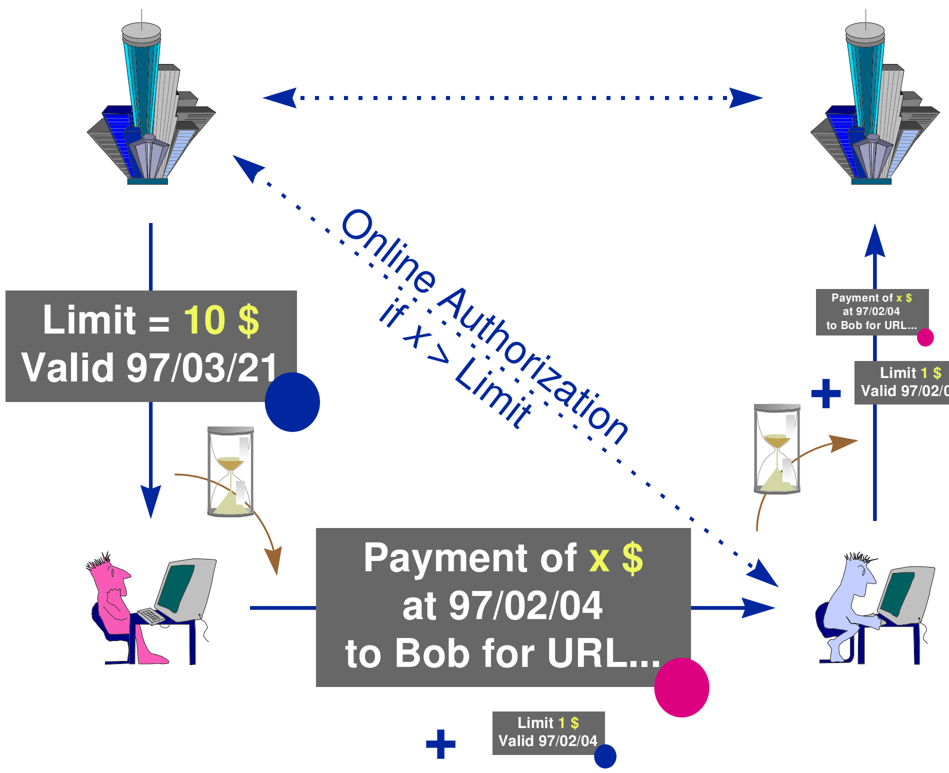
– – –

- **Alice and Bob have to trust 1 billing server**
- **Online third party**
  - bottleneck
  - must be trustworthy

- **Also called "Payment Server", "Proxy Server"**
- **Digital signatures instead of MACs for improved scalability and trust management**

Micropayments

IBM MiniPay



IBM MiniPay

Micropayments

+ + +	− − −
◆ Message efficient ◆ Offline in most cases ◆ Signature based ◆ non-repudiation of payment order	◆ Signature based ◆ not for high frequency ◆ Fraud not 100% prevented

