

E-Privacy – Privacy in the Electronic Society

Anonymous Credentials II

Jan Camenisch

IBM Zurich Research Laboratory

Rüschlikon

jca@zurich.ibm.com

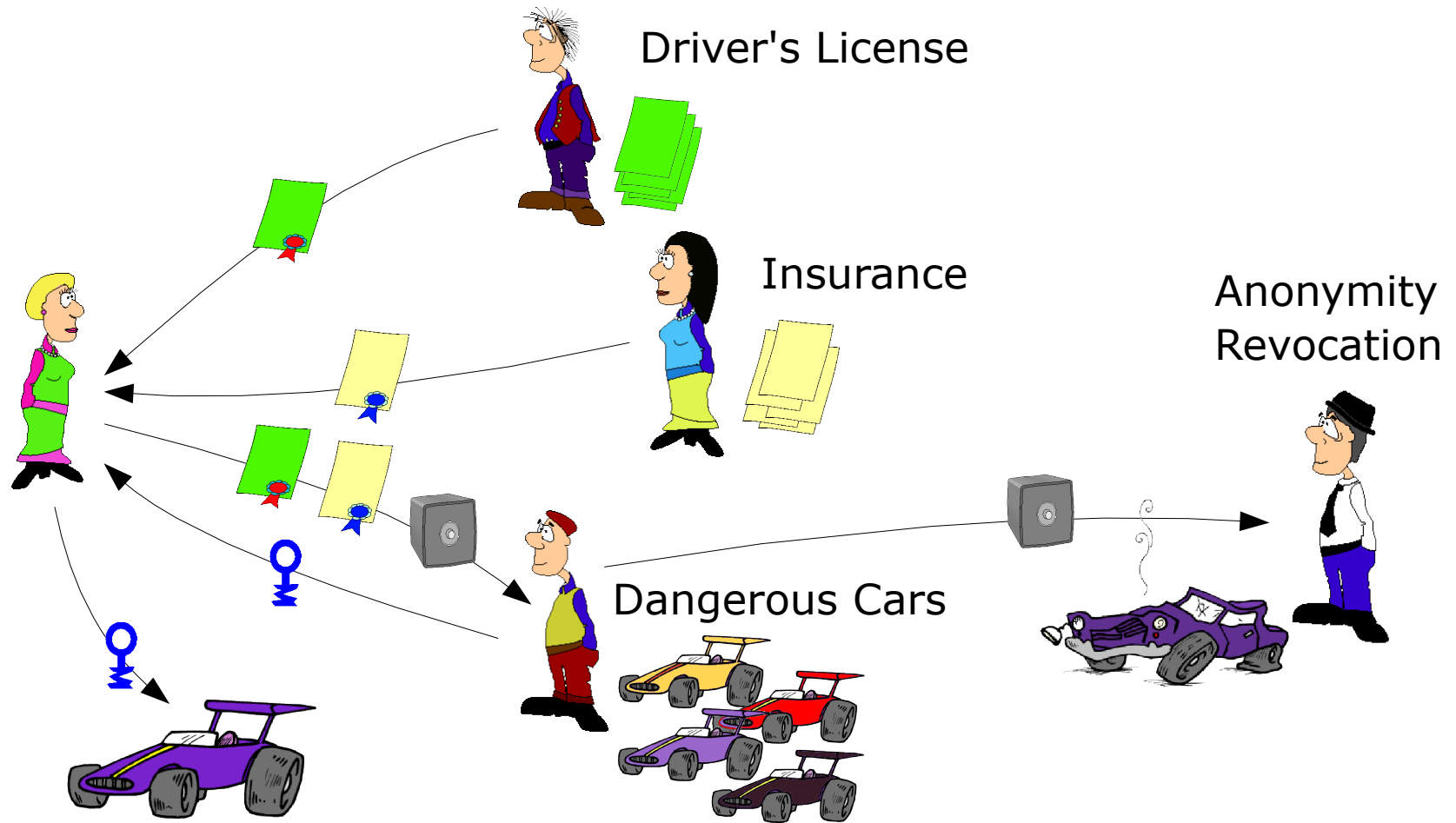
Interested in a Master Thesis @ IBM ?

- Incentive Systems for Wiki comments
 - Wiki content is not trustworthy
 - One way to achieve this is reviews
 - No one wants to do reviews
 - Need incentive system
 - Need privacy!
 - Theory, Design, Implementation

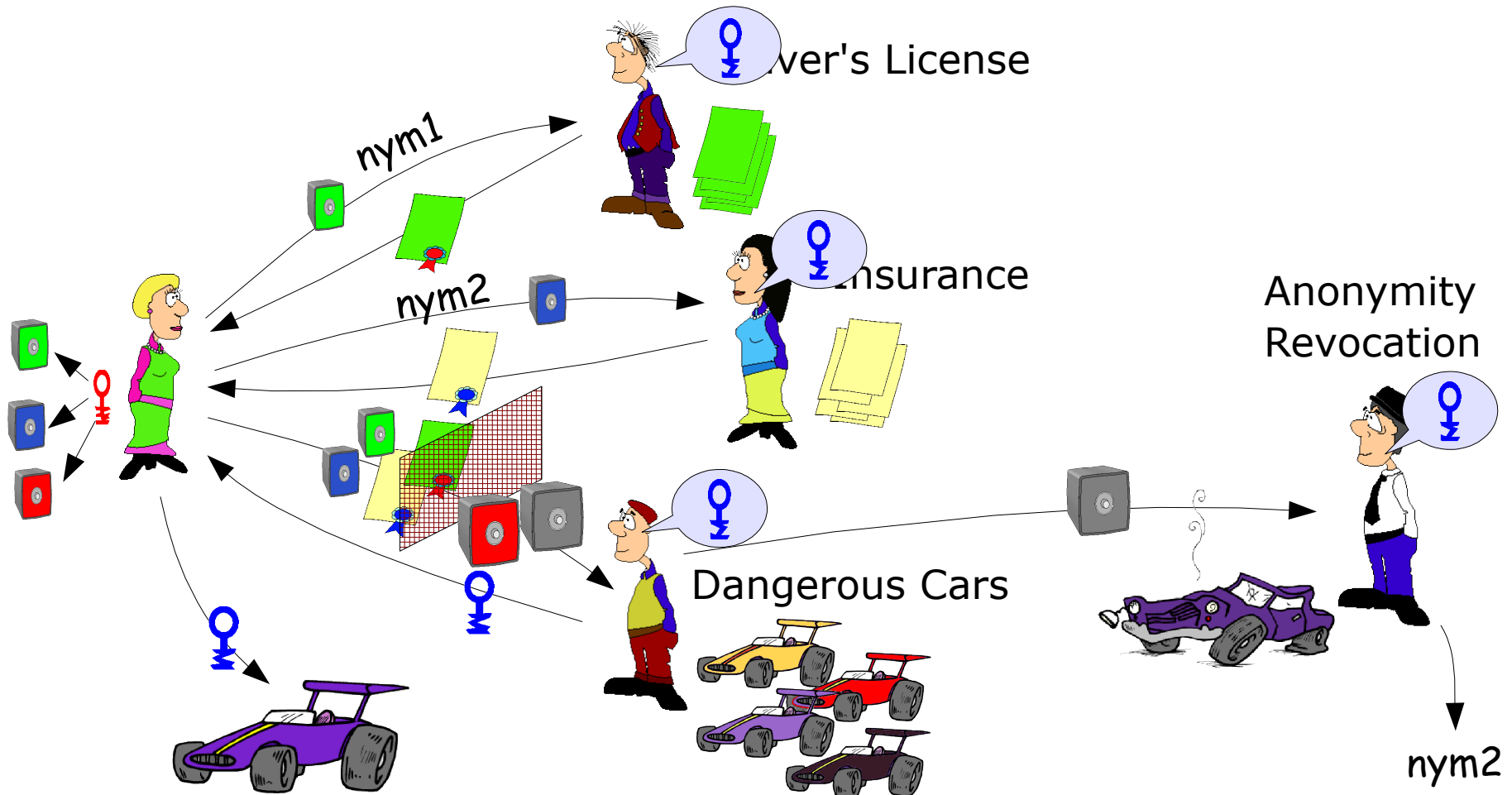
Outline

- Recap Credential Systems & Tools
- Building Blocks and Example Apps (Ecash)
 - Commitment scheme
 - RSA signatures
 - CL signatures
 - ECash



Credential System



Abstract Solution



Basic Requirements of Pseudonym System

- Protection of user's privacy
 - anonymity: use of different pseudonyms  and because of zero-knowledgeness of proofs
 - unlinkeability (multi-use): zero-knowledgeness of proofs
- Unforgeability of credentials: unforgeability of underlying signature scheme
- Consistency of credentials (no pooling): each user has one secret key, proof that all  contain same value.

Extra Requirements of Pseudonym System

- Sharing of credentials: not considered
- Anonymity revocation: security of proof, signature scheme, and encryption scheme
- Revocation of credentials: not considered
- Encoding of attributes: additional messages signed, can be selectively revealed.
- One-show credential (e-cash): not considered.

Technologies

Efficient Zero-Knowledge Proofs

Zero Knowledge Proofs

Given group $\langle g \rangle$ and element $y \in \langle g \rangle$.

Prover wants to convince verifier that she *knows* $x = \log_g y$ such that verifier only learns y and g .



Prover:

$$PK\{(a): y = g^a\}$$

Verifier:



random r

$$t := g^r$$

t



c



$$s := r - cx$$

s



random c

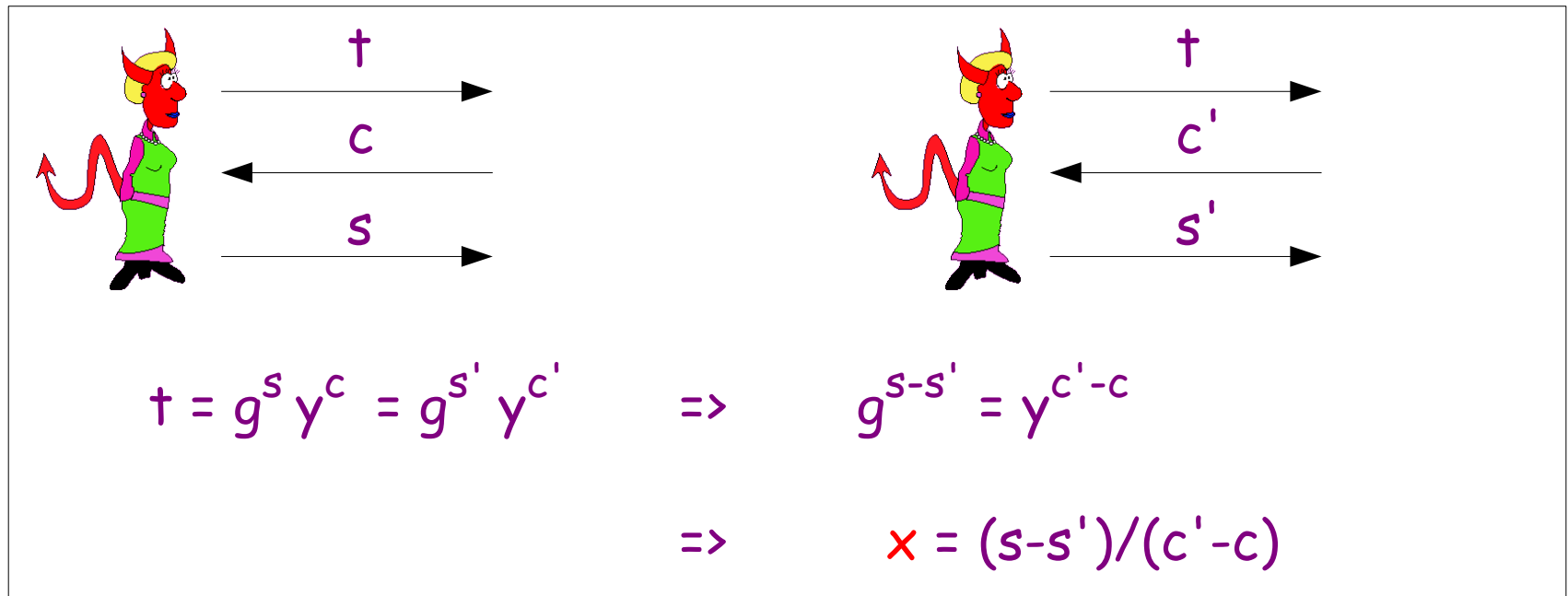
$$t' = g^s y^c$$

Zero Knowledge Proofs: Security

Proof of Knowledge Property:

If prover is successful, then she “knows” $x = \log_g \gamma$, i.e., one can extract x from her.

Idea: run two copies of Alice.



(One would need to consider success probabilities.....)

Zero Knowledge Proofs: Security

Zero-knowledge property:

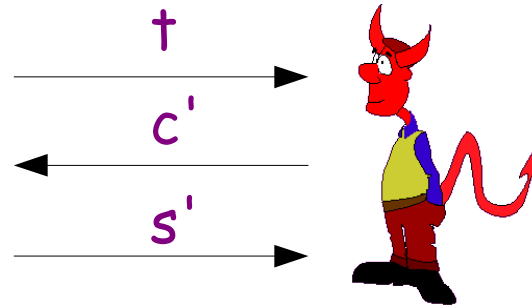
If verifier does not learn anything (except the fact that Alice knows $x = \log_g y$)

Idea: One can simulate whatever Bob "sees".

Choose random c, s

compute $t := g^s y^c$

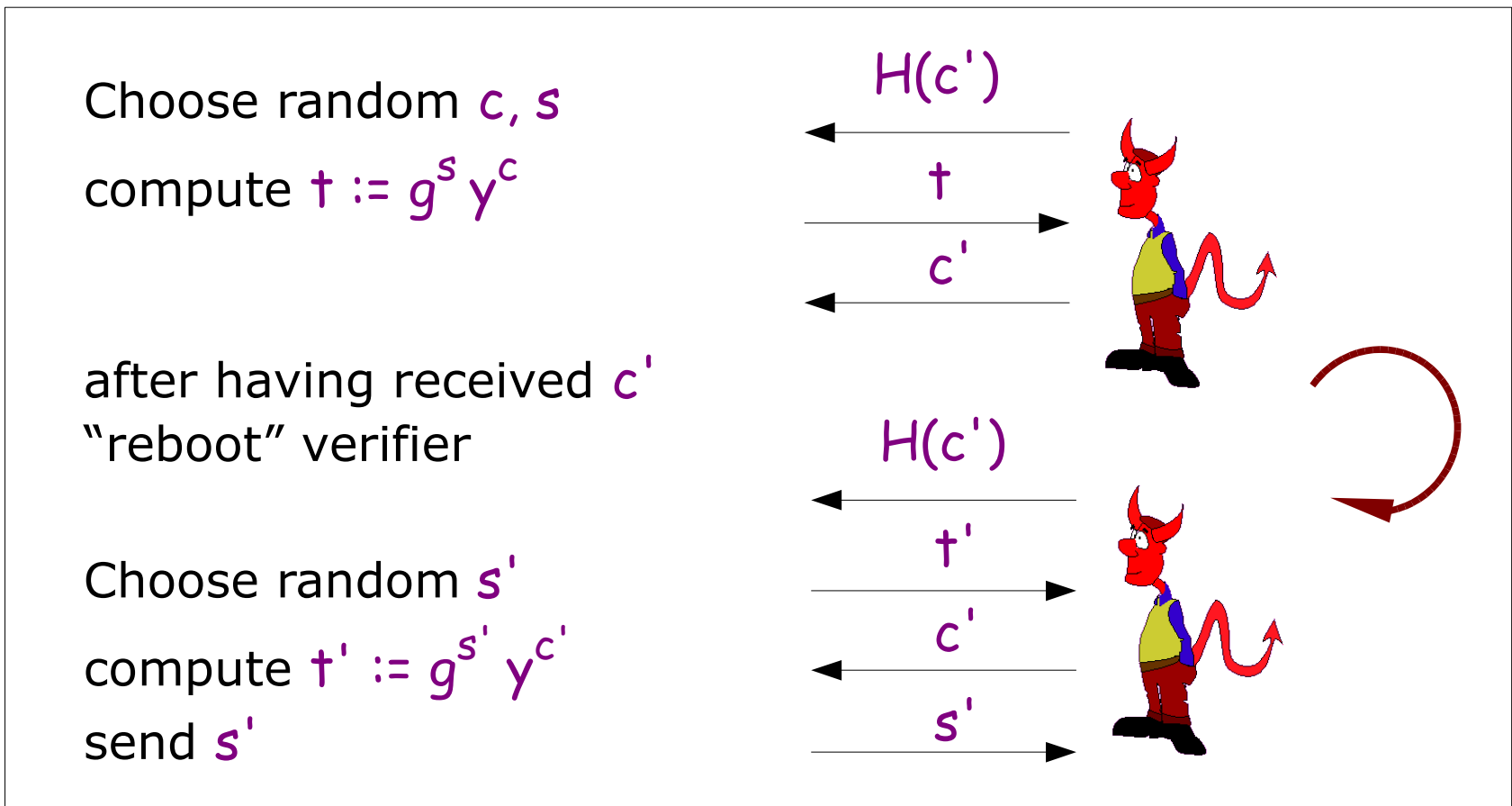
if $c = c'$ send $s' = s$,
otherwise restart



(works only if c' comes from small domain, but can be overcome by various means.)

