



## Outline

### Attribute-based Authorisation

Bart Jacobs

Institute for Computing and Information Sciences – Digital Security  
Radboud University Nijmegen

Joint work with Wojciech Mostowski & Pim Vullers

16 nov. 2011

Govcert'11

Background on identities & attributes

Different systems for attributes

Mathematics behind U-Prove & Idemix  
Self-blindable credentials

Developments

United States  
Germany  
The Netherlands

Conclusions

## Motivation & support for the work

- **Original motivation:** develop privacy-friendly alternative for OV-chipkaart (in public transport)
  - using electronic **attributes** instead of **identities**
  - card says only e.g. "I'm a valid 2nd class year pass"
  - involves non-trivial cryptographic protection (see later)
- Research project supported by:
  - NLnet Foundation ([nlnet.nl](http://nlnet.nl))
  - Open Ticketing / TLS ([openticketing.nl](http://openticketing.nl))
- **Later, broader motivation:** develop attribute infrastructure for e-Identity card
  - citizens can obtain attributes from attribute-providers (age  $\geq 18$ , age  $\geq 65$ , student, address, bank account, ip-address)
  - and use them for online transactions

## Who are you? Identities and attributes

- If you wish to buy a bottle of whiskey, you have to show that you are over 18 — fair enough
- In practice (offline) you wave an identity card in front of the shopkeeper
- But what if the shopkeeper would make a photocopy, or read your identity document electronically?
  - online this becomes even more problematic
- The transaction only requires the **attribute** "over 18"
  - and not your **identity** (whatever that is)
  - any additional information, besides "over 18", can be **abused** (identity fraud, profiling)
  - attribute-usage fits in data minimalisation requirements

## Identities & attributes I

- Difficult question: *What is your identity?*
  - is it your social security number (BSN, in Dutch)?
  - who knows his/her number by heart?
  - does it make you feel: this is me?
- Possible way out:

Your identity is the collection of attributes that hold for you

## Identities & attributes II

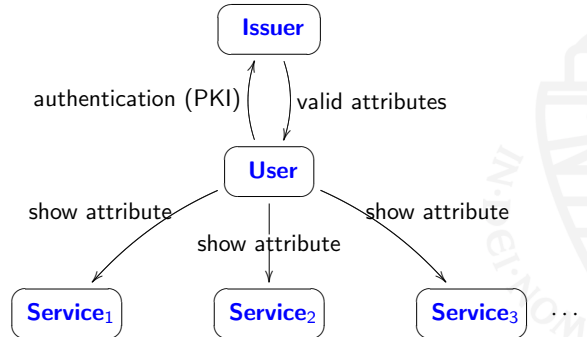
- Some attributes are **identifying**
  - like your social, security number or bank account, or OV-chipcard number
  - they are different for different people
- Other attributes are **non-identifying** (anonymous)
  - like your gender, whether you're over 18, your home-town
  - whether you have a valid ticket to travel by bus
  - whether you are a nurse or a doctor
- Sometimes your identity is understood as a (small) set of identifying attributes, like on your passport
- When going digital, attributes are often replaced by identities, like in public transport
  - why do I have to tell who I am when I get on the bus
  - more unnecessary surveillance / profiling / fraud risk



## Attribute-based authentication & authorisation

## Attribute issuance-usage model

- Many transactions can be performed on the basis of **non-identifying attributes**
  - a cheaper hair-cut for a student, or cheaper public transport for senior citizens
  - participation in local referendum for locals
  - buying games online (over 16, or over 18)
  - viewer restrictions for missed TV-program website
- **Attribute-based** extends **role-based** access control
  - the captain of the ship can turn the ship's wheel
  - very relevant in the medical sector (access to files)
  - or in the military, or in any other organisation with different authorisations for different hierarchies/roles
- Typical transactions involve a **combination of attributes**
  - address, possibly with bank account, for pizza delivery
  - age + bank account for online gambling / XXX / ...
  - "doctor" status + medical registration number for write-access to medical record



One may also have **multiple issuers** (government, banks, isp's, ...)



