Anonymous Credentials

Leonardo A. Martucci

PriSec
Privacy & Security

* part of this set is based on J. Camenisch slides (IBM Zurich)

In this Session

PriSec
Privacy & Security

- Anonymous credentials
  - Building Blocks
  - **WHAT** are they?
  - **WHY** do we need them?
  - **HOW** to build them?

- Source*:

  Jan Camenisch, Gregory Neven and Anja Lehmann
  IBM Research - Zurich

* the slides are not the original: they were redesigned for this presentation

Cryptographic Building Blocks

- Blinding Signatures
- Zero-knowledge proofs
- Commitments

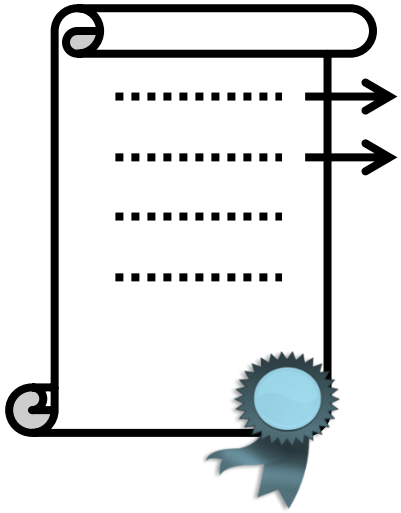
Today

PriSec
Privacy & Security

- Credentials and anonymous credentials
    - **WHAT** are they?
    - **WHY** do we need them?
    - **HOW** to build them?
    - Their building blocks
    - How can you use them