Zero Knowledge Proofs: Security

One way to get large c :



(One would need to consider success probabilities.....)

Zero Knowledge Proofs

Non-interactive (Fiat-Shamir heuristic):

$$\text{PK}\{(\alpha): y = g^\alpha\}(m)$$

Logical combinations:

$$\text{PK}\{(\alpha, \beta): y = g^\alpha \wedge z = g^\beta \wedge u = g^\beta h^\alpha\}$$

$$\text{PK}\{(\alpha, \beta): y = g^\alpha \vee z = g^\beta\}$$

Intervals and groups of different order (under SRSA):

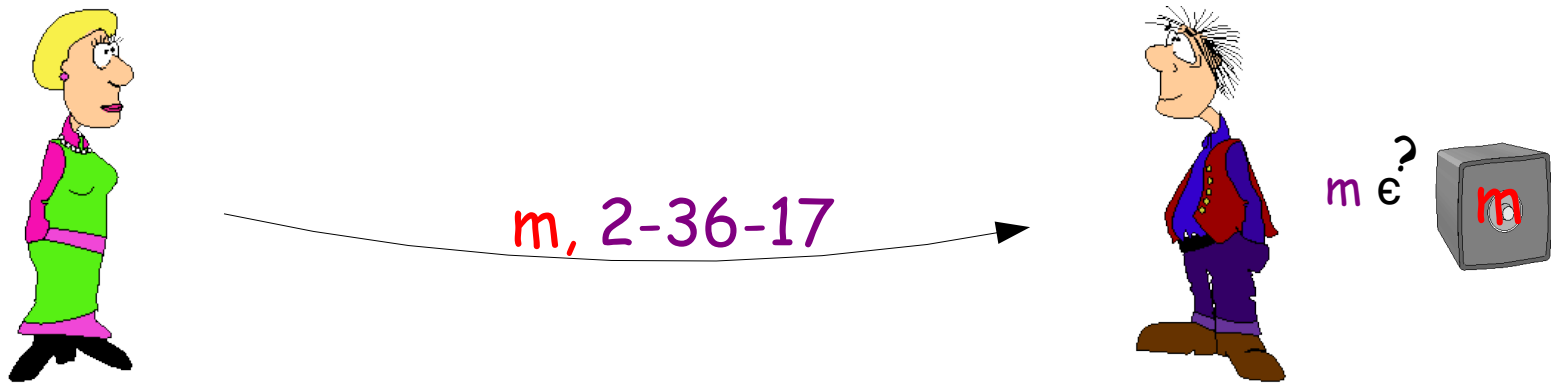
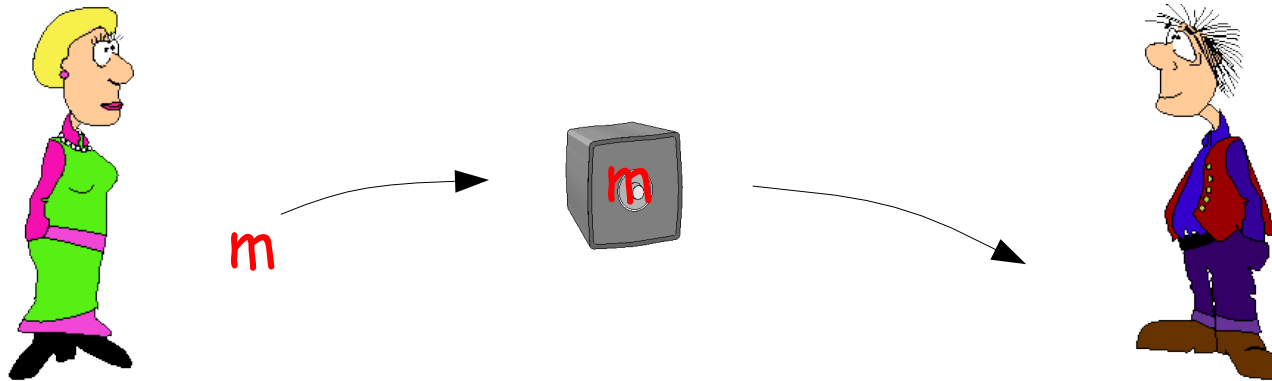
$$\text{PK}\{(\alpha): y = g^\alpha \wedge \alpha \in [A, B]\}$$

$$\text{PK}\{(\alpha): y = g^\alpha \wedge z = g^\alpha \wedge \alpha \in [0, \min\{\text{ord}(g), \text{ord}(g)\}]\}$$

Technologies

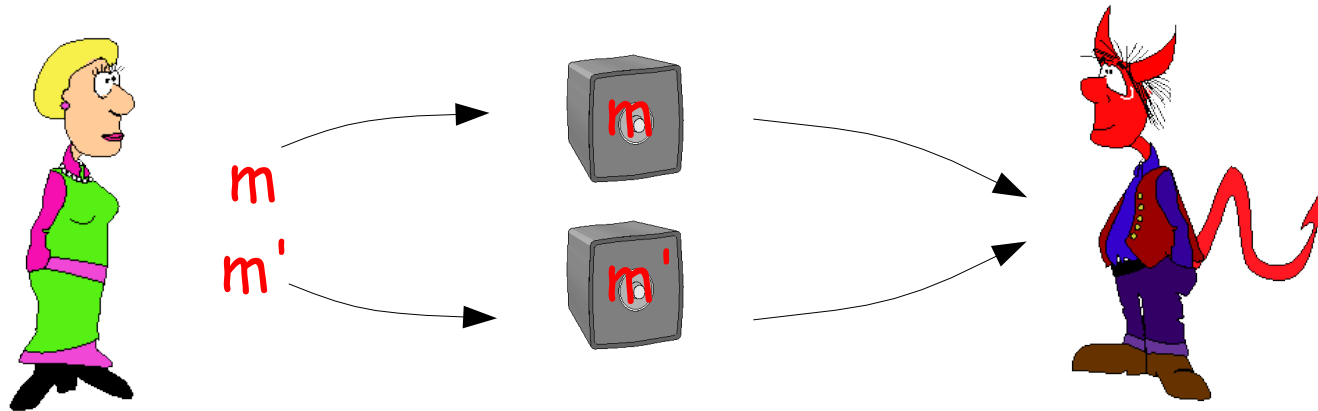
Commitment Schemes

Commitment Scheme: Functionality



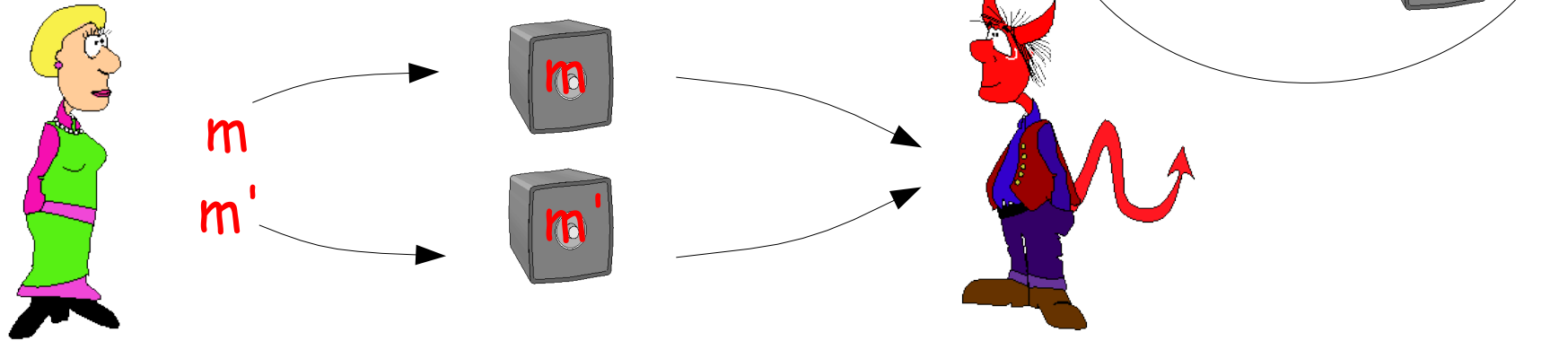
Commitment Scheme: Security

Hiding

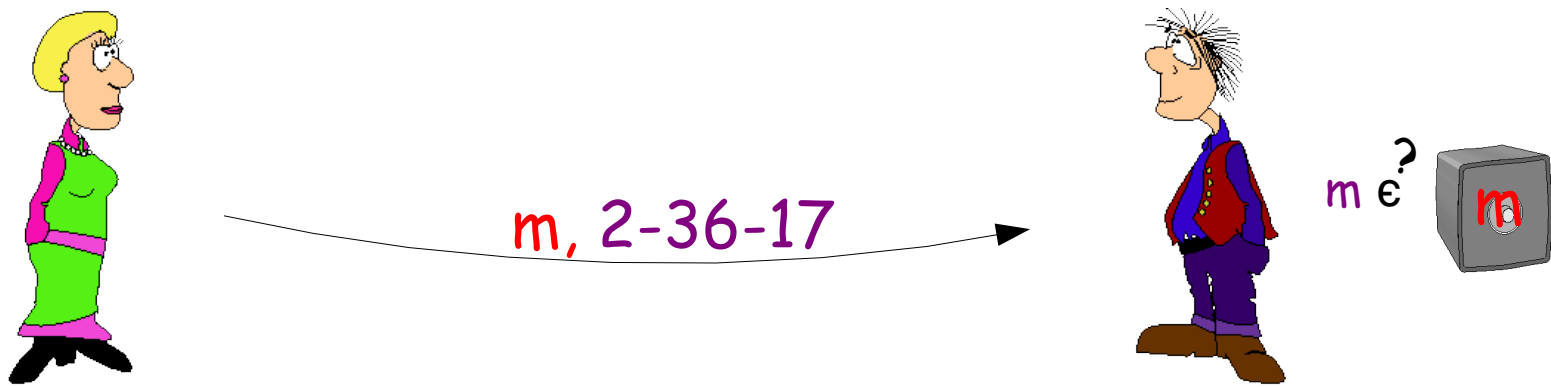
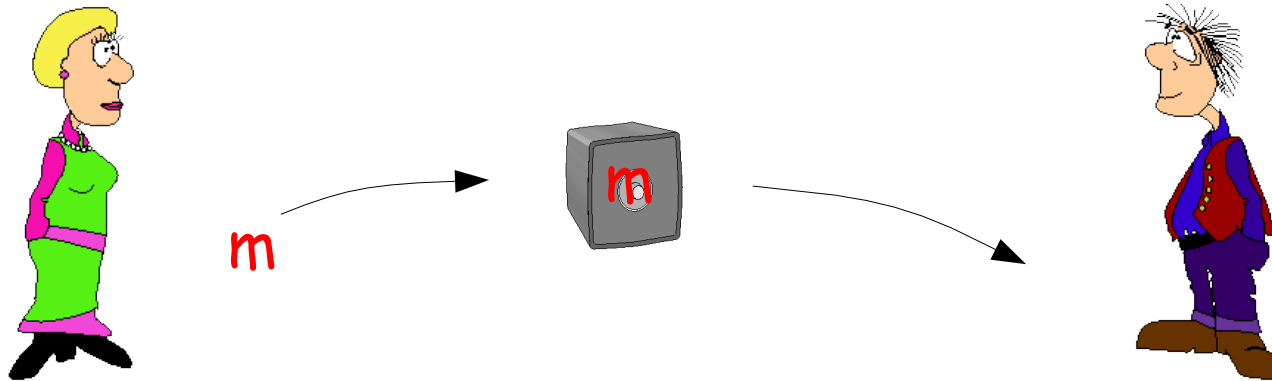