## Requirements for attribute-based systems

## Three main systems

- **Non-transferability**: my little nephew should not be able to get my "over 18" attribute (and go to XXX sites)
  - realised via binding to my private key
- **Issuer-unlinkability**: the issuers should not be able to track where I use which attribute
  - typically realised via blind signature
- **Multi-show unlinkability**: service providers should not be able to connect usage (at different providers)
  - realised via zero-knowledge proofs, or via "self-blindable" credentials
- **Revocation**: rogue attributes (via stolen/lost cards) should be blockable.
  - most difficult, partly in conflict with previous requirements

- **U-Prove**
  - developed by Stefan Brands (Credentica), bought by Microsoft
  - specification available, under the Open Specification Promise
  - open source reference toolkits in C# and Java
  - multiple attributes in single (traceable) token, selective disclosure
- **Idemix** ("Identity Mixer")
  - developed by Camenisch & Lysyanskaya, IBM Research Zürich
  - specs & sources also openly available
  - most properties, including revocation (by users, not by issuers)
  - most complicated (even "over-engineered")
- **Self-blindable certificates**
  - developed by Verheul (Radboud Univ. & PWC) and others
  - only one attribute per token
  - uses bilinear pairings on elliptic curves
  - open implementation available



## Smart card implementations of attributes

## U-Prove & Idemix

Main focus of the work at Nijmegen (esp. Vullers & Mostowski)

- **Self-blindable certificates**
  - implemented on a Java smart card, with ECC
  - transaction (showing) times: 500-1000ms
  - too slow for public transport (requires 300ms); next, faster generation of cards is needed
- **U-Prove**
  - low-level implementation on MULTOS smart card
  - with low-level access to the cryptographic co-processor
  - times: 500-1000ms, depending of number of shown attributes
  - major improvement over Microsoft's device-binding approach
- **Idemix**
  - MULTOS implementation under development
  - (partial & slow IBM-implementation on Java card exists)

- Both involve tokens with multiple attributes  $a_1, \dots, a_k$ , encoded as **multi-exponent**  $g_1^{a_1} \dots g_k^{a_k}$
- Such tokens are **blindly** signed by the issuer
  - using Schnorr-style signature scheme in U-Prove
  - via Camenisch-Lysyanskaya signature for Idemix
- Proving proceeds in both cases via **zero knowledge proof**
- In Idemix tokens are **self-blindable** (aka. **randomisable**: owner can transform them so that they look different in each showing proof)
- In U-Prove, tokens are not self-blindable and can be used to trace the owner
  - tokens thus work as **pseudonyms**
  - show-unlinkability still possible by using multiple tokens, or by using them only once
  - revocation of tokens thus possible via blacklisting



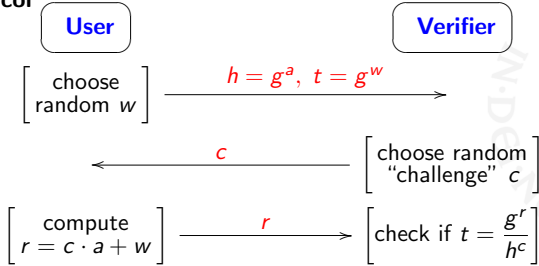


## Basic protocol: Schnorr's proof of knowledge

### Assumptions

- User has attribute  $a$ , encoded as  $h = g^a$  (in some cyclic group)
- Verifier must be convinced of knowledge of  $a$  in  $h$ , without revealing  $a$

### Protocol



Relevant computation:  $g^r = g^{c \cdot a + w} = (g^a)^c \cdot g^w = h^c \cdot t$

