Insecurity of Anonymous Login with German Personal Identity Cards

Lucjan Hanzlik, Kamil Kluczniak, Mirosław Kutyłowski[1]

Wrocław University of Technology, Poland

SocialSec 2015, Hangzhou

[1] speaker

Electronic ID documents

E-ID

Necessity for ID documents with a chip

- traditional security printing is not reliable enough:
  - race between authorities and sophisticated forgers
  - a personal ID document should be used for (10) years
- **cryptographic protection – independent and relatively long lasting**

Identity document with a memory chip - a simplest solution

- the printed data stored also on the chip,
- . . . and signed by the document issuer

E-ID

Identity document with a memory chip - a simplest solution

- the printed data stored also on the chip,
- . . . and signed by the document issuer

Side effect: severe privacy problems

E-ID

Identity document with a memory chip - a simplest solution

- the printed data stored also on the chip,
- . . . and signed by the document issuer

Side effect: severe privacy problems

- personal data signed by the state authorities are attractive for **illegal trading** – quality is guaranteed!

E-ID

Identity document with a memory chip - a simplest solution

- the printed data stored also on the chip,
- . . . and signed by the document issuer

Side effect: severe privacy problems

- personal data signed by the state authorities are attractive for **illegal trading** – quality is guaranteed!
- for durability reasons, the chip of the e-passport should communicate via a wireless interface
  – so **skimming is possible**

Privacy protection consequences

requirements

- access to data stored in the eID must be secured by the chip of eID
- the eID has to verify that the terminal asking for data has the right to get this data

$\Rightarrow$ nontrivial (cryptographic) procedures

Privacy protection consequences

requirements

- access to data stored in the eID must be secured by the chip of eID
- the eID has to verify that the terminal asking for data has the right to get this data

⇒ nontrivial (cryptographic) procedures

consequences:

- eID chip has to execute cryptographic protocols (crypto coprocessor is a MUST)

Privacy protection consequences

conclusion

we have to employ strong cryptography for eID documents, so why not use it **online** ?

ICAO standard solutions

BAC - Basic Access Control: session key derived from a personal data readable via an optical channel (relatively insecure protocol)

ICAO standard solutions

BAC - Basic Access Control: session key derived from a personal data readable via an optical channel (relatively insecure protocol)

EAC - Extended Access Control: both Chip Authentication and Terminal Authentication - to authenticate both the eID chip and the terminal in a cryptographic way

E-Passports

ICAO standard solutions

BAC - Basic Access Control: session key derived from a personal data readable via an optical channel (relatively insecure protocol)

EAC - Extended Access Control: both Chip Authentication and Terminal Authentication - to authenticate both the eID chip and the terminal in a cryptographic way

PACE - Password Authenticated Communication Establishment: the user has to enter the password to the reader, protocol immune against offline attacks

ICAO standard solutions

- **BAC** - Basic Access Control: session key derived from a personal data readable via an optical channel (relatively insecure protocol)
- **EAC** - Extended Access Control: both Chip Authentication and Terminal Authentication - to authenticate both the eID chip and the terminal in a cryptographic way
- **PACE** - Password Authenticated Communication Establishment: the user has to enter the password to the reader, protocol immune against offline attacks
- **CAM** - PACE combined with Chip Authentication, but more efficient than the protocol executed separately

the focus of ICAO specification

- border control - document inspection
- enabling automatic border control
- no anonymity

German personal ID card

Main components

- **Terminal Authentication** - checking terminal's access rights
- **Chip Authentication** - checking originality of a chip
- **Restricted Identification** - anonymous authentication
- **PACE** - enabling chip operation with a password

as well as place for qualified signatures

Specifications:

> *BSI Technische Richtlinie 03110: Advanced Security Mechanisms for Machine Readable Travel Document*