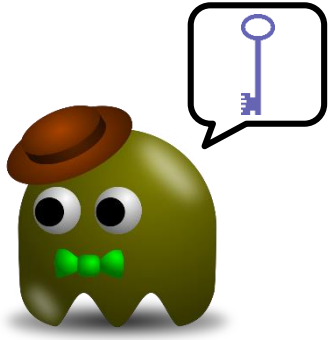
Credentials and Certificates

- A signed list of attribute-value pairs

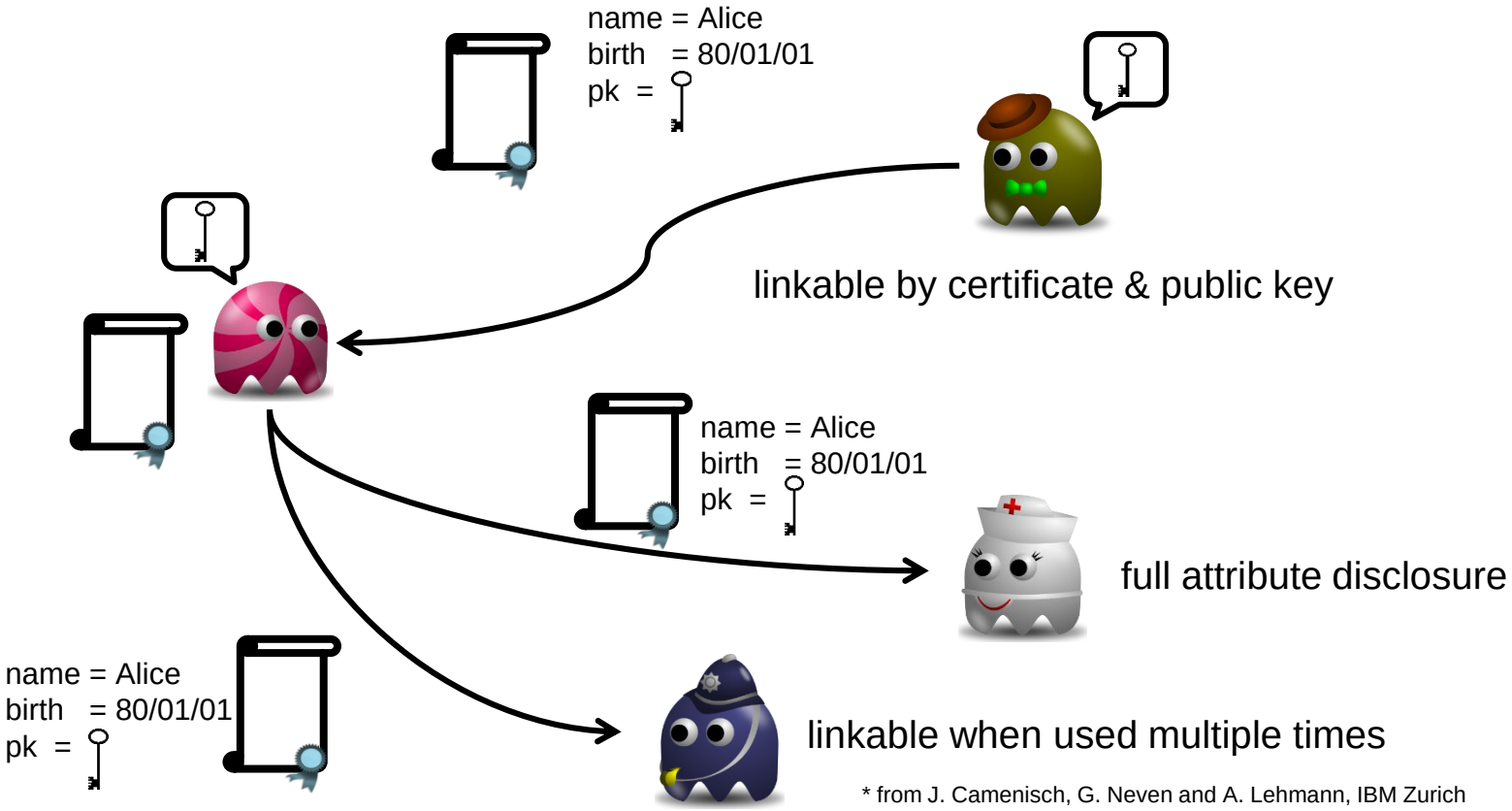


- name = Alice
- birth date = 1990 / 01 / 01

- signed by the issuer



Problems with X.509 Public-Key Certificates



* from J. Camenisch, G. Neven and A. Lehmann, IBM Zurich

Anonymous Credentials

- **ARE** privacy-enhancing attribute based credentials
  (privacy-ABCs)
  ≈ minimal disclosure token

a sort of identification
(similar to a certificate)

- There are different ways to build them
  e.g. Identify Mixer, U-Prove

Anonymous Credentials: Basic Functionality

- Protects the users privacy
  - Anonymity
  - Unlinkability (multi-use)
  - Selective disclosure

evil siblings

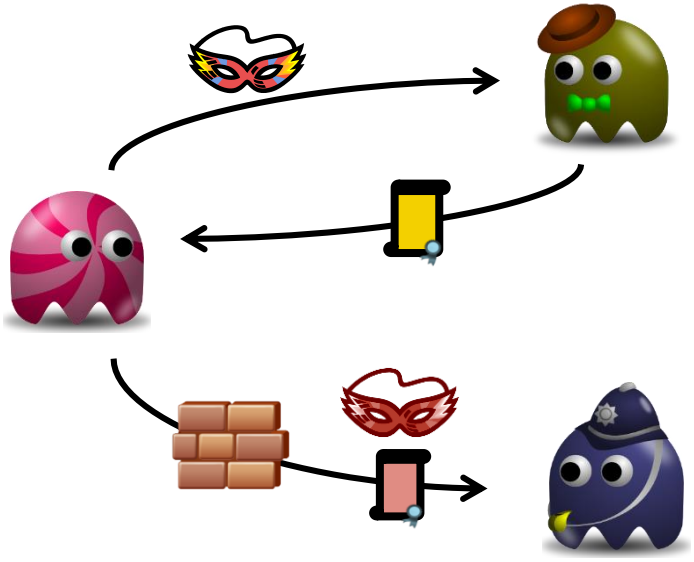
evil Alice

Unforgeability of credentials

Consistency of credentials (no sharing)

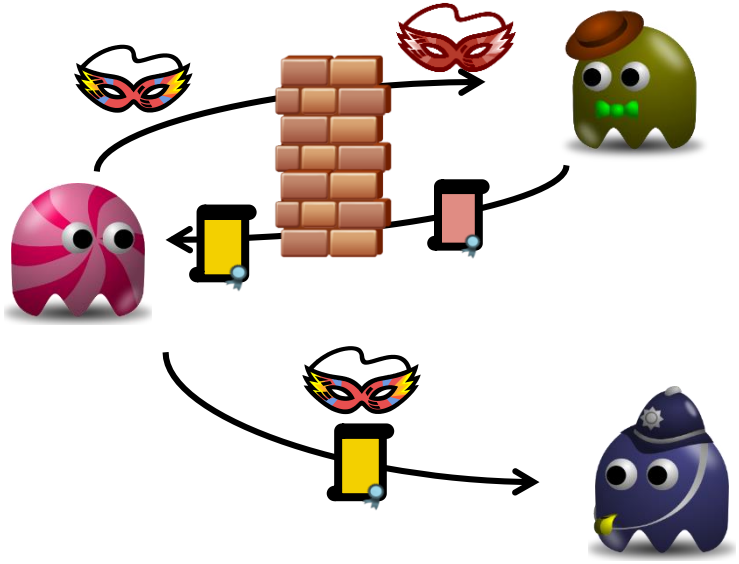
The two-ways to build them

Zero-Knowledge Proofs



Identity Mixer

Blind Signatures



U-Prove

* from J. Camenisch, G. Neven and A. Lehmann, IBM Zurich

Blind Signatures and Zero-Knowledge Proofs

- Blind Signatures
  - The problem:

    How can a Signer sign something without seeing it ?

- Zero-Knowledge Proofs
  - The problem:

    How to prove that I know the answer to a problem

    without telling the answer ?

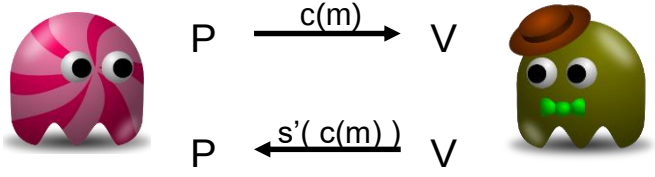
Blind Signatures

- Enables a signer to sign a message where the content has been blinded (i.e. hidden) prior to the signature

- A verifier that is:
  - *The signer* can verify the signature
    BUT
    not link it to the message it signed earlier
  - *A third party* can also verify the signature

Blind Signatures

- Given the functions
  - s(x) (public) and (the inverse of s) s'(x) (secret) Signer V
  - c(x) (public) and (the inverse of c) c'(x) (secret) Prover P



c' (s'(c(m))) = s'(m) ➡ Blinded Signature!