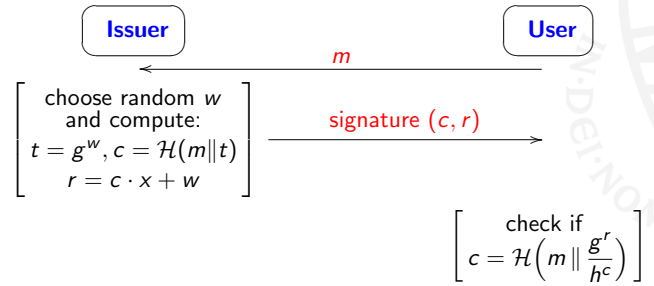


## Basic protocol: Schnorr's signature

### Assumptions

- Issuer has private key  $x$ , with associated public key  $h = g^x$
- User wants signature on hash  $\mathcal{H}(m)$  of message  $m$

### Protocol



## More on U-Prove and Idemix

## Self-blinding via Elliptic Curve Cryptography (ECC)

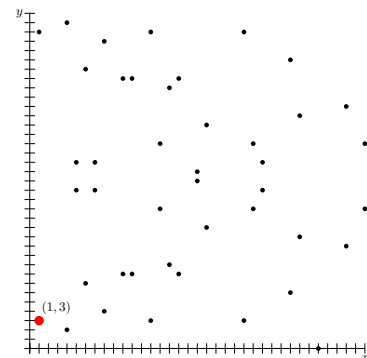
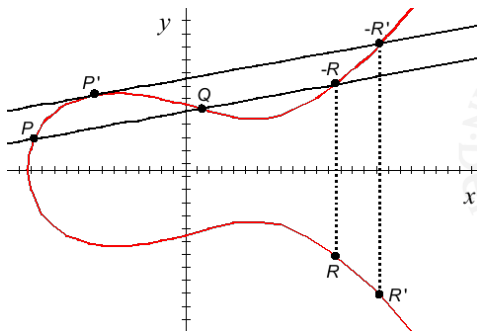
- These two protocols only describe basic building blocks
- The real U-Prove protocols are more complicated than this
- For more info:
  - Stefan Brands, *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*, MIT, 2000. Freely available via [credentica.com](http://credentica.com)
  - [microsoft.com/u-prove](http://microsoft.com/u-prove)
- For information about Idemix, see:
  - [zurich.ibm.com/security/idemix](http://zurich.ibm.com/security/idemix)

- Koblitz and Miller proposed the use of elliptic curves for cryptography in the mid 1980's
- Nowadays this technology is widely accepted
- Provides the functionality of RSA and more
  - Smaller keys
  - Pairings
- Standard public key cryptography for embedded platforms
  - used now e.g. in e-passport chip

## The operations, over the real numbers

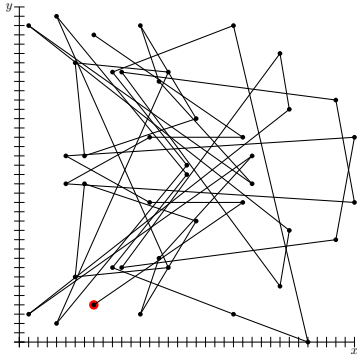
## Example curve: $y^2 = x^3 + 2x + 6$ over finite field $\mathbb{F}_{37}$

- Point addition:  $P + Q = R$
- Point doubling:  $2P = R'$





## Repeated addition: $n \cdot P$ goes everywhere



Given  $Q = n \cdot P$ , finding  $n$  involves basically trying all options.

## Elliptic Curve Cryptosystem

- Point multiplication (repeated addition):  $k \cdot P = Q$
- Easy to compute (double and add)
- EC **discrete log** problem: Given  $P$  and  $Q = k \cdot P$ , determine  $k$
- This problem is believed to be **hard**
- Point multiplication is a **one way function** which can be used to build public key cryptosystems
- The **public** key is  $Q$ ; and the **private** key is  $k$
- Allows for key agreement (Diffie-Hellman), signatures (DSA), encryption (ElGamal), and more ...



