# Electronic identity documents

Mirosław Kutyłowski

Wrocław University of Technology

INSCRYPT 2012, Beijing, Nov. 29
Tutorial

# Electronic Identification Documents – e-ID Introduction

## Graphical protection

it is harder and harder to secure against forgery with graphical means:

- document inspection requires knowledge ...
- ... and good eyes
- different countries use different methods

# Personal identity documents

## smart cards for eID

- electronic chip inside the card
- energy from outsize source
  (contacts or antenna - electromagnetic induction)
- communication wireless or traditional
- price falling down
  ($<$ 10 USD production cost for a reasonable e-ID card)
- small memory on the card (e.g. 64K for everything)

- de facto standard of **ICAO** (International Civil Aviation Organization)
- chip in the cover page, **wireless communication**
- **electronic copy** of owner's (printed) data
  additionally: **biometric data**
- **electronic signature** of the document issuer for owner's data
- option: **active authentication** with a private key stored in the passport's chip

# eID - functionalities

## Opportunities

- **preventing forgery:**
  - repeating the same data in electronic layer
  - cryptographic protection - like signing *data groups*
- **control over the system:** distribution, activation, . . .
- **electronic inspection:** automatic border control, biometric authentication (data checked on-site)
- **remote services:**
  - a service provider can check that an **ID card is present** on the other side of a remote link
  - eID can serve as a **personal cryptographic suite**

# Why personal ID cards?
## opportunities

### Advantages

- **one user – one eID card**
- issuing ID cards under **strict control** of the state
- well trained **proper behavior** of the eID owners
- a chance for standardization

### Limitations

- **limited memory, slow computation**
- **slow communication**
- **no own energy source**
- **dependence on terminals** (master-slave mode)
- **loosing ID cards**: forgotten, stolen, machine washed, damaged, . . .

# Requirements I



## Confirm data of the eID owner

- guarantee that the data printed are those entered **by the document issuer**
- **provide additional information** (like high resolution photo, fingerprints, . . . ) - and **confirm** by the document issuer

## Confirm validity of the eID

- check that the **document presented is a valid eID**
- check for **whom** the eID has been issued

# Requirements II



**Consent of the user**

- **prevent** using an eID by **third parties**,
- check that the **owner is willing** to present an e-ID or use it

**Confirm presence of the eID**

- check that **the eID is used for** establishing **a remote connection**

# Requirements III



**Privacy and Data Protection**

- **prevent illegal tracing** of eID holders in electronic way
- **prevent illegal collecting evidence** of legally performed interactions with an eID

# Privacy Issues of Electronic Identity

# Data Protection

## inspection of an eID

### Approach I

1. data stored on the chip are signed by the card issuer
2. data presented for inspection with signatures

### Approach II

1. data stored on the chip without signatures
2. data presented to the terminal via a secure channel, after strong authentication of the chip and channel creation

# Data Protection
## comparison



### Approach I

1. data stored on the chip are signed by the card issuer
2. data presented for inspection with signatures

### Approach II

1. data stored on the chip without signatures
2. data presented to the terminal via a secure channel, after strong authentication of the chip and channel creation

### Tamper resistance

Approach I: secure even if chip's memory is unprotected

Approach II: fully depends on security of the chip

# Data Protection

## comparison

### Approach I

1. data stored on the chip are signed by the card issuer
2. data presented for inspection with signatures

### Approach II

1. data stored on the chip without signatures
2. data presented to the terminal via a secure channel, after strong authentication of the chip and channel creation

### Protection against data misuse

Approach I: signed data can be (mis)used by third parties

Approach II: no proof of authenticity against third parties

# Data Protection
## comparison

### Approach I

1. data stored on the chip are signed by the card issuer
2. data presented for inspection with signatures

### Approach II

1. data stored on the chip without signatures
2. data presented to the terminal via a secure channel, after strong authentication of the chip and channel creation

### Complexity

Approach I: easy to implement

Approach II: requires careful design of protocols

# Authentication of eID

## Approach I

1. challenge-response protocol
2. response with the private key of eID

## Approach II

1. key exchange, zero-knowledge protocol
2. proof of possession of the secret key via derivation of session key(s)