Commitment Scheme: Security

Hiding

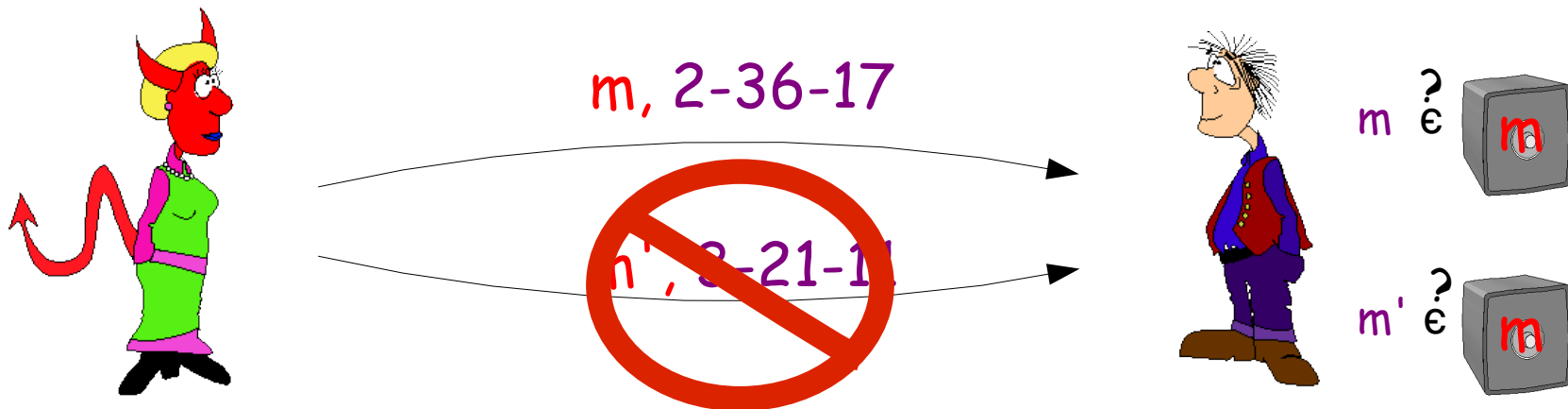


Commitment Scheme: Functionality



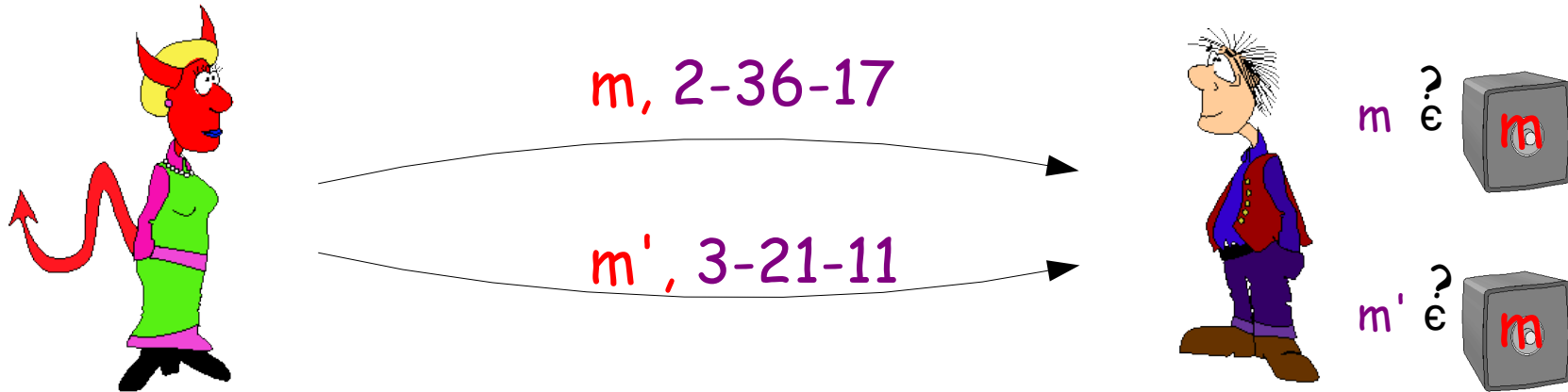
Commitment Scheme: Security

Binding



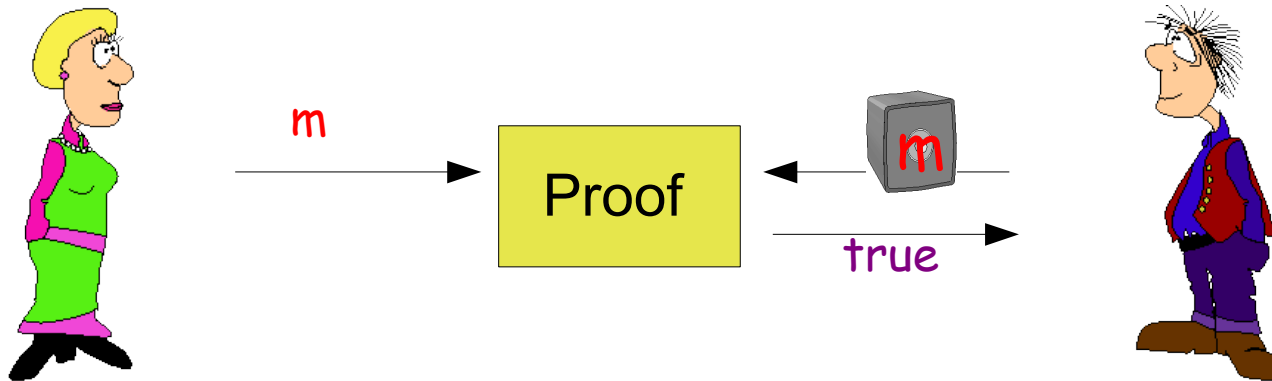
Commitment Scheme: Security

Binding

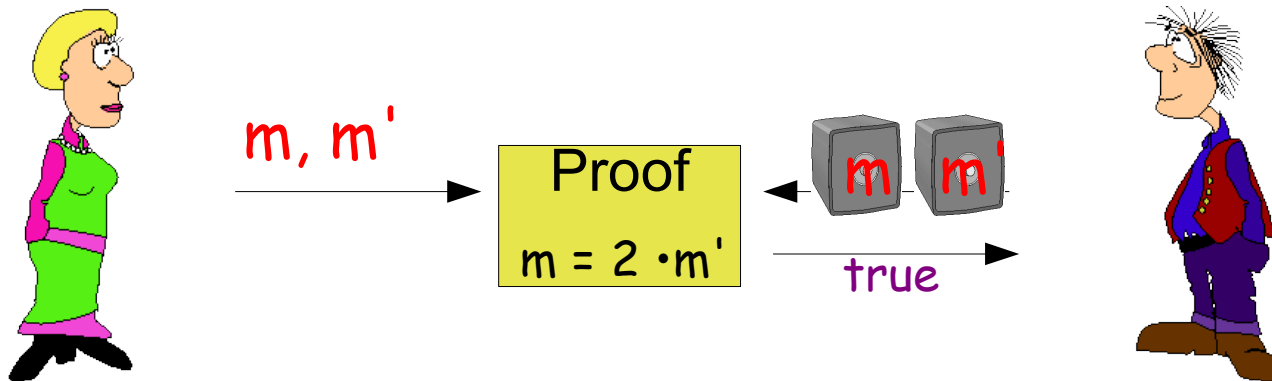


Commitment Scheme: Features

Proof of Knowledge of Contents



Proof of Relations among Contents



Commitment Schemes

Group $G = \langle g \rangle = \langle h \rangle$ of order q

To commit to element $x \in \mathbb{Z}_q$:

- Pedersen: perfectly hiding, computationally binding

choose $r \in \mathbb{Z}_q$ and compute $c = g^x h^r$

- ElGamal: computationally hiding, perfectly binding:

choose $r \in \mathbb{Z}_q$ and compute $c = (g^x h^r, g^r)$

To open commitment:

- reveal x and r to verifier

- verifier checks if $c = g^x h^r$

Pedersen's Commitment Scheme

Pedersen's Scheme:

Choose $r \in \mathbb{Z}_q$ and compute $c = g^x h^r$

Perfectly hiding:

Let c be a commitment and $u = \log_g h$

$$\begin{aligned} \text{Thus } c = g^x h^r &= g^{x+ur} = g^{(x+ur') + u(r-r')} \\ &= g^{x+ur'} h^{r-r'} \quad \text{for any } r'! \end{aligned}$$

I.e., given c and x' here exist r' such that $c = g^{x'} h^{r'}$

Computationally binding:

Let $c, (x', r')$ and (x, r) s.t. $c = g^{x'} h^{r'} = g^x h^r$

Then $g^{x'-x} = h^{r-r'}$ and $u = \log_g h = (x'-x)/(r-r') \pmod q$

Commitment Schemes: Integers

To commit to integer $x \in \mathbb{Z}$: similarly, if order of G is not known, e.g., $G = \mathbb{QR}_n$ for an RSA modulus n , e.g.:

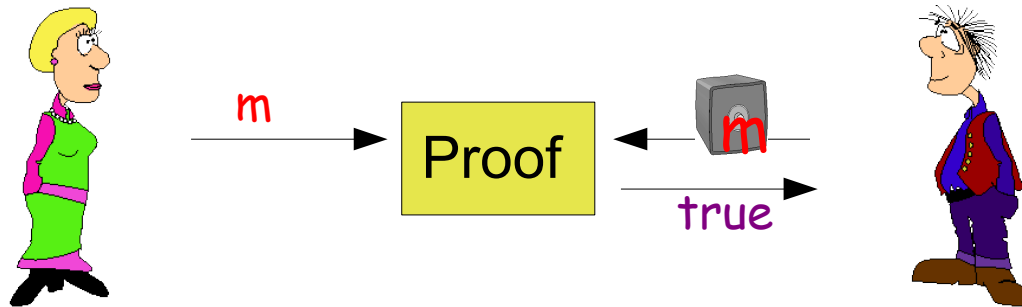
Let h, g be elements of \mathbb{QR}_n modulus n , then to commit to any integer x , choose $r \in [0, n+2^\ell]$ and compute

$$c = g^x h^r \pmod n$$

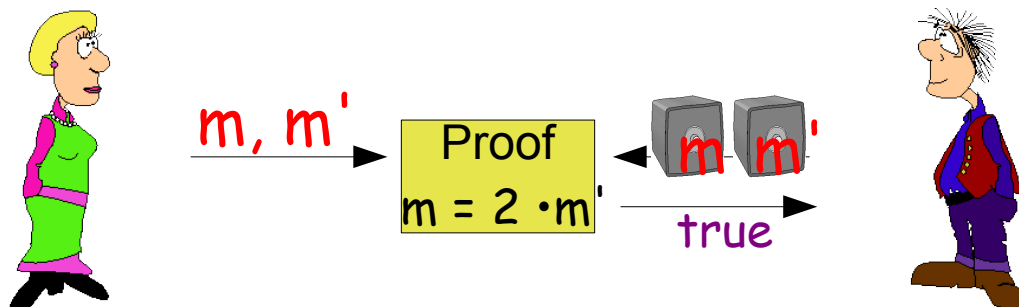
(Normally, one would choose $r \in [0, \phi(n)-1]$, i.e., between 0 and the order of the group. Here, however, the order is not known. Choosing $r \in [0, n+2^\ell]$ results in a distribution of h^r that is statistically close to the uniform distribution over of $\langle h \rangle$, i.e., almost uniform.)

Commitment Scheme: Features

Let $c = g^m h^r$ and $c' = g^{m'} h^r$ then:



$$\text{PK}\{(\alpha, \beta): c = g^\beta h^\alpha\}$$



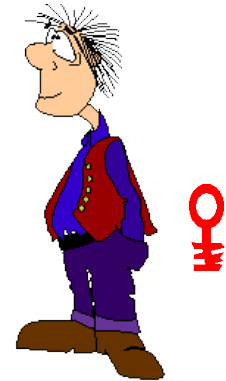
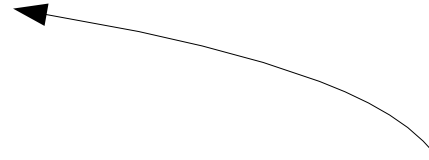
$$\text{PK}\{(\alpha, \beta, \gamma): c' = g^\beta h^\alpha \wedge c = g^{2\beta} h^\gamma\}$$