IBM

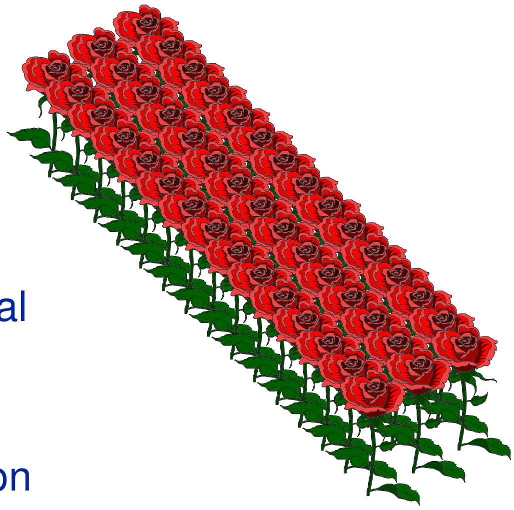
Existing Systems and Proposals

	Online Payments	Offline Payments
Traceable	◆ **Crypto-less Systems:** just-do-it, *FirstVirtual* ◆ **Encrypted CC:** just-do-it, e.g., *SSL* ◆ **Secure CC:** *CyberCash, iKP/IPay, SET* ◆ **Secure Cheques:** *FSTC e-check* ◆ **Billing server for Micro payments:** *NetBill, NetCash* ◆ **Micro payments:** *CAFE Phoneticks, Micro-iKP, Millicent, MicroMint, MiniPay*	◆ **Smartcard-based electronic purses** ◆ **MAC-based:** *Danmont, Proton, ZKA, Chipper, ...* ◆ **Signature-based:** *CLIP, Mondex, ...*
Untraceable	◆ **Anonymous “exchangers”:** *Anonymous Credit Cards, NetCash* ◆ **Blind Signatures:** *ecash*	◆ **Smartcard-based or Electronic Wallets + Blind Signatures:** *CAFE*

IBM

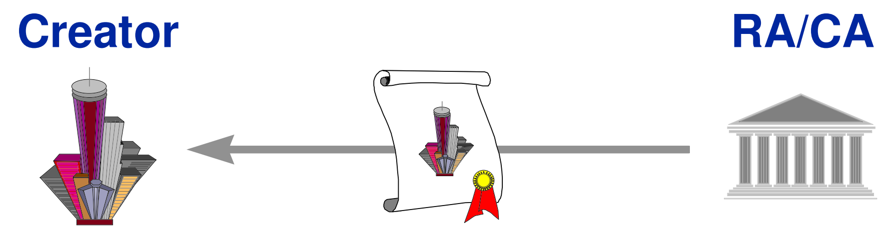
+ many other proposals and projects ...

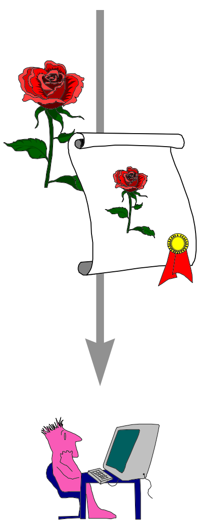
Secure Transfer of Digital Originals



- **Owner / seller**
  - Protection of Intellectual Property Rights
  - Prevent *mass-fraud*
  - Cost-effective protection
- **Buyer**
  - Secure against wrong accusations by sellers
  - Authenticity of copies bought
  - No negative impact on quality, performance

Authenticity of a Copy





- **Registration of Creators**
  - by whom?
  - semantics of certificate?
- **Achieves "accountability"**
- **No guarantee for correctness**
- **No guarantee for honesty**
- **Several "codesigning" proposals**

Digital Originals

Copy Control

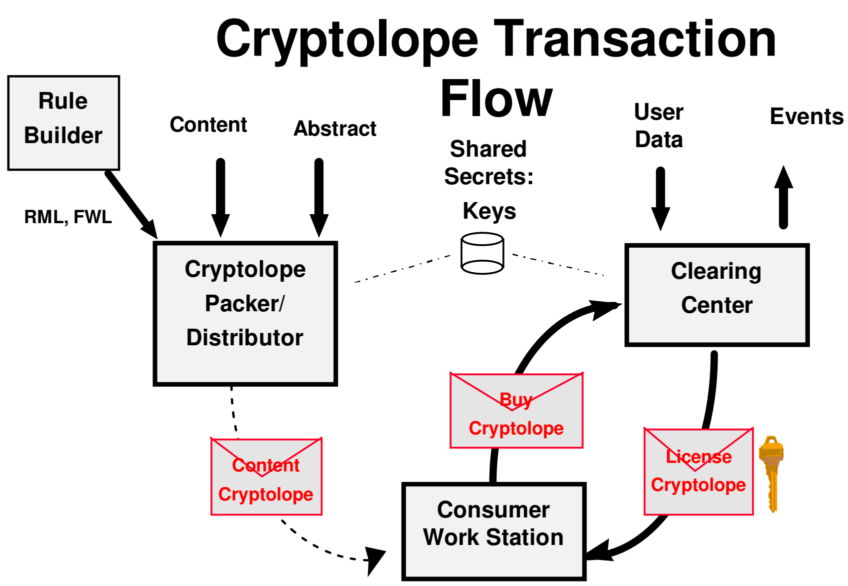
- **Decrease Incentive of Making Illegal Copies**
  - Customization: specific info, interactive presentation
  - Short lifetime of information
  - Cheap access to information
- **Increase Risk of Distributing Illegal Copies**
  - Individualized authentication, watermarking
  - **Fingerprinting**
- **Prevent Illegal Copies**
  - low-tech attackers: **Superdistribution**
  - high-tech attackers: Tamper-resistant hardware

