Security and privacy issues for electronic ID cards

Przemysław Kubiak Mirosław Kutyłowski

Wrocław University of Technology
Institute of Mathematics and Computer Science

Warsaw, 18.03.2010

Agenda

Plan of this talk:

1. e-ID and privacy issues
2. anonymity: sector identities, restricted identification
3. e-signature for public administration

e-ID & Authentication versus Data Protection

Privacy problems

naïve implementation

Authentication

1. data fields of the e-ID digitally signed by the issuer of the e-ID card
2. an e-ID card authenticates itself by signing a random challenge with its private authentication key

Privacy problems
naïve implementation - personal data harvesting

Threats for signed data

uncontrolled access signed data fields collected (even illegally):

- high quality data, undeniable
- hot potato transmission protocols, untraceable thanks to P2P & crypto technology

Privacy problems
naïve implementation - personal data harvesting

Threats for signed data

uncontrolled access signed data fields collected (even illegally):
- high quality data, undeniable
- hot potato transmission protocols, untraceable thanks to P2P & crypto technology

controlled access only authorized readers can access signatures:
- non trivial infrastructure: PKI for entitled readers, with no internal clock in e-ID cards
- is it really necessary? (online access to registries is possible)

Privacy problems

naïve implementation - authentication

Threats of Big Brother

undeniable evidence of location of the e-ID owner:

random challenge: implemented as pseudo-random and indistinguishable from random challenge, it hides data readable for insiders

pseudo-random, verifiable: as above but readable to all

Privacy problems
naïve implementation - authentication

Threats of Big Brother

undeniable evidence of location of the e-ID owner:

random challenge: implemented as pseudo-random and indistinguishable from random challenge, it hides data readable for insiders

pseudo-random, verifiable: as above but readable to all

A more general problem

- strong cryptography on tiny artefacts itself is a threat
- high quality undeniable data available where so far gathering data on a massive scale was hard or impossible

General Approach

zero knowledge protocols

Zero-knowledge protocols

1. authenticate with a secret stored in the device
2. challenge response protocol, but easy to create fake protocol transcripts:
   - when executed, the protocol yields a good proof for the current participants,
   - when presented afterward to the third party, it is useless

General Approach
zero knowledge protocols

Zero-knowledge protocols

1. authenticate with a secret stored in the device
2. challenge response protocol, but easy to create fake protocol transcripts:
   - when executed, the protocol yields a good proof for the current participants,
   - when presented afterward to the third party, it is useless

Problems

- zero-knowledge protocols are quite universal...
- but “heavy” and not practical in most cases

Positive example

Static DH authentication

Authentication protocol

- e-ID holds a secret x and a certificate of public key $y = g^x$
  (all computations in a group with hard DL problem)
- protocol steps:
  1. the reader generates a at random, computes $z = g^a$ and sends z to the card,
  2. the e-ID computes $K := F(z^x)$
  3. the reader computes $K := F(y^a)$
  4. the reader and the tag communicate over a channel encrypted with K,
     implicit authentication by correct decryption

note that:

$$z^x = (g^a)^x = (g^x)^a = y^a$$

Positive example
Static DH authentication

Zero-knowledge properties

1. in order to compute the session key K, the e-ID card has to know the secret key x
2. it is quite easy to create the transcript of a session – it suffices to write the responses of the e-ID by himself!

Data protection

1. relatively easy, transcripts of communication have no proof value for the third party,
2. authentication convinces that data told by an e-ID holding a secret key confirmed by the issuer

Disadvantage

1. protection is based on memory protection

Sector Identities

Sector Identities

Requirements

1. owner of an e-ID card authenticates himself against diverse systems,
2. identities of the same person in different sectors should be unlinkable
3. in particular, the sector identities should be unlinkable to the ID of the owner

Applications

1. access to medical data,
2. contact with law enforcement authorities,
3. eBay, allegro, . . .
4. . . .