Blind Signatures: RSA

- signature $s = m^d \ mod \ n$
- public key (e, n) and $n = p \, . \, q$ (large primes)
- private key (d, n)

$$\varphi(n) = (p-1)(q-1)$$
$$d \ = \ e^{-1} \ mod \ \varphi(n)$$

- P calculates $m' = m \, . \, r^e \ mod \ n$

blinding factor

P —m'→ V

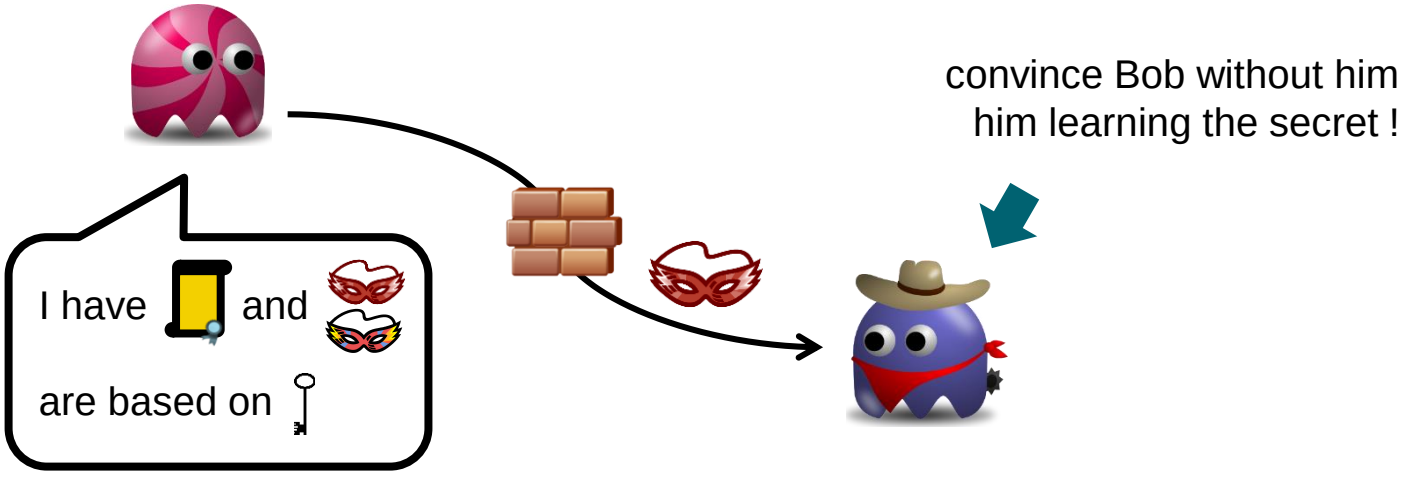
P ←s'— V where $s' = (m')^d \ mod \ n$

$$s' = (m \, . \, r^e)^d \ mod \ n = m^d \ r^{e.d} \ mod \ n = m^d \ r \ mod \ n$$
$$s = s' . r^{-1} = m^d \ mod \ n$$

⇒ Blinded Signature

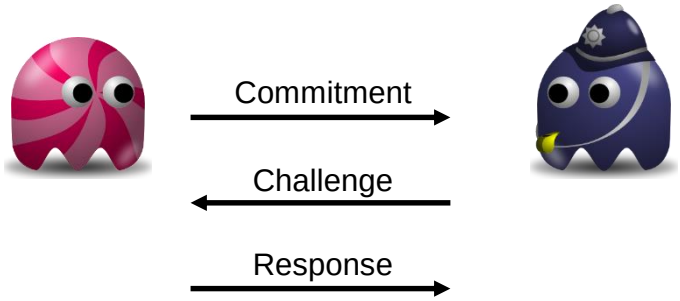
Zero-Knowledge Proofs

- A provider wants to convince a verifier that she knows a secret *without* revealing the secret!