IBM

Digital Originals

Superdistribution

Cryptolope CONTAINERS



IBM

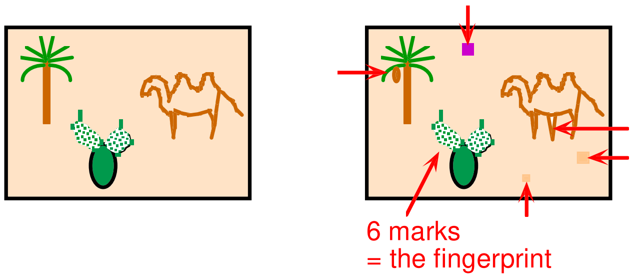
Digital Originals

Superdistribution

- **Content Cryptolope**
  - Bill of Materials
  - Abstract, Metadata
  - Encrypted Files + Key Records
  - Rights Management Language
  - Fingerprinting / Watermarking
  - Digital Certificates
- **Cryptolope Types**
  - Content
  - Buy = Spec's + Consumer Environment Info
  - License = Keys + F/W Details

Digital Originals

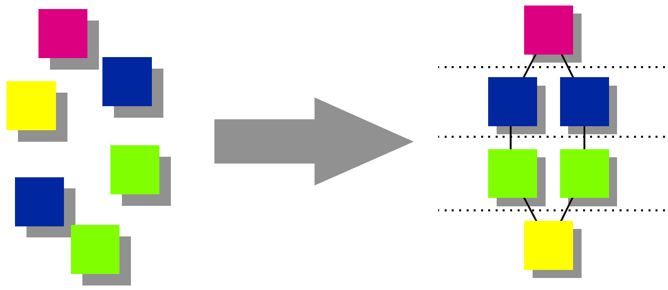
Fingerprinting



- **Different in each sold "copy"**
- **No significant change to data**
  - Usefulness for buyer
  - Marks hard to find for traitors
- **More tolerant against transformations than data**
- **Collusion tolerance**
- **Asymmetric Fingerprinting**
- **Anonymous Fingerprinting**

Frameworks

Frameworks for Secure Electronic Commerce



- **Interoperability of Components**
- **Consistent and coherent user interface**
- **Open for new components, e.g., new services, new protocols, new implementations**
- **System, application, vendor neutral**

IBM

Frameworks

Proposed Frameworks

- **Vendor-specific Frameworks**
  - IBM (SecureWay)
  - Microsoft (Internet Security Framework)
  - Netscape
- **Java Electronic Commerce Framework** (Sun)
- **Research Projects**
  - SEMPER (EU ACTS Project)
    Secure Electronic Marketplace for Europe
  - ECO-Sys (CommerceNet)