Terminal Authentication v. 2
protocol specification of BSI

	terminal		e-ID chip
		$\xrightarrow{cert(PK_{PCD})}$	
1.			verify $cert(PK_{PCD})$, extract PK_{PCD}
2.	choose $\widetilde{SK_{PCD}}$ at random $\widetilde{PK_{PCD}} := g^{\widetilde{SK_{PCD}}}$ compute commitment $Comp(\widetilde{PK_{PCD}})$		
		$\xrightarrow{Comp(\widetilde{PK_{PCD}})}$	
3.			choose r at random
		$\xleftarrow{r}$	
4.	$s := Sign_{SK_{PCD}}(ID_{PICC} \| r \| Comp(\widetilde{PK_{PCD}}) \|)$		
		$\xrightarrow{s}$	
5.			verify s

Chip Authentication

	terminal		e-ID chip
			static key pair (SK_{PICC}, PK_{PICC})
6.		$\xleftarrow{PK_{PICC}}$	
7.		$\xrightarrow{\widetilde{PK_{PCD}}}$	
8.	$\mathcal{K} := (PK_{PICC})^{\widetilde{SK_{PCD}}}$		$\mathcal{K} := (\widetilde{PK_{PCD}})^{SK_{PICC}}$
9.		$\xleftarrow{TAG, r'}$	choose r' at random $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ $TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$
10.	$\mathcal{K}' := Hash_1(\mathcal{K}, r')$ $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$		
11.	$TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$		

Restricted Identification

Restricted Identification concept

Domains
each domain is an autonomous system such that

- user's personal data are **processed only within the system** (unless a special event occurs)
- within a domain the user appears under his **domain specific identity/pseudonym**
- it should be **infeasible to link** identities of one user in two different domains

Restricted Identification concept

Domains

each domain is an autonomous system such that

- user's personal data are **processed only within the system** (unless a special event occurs)
- within a domain the user appears under his **domain specific identity/pseudonym**
- it should be **infeasible to link** identities of one user in two different domains

Background

- full disclosure of identity is not really necessary
- unnecessary data flow is a privacy risk
- a kind of privacy-by-design

Restricted Identification:

1. e-ID card computes a unique password for each domain
2. the terminal of the domain:
   a) checks that it is talking with an e-ID card
   b) receives a password
   c) checks the password against its blacklist

Restricted Identification

Core RI procedure
(notation according to BSI specification)

Terminal		e-ID chip
holds $\mathcal{K}'$		holds $\mathcal{K}'$
$\sigma := \mathrm{ENC}_{\mathcal{K}'}(PK_{\mathrm{sector}})$	$\xrightarrow{\sigma}$	
		$PK_{\mathrm{sector}} := \mathrm{DEC}_{\mathcal{K}'}(\sigma)$
		$I_{ID}^{\mathrm{sector}} := \mathrm{Hash}(\,(PK_{\mathrm{sector}})^{SK_{ID}})$
		$\sigma' := \mathrm{ENC}_{\mathcal{K}'}(I_{ID}^{\mathrm{sector}})$
$I_{ID}^{\mathrm{sector}} := \mathrm{DEC}_{\mathcal{K}'}(\sigma')$	$\xleftarrow{\sigma'}$	
check if I_{ID}^{sector} is on sector's black-list		

$\mathcal{K}'$ is a shared key that must be established **before** running RI

Security background

- since the chip is assumed to be secure, we have to believe that the eID really sends $I_{ID}^{\mathrm{sector}} := \mathrm{Hash}((PK_{\mathrm{sector}})^{SK_{ID}})$ using its private RI key SK_{ID}

Insecurity of RI

Hanzlik, Kluczniak, Kutyłowski

ID documents
E-ID
TA and ChA
RI
RI concept
RI concept
RI on German eID
Possible defense
BSI eIDAS

Blacklist

- a list of values $\mathrm{Hash}((PK_{\mathrm{sector}})^x)$, where x belongs to a banned person