Technologies

Signature Scheme

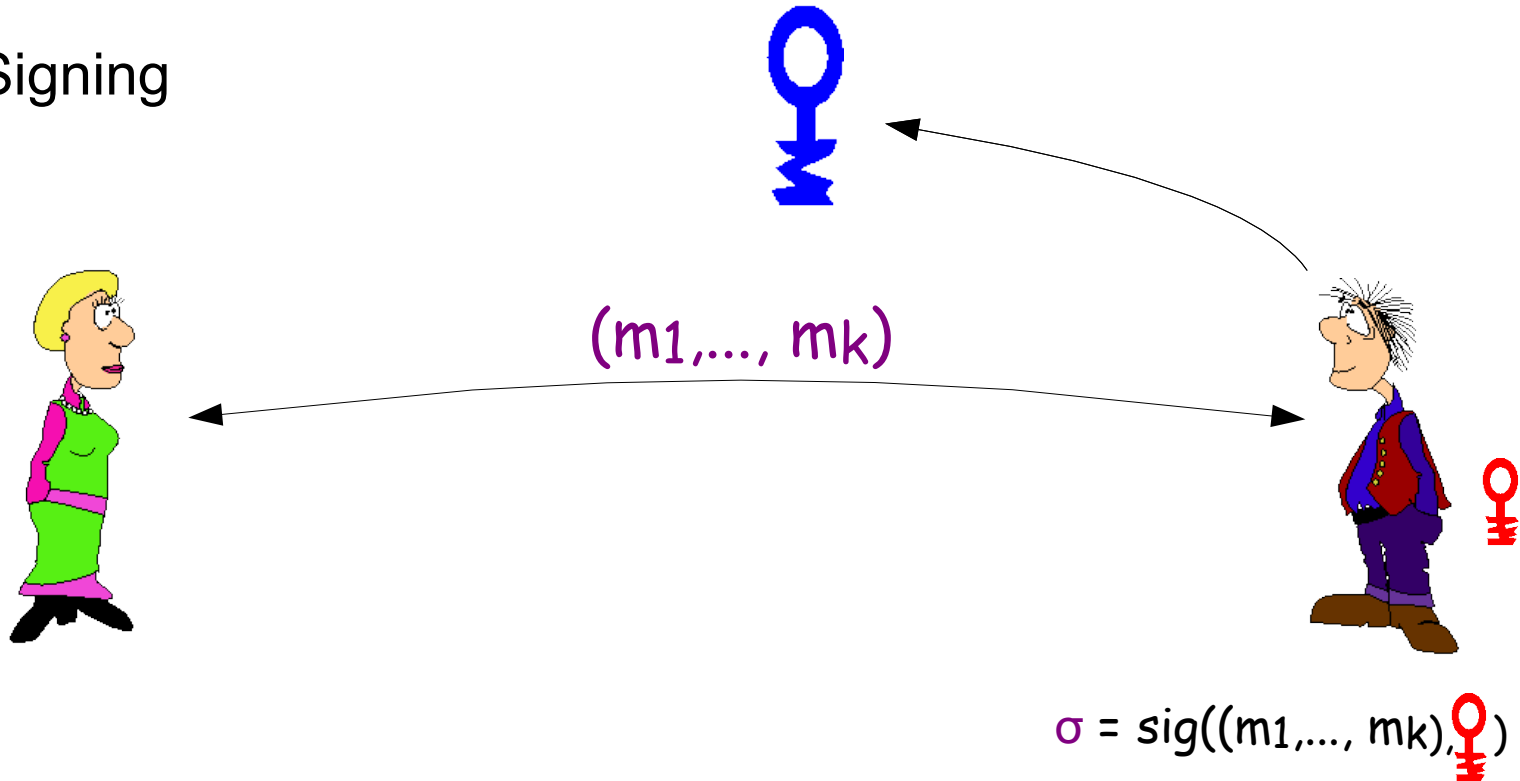
Signature Scheme: Functionality

Key Generation



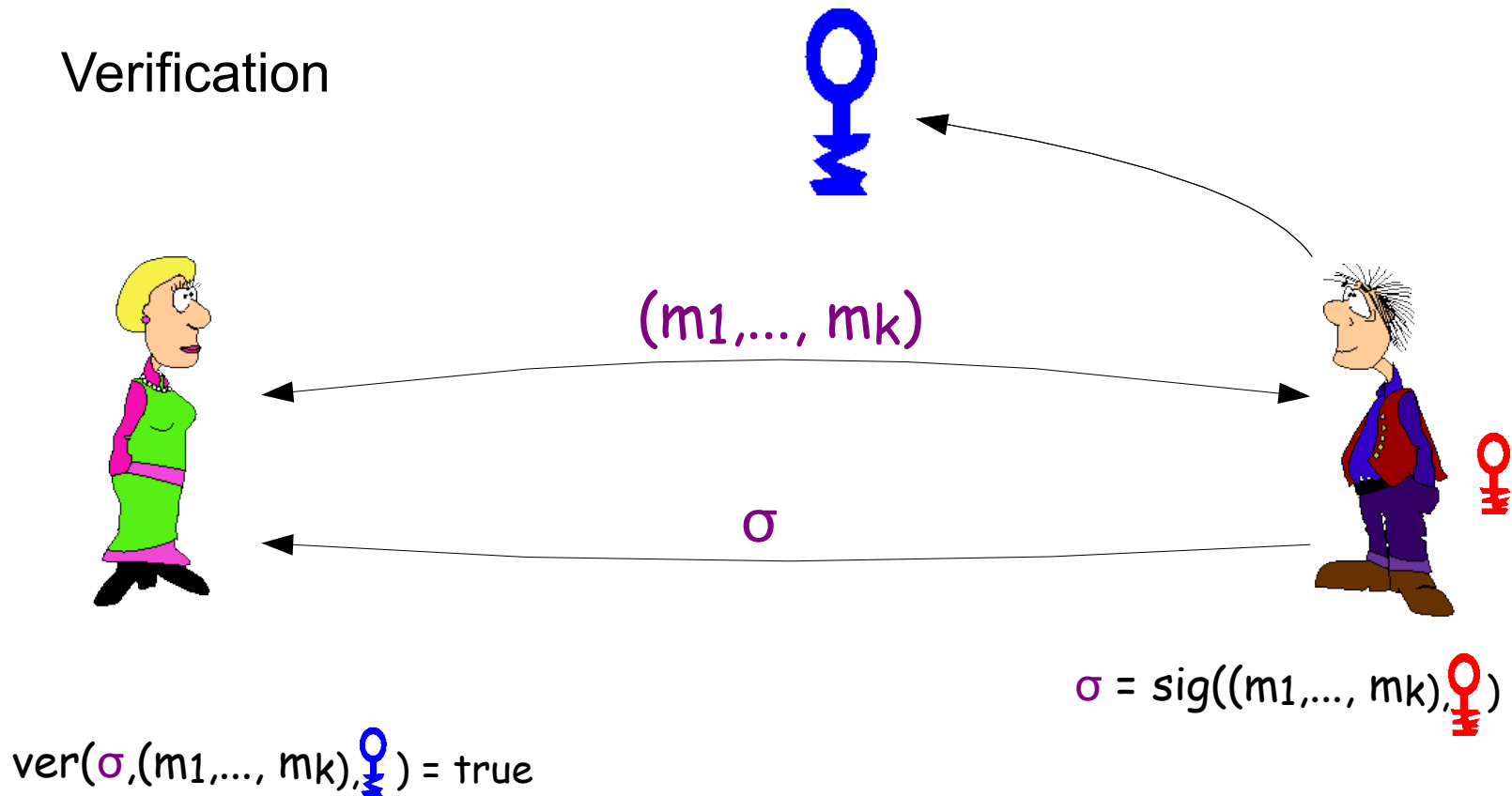
Signature Scheme: Functionality

Signing



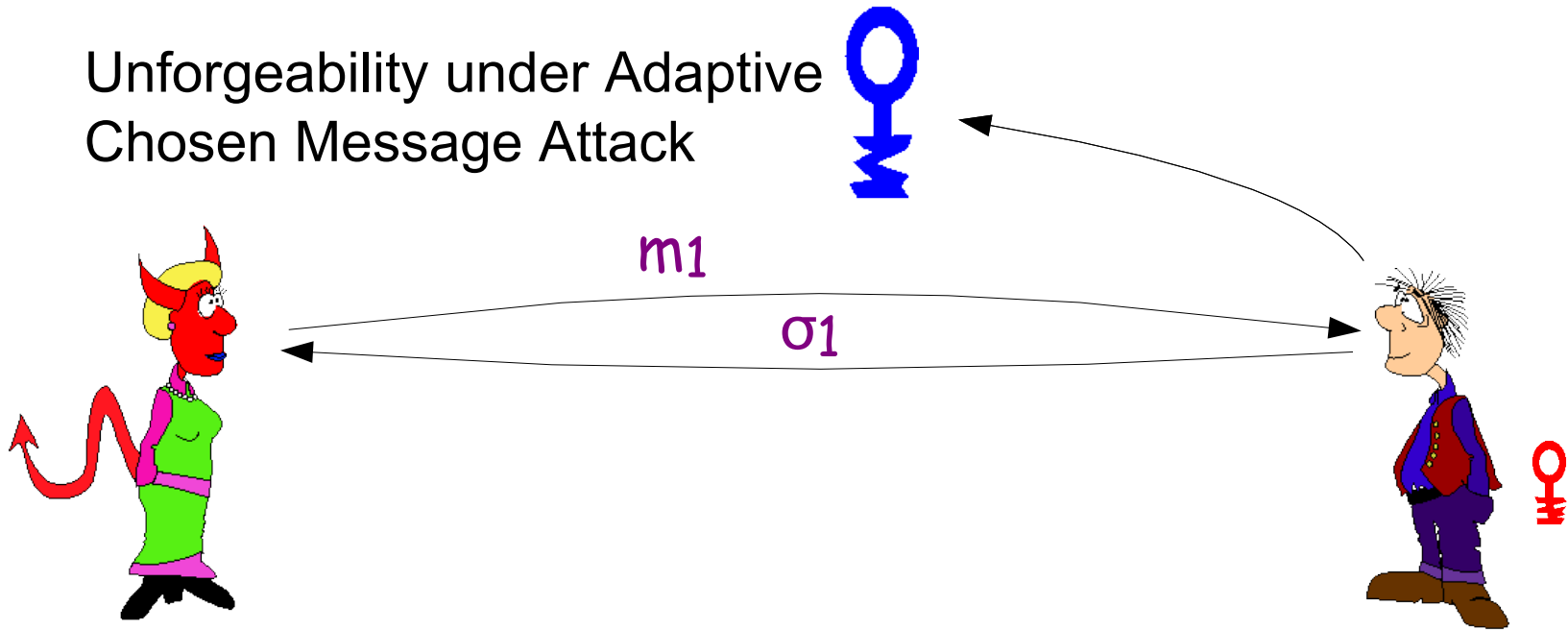
Signature Scheme: Functionality

Verification



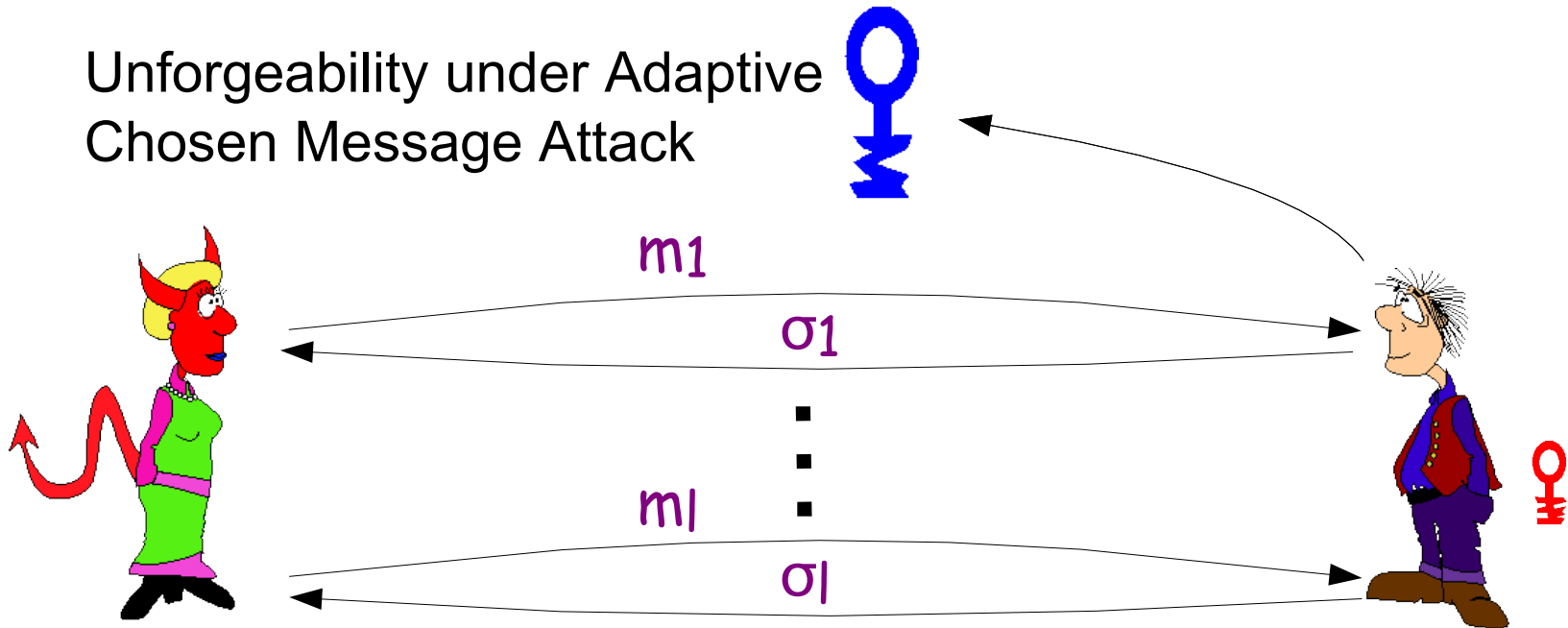
Signature Scheme: Security

Unforgeability under Adaptive Chosen Message Attack



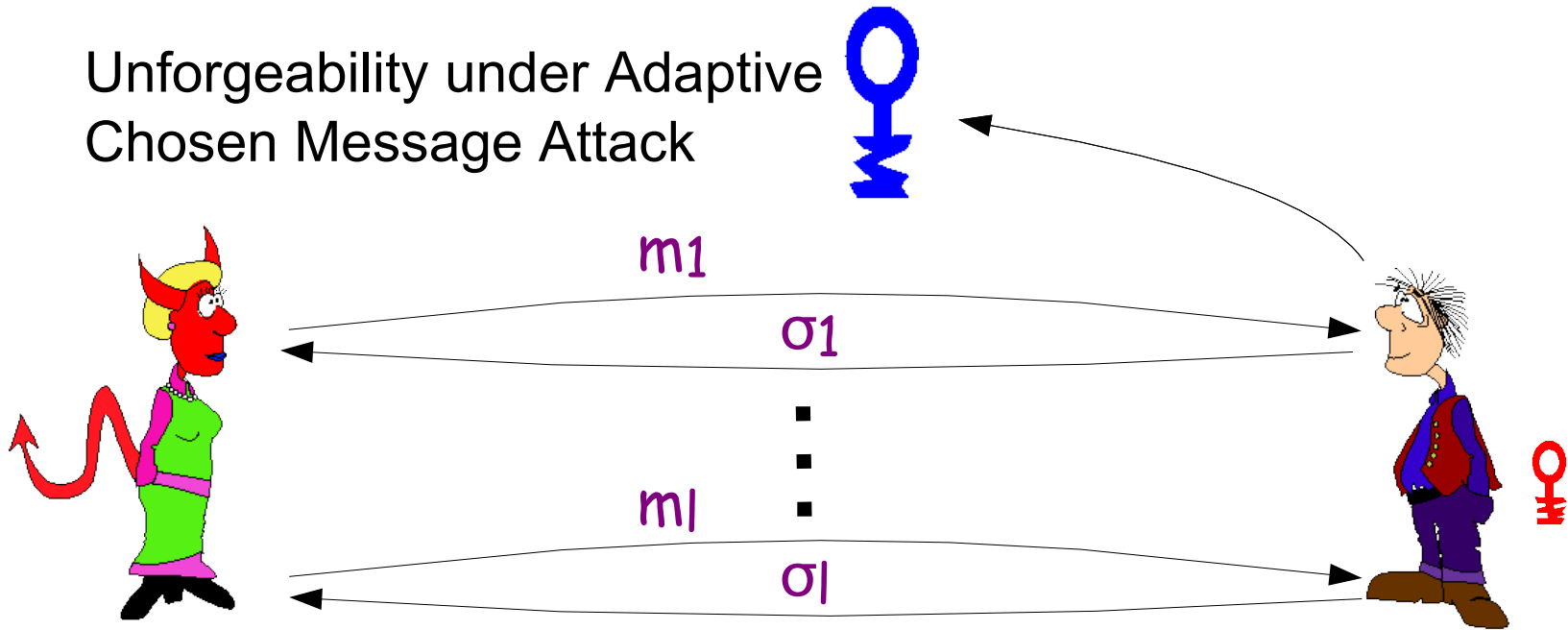
Signature Scheme: Security

Unforgeability under Adaptive Chosen Message Attack



Signature Scheme: Security

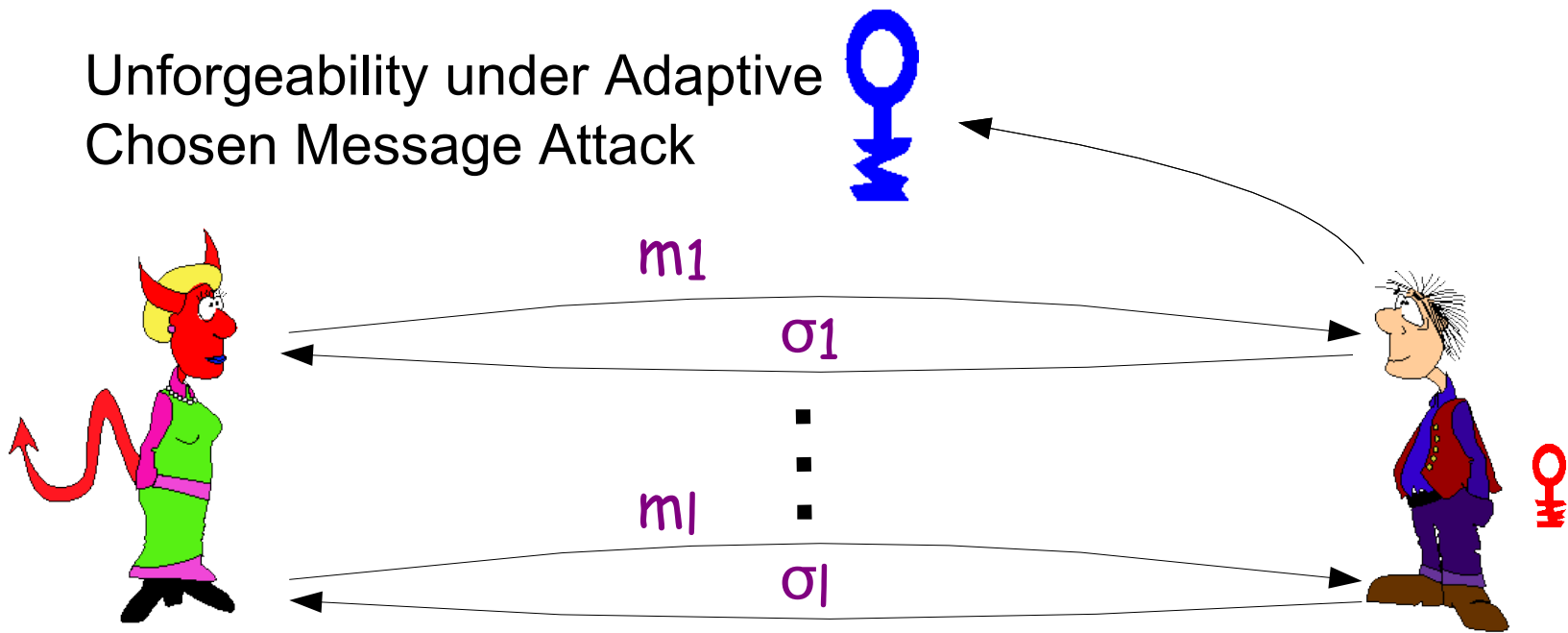
Unforgeability under Adaptive Chosen Message Attack