IBM

Secure Electronic Marketplace for Europe

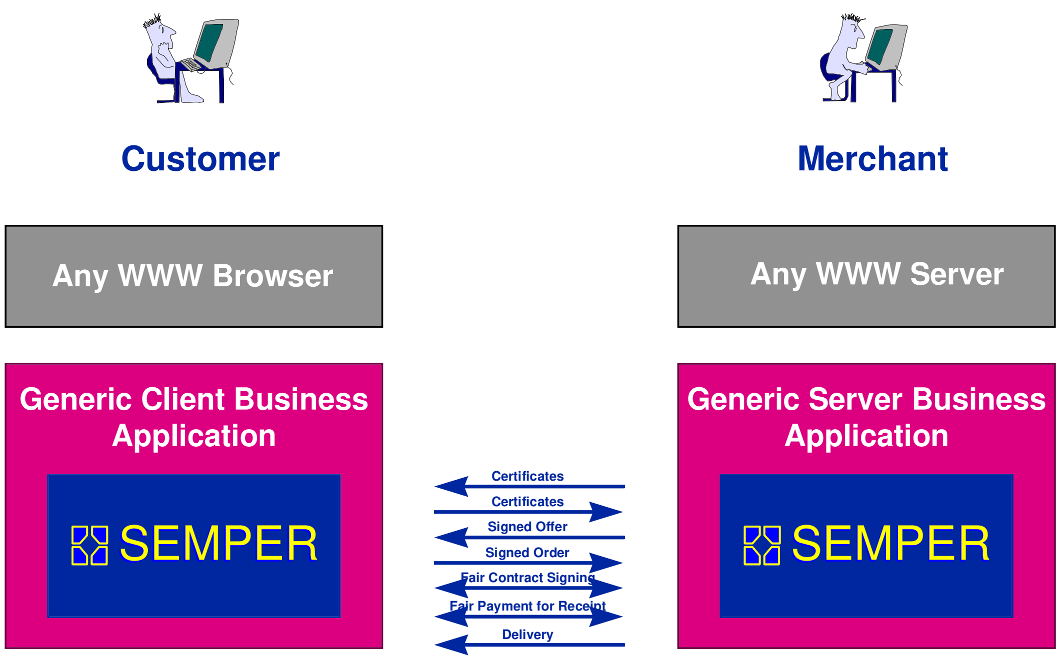
SEMPER

- EU ACTS Project, 20 Partners, 1995 - 1998
- Objective: "Establish an open and generic security architecture for the global marketplace"
  - Define a model of the marketplace
  - Specify the architecture and associated services
  - Evaluate the architecture and the services
    - Develop a prototype
    - Run realistic trials
    - Evaluate security and consumer response
  - Disseminate the results
    - Publish the architecture and the specifications
    - Contribute to standardisation efforts

IBM

SEMPER Architecture

SEMPER



IBM

Frameworks

SEMPER Architecture

SEMPER

Business Applications

"Generic BA" used by Otto-Versand, FOGRA, Eurocom

Commerce Block

Standard Business Processes, "Scripts" and "Templates"

Transfers & Fair Exchanges

"Containers" + Time Stamping, Contracts, Certified Mail, etc.

Payments

"Money"

Certificates

"Credentials"

Statements

"Documents"

Supporting Services

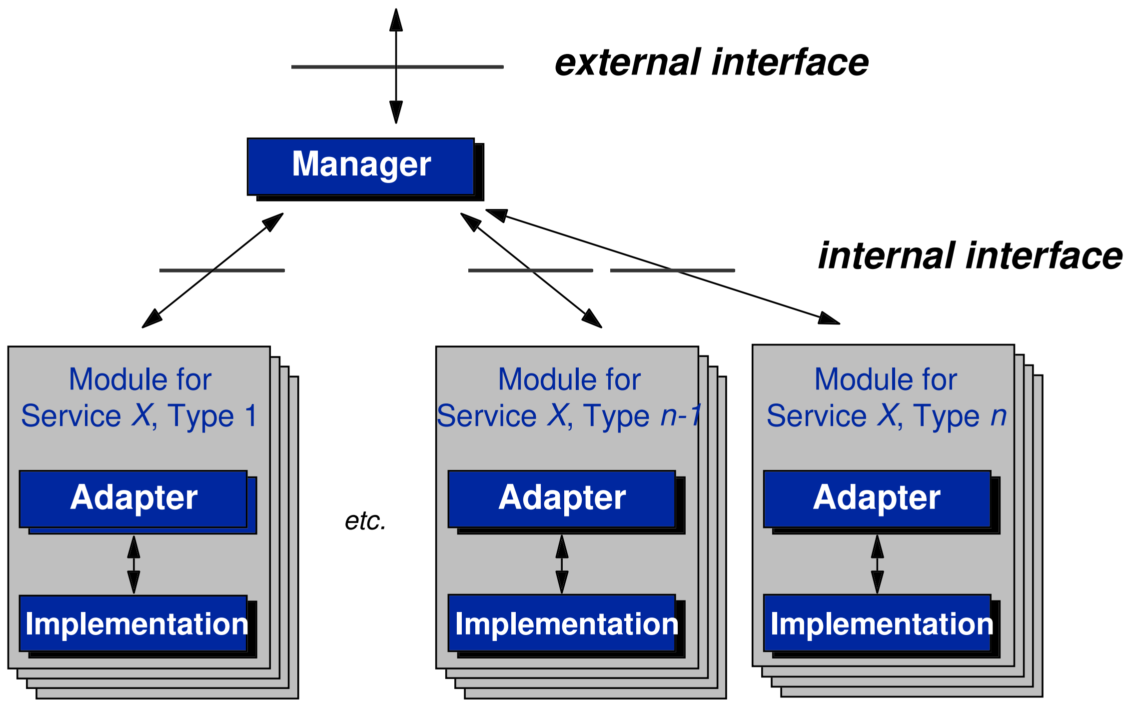
Communication, Crypto Engine, Trusted User I/O (TINGUIN), Archive, Access Control, Preferences