Blacklisting a user

- the Issuing Authority holds the public key $PK = g^x$ of that user
- $PK_{\mathrm{sector}} = g^{r \cdot R}$, where
  - r is known to the Issuing Authority
  - R is known to the domain authority
- two steps:
  - the Issuing Authority computes $P_1 = PK^r$
  - the domain authority computes P_1^R

note that $P_1^R = PK^{r \cdot R} = g^{x \cdot r \cdot R} = (g^{r \cdot R})^x = (PK_{\mathrm{sector}})^x$

Blacklisting properties:

- the Issuing Authority does not learn the password of the revoked user
- **the terminal has to know that it is really talking with a valid eID**
  otherwise a random response would be accepted as a valid pseudonym – it is unlikely that it appear on the blacklist

Challenge

- the terminal must check that it is talking with a valid eID
- **there are many authentication protocols – but how to hide the identity of the chip?**
  standard solutions use something (e.g. a public key) that would link RI passwords in different domains

Group key

Design decision

- authentication of an eID via Chip Authentication with a **group key**
  it does not mean using group signatures
- a large number of eIDs share the same group key – a big *anonymity set*

Realistic attack assumptions

Are group keys really protected?

- a really powerful adversary can break into an eID chip and read its secrets
  – breaking into just one eID of the group is enough!
- if a group key has to be installed in a large number of devices, it must be stored and protected outside the eIDs
- it suffices to provide just one tampered raw eID for personalization – it would reveal the secret (group key) in response to a secret command

what would be the consequences?

Known Attack: creating a fake ID

A fake eID

- contains a valid group key
- provides a random password during execution of the RI protocol

Properties

- **the fake eID works as long as RI is used**
- **impossible to blacklist the fake eID**

Main Attack

A powerful adversary

- learns the group key
- eavesdrops the communication with a domain server

	terminal		e-ID chip
	SK_{PCD} chosen at random		group key (SK_{group}, PK_{group})
6.		$\xleftarrow{PK_{group}}$	
7.		$\xrightarrow{\widetilde{PK_{PCD}}}$	
8.	$\mathcal{K} := (PK_{group})^{\widetilde{SK_{PCD}}}$		$\mathcal{K} := (\widetilde{PK_{PCD}})^{SK_{group}}$
9.			choose r' at random $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$
		$\xleftarrow{TAG, r'}$	$TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$
10.	$\mathcal{K}' := Hash_1(\mathcal{K}, r')$ $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$		
11.	$TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$		

Observation

- **the eID derives the session key with the group key SK_{group} - no ephemeral random values used**

$\Rightarrow$ **Adversary knowing SK_{group} can derive the session key K from eavesdropped communication**

Main Attack
RI Phase

Terminal		e-ID chip
holds $\mathcal{K}'$		holds $\mathcal{K}'$
$\sigma := \mathrm{ENC}_{\mathcal{K}'}(PK_{\mathrm{sector}})$	$\xrightarrow{\sigma}$	
		$PK_{\mathrm{sector}} := \mathrm{DEC}_{\mathcal{K}'}(\sigma)$
		$I_{ID}^{\mathrm{sector}} := \mathrm{Hash}(\,(PK_{\mathrm{sector}})^{SK_{ID}})$
		$\sigma' := \mathrm{ENC}_{\mathcal{K}'}(I_{ID}^{\mathrm{sector}})$
$I_{ID}^{\mathrm{sector}} := \mathrm{DEC}_{\mathcal{K}'}(\sigma')$	$\xleftarrow{\sigma'}$	
check if I_{ID}^{sector} is on sector's black-list		

Observation

- **the Adversary knows K'!**
- **the Adversary can decrypt σ' and get the domain password I_{ID}^{sector} of this user**

The Adversary

- connects to the server with user's account
- runs the RI protocol, with minor change:

Terminal		e-ID chip
holds $\mathcal{K}'$		holds $\mathcal{K}'$
$\sigma := \mathrm{ENC}_{\mathcal{K}'}(PK_{\mathrm{sector}})$	$\xrightarrow{\sigma}$	
		$PK_{\mathrm{sector}} := \mathrm{DEC}_{\mathcal{K}'}(\sigma)$ [2]
		take I_{ID}^{sector} learned in the previous phase of the attack
		$\sigma' := \mathrm{ENC}_{\mathcal{K}'}(I_{ID}^{\mathrm{sector}})$
$I_{ID}^{\mathrm{sector}} := \mathrm{DEC}_{\mathcal{K}'}(\sigma')$	$\xleftarrow{\sigma'}$	
check if I_{ID}^{sector} is on sector's black-list		

[2] the step may be ignored, as the Adversary knows PK_{sector}

Main Attack

Attack potential

an attacker may login to the user's account after a purely passive attack

It looks like an obvious trapdoor in the German personal identity cards.

Possible defense

Modified version of the protocol
Chip Authentication phase

	Terminal i		Chip
	Chip Authentication Phase		
6.			choose ρ at random
			$Y'_{group} := Y^{\rho}_{group}$
		$\xleftarrow{Y'_{group}}$	
	$\mathcal{K} := (Y'_{group})^{\widetilde{x_{i,T}}}$		$\mathcal{K} := (\widetilde{Y_i})^{x_{group} \cdot \rho}$
	choose r' at random		
	$\mathcal{K}_{\mathrm{MAC}} := \mathrm{Hash}(\mathcal{K}, r')$		
	$\mathrm{Tag} := \mathrm{MAC}(\mathcal{K}_{\mathrm{MAC}}, Y'_{group})$	$\xrightarrow{\mathrm{Tag}, r'}$	
			$\mathcal{K}_{\mathrm{MAC}} := \mathrm{Hash}(\mathcal{K}, r')$
			$\mathrm{Tag} \stackrel{?}{=} \mathrm{MAC}(\mathcal{K}_{\mathrm{MAC}}, Y'_{group})$
	$\mathcal{K}_{\mathrm{Enc}} := \mathrm{Hash}_1(\mathcal{K}, r')$		$\mathcal{K}_{\mathrm{Enc}} = \mathrm{Hash}_1(\mathcal{K}, r')$

Modified version of the protocol

Restricted Identification phase

	Terminal i		Chip
	Restricted Identification phase		
7.	$\sigma := \mathrm{Enc}_{\mathcal{K}_{\mathrm{Enc}}}(Y_{\mathrm{sector}})$	$\xrightarrow{\sigma}$	
8.			$Y_{\mathrm{sector}} := \mathrm{Dec}_{\mathcal{K}_{\mathrm{Enc}}}(\sigma)$ $ID_{\mathrm{User}} := \mathrm{Hash}_2((Y_{\mathrm{sector}})^{x_{\mathrm{RI}}})$ $\sigma' := \mathrm{Enc}_{\mathcal{K}_{\mathrm{Enc}}}(ID_{\mathrm{User}})$ $\sigma'' := \mathrm{Enc}_{\mathcal{K}_{\mathrm{Enc}}}(\rho)$ $\sigma''' := \mathrm{Enc}_{\mathcal{K}_{\mathrm{Enc}}}(Y_{group})$
		$\xleftarrow{\sigma',\sigma'',\sigma'''}$	
9.	$ID_{\mathrm{User}} := \mathrm{Dec}_{\mathcal{K}}(\sigma')$ is ID_{User} on sector's black-list? $\rho := \mathrm{Dec}_{\mathcal{K}}(\sigma'')$ $Y_{group} := \mathrm{Dec}_{\mathcal{K}}(\sigma''')$ check if $Y^{\rho}_{group} \stackrel{?}{=} Y'_{group}$		
	data exchange		
	whole communication secured by encryption with key $\mathcal{K}_{\mathrm{Enc}}$		

Modified version of the protocol

properties

Properties

- authentication of ChA phase becomes effective after establishing a secure channel
- the session key resulting from Chip Authentication depends on ephemeral values on the side of eID and therefore **cannot be derived from the group key alone**