# Authentication of eID

## comparison



### Approach I

1. challenge-response protocol
2. response with the private key of eID

### Approach II

1. key exchange, zero-knowledge protocol
2. proof of possession of the secret key via derivation of session key(s)

### Proof transferability

Approach I: creates undeniable proof of presence

Approach II: authentication not transferable

# Authentication of eID
## comparison

### Approach I

1. challenge-response protocol
2. response with the private key of eID

### Approach II

1. key exchange, zero-knowledge protocol
2. proof of possession of the secret key via derivation of session key(s)

### Eavesdropping

Approach I: additional protection necessary

Approach II: immune by design

# Authentication of eID
## comparison

**Approach I**

1. challenge-response protocol
2. response with the private key of eID

**Approach II**

1. key exchange, zero-knowledge protocol
2. proof of possession of the secret key via derivation of session key(s)

**PKI**

necessary confirmation of private keys used by the eID

# Anonymous Authentication

## Approach I - anonymous credentials

1. credentials issued and uploaded to eID
2. authentication protocols executed - no direct connection to eID
3. unlimited number of anonymous identities for a user, many identities for the same attributes for one person possible

## Approach II - restricted identification

1. one key on eID for all attributes
2. protocol executed from an eID only
3. one attribute - one anonymous identity

# Anonymous Authentication

## Unlinkability

it is infeasible to decide whether two anonymous identities for different attributes represent the same person:

- adversary can analyze the protocols
- .. even as a terminal

## Complexity

AC still quite heavy

RI simple, cheap

# Electronic Passport

# Basic Access Control and Active Authentication

# Basic Access Control
the basic scheme for biometric passports



## Protocol overview

- each passport has a **"private" key for communication**
- this key is **derived directly** from the passport data read from MRZ (Machine Readable Zone)
- **authentication** of the passport based on **knowledge of this key**
- establishing communication (mutual authentication) with session key derivation so that **different sessions do not mix**

strong points: very **easy** to deploy

weak points: **serious** security problems

ICAO standard

# BAC Protocol Description

## “private key” derivation and usage



| | Terminal | | Chip |
|---|---|---|---|
| 1. | optical reading | $D$ <br> $<\!\!-\!\!-$ | $D$ in MRZ |
| 2. | $K_{\mathrm{MAC}} := F_1(D)$ <br> $K_{\mathrm{ENC}} := F_2(D)$ | | |
| 3. | | $\mathrm{RND}_{\mathrm{ICC}}$ <br> $<\!\!-\!\!-$ | choose random $\mathrm{RND}_{\mathrm{ICC}}$ |
| 4. | choose random $K_{\mathrm{IFD}}, \mathrm{RND}_{\mathrm{IFD}}$ | | |
| 5. | $S := \mathrm{RND}_{\mathrm{ICC}} \| \mathrm{RND}_{\mathrm{IFD}} \| K_{\mathrm{IFD}}$ | | |
| 6. | $E_{IFD} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(S)$ <br> $M_{IFD} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{IFD})$ | $E_{IFD}, M_{IFD}$ <br> $-\!\!-\!\!>$ | |
| 7. | | | decrypt at check if $\mathrm{RND}_{\mathrm{ICC}}$ obtained |
| 8. | | | choose random $K_{\mathrm{ICC}}$ |
| 9. | | | $R := \mathrm{RND}_{\mathrm{IFD}} \| \mathrm{RND}_{\mathrm{ICC}} \| K_{\mathrm{ICC}}$ |
| 10. | | $E_{ICC}, M_{ICC}$ <br> $<\!\!-\!\!-$ | $E_{ICC} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(R)$ <br> $M_{ICC} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{ICC})$ |
| 11. | check MAC, decrypt <br> check $\mathrm{RND}_{IFD}$ | | |
| ... | Secure Messaging | with | $KS_{ENC}$ $KS_{MAC}$ |