IBM

Frameworks

Service Blocks

SEMPER

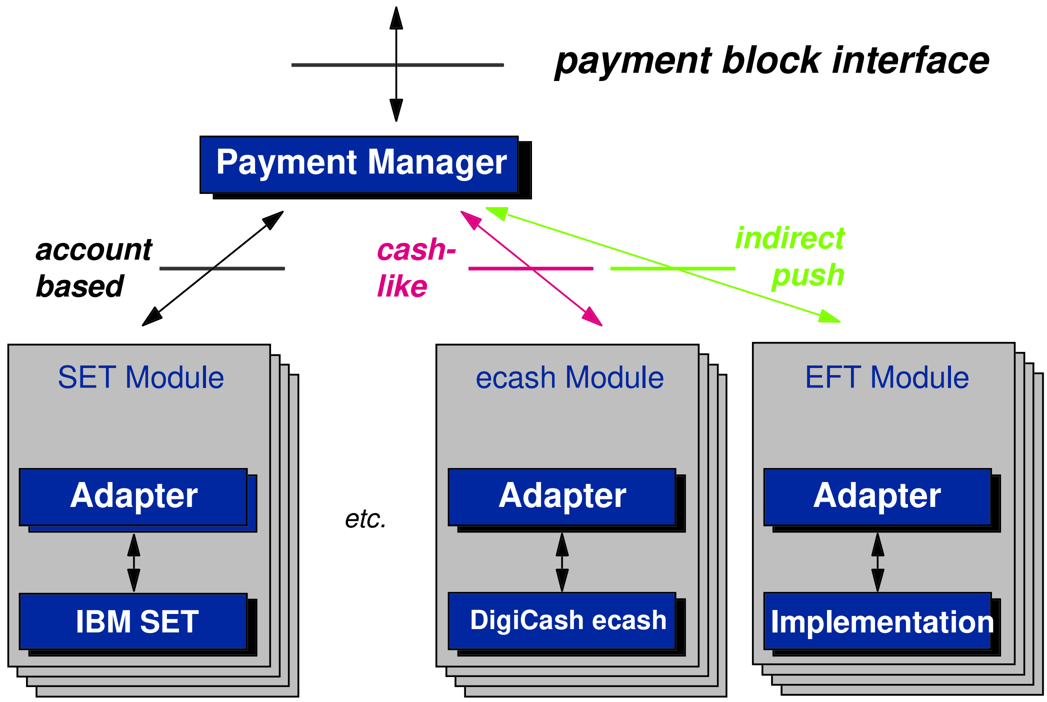


IBM

© IBM 2/15/97 <wmi@zurich>

Frameworks

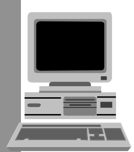
For Instance: Payments SEMPER



© IBM 2/15/97 <wmi@zurich>

Outlook

Summary and Outlook



- **Today and tomorrow**
  - SSL for communication security; but no sound global certification infrastructure available yet
  - Credit cards, unprotected or secured via SSL
  - Software-only solutions
  - Closed systems; shopping, home banking, trading

- **Short-term**
  - Global frameworks, national CA's, digital signature laws
  - SET, e-cash, e-cheques, untraceable e-cash
  - Smartcards to prevent some fraud, and increase mobility
  - Citizen-government, individual-individual, SME-SME

- **Mid-term**
  - Global marketplace, global PK infrastructure
  - Unified interfaces, open systems: SEMPER, ECO, JECF, ...
  - Secure mobile user devices
  - General business