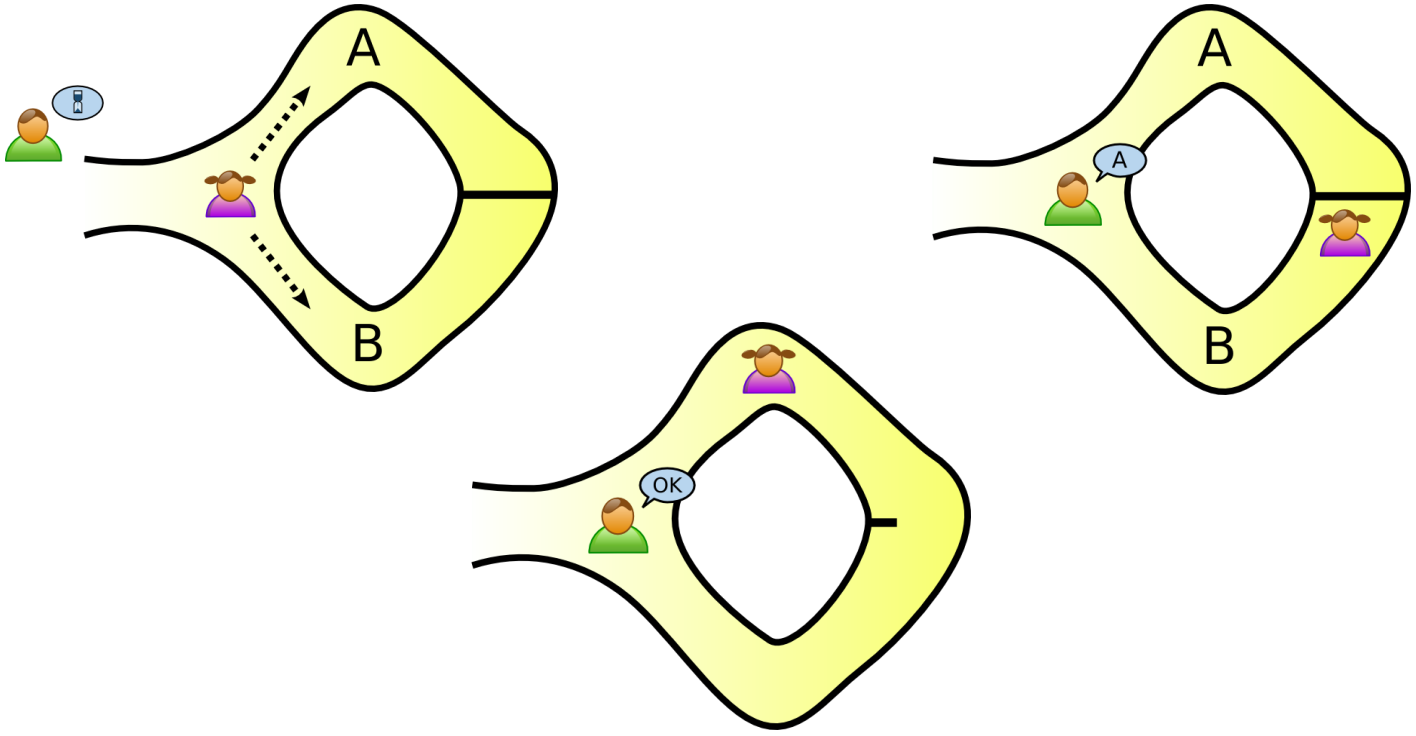
Zero-Knowledge Proofs

- Interactive proof



- Properties:
  - Zero-knowledge — verifier learns nothing about the prover's secret
  - Soundness — prover can convince verifier only if she knows the secret
  - Completeness — if prover knows the secret she can always convince the verifier

Zero-Knowledge Proofs in Pictures



* Quisquater, 1990

Zero-Knowledge Proofs with Discrete Logs

- Given group G and element $y \epsilon G$
  - prove knowledge of $x = \log_g y$ such that verifier only learns g and y

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

(x, g, y)
Prover

(g, y)
Verifier

random r

$t := g^r$

t →

← c

random c

$s := r - x.c$

s →

accept if $t := g^s y^c$

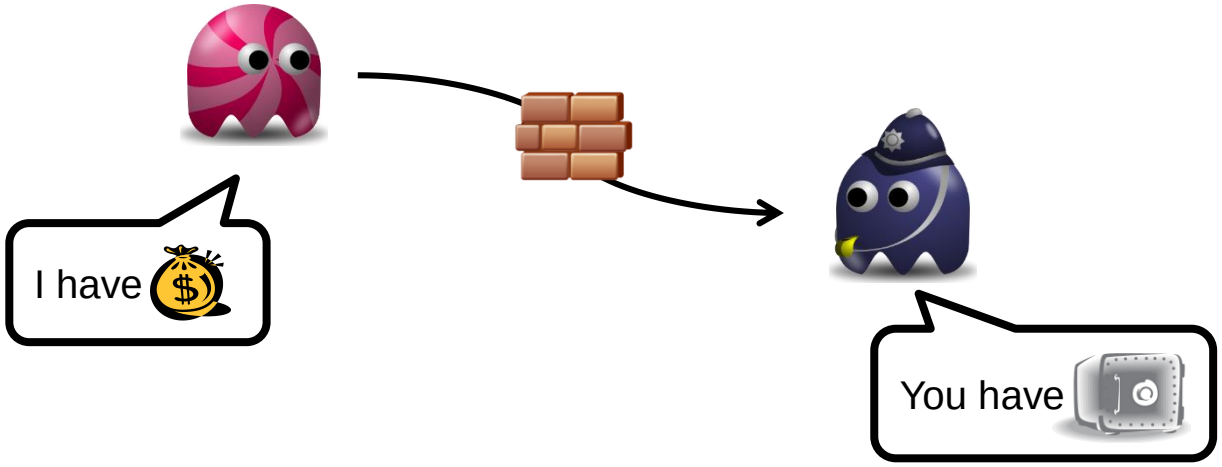
Proof of completeness: $g^s y^c = g^{r-x.c} y^c = g^{r-x.c} g^{x.c} = g^r = t$

* Schnorr, 1991

Commitment Schemes

- The problem:

  How to prove that you hold (or selected) a value, and
  commit to it (so it can be proved later), without showing it ?



Commitment Schemes

- Allows a sender to commit to a value (the commitment) towards a receiver, without revealing the value
    - the sender can later reveal (open) the value to the receiver

$$c \leftarrow Commit(\, m \,)$$

$$\{true, false\} \leftarrow Open(\, c, m \,)$$

- The commitment has to be binding
- The analogy: coin flipping over the telephone

Commitment Schemes
A coin flipping over the telephone

- Bob flips the coin and tells the result to Alice
  - we use bits instead

Not fair!