σ' and $m' \neq m_i$ s.t.
 $\text{ver}(\sigma', m', \text{♀}) = \text{true}$

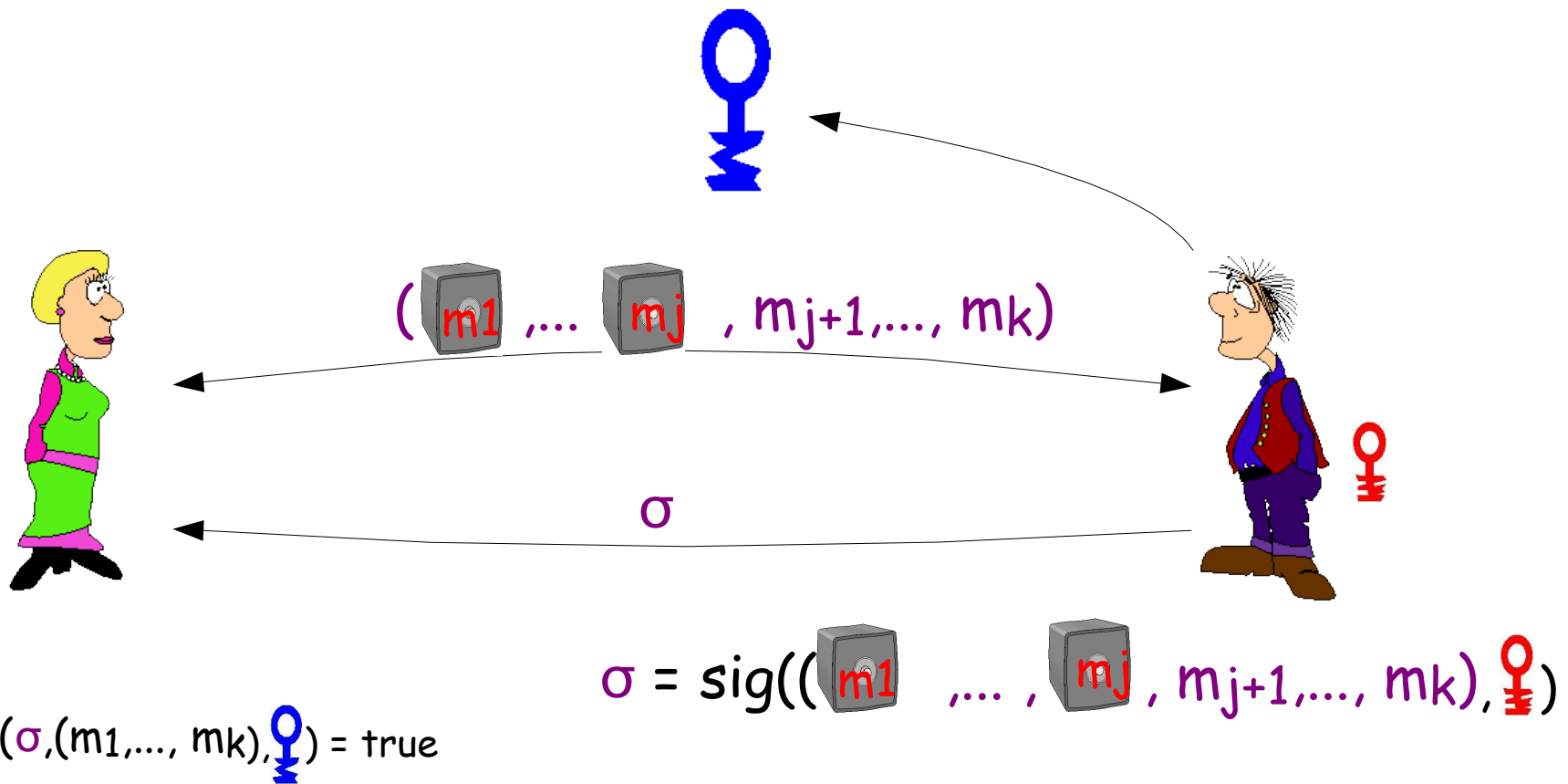
Signature Scheme: Security

Unforgeability under Adaptive Chosen Message Attack



~~m' and $m' \neq m_i$ s.t.
 $\text{Verify}(\sigma, m', \text{PK}) = \text{true}$~~

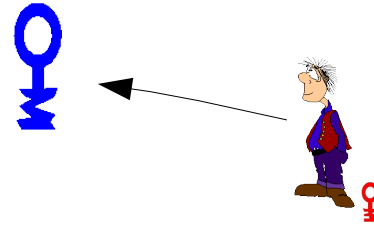
Signature Scheme: Signing Hidden Messages



Verification remains unchanged!
Security requirements basically the same

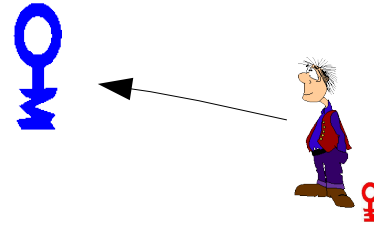
Proving Possession of a Signature

σ on (m_1, \dots, m_k)



Proving Possession of a Signature

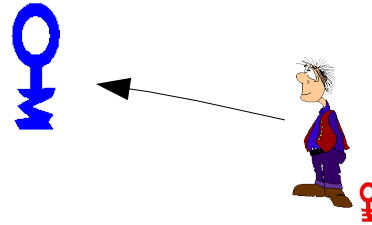
σ on (m_1, \dots, m_k)



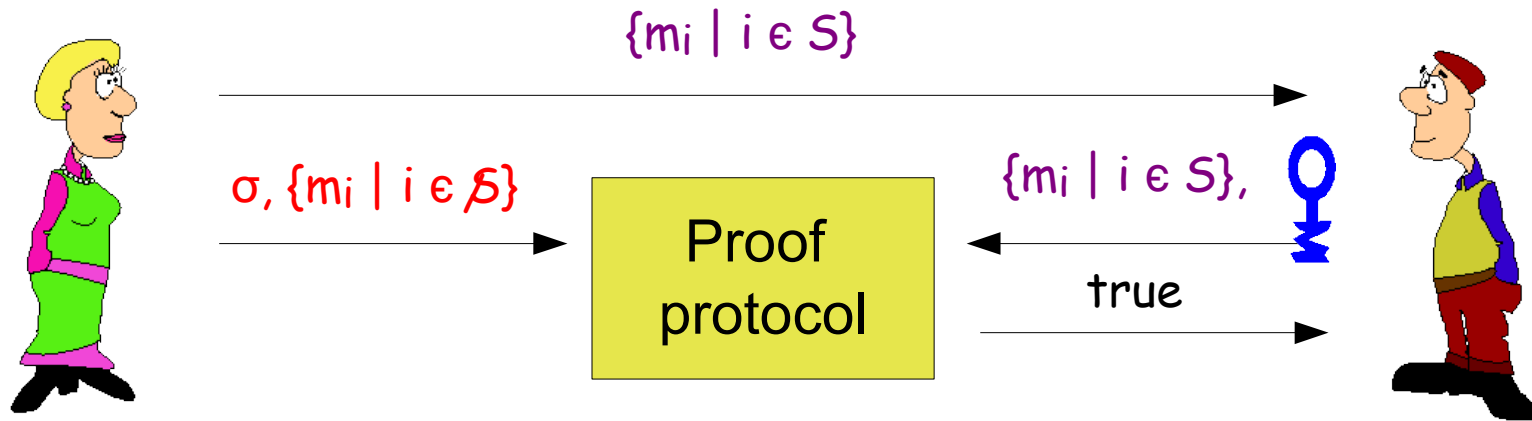
$\{m_i \mid i \in S\}$



Proving Possession of a Signature



σ on (m_1, \dots, m_k)



•Variation:

- send also $\{ \text{mi} \notin \mathcal{S} \}$ to verifier and
- prove properties about hidden m_i

Technologies

Signature Schemes: RSA
(For Reference)

RSA Signature Scheme

Rivest, Shamir, and Adleman 1978

Secret Key: two random primes p and q

Public Key: $n = pq$, prime e ,
and collision-free hash function

$$H: \{0,1\}^* \rightarrow \{0,1\}^{\ell}$$

Computing signature on a message $m \in \{0,1\}^*$

$$d = 1/e \pmod{(p-1)(q-1)}$$

$$s = H(m)^d \pmod{n}$$

Verification signature on a message $m \in \{0,1\}^*$

$$s^e = H(m) \pmod{n}$$

RSA Signature Scheme: Security

1. Hash function finding m and m' s.t. $H(m) = H(m')$

$$s^e = H(m) = H(m') \pmod{n}$$

2. Factoring $n \Rightarrow p$ and q

2'. RSA assumption:

cannot compute s given $n, e,$ and $H(m')$

- Signature scheme secure under the RSA assumption
- NOT equivalent to factoring

Technologies

Signature Schemes:

Caménisch & Lysyanskaya

The Strong RSA Assumption

Flexible RSA Problem: *Given RSA modulus n and $z \in QR_n$
find integers e and u such that*

$$u^e = z \pmod n$$


- Introduced by Barić & Pfitzmann '97 and Fujisaki & Okamoto '97
- Hard in generic algorithm model [Damgård & Koprowski '01]

Cf. RSA assumption: e is fixed there.

$$QR_n = \{ x : \text{exist } y \text{ s.t. } y^2 = x \pmod n \}$$

Signature Scheme based on SRSA

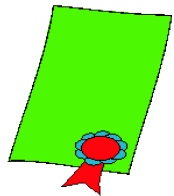
Public key of signer: RSA modulus n and $a_i, b, d \in \mathbb{Q}\mathbb{R}_n$, 

Secret key: factors of n 

To sign k messages $m_1, \dots, m_k \in \{0,1\}^\ell$:

- choose random *prime* $e > 2^\ell$ and *integer* $s \approx n$
- compute c :

$$c = (d / (a_1^{m_1} \cdot \dots \cdot a_k^{m_k} b^s))^{1/e} \text{ mod } n$$

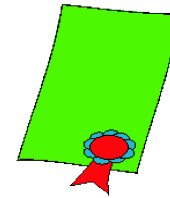


- signature is (c, e, s)

Signature Scheme based on SRSA II

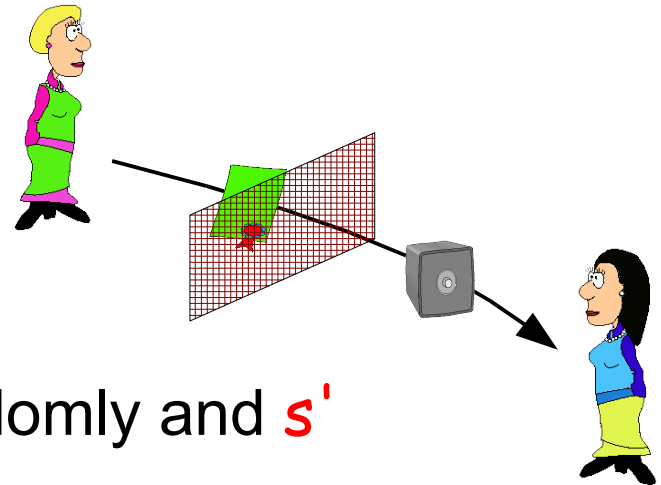
To verify a signature (c, e, s) a messages m_1, \dots, m_k :

- $m_1, \dots, m_k \in \{0,1\}^\ell$:
- $e > 2^\ell$
- $d = c^e a_1^{m_1} \dots a_k^{m_k} b^s \pmod n$



Theorem: *Signature scheme is secure against adaptively chosen message attacks under SRSA assumption.*

Proving Knowledge of a Signature



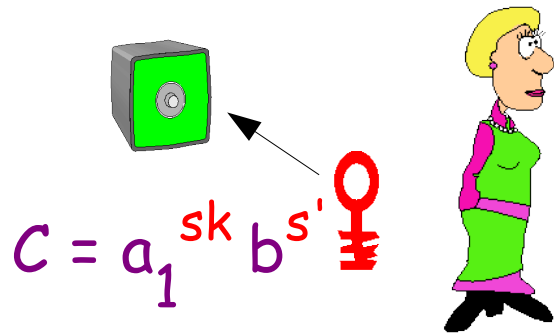
Observe:

- Let $c' = c b^{s'}$ mod n with randomly and s'
- then $d = c'^e a^m b^{s-es'}$ (mod n), i.e.,
($c', e, s^* = s-es'$) is also signature on m

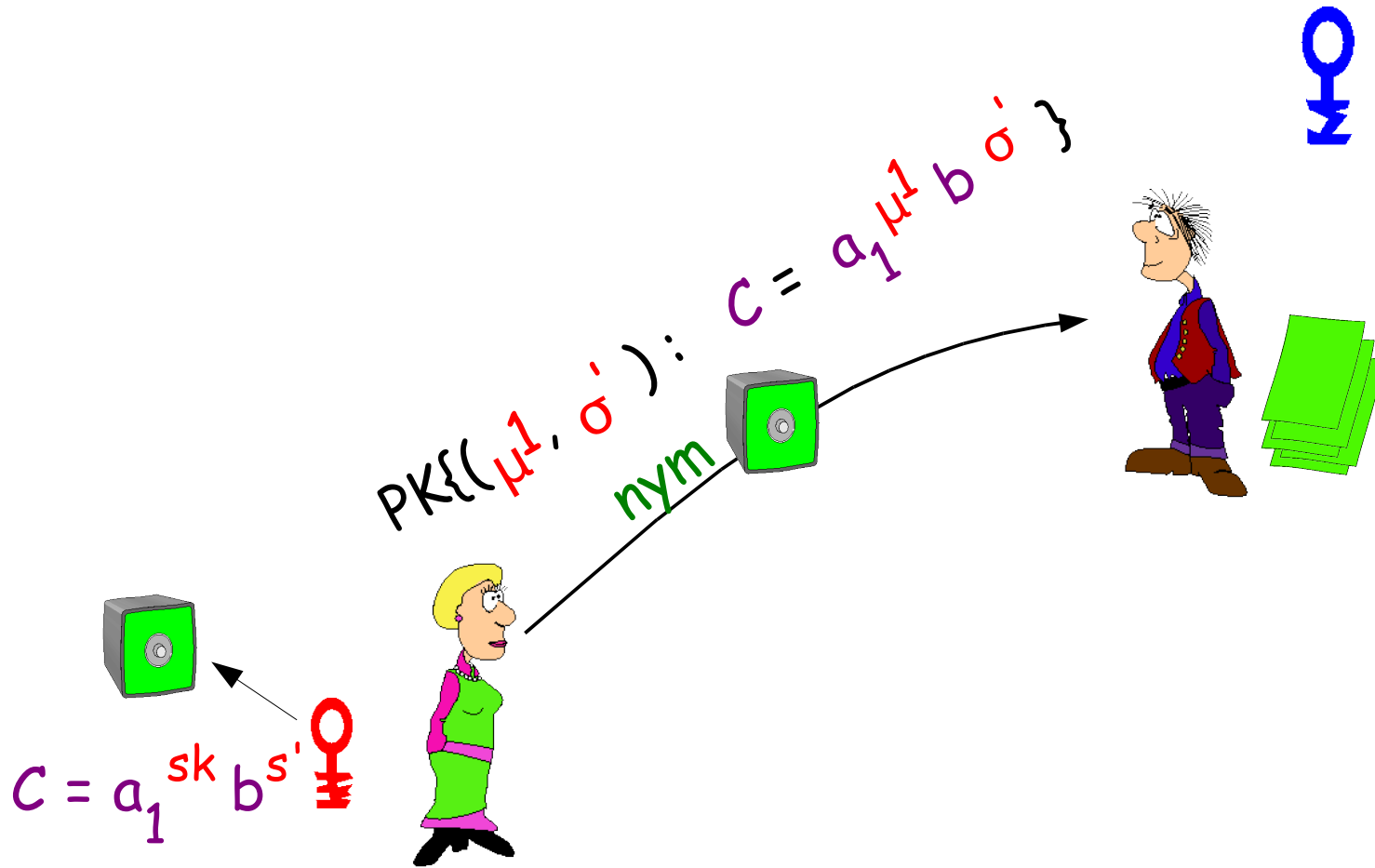
To prove knowledge of signature (c', e, s^*) on some m

- provide c'
- $PK\{(\varepsilon, \mu, \sigma) : d := c'^\varepsilon a^\mu b^\sigma\}$