Austria

sketch of the solution

Symmetric solution - automatic way of deriving sector logins

ID for each sector computed from the personal ID number, sector ID and the secret key of the authority

$$ID_{i,s} := H(i, s, K)$$

recomputed on the fly by a central authority
solution analogous to ATM PIN mechanism

Disadvantages

1. replay attack
2. impersonation attack (by the recipient)

Germany

restricted identification, main protocol

Main properties

1. strong mutual authentication based on a three-step procedure: PACE, terminal authentication, chip authentication
2. sector ID

Discussion

1. rather heavy
2. finally anonymity is limited

Lightweight Restricted Identification

Keys

1. each e-ID holds a single secret key x for many sectors,
2. a sector S holds a base key g^r, for $r = r_S$
3. the public keys used in the sector with the base key g^r derived as

$$y_1^r, y_2^r, \ldots$$

from the public keys of the e-ID owners:

$$y_1, y_2, \ldots$$

Authentication

1. static DH
2. public key y_i^r and private key x used

Lightweight Restricted Identification

Unlinkability issues

problem given the lists $y_1, y_2, \ldots$ and $y_1^r, y_2^r, \ldots$ after sorting them,
is it possible to link any y_i with y_i^r?

linking DH Problem security question reduces to Linking Diffie-Hellman Problem:
given g^r, (g^a, g^b) and g^{z_1}, g^{z_2} where $\{g^{z_1}, g^{z_2}\} = \{g^{ra}, g^{rb}\}$
find i such that $g^{ra} = g^{z_i}$.

Lightweight Restricted Identification

Unlinkability issues

reduction it can be formally proved that

Linking Diffie-Hellman Problem can be broken
iff
Decisional Diffie-Hellman Problem can be broken

corollary the construction does not introduce any new threat.

Architectures based on mediated solutions

Why X.509 PKI fail?

Crypto card as a single point of failure

1. signing key secured by PIN only (just 4 digits!) security level unacceptable as password even for non-sensitive systems . . .

Why X.509 PKI fail?

Crypto card as a single point of failure

1. signing key secured by PIN only (just 4 digits!) security level unacceptable as password even for non-sensitive systems . . .
2. securing keys on a crypto card - a never ending game between security features and attack methods

Why X.509 PKI fail?

Crypto card as a single point of failure

1. signing key secured by PIN only (just 4 digits!) security level unacceptable as password even for non-sensitive systems . . .
2. securing keys on a crypto card - a never ending game between security features and attack methods
3. black box - the manufacturer claims that everything is fine, but a citizen must blindly trust the manufacturer and certification authorities

Why X.509 PKI fail?

Crypto card as a single point of failure

1. signing key secured by PIN only (just 4 digits!) security level unacceptable as password even for non-sensitive systems . . .
2. securing keys on a crypto card - a never ending game between security features and attack methods
3. black box - the manufacturer claims that everything is fine, but a citizen must blindly trust the manufacturer and certification authorities
4. the owner cannot really maintain the crypto card under his or her sole control

Why X.509 PKI fail?

Crypto card as a single point of failure

1. signing key secured by PIN only (just 4 digits!) security level unacceptable as password even for non-sensitive systems . . .
2. securing keys on a crypto card - a never ending game between security features and attack methods
3. black box - the manufacturer claims that everything is fine, but a citizen must blindly trust the manufacturer and certification authorities
4. the owner cannot really maintain the crypto card under his or her sole control
5. no control over signing time, so revocation does not solve the problem

Requirements

Minimal security properties

1. independent components necessary for signing
2. it must be possible to “deactivate” signing possibility when needed
3. signing time must be undeniable based on digital data

Mediated signature *mRSA*

key generation

RSA, setup for 3 components

1. keys n, e, d generated exactly as for the standard RSA

Mediated signature *mRSA*
key generation

RSA, setup for 3 components

1. keys n, e, d generated exactly as for the standard RSA
2. certificate for the public key n, e

Mediated signature *mRSA*

key generation

RSA, setup for 3 components

1. keys n, e, d generated exactly as for the standard RSA
2. certificate for the public key n, e
3. finalization keys f_1, f_2 generated **independently** from n, e, d

Mediated signature *mRSA*

key generation

RSA, setup for 3 components

1. keys n, e, d generated exactly as for the standard RSA
2. certificate for the public key n, e
3. finalization keys f_1, f_2 generated **independently** from n, e, d
4. the private key d updated to $d' = d - (f_1 + f_2)$

Mediated signature *mRSA*

key generation

RSA, setup for 3 components

1. keys n, e, d generated exactly as for the standard RSA
2. certificate for the public key n, e
3. finalization keys f_1, f_2 generated **independently** from n, e, d
4. the private key d updated to $d' = d - (f_1 + f_2)$
5. integer d' placed in the signing card, f_1 in the laptop of the owner, f_2 in the remote mediator server

Mediated signature *mRSA*
signing and verification

Signing procedure

1. to sign m use the card and d' in the regular way (if $d' < 0$ then $s' = ((\mathrm{padding}(m))^{|d'|})^{-1}$ mod n), send the result s' to the laptop
2. the laptop uses f_1 and sends $s'' = s' \cdot (\mathrm{padding}(m))^{f_1}$ mod n and $\mathrm{padding}(m)$ to the mediator server
3. the mediator server uses f_2 and creates the final signature $s = s'' \cdot (\mathrm{padding}(m))^{f_2}$ mod $n = (\mathrm{padding}(m))^{d}$ mod n

Verification

standard RSA, use the public key

Mediated signature *mRSA*

security features

Physical security

In order to create a signature against the will of the owner it is necessary to:

1. steal the crypto card
2. steal the laptop
3. break into mediator server (when the user blocks)

Mediated signature *mRSA*
security features 2

Disabling

In order to disable signing possibility:

1. block it in the laptop (login + password necessary to unblock), and/or
2. block it in mediator server (password+ presence of the E-ID card necessary to unblock)

Monitoring

supervision systems blocking/suspending suspicious transactions
(e.g. requests coming from different IP's at the same time)

Mediated signature *mRSA*

security reduction

Formal security proof

reduction security of mRSA in the form presented can be reduced to standard RSA

corollary mRSA as secure as RSA

Key privacy

key usage all keys MUST be used to create a signature,

compromised card exposing d shows neither f_1 nor f_2
in the standard RSA d shows everything and the scheme completely broken

compromised laptop exposing f_1 shows no information on d and f_2

compromised mediator server exposing f_2 shows no information on d and f_1

Mediated signature *mRSA*

traces

Hash chain

The mediator server may store cryptographic/physical/published logs of the operations performed so that:

1. existence of a transaction undeniable even if crypto broken after some years
2. no way to *sign in the past*

Mediated signature *mRSA*

signing time

Undeniability of signing time

1. The mediator server refuses to finalize signature if declared signing time not fresh.
2. traces left in the hash chain.

Verification

1. just execute RSA verification

Mediated signature *mRSA*

comparison with Oasis DSS

Features

1. unlike classical X.509 solutions both guarantee signing time
2. VA-DSS incorporates any signing scheme as plug-in, mediated architecture trivial for RSA but requires more complicated computing procedure for DSA
3. VA-DSS quite heavy (like any universal tool)
4. obviously, one can incorporate fields from VA-DSS to mediated signature, if needed (time stamp,...)

Mediated signature *mRSA*

comparison with DSA, ECDSA

Mediated signatures with DSA

1. formal security proofs are published
2. modifications in the signing procedure less trivial, require 2 HSM's instead of one (this method)

Long time security
general situation

Perspectives of long time security

1. from scientific point of view advances in cryptanalysis unpredictable
2. computing power more predictable, but this is not decisive here due to high security margins
3. any guarantees are not honest, recommendations (like formulated by German authorities) are just orientation point and can change rapidly
4. the system must be prepared to rapid changes (e.g. freezing all e-ID cards until the case is cleared)

Long time security
comparisons

DSA, ECDSA, RSA

hard to say which technology will last longer

1. no serious results showing definite advantage of one technology over the other one
   it is an area of science fiction
2. DSA and ECDSA practically harder to audit since failure of randomness means immediately key exposure,
   it is mathematically impossible to prove that a string is random

Conclusions

1. re-using the approach designed for qualified signatures for some crucial components of e-ID seems to be the worst solution
2. practical security level can be much higher than for the classical solutions
3. the users should have the right to use mediated solutions

Thanks for your attention!

Contact data

1. Miroslaw.Kutylowski@pwr.wroc.pl
2. http://kutylowski.im.pwr.wroc.pl
3. +48 71 3202109, fax: +48 71 320 2105

Acknowledgment

we are grateful for the support from Polish Ministry of Science and Higher Education for support given to Wrocław University of Technology and TICONS,
many thanks for CryptoTech for cooperation in building feasibility prototype