BSI eIDAS

Patch by BSI

Idea: authenticating the chip via **domain signature**

- the terminal can check that the signature comes from a chip personalized by the document issuer
- no unique public key for a chip
- the public key used for signature verification derived separately for each domain (sector)

Patch by BSI

Idea: authenticating the chip via **domain signature**

- the terminal can check that the signature comes from a chip personalized by the document issuer
- no unique public key for a chip
- the public key used for signature verification derived separately for each domain (sector)

Properties: of the solution from BSI TR

- keys for an eID chip derived from **group secret key**
- . . . yet each eID holds different keys
- **leaking secret group key does not enable to impersonate a user**

Domain signatures

of course domain signatures have also different applications

a good topic for another (long) talk

in BSI TR 03110 renamed as *pseudonymous signatures*

Issuer's setup

- the secret keys z and x
- public keys $g_1, \quad g_2 = g_1^z, \quad y = g_1^x$

BSI algorithm
core algorithm

Issuer's setup

- the secret keys z and x
- public keys $g_1, \quad g_2 = g_1^z, \quad y = g_1^x$

Issuing an eID for user i

- choose $x_{2,i} \in \mathbb{Z}_p$ at random
- compute $x_{1,i} = x - z \cdot x_{2,i}$
- install $(x_{1,i}, x_{2,i})$ in the eID of the user i.

BSI algorithm

core algorithm

Issuer's setup

- the secret keys z and x
- public keys $g_1, \quad g_2 = g_1^z, \quad y = g_1^x$

Issuing an eID for user i

- choose $x_{2,i} \in \mathbb{Z}_p$ at random
- compute $x_{1,i} = x - z \cdot x_{2,i}$
- install $(x_{1,i}, x_{2,i})$ in the eID of the user i.

Signing m by Alice for domain D

- create domain specific pseudonym $dsnym = D^{x_{1,i}}$
- choose t_1, t_2 at random, $a_1 = g_1^{t_1} g_2^{t_2}$, $a_2 = D^{t_1}$
- $c = \mathrm{Hash}(D, dsnym, a_1, a_2, m)$
- $s_1 = t_1 - c \cdot x_{i,1}$, $s_2 = t_2 - c \cdot x_{i,2}$
- output the signature (c, s_1, s_2)

Signing m **by Alice for domain** D

- create domain specific pseudonym $dsnym = D^{x_{1,i}}$
- choose t_1, t_2 at random, $a_1 = g_1^{t_1} g_2^{t_2}$, $a_2 = D^{t_1}$
- $c = \mathrm{Hash}(D, dsnym, a_1, a_2, m)$
- $s_1 = t_1 - c \cdot x_{i,1}$, $s_2 = t_2 - c \cdot x_{i,2}$
- output the signature (c, s_1, s_2)

Signature verification

- compute $a_1 = y^c \cdot g_1^{s_1} \cdot g_2^{s_2}$, $\quad a_2 = dsnym^c \cdot D^{s_1}$
- output `valid` if $c = \mathrm{Hash}(D, dsnym, a_1, a_2, m)$ and $dsnym$ not on a blacklist

Seclusiveness problem

Attack:

- **break into just two eIDs**
- use private keys $x_{1,i}, x_{2,i}$ and $x_{1,j}, x_{2,j}$ to compute x, z based on the equations

$$x = x_{1,i} + z \cdot x_{2,i}$$
$$x = x_{1,j} + z \cdot x_{2,j}$$

- . . . and **create any number of fake eIDs** that would create proper domain signatures

Undeniability problem

Proof of interaction:

- every authentication based on signature leaves undeniable proof of user's activity
- sometimes the proof is required but otherwise it is a security threat in the system as the signature can serve as evidence against third parties

security rule: one should avoid generating data that can be misused

Thanks for your attention!

Contact data

1. Miroslaw.Kutylowski@pwr.edu.pl
2. http://kutylowski.im.pwr.edu.pl