## Pairings

## Self-blindable certificates

- A **bilinear pairing** is a map  $e: G_1 \times G_2 \rightarrow G_T$  which is bilinear, that is, linear in both components:

$$e(P + P', Q) = e(P, Q) \cdot e(P', Q)$$

and

$$e(P, Q + Q') = e(P, Q) \cdot e(P, Q')$$

- As a result,  $e(n \cdot P, m \cdot Q) = e(P, Q)^{nm}$

### Pairing-based Signatures

- Signature  $S = s \cdot P$  over a point  $P$  is **multiplication** by a private key  $s$
- Check  $e(S, Q) \stackrel{?}{=} e(P, s \cdot Q)$  to verify a signature  $S$  over  $P$ : both sides are equal to  $e(P, Q)^s$ .

### This certificate is stable under self-blinding

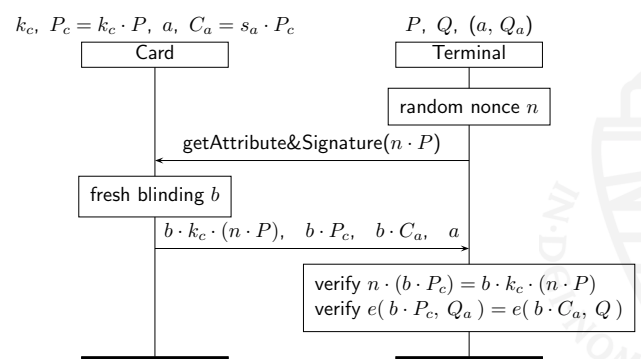
- card generates random blinding number  $b$
- forms new pair
 
$$P_b = b \cdot P \quad \text{with signature} \\ S_b = b \cdot S = b \cdot s \cdot P = s \cdot b \cdot P = s \cdot P_b$$
- Verification of blinded  $P$  and  $S$ :  $e(b \cdot P, s \cdot Q) = e(b \cdot S, Q)$



## Set up for attribute proving

## Protocol for Attribute-proving

- Public fixed point  $Q$  and a **finite set of attributes**
- A secret key  $s_a$  and public key  $Q_a = s_a \cdot Q$  for each attribute  $a$
- The associated pairs  $(a, Q_a)$  are publicly know, and stored in all terminals together with the fixed point  $Q$
- Card  $c$  generates a key pair  $k_c$  and  $P_c = k_c \cdot P$
- Private key  $k_c$  is stored in a protected manner, in the card
- Card  $c$  receives an attribute together with a **certificate**  $C_a = s_a \cdot P_c$  linking its public key  $P_c$  to the attribute  $a$
- (alternative "Boneh-Boyer" certificates exist)



This runs in a Java smart card in 500-1000 msec.



## USA: NSTIC

## NSTIC, continued

- NSTIC = *National Strategy for Trusted Identities in Cyberspace*
  - released by the White House in April 2011
  - discussed by Cyber Security Czar Howard Schmidt in his blog "We have an opportunity to design privacy directly into the fabric of the Identity Ecosystem."
- NSTIC refers to privacy-enhancing technologies for implementing credentials
  - U-Prove & Idemix are not mentioned explicitly
  - but probably intended implicitly
- Issuer- and multi-show unlinkability explicitly required (and also partial information, like "over 18")

- Discussion by Francisco Corella on [pomcor.com](http://pomcor.com)

"Government-issued credentials will only be acceptable if they incorporate all available privacy protections. That makes the use of privacy-enhancing technologies essential to the success of NSTIC."
- Talking about Nijmegen's U-Prove implementation:
 

"A non-Microsoft implementation of U-Prove on a MULTOS smart card, where all the cryptographic computations are carried out by the card with impressive performance (close to 0.3 seconds in some cases), can be found in ..."
- Good intentions, but the current status/progress of NSTIC is unclear.



## Germany: *Neue Personalausweis (nPA)*

## *Neue Personalausweis*

- RFID smart card issued to German citizens since nov. 2010
  - non-trivial protocols developed by BSI (a.o. Dennis Kügler)
- Acces to card involves **terminal authentication** (part of "EAC")
  - only authorised parties can read (part of the data)
  - eg. webshops need to get certificates to access the card
- Card is protected by **6-digit PIN**, with fallback via number on card & separate PUK
- Also **pseudonym** generation is included, providing linkability ("restricted identification") for returning customers
  - authentication of cards happens via private key ...
  - that is same in batches of 100.000s of cards

- Authenticity & integrity of data in chip not guaranteed via signature, like in "passive authentication", to prevent third party usage, but implicitly via chip-authentication
  - Thus, once the authenticity of the chip is established
  - the reader trusts the provided data
  - these **unsigned** data are less valuable
- **Signing keys** are not included, but can be added by the card owner, following own choice of certificate provider
  - hence it is more difficult to force citizens to sign everything, and make themselves traceable via their certificates
- Attacks have been reported, mainly focusing on the interface
  - notably entering the PIN via the computer (key logging!)



## Modern smart card reader with PIN pad

## The Netherlands: eNIK and DigiD

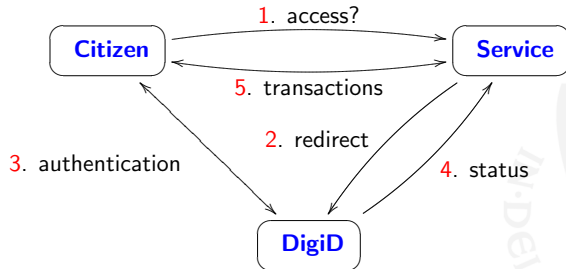


- This one is available for the German e-Identity card
- Interfaces for both **contact** and **contactless** cards
- Certified by BSI; cost: 30-100 €

- Tender for citizen smart card organised in 2007, but aborted after legal battle
  - smart card would only do authentication & signing
  - only within the government domain
- Currently, these plans are re-emerging
- No great need so far, since NL has a centralised **authentication service**
  - called **DigiD**, for Digital Identity
  - operational since 2005; widely used, only for public sector
  - works with two levels of authentication (password/SMS-OTP)



## DigiD essentials



In step 4 the status message says: "social security number  $i$  with certainty level  $j$ ". Afterwards DigiD has no role in the transactions in step 5.

## DigiD: main points

- DigiD is a central **authentication server**
  - a bit like **Kerberos**
  - DigiD is a **hotspot** for traffic data (not for transactions)
  - it does not provide signatures (non-repudiation)
- Three levels of authentication are foreseen
  - username + password (traditionally weak)
  - one-time code via SMS (can be intercepted, now GSM is broken)
  - smart card authentication (**eNIK**) not present yet
- DigiD has several weaknesses:
  - no **face-to-face** authentication at registration
  - "services" side is sometimes vulnerable (SQL/XSS)



## DigiD reflects centralistic mentality

### Citizen is reduced to authenticator

- we administer all the information that we need about you
- you, as citizen, only need to authenticate (prove who you are)

### How far should one push this centralistic approach?

- Suppose I wish to enter my own house; at the front door, ...
- I push a button and authenticate to DigiD ...
- which remotely opens the door for me!

Maybe a few things should be done de-centrally

## eNIK 2012

- DigiD is the dominant (centralistic) paradigm, so smart cards are not well-understood — except for authentication
- Thus, the main motivation for a eNIK is to provide a third authentication level within DigiD
  - additional value of signature via smart card is not recognised
  - some even promote **server-side** "signatures" !!! ☹️
  - i.e. after authentication you tell a server to sign on your behalf
- Chosen card functionality for 2012 like for 2007
  - only authentication & signature (30 year old technology)
  - no ambition noticeable so far to use modern attribute mechanisms
- Attitude towards attributes: **put them on the server**
  - vulnerable e.g. for doctor accessing local files
  - complete trust in administration required



## Main points

- Attribute-based** authentication/authorisation is natural and privacy-friendly (data-minimisation), see e.g. in NSTIC
- Cryptographic basis exists for  $\geq 10$  years (U-Prove/Idemix)
  - open & supported by major players (Microsoft, IBM)
- ECC-based self-blindable credentials more recent (and fancy)
- Fast **smart card** support available ( $< 1$  sec for showing)
  - That is, for U-Prove & self-blindable certificates
  - Idemix card implementation is ongoing work
- Demo available (but not shown here)



## Questions / Remarks?