- Alice picks a random bit a and sends it to Bob and Bob picks a random bit b and sends it to Alice
  - the value of the coin is $= a \oplus b$

Who goes first?

- Alice commits to her bit $c \leftarrow Commit(\ a\ )$
  Bob sends b to Alice
  Alice sends a to Bob and Bob opens the commitment
  $\{true, false\} \leftarrow Open(\ c, a\ )$

Pedersen Commitment Scheme

- A scheme that offers
  - unconditional hiding → does not leak information about m
  - computational binding

    given c, m, r it is hard to compute $m' \neq m$ and $r' \neq r$ such that

$$true \leftarrow Open(\, c, m', r')$$

  - homomorphic function

$$Commit(\, a, r\,) \,.\, Commit(\, b, s\,) = Commit\,(\, a + b, r + s\,)$$

  - To commit m, pick random r and compute $c = g^m h^r$

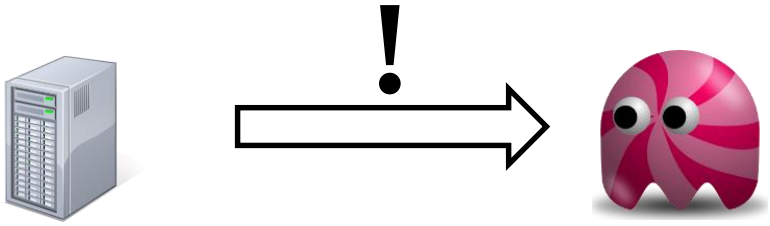
    and for opening the commitment, reveal (m, r)

Oblivious Transfer

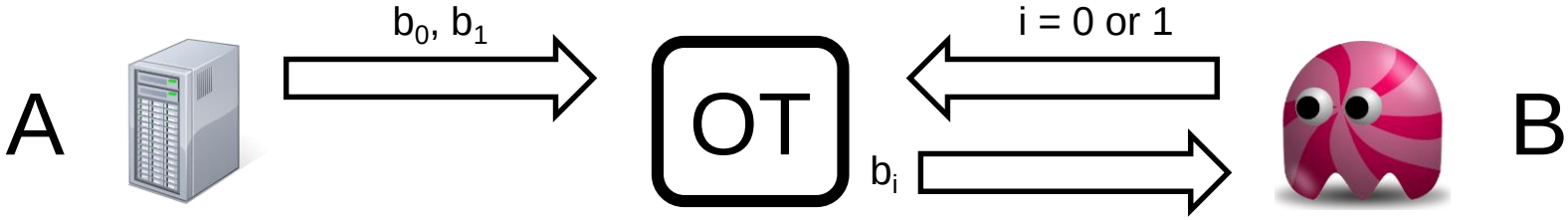
- The problem:

  How to transfer information to a receiver,
  and not know what information was transferred?

  (i.e. to protect the receiver's privacy)



Oblivious Transfer: the basic idea



- A inputs 2 bits and B inputs the index of one of A's bits
- B learns his chosen bit, A learns nothing

  A does not learn which bit B has chosen

  B does not learn the value of the bit that he did not choose

- Generalizes to bitstrings (n instead of 2)

* from on Vitaly Shmatikov

Oblivious Transfer
the 1-2 OT protocol

A

2 messages m_0, m_1

x_0, x_1 (d, e, N)

B

random x_0, x_1

x_0, x_1 →

picks b $(x_0 \, or \, x_1)$,
random k, blinds x_b
$v = (x_b + k^e)$

← v

doesn't know b
$k_0 = (v - x_0)^d$ and
$k_1 = (v - x_1)^d$.

$m'_0 = m_0 + k_0$
$m'_1 = m_1 + k_1$ →

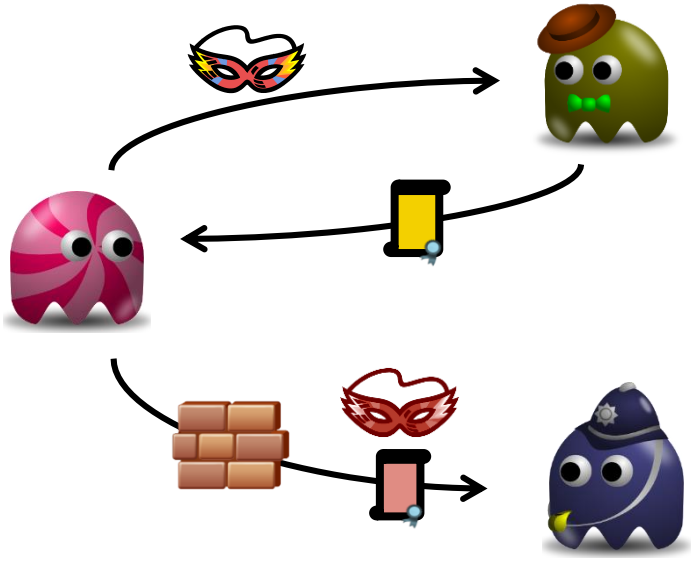
$m_b = m'_b - k$

A does not know which message B could unblind

and now back to Anonymous Credentials

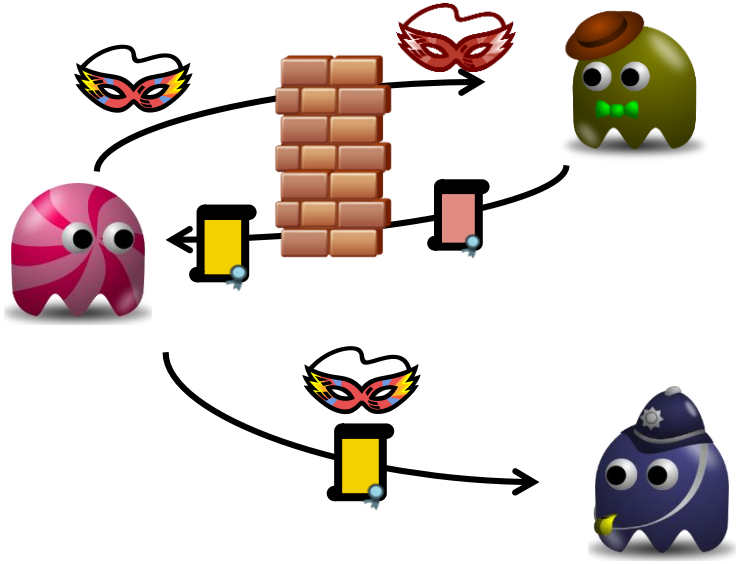
The two-ways to build them

Zero-Knowledge Proofs



Identity Mixer

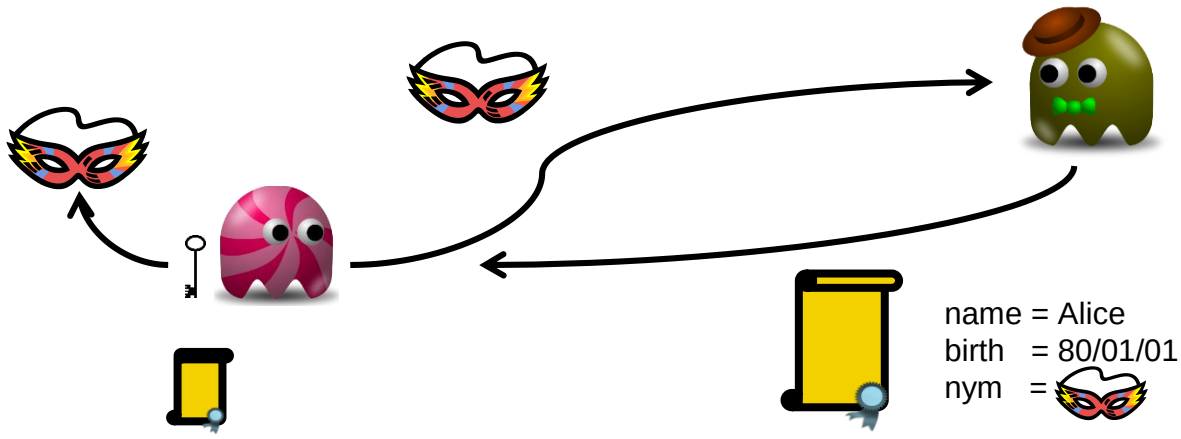
Blind Signatures



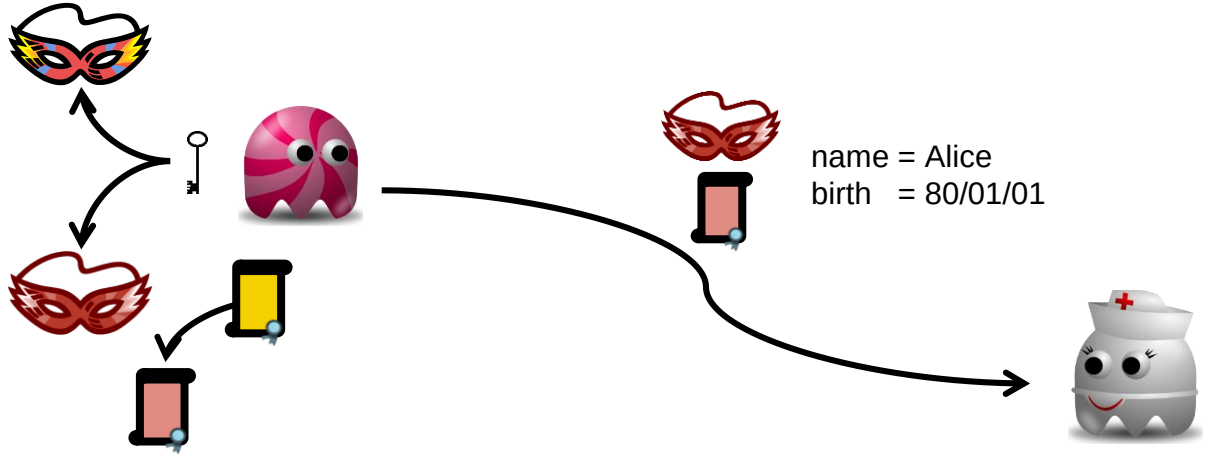
U-Prove

* from J. Camenisch, G. Neven and A. Lehmann, IBM Zurich

Anonymous Credentials Generation

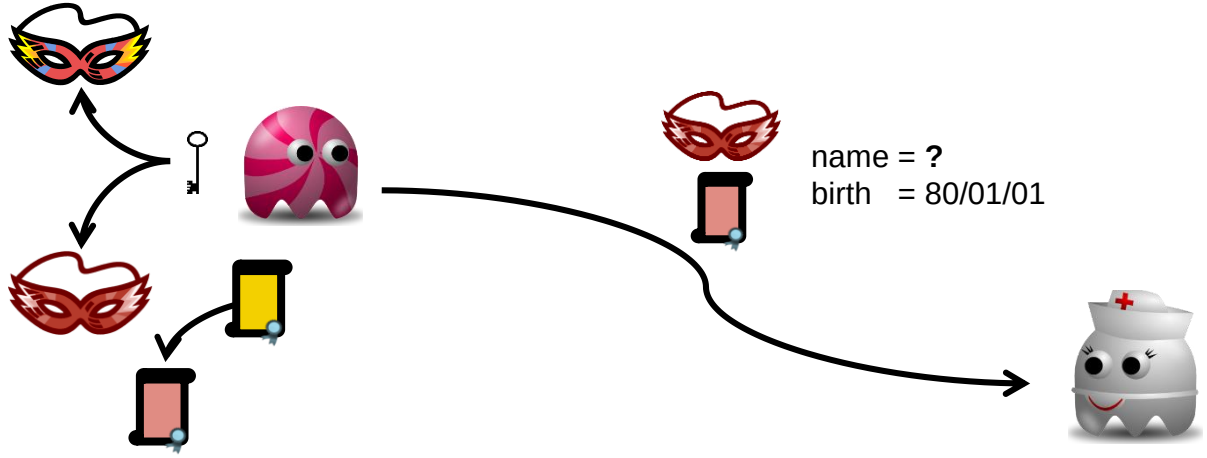


Anonymous Credential
Using them



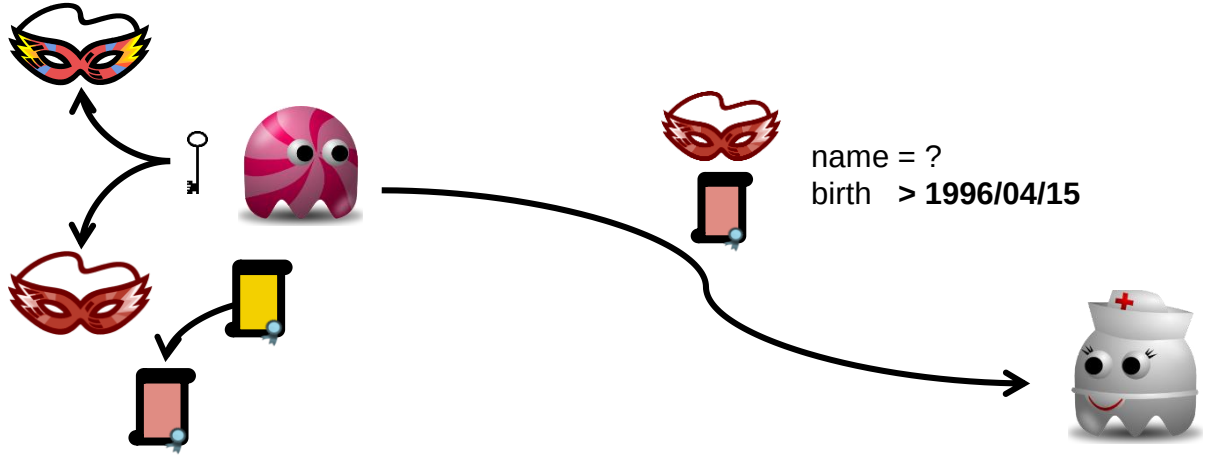
issuing and showing are unlinkable

Anonymous Credential
Using them



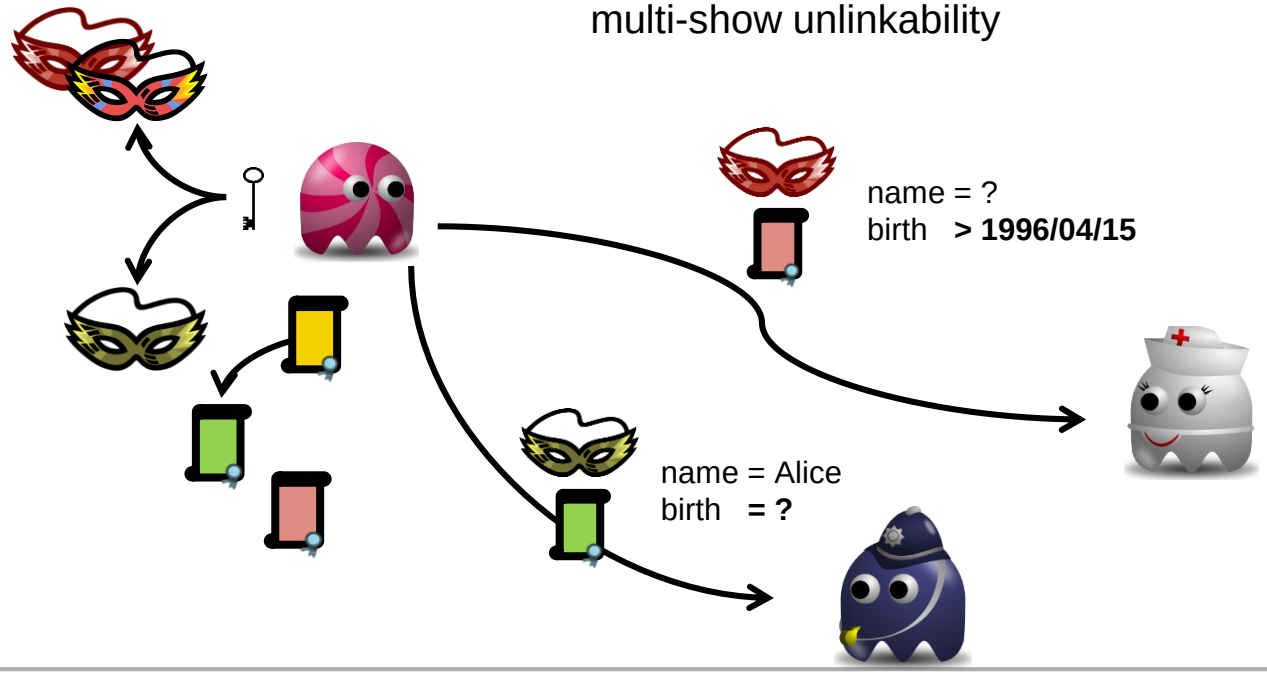
selective attribute disclosure

Anonymous Credential
Using them

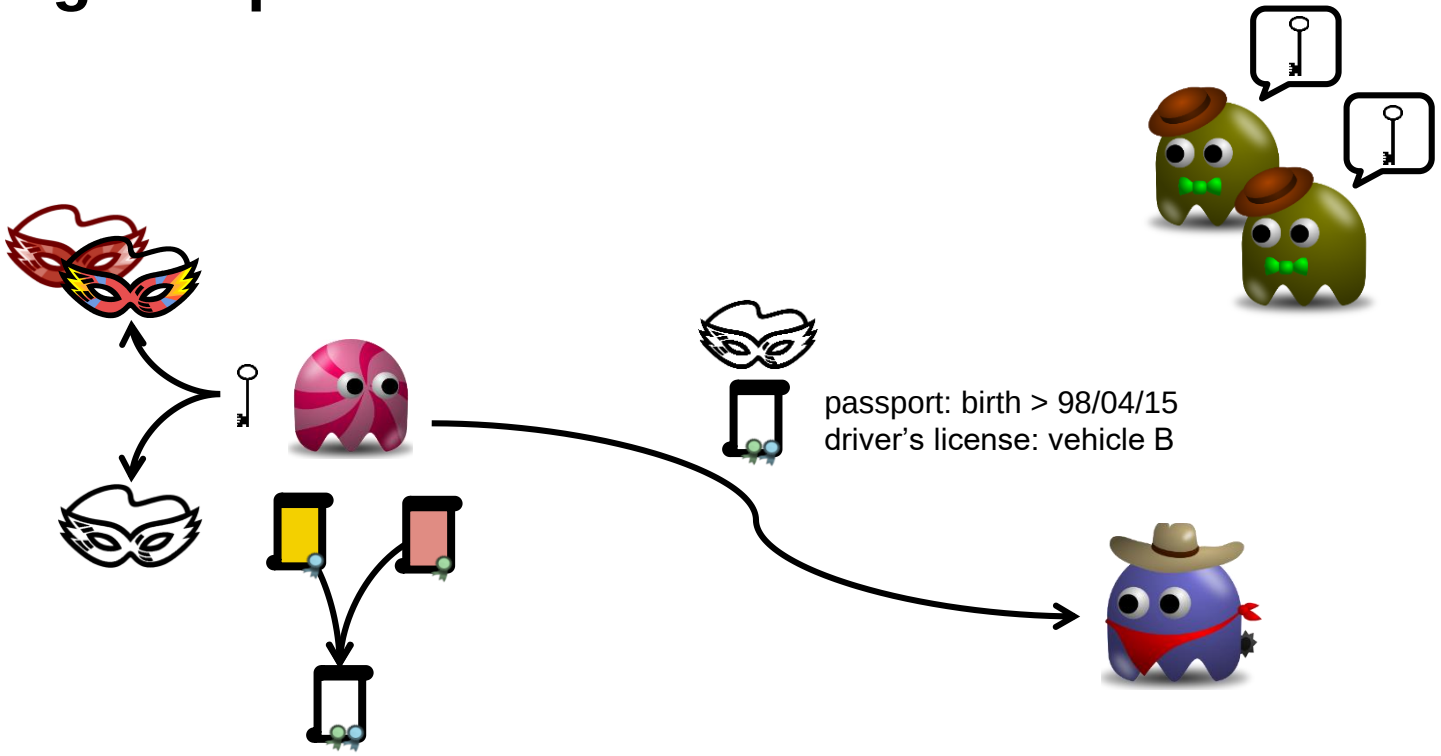


attribute predicate disclosure

Anonymous Credential
Using them



Anonymous Credential
Using multiple of them



Summary

- Anonymous credentials enable data minimization
  - Only reveal the attributes needed
    or just prove some predicate over them
- Strong yet privacy preserving authentication
- Identity Mixer and U-Prove are similar
  - implementations are available
  - various extensions are possible

Questions?

Leonardo Martucci
leonardo.martucci@kau.se

Extended Functionalities of Anonymous Credentials

- Credentials on hidden attributes
- Tracing of user/attribute
- Revocation of credentials
  - Accumulators, signed intervals, validity time
- Limited spending
  - Hidden serial number, domain pseudonyms, offline identity recovery, verifiable random functions