# BAC Protocol Description

## mutual authentication



| | Terminal | | Chip |
|---|---|---|---|
| 1. | optical reading | $D$<br>$\longleftarrow$ | $D$ in MRZ |
| 2. | $K_{\mathrm{MAC}} := F_1(D)$<br>$K_{\mathrm{ENC}} := F_2(D)$ | | |
| 3. | | $\mathrm{RND}_{\mathrm{ICC}}$<br>$\longleftarrow$ | choose random $\mathrm{RND}_{\mathrm{ICC}}$ |
| 4. | choose random $K_{\mathrm{IFD}}, \mathrm{RND}_{\mathrm{IFD}}$ | | |
| 5. | $S := \mathrm{RND}_{\mathrm{ICC}} \mid \mathrm{RND}_{\mathrm{IFD}} \mid K_{\mathrm{IFD}}$ | | |
| 6. | $E_{IFD} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(S)$<br>$M_{IFD} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{IFD})$ | $E_{IFD}, M_{IFD}$<br>$\longrightarrow$ | |
| 7. | | | decrypt at check if $\mathrm{RND}_{\mathrm{ICC}}$ obtained |
| 8. | | | choose random $K_{\mathrm{ICC}}$ |
| 9. | | | $R := \mathrm{RND}_{\mathrm{IFD}} \mid \mathrm{RND}_{\mathrm{ICC}} \mid K_{\mathrm{ICC}}$ |
| 10. | | $E_{ICC}, M_{ICC}$<br>$\longleftarrow$ | $E_{ICC} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(R)$<br>$M_{ICC} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{ICC})$ |
| 11. | check MAC, decrypt<br>check $\mathrm{RND}_{IFD}$ | | |
| ... | Secure Messaging | with | $KS_{ENC}$ $KS_{MAC}$ |

# BAC Protocol Description

## mutual authentication

| | Terminal | | Chip |
|---|---|---|---|
| 1. | optical reading | $D$ <br> $<\!\!\longleftarrow$ | $D$ in MRZ |
| 2. | $K_{\mathrm{MAC}} := F_1(D)$ <br> $K_{\mathrm{ENC}} := F_2(D)$ | | |
| 3. | | $\mathrm{RND}_{\mathrm{ICC}}$ <br> $<\!\!\longleftarrow$ | choose random $\mathrm{RND}_{\mathrm{ICC}}$ |
| 4. | choose random $K_{\mathrm{IFD}}, \mathrm{RND}_{\mathrm{IFD}}$ | | |
| 5. | $S := \mathrm{RND}_{\mathrm{ICC}} \mid \mathrm{RND}_{\mathrm{IFD}} \mid K_{\mathrm{IFD}}$ | | |
| 6. | $E_{IFD} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(S)$ <br> $M_{IFD} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{IFD})$ | $E_{IFD}, M_{IFD}$ <br> $\longrightarrow\!\!>$ | |
| 7. | | | decrypt at check if $\mathrm{RND}_{\mathrm{ICC}}$ obtained |
| 8. | | | choose random $K_{\mathrm{ICC}}$ |
| 9. | | | $R := \mathrm{RND}_{\mathrm{IFD}} \mid \mathrm{RND}_{\mathrm{ICC}} \mid K_{\mathrm{ICC}}$ |
| 10. | | $E_{ICC}, M_{ICC}$ <br> $<\!\!\longleftarrow$ | $E_{ICC} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(R)$ <br> $M_{ICC} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{ICC})$ |
| 11. | check MAC, decrypt <br> check $\mathrm{RND}_{IFD}$ | | |
| ... | Secure Messaging | with | $KS_{ENC}$ $KS_{MAC}$ |

# BAC Protocol Description

## Derivation of session keys



| | Terminal | | Chip |
|---|---|---|---|
| 1. | ... | ... | ... |
| 2. | ... | ... | ... |
| 3. | ... | ... | ... |
| 4. | choose random $K_{\mathrm{IFD}}$, $\mathrm{RND}_{\mathrm{IFD}}$ | | |
| 5. | $S := \mathrm{RND}_{\mathrm{ICC}} \| \mathrm{RND}_{\mathrm{IFD}} \| K_{\mathrm{IFD}}$ | | