IRMA Math

Bart Jacobs

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

May 17, 2013, Kaleidoscoop, Nijmegen
irmacard.org

Outline

Background about IRMA

- IRMA = "I Reveal My Attributes"
- A research project on attribute-based authentication
  - within Nijmegen's Digital Security group
- Many (local) people involved: Gergely Alpár, Jaap-Henk Hoepman, Wouter Lueks, Roland van Rijswijk, Peter Schwabe, Pim Vullers, Ronny Wichers Schreur
- And several partner organisations: TNO, Surfnet, SIDN
  - Esp. Maarten Everts, Martijn Oostdijk

Mathematical basis of IRMA

- Idemix = "Identity Mixer", underlying cryptographic system
  - developed at IBM Zürich in the early 2000s
  - notably by Jan Camenisch
- Idemix uses powerful cryptograpic primitives
  - the main points will be described later on
  - actual calculations are non-trivial (take quite some time)
- There are comparable cryptographic systems:
  - U-prove, developed by Stefan Brands, now owned by Microsoft
  - Self-blindable signatures, by Eric Verheul, using bilinear pairings on elliptic curves

Nijmegen's contribution

- Fast(est) smart card implementation for all three approaches, by Pim Vullers
- Additional protocol development (eg. tunnels, revocation)
- Practical realisation initiative "IRMA", based on Idemix
  - not all Idemix features, emphasis on *selective disclosure*
  - active role in discussion about next eID in NL
- Middleware development to create eco-system for attributes
  - attribute verification, issuing, management; registration
  - integration in websites, NFC phones & tables, POS terminal
  - experimental attribute issuing via government website
- Small pilot for own "Kerckhoffs" master students (±100), starting soon.

What is authentication?

To authenticate yourself can mean:

1. proving who you are, eg. via your passport or password
2. proving some property ("attribute") about yourself
   - Eg. proving you are "over 18" when buying liquor

In the IRMA context:

- we concentrate on authentication via attributes
- it may uniquely identify you, eg. via a bank-acount-attribute

Offline authentication

- In the offline world (real-life) authentication often happens implicitly
- Eg. in a hospital you expect that
  - a woman in a white uniform is a ...
  - a man in a white uniform is a ...

Still there are **rules**: eg. if someone dressed like a police officer asks for your identity document

- you have the right to require that (s)he first proves to be a police officer indeed

Online authentication

- Authentication online is more delicate because the context is
  - either absent (how to recognise an online doctor?)
  - easy to manipulate (eg. fake bank-website)
- Cryptographic techniques can be helpful in authentication
  - most well-known for authentication of websites: 
  - but also for users, eg. via smart cards

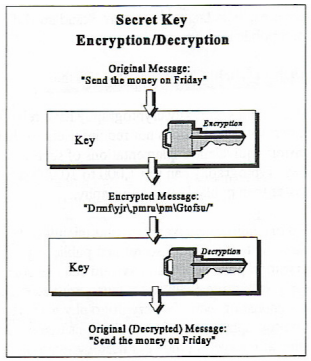
Online authentication via DigiD

- DigiD is central authentication service in NL
  - it is heavily used for lots of (semi)public services
- DigiD works in several steps:
  1. user goes to website of "relying party", where login is required
  2. user gets redirected to DigiD and authenticates
  3. user returns to relying party, with "ticket" saying

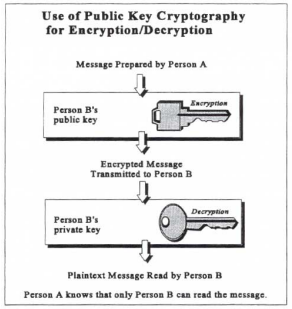
     with certainty level x, this user has BSN y

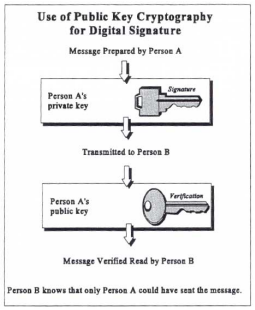
     This ticket is digitally signed by DigiD
- Three authentication levels are foreseen; only 1&2 available
  1. username + password
  2. one-time password (OTP) via SMS
  3. cryptographic authentication (via smart card)

Secret key crypto: one *shared* key



Public key crypto: encryption & signing via *two keys*





Public key notation

- **Keys**: a keypair of a participant X consists of two keys:

$$\begin{cases} \text{public key:} & e_X \\ \text{private key:} & d_X \end{cases}$$

- **En/de-cryption**: Let M be an arbitrary message

$$\begin{cases} \text{encryption for } X\text{:} & \mathrm{Enc}(M; e_X) \\ \text{decryption/signing by } X\text{:} & \mathrm{Dec}(m; d_X),\ \mathrm{Sign}(M; d_X) \end{cases}$$

- **Certificate**: statement of public key possession, signed by some authority CA

$$\begin{aligned} C_X &= \mathrm{Sign}(\text{key-claim}; d_{CA}) \\ \text{key-claim} &= \text{"public key } e_X \text{ belongs to } X\text{"} \end{aligned}$$

Authentication with public key crypto

$A \longrightarrow B$: "hi, I'm A; this is my certificate", C_A
$B \longrightarrow A$: R, (a large random number)
$A \longrightarrow B$: $\mathrm{Sign}(R; d_A)$

Different keypairs for different functions

- There are three main roles of public key crypto: **encryption, signing, authentication**
- for each of these roles a separate keypair should be used

RSA essentials

1. A user chooses:
   - two large primes p, q (each at least 1024 bits)
   - a number $e \in \mathbb{Z}^*_\phi$ where $\phi = \phi(p \cdot q) = (p-1) \cdot (q-1)$
2. The public key is now (n, e), where $n = p \cdot q$
3. The private key is (n, d), where $d = \frac{1}{e} \in \mathbb{Z}^*_\phi$, so that $e \cdot d \equiv 1 \mod \phi$

Usage, on message M:

- **Encryption**: $\mathrm{Enc}(M; (n, e)) = M^e \mod n$
- **Decryption**: $\mathrm{Dec}(M; (n, d)) = M^d \mod n$
- **Signing**: $\mathrm{Sign}(M; (n, d)) = M^d \mod n$,
  - More efficiently: $\mathrm{Sign}(M; (n, d)) = h(M)^d \mod n$, where h is a hash function.

Blind signatures with RSA (after Chaum)

Let (n, e) be the public key of B, with private key (n, d).

1. A wants to get a blind signature on M; she generates a random r, takes $M' = (r^e) \cdot M \mod n$, and gives M' to B.
2. B signs M', giving the result $N = \mathrm{Sign}(M'; (n, d)) = (M')^d \mod n$ to A
3. A computes:

$$\frac{N}{r} = \frac{(M')^d}{r} = \frac{(r^e \cdot M)^d}{r} = \frac{r^{ed} \cdot M^d}{r} \equiv \frac{r \cdot M^d}{r} = M^d = \mathrm{Sign}(M; (n, d))$$

Thus: B signed message M without seeing it!
This will be useful later.

Identities versus attributes

- Identity management seems to revolve around identities
  - In practice this means uniquely identifying numbers, like social security number, or passport number
  - high-value targets for profiling & identity fraud
- But a more flexible identity ecosystem uses attributes
  - 'over 18', 'over 21', 'over 65', 'under 15', 'male', 'female'
  - 'student', 'doctor', 'president', 'top secret clearance'
  - 'NL-citizen', 'resident of Nijmegen'
  - 'home address', 'owner of bankaccount nr. ...'

Your **identity** is the collection of attributes that hold for you

Key idea in attribute-based IdM

- Each transaction only requires a **subset** of your attributes for authentication
  - the subset should be small & proportional: data minimisation
  - this also offers some protection against identity fraud
- Attributes support **contextual privacy**
  - an essential aspect of privacy is being able to reveal different aspects of yourself in different contexts
  - attributes support such "partial identities" or "personas"

Demo time

Let's see some running code:

1. attribute verification, on NFC-enabled tablet
2. age bound for spicy website, using NFC phone as card reader

Requirements for attribute-bases 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(able) signature
- Multi-show unlinkability: service providers should not be able to connect usage (at different providers)
  - realised via zero-knowledge proofs, or via "self-blinding"
- Revocation: rogue attributes (via stolen/lost cards) should be blockable.
  - most difficult, partly in conflict with previous requirements
  - efficient solutions are still very much a research topic

Attributes in certificates (naive version)

- Instead of identities, one can put attributes in certificates:

$$\begin{array}{rcl} C_{\geq 18} & = & \mathrm{Sign}(\text{age-claim}; d_{\mathrm{Auth}}) \\ \text{age-claim} & = & \text{"public key } e \text{ belongs to someone who is over 18"} \end{array}$$

- Age proof protocol:

$$\begin{array}{rl} A \longrightarrow B: & \text{"hi, I'm over 18; this is my certificate"}, C_{\geq 18} \\ B \longrightarrow A: & R \\ A \longrightarrow B: & \mathrm{Sign}(R; d) \end{array}$$

Traceability problems

- by issuer, who can follow usage of certificate $C_{\geq 18}$
  - this can be solved via a blind signature
- by verifiers, who can link different usages of $C_{\geq 18}$.

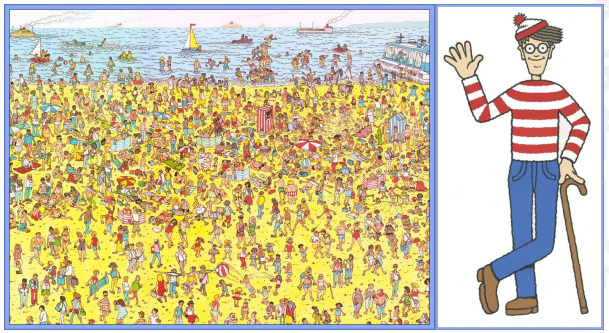
Solving traceability problems

- Via blindable signatures
  - use special signatures that remain valid when multiplied with some blinding factor; must be done after each usage
  - this can be done eg. via bilinear pairings on elliptic curves; elaborated by Eric Verheul (2001)
  - used in Idemix, and hence also in IRMA cards
- Via zero-knowledge proofs
  - also used in Idemix & IRMA

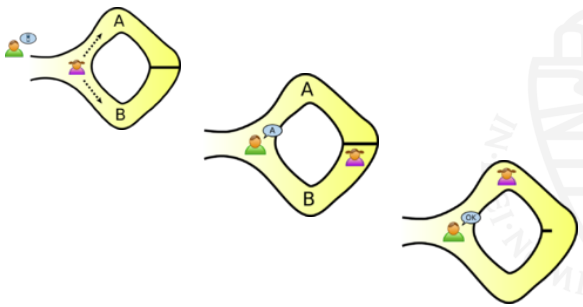
What are zero-knowledge proofs

- Typically, one proves to have knowledge of a certain password or PIN by typing it in
  - using this secret in-the-clear is a bad idea
  - the secret information can easily be eavesdropped
- It would be best to prove knowledge of the right password, without revealing it.
- This is precisely what zero-knowledge proofs allow you to do!

Zero-knowlegde proof of "Wally's location"



Quisquater's cave explanation



Discrete log problem

- The security of RSA depends on the difficulty of prime factorisation
- Another mathematical difficulty that is useful in cryptography is the discrete log problem
  - this applies to (multiplicative) groups like $\mathbb{Z}_N^*$
  - but also to (additive) groups of points on an elliptic curve
- **Discrete log problem**: given an element h in a finite group G of order N with generator $g \in G$, find $n < N$ with $h = g^n$
- In general, this discrete log problem is computationally hard. Intuitively, there is no better way than trying out all g^i.

Schnorr's proof of knowledge

- Assume a generator $g \in G$ in a finite group of prime order q, with publicly given $h = g^x \in G$, where $x \in \mathbb{Z}_q^*$.
- Peggy (P) wants to prove to Victor (V) that she knows x, of course, without revealing it.

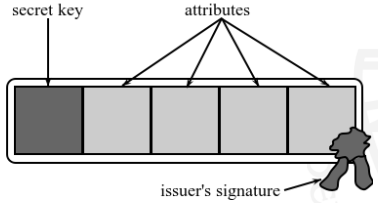
$$\begin{array}{lll} P \longrightarrow V: & a \stackrel{\text{def}}{=} g^w \in G & \text{with } w \in \mathbb{Z}_q^* \text{ random} \\ V \longrightarrow P: & c \in \mathbb{Z}_q & \text{a random challenge} \\ P \longrightarrow V: & r \stackrel{\text{def}}{=} c \cdot x + w & \\ & \text{V now checks } g^r \stackrel{??}{=} h^c \cdot a & \end{array}$$

Note that V can prove nothing to others: anyone can produce values r and a with $g^r = h^c \cdot a$.

Credentials and attributes on IRMA cards

A card contains multiple credentials, each with multiple attributes:

Credential =



- The secret key is securely stored in the smart card, making credentials non-transferable; required in "showing attributes"
- The issuer's signature guarantees authenticity and integrity
- Any subset of the attributes can be shown in transactions. This is called selective disclosure.