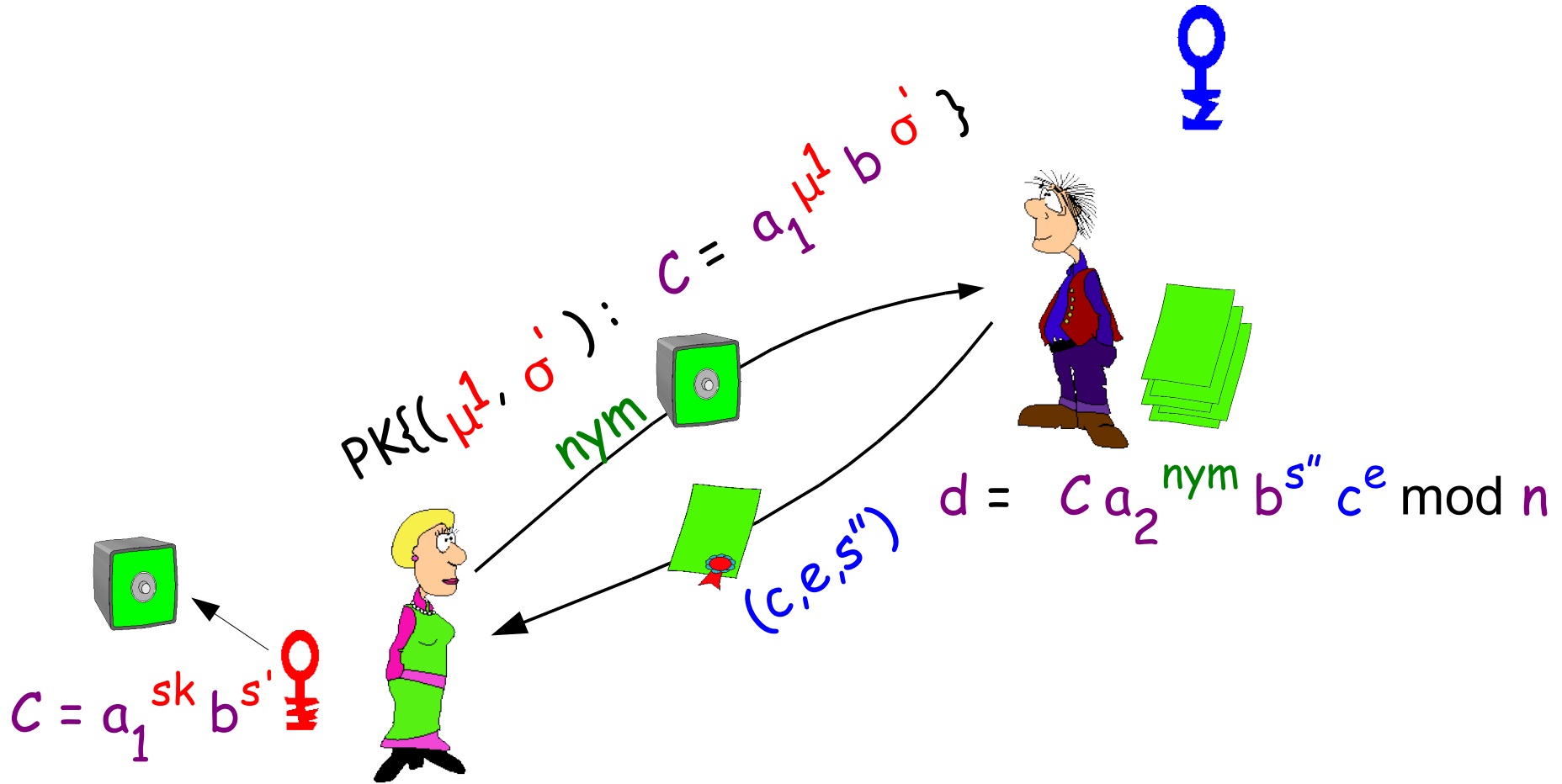
Getting a Signature on a Secret Message



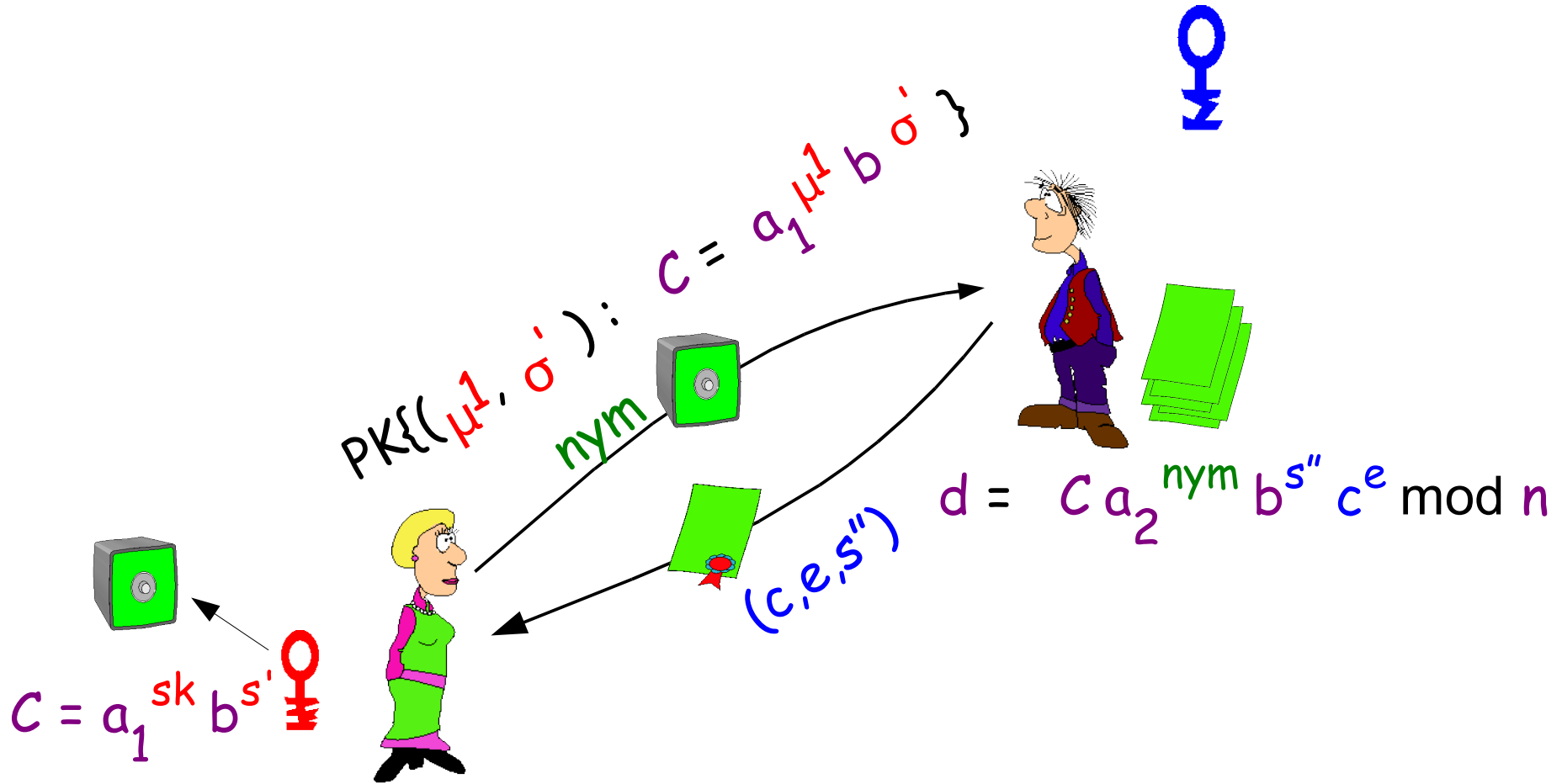
Getting a Signature on a Secret Message



Getting a Signature on a Secret Message



Getting a Signature on a Secret Message

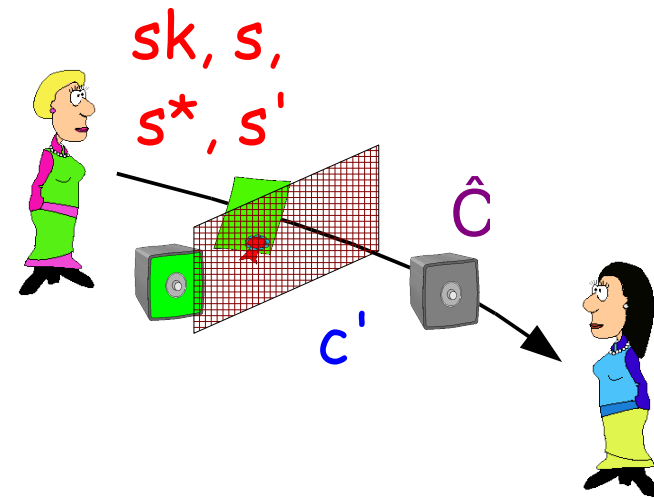


$$d = c^e a_1^{sk} a_2^{nym} b^{s'' + s'} \pmod n$$

Integrated Proof of Knowledge of a Signature

Let $\hat{a}_1, \dots, \hat{a}_k, \hat{b}, \hat{n}$ be the PK of verifier.

Let $\hat{C} = \hat{a}_1^{sk} \hat{b}^{s'}$ be a commitment on sk .

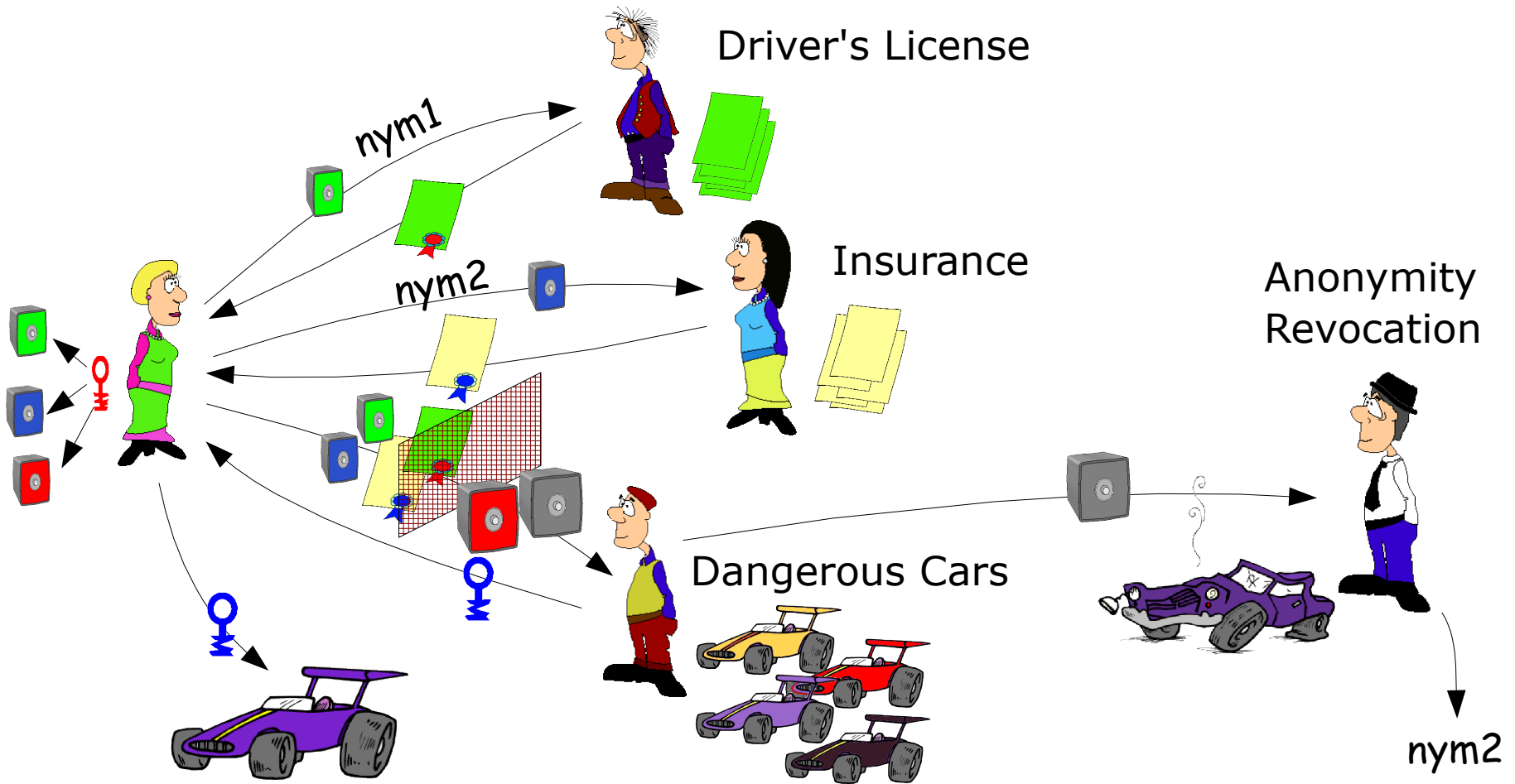


Prover computes $c' = c b^{s^*} \bmod n$ with randomly s^* , releases c' , and executes

$$\text{PK}\{(\varepsilon, \rho, \mu, \sigma, \sigma') : d := c'^{\varepsilon} a^{\mu} b^{\sigma} \bmod n \\ \wedge \hat{C} = \hat{a}_1^{\mu} \hat{b}^{\sigma'} \bmod \hat{n} \}$$

with the verifier.

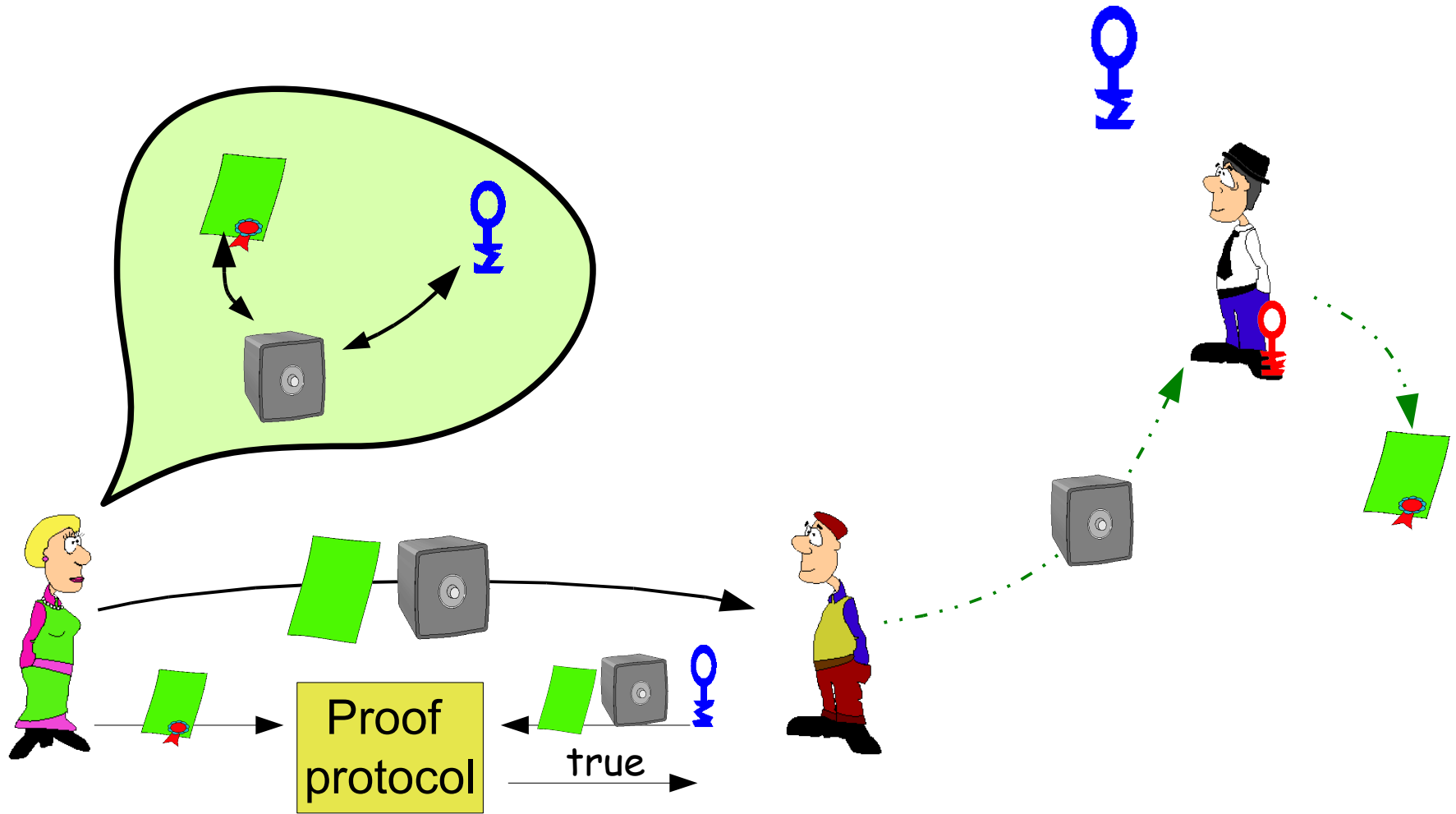
Everything Put Together



Technologies

Verifiable Encryption

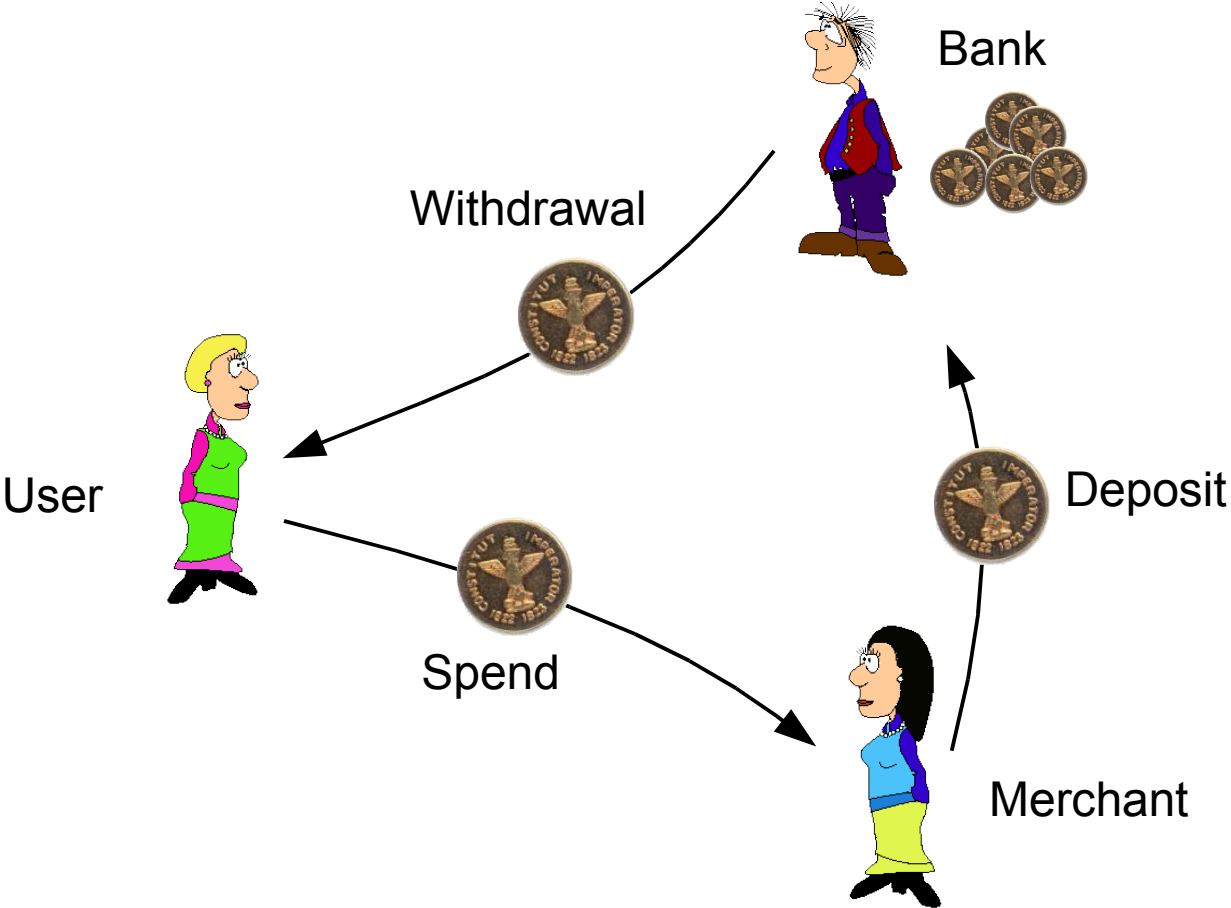
Verifiable Encryption



Application

E-cash (On-Line)

E-Cash Setting



E-Cash Setting

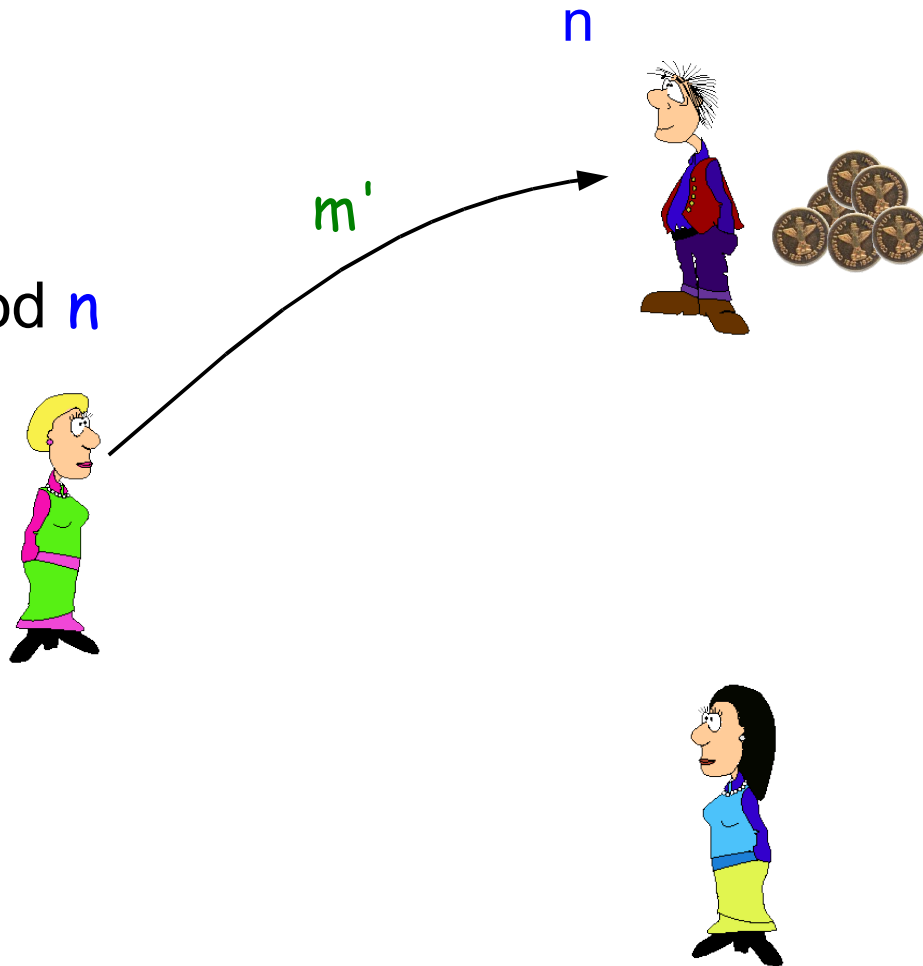
- Anonymity:
Withdrawal and Deposit must be unlinkable
- Double Spending:
Coin is bit-strings, can be spend twice
 - Bank *on-line*: merchant checks with bank whether some coin has previously been spend before accepting
 - Bank *off-line*: merchant accepts, get credited, but bank can reveal anonymity of double (multiple) spenders
 - Need to have scheme where spending once is anonymous
 - Spending more than once is not anonymous

E-cash with (Blind) RSA Signatures

[Chaum 82]

random m and b

$$m' = H(m) b^e \text{ mod } n$$



E-cash with (Blind) RSA Signatures

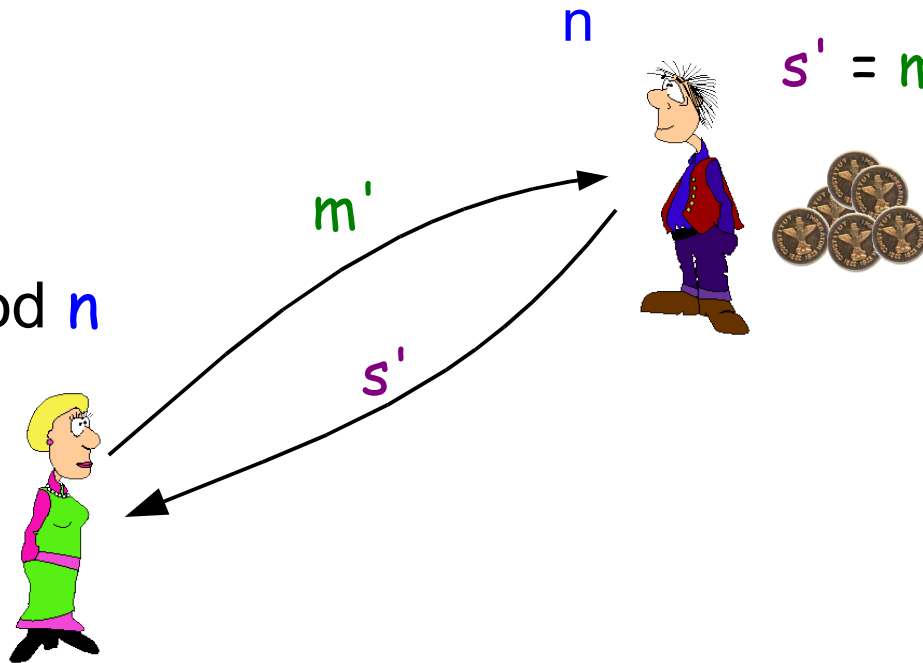
[Chaum 82]

$$d = 1/e \text{ mod } (p-1)(q-1)$$

$$s' = m' d \text{ mod } n$$

random m and b

$$m' = H(m) b^e \text{ mod } n$$



$$s = s' / b \text{ mod } n$$

$$\Rightarrow s^e = H(m) \text{ mod } n$$

E-cash with (Blind) RSA Signatures

[Chaum 82]

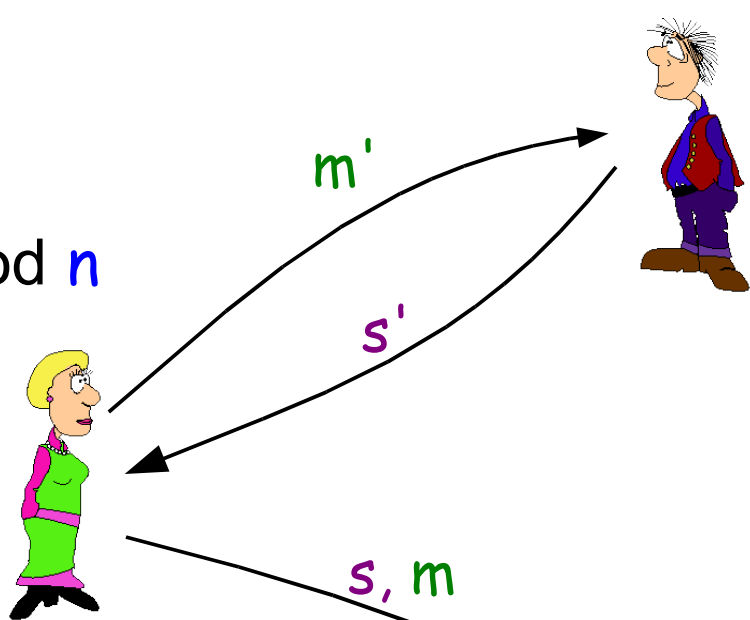
$$d = 1/e \text{ mod } (p-1)(q-1)$$

$$s' = m' d \text{ mod } n$$

random m and b

$$m' = H(m) b^e \text{ mod } n$$

n



$$s = s' / b \text{ mod } n$$

$$\Rightarrow s^e = H(m) \text{ mod } n$$

$$s^e = H(m) \text{ mod } n ?$$

E-cash with (Blind) RSA Signatures

[Chaum 82]