Example credential: address

Address
Country
City
Street + Number
Postal code

Separately usable attributes

Issued by: public authorities

- Name is not included, stored elsewhere (no credential overlap)
- Expiry info is omitted

Attribute representation I

System parameters:

- $n = pq$, for large "safe" primes: p, q, where $p = 2p' + 1, q = 2q' + 1$, with also p', q' prime

  The pair (p, q) is the secret key of the credential issuer
- quadratic residues: $R_0, R_1, R_2, R_3, R_4, S, Z \in QR_n \subseteq \mathbb{Z}_n^*$ (5 R's, for 4 attributes per credential, plus the card's secret key)

A 4-tuple (a_1, a_2, a_3, a_4) of attributes a_i is represented via a multi-exponent:

$$R_1^{a_1} \cdot R_2^{a_2} \cdot R_3^{a_3} \cdot R_4^{a_4} \in \mathbb{Z}_n$$

This multi-exponent must be *randomised* and *signed*, via a so-called Camenisch-Lysyanskaya signature (2002).

Attribute representation II

- Let k be the secret key of the card. A credential is a triple:

$$v,\ e,\ C \ = \ \left(\frac{Z}{S^v R_0^k R_1^{a_1} R_2^{a_2} R_3^{a_3} R_4^{a_4}}\right)^{\frac{1}{e}}$$

  where v, e are random, with $e \cdot \frac{1}{e} \equiv 1 \mod \phi(n)$.
- The crucial signature verification equation is:

$$Z \ \equiv \ C^e \cdot S^v \cdot R_0^k \cdot R_1^{a_1} \cdot R_2^{a_2} \cdot R_3^{a_3} \cdot R_4^{a_4} \mod n$$

Blinding of the signature/credential

- the equation still holds for $v' := v + e \cdot r$, $C' := C \cdot S^{-r}$
- e remains the same, requires zero-knowledge proof

Selective disclosure

- Assume I wish to disclose attributes a_1, a_3, but not a_2, a_4.
  - The blinding of the credential e, v, C is skipped here
- I reveal attribute values a_1, a_3 and credential (parts) v, C
- Via a zero-knowledge proof I show that I know exponents $\varepsilon, \kappa, \alpha_2, \alpha_4$ with:

$$\frac{Z}{R_2^{a_2} \cdot R_4^{a_4}} \equiv C^{\varepsilon} \cdot S^{v} \cdot R_0^{\kappa} \cdot R_2^{\alpha_2} \cdot R_4^{\alpha_4} \mod n$$

Attribute access for verifiers

NOTE: there is a risk that verifiers read too many attributes

Governance model

- prospective verifiers register at an "IRMA foundation", stating their goals & requesting access to certain attributes
- if the request is proportional, access is approved
- verifier obtains certificate capturing acces to these attributes
- IRMA cards check such certificates first, before they reveal any attributes

Card-to-cardholder binding

Two mechanisms:

1. Photo of cardholder
2. PIN

Card view

Front

Back

- There is only a picture on the frontside, nothing else
- There is a (random) card number on the back, which is:
  - not present inside the (chip in the) card
  - useful for "lost-and-found" scenarios
    (The card has a randomised UID)

PIN issues

Each card comes with two PINs

- **One for attribute reading**
  - Which attributes should be protected by PIN?
  - Balance between: ease-of-use, ease-of-abuse, confidentiality
  - over 18: *yes*, medical data: *no* (restrict read-certificates)
- **One for personal card management**
  - card owner can manage own attributes on card
    (like apps on smart phone)
  - also access to card logs

eID developments in NL

- NL is late — through legal fight over earlier tender
  - delay may actually be an advantage
  - existing PKI cards are hardly used in practice (except Estonia)
- NL has social security number ("BSN")
  - but its usage is restricted to the public sector
  - it forms the basis for much-used national authentication system (DigiD), based on password and/or OTP via text message
- Two (main) eID requirements in NL:
  - strong authentication, based on smart cards
  - usable both in the public and private sector
- Two important options:
  1. copy German card, using pseudonyms
  2. introduce IRMA cards, using attributes

Option 1 is safest bet, but option 2 is most flexible

Main points

- IRMA work is based on:
  - clever cryptographic basis (Camenisch-Lysyanskaya)
  - smart & fast implementation, and middleware (Nijmegen)
- This approach offers privacy & security, much user control
- Scaling-up attribute use requires carefully designed & controlled identity eco-system
- Open character can be innovation motor, leading to many, now unforeseen, applications.
- Recommended next step: organisation of large scale pilot, with ten thousands of users, like in university pass, or city pass.
- This technology gives policy makers & regulators the tools to enforce privacy & security by design!

Learning / doing more . . .

- Check out the website irmacard.org
- Let us know if you wish to join, eg. with development work
- You can even join our computer security master programme:

k3rckhoffs 1nst1tute named after 

See kerckhoffs-institute.org

Thanks for your attention. Questions/remarks?