$$d = 1/e \text{ mod } (p-1)(q-1)$$

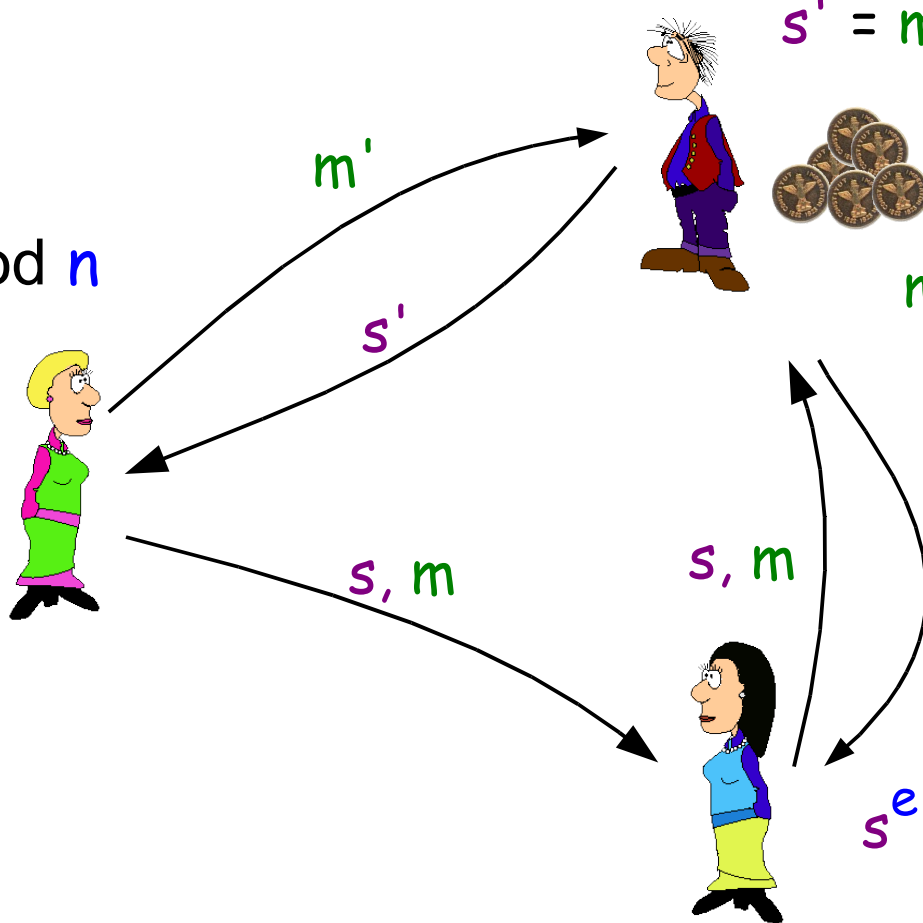
$$s' = m' d \text{ mod } n$$

random m and b

$$m' = H(m) b^e \text{ mod } n$$

$$s = s' / b \text{ mod } n$$

n



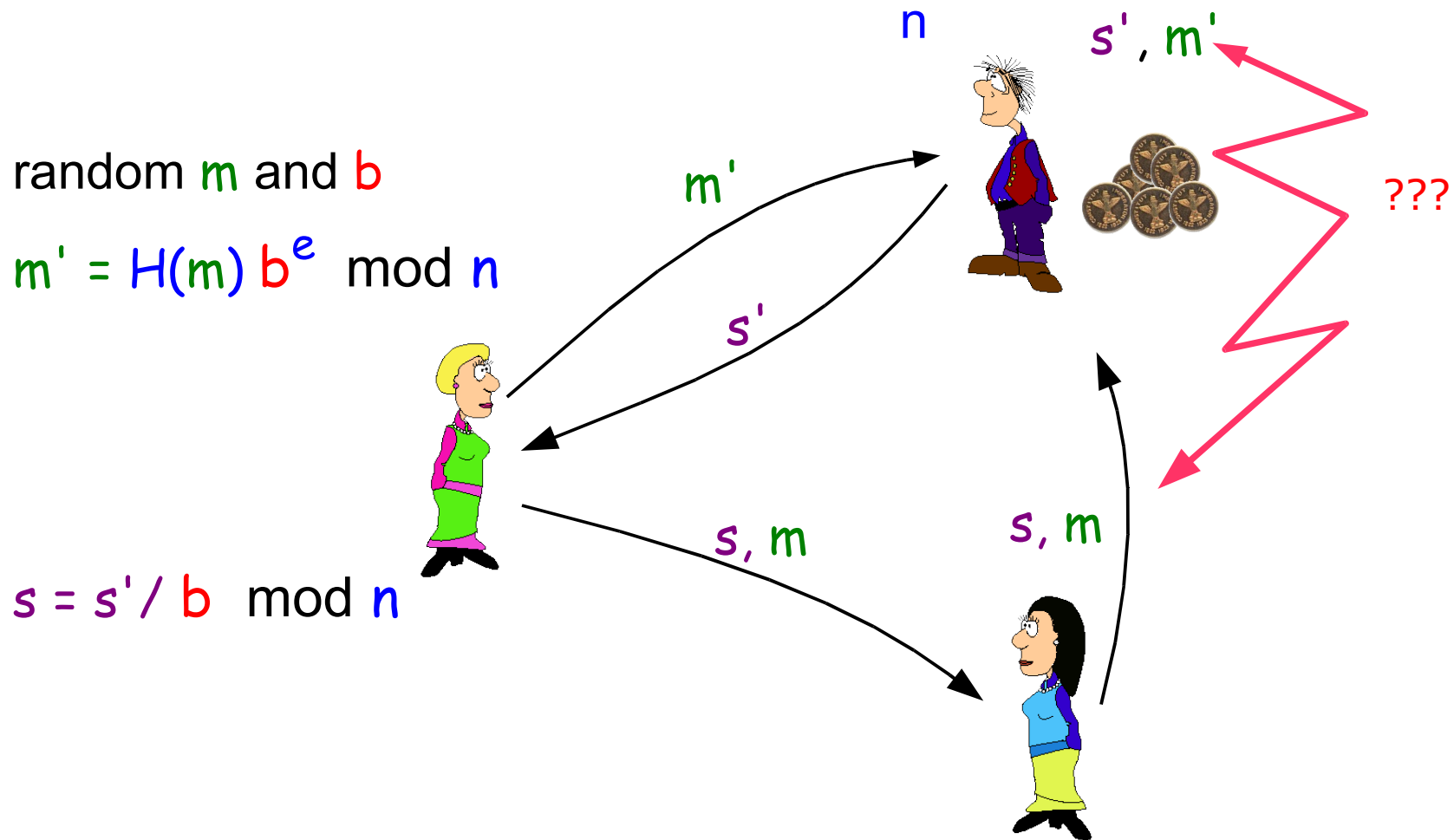
$m \in L?$

OK/ not OK

$$s^e = H(m) \text{ mod } n ?$$

Blindness of Blind RSA Signatures

[Chaum 82]



Blindness of Blind RSA Signatures

[Chaum 82]

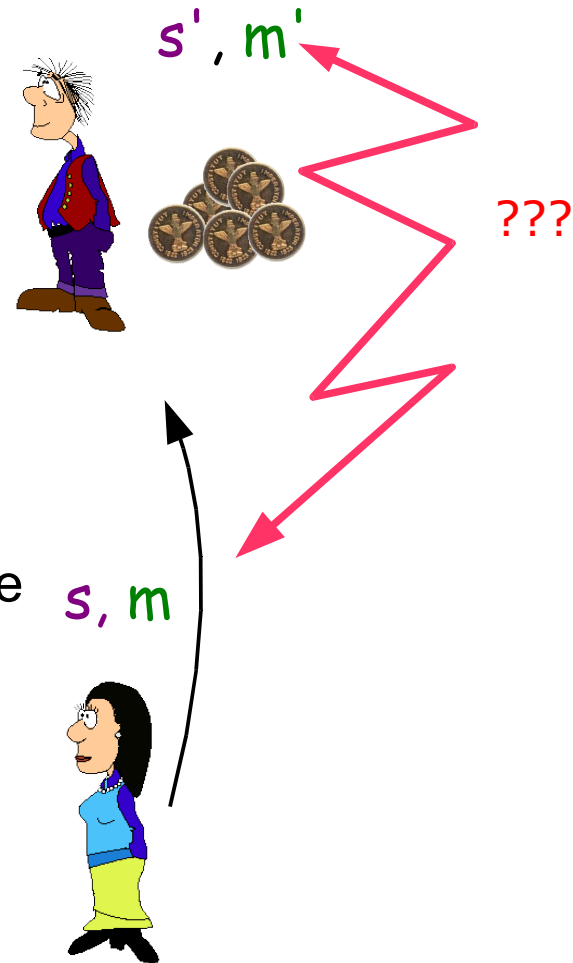
Given any two pairs (s, m) and (s', m') .

Let $b = s'/s \pmod n$

Then:

$$\begin{aligned} m' &= s'^e = (s b)^e = s^e b^e = \\ &= H(m) b^e \pmod n \end{aligned}$$

Therefore the pairs could stem from the same user! Thus the two transactions are perfectly unlinkeable and signature scheme is perfectly blind.



Security of Blind RSA Signatures

[Michels, Stadler, Sun 98]

Problem:

Signer computes e -th root of anything!

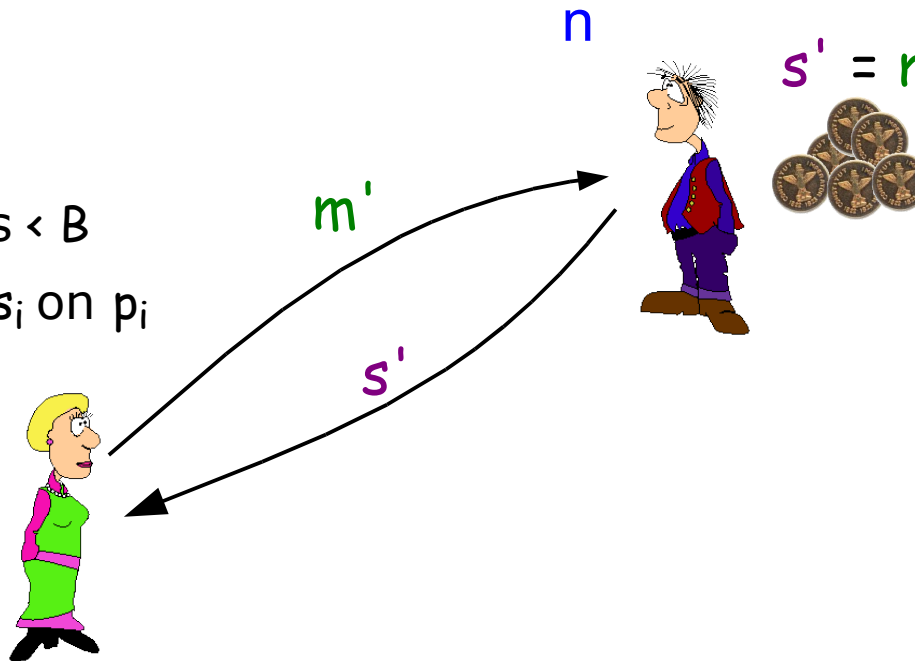
$$d = 1/e \pmod{(p-1)(q-1)}$$

$$s' = m'^d \pmod{n}$$

Attack:

Let p_i be primes $< B$

get signatures s_i on p_i





find message m and set S s.t. $H(m) = \prod_{i \in S} p_i$ (i.e., factor $H(m)$)

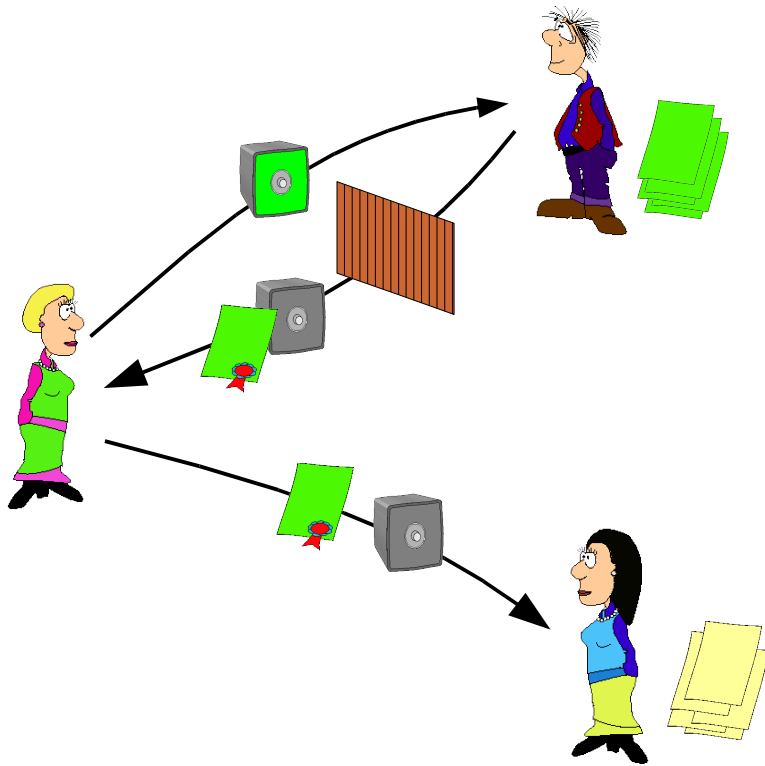
then for $s = \prod_{i \in S} s_i$ we have

$$s^e = \prod_{i \in S} s_i^e = \prod_{i \in S} p_i = H(m) \pmod{n}.$$

E-cash with (Blind) RSA Signatures

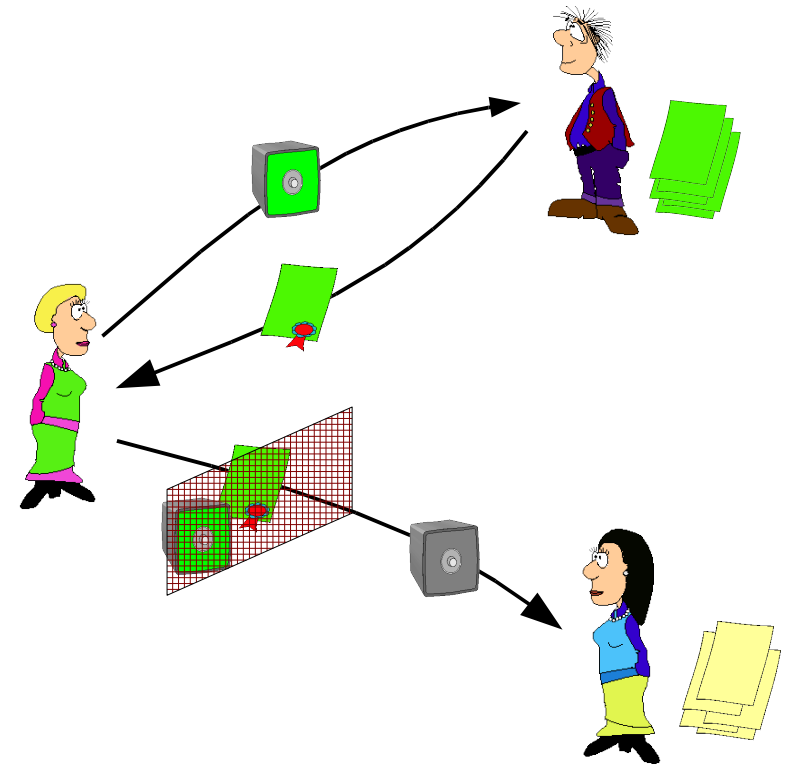
- Attack can be overcome by requiring $H': \{0,1\}^* \rightarrow [0,n-1]$ instead of $H: \{0,1\}^* \rightarrow \{0,1\}^\ell$
Called full-domain hash, e.g., can be obtained padding
$$H'(m) = H(1||m) || H(2||m) || H(3||m) || \dots || H(k||m)$$
- Security has been proved (in a somewhat weak model) [Bellare et al. 03]
- Can be used for credential scheme:
 - Credential: blind signature s on message $m =$ 
 - Showing credential:
 - reveal s and m
 - prove knowledge of secret contained in 
 - But credential can be used only once! :-)

Using Blind Signatures vs. Proving Possession



Certificates can be used only *once*!

- Chaum 82
- Brands 90'ies



Certificates can be used *multiple* times!

- Camenisch, Lysyanskaya 2000

Proving Possession of Signature

In principle:
Prove of Knowledge of
message m and signature s s.t.

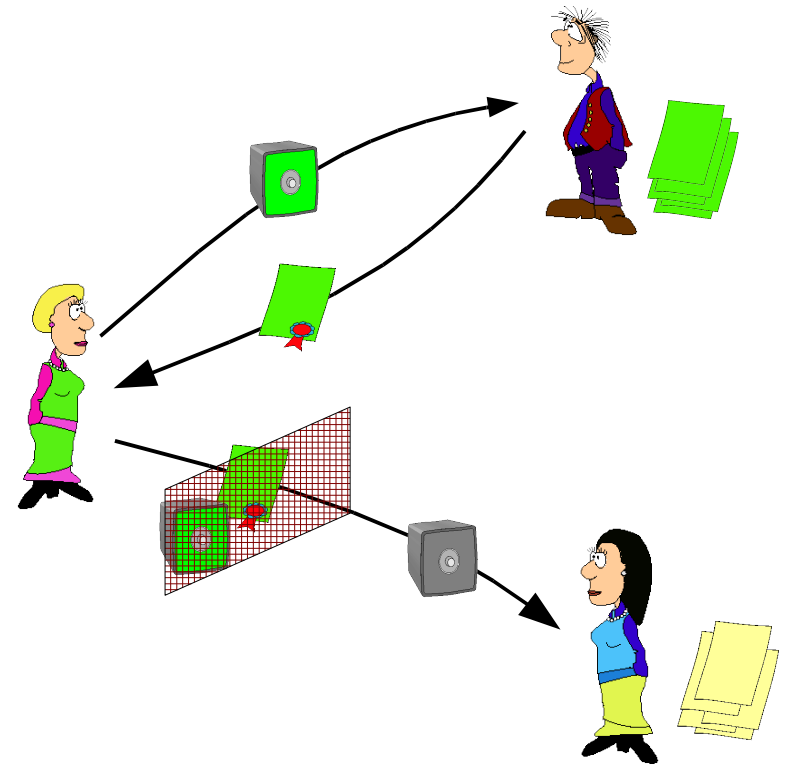
$$\text{ver}(s, m, PK_{\text{signer}}) = 1$$

In the case of RSA:

$$s^e = H(m) \pmod n$$

which, however, would not be efficient
because of hash function, i.e., proof of
knowledge of pre-image under hash-
function -> Requires proof over
representation of hash-function as gates.

=> need other signature scheme!



Certificates can be used *multiple* times!

- Camenisch, Lysyanskaya 2000

References

- M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko: *The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme*. Journal of Cryptology, Volume 16, Number 3. Pages 185 -215, Springer-Verlag, 2003.
- E. Bangerter, J. Camenisch and A. Lyskanskaya: A Cryptographic Framework for the Controlled Release Of Certified Data. In Twelfth International Workshop on Security Protocols 2004. www.zurich.ibm.com/~jca/publications
- Stefan Brands: Untraceable Off-line Cash in Wallets With Observers: In Advances in Cryptology – CRYPTO '93. Springer Verlag, 1993.
- J. Camenisch and A. Lyskanskaya: Efficient Non-transferable Anonymous Multi-show Credential System with Optional Anonymity Revocation. www.zurich.ibm.com/~jca/publications
- David Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In Communications of the ACM, Vol. 24 No. 2, pp. 84–88, 1981.
- David Chaum: Blind Signatures for Untraceable Payments. In Advances in Cryptology – Proceedings of CRYPTO '82, 1983.
- David Chaum: Security Without Identification: Transaction Systems to Make Big Brother obsolete: in Communications of the ACM, Vol. 28 No. 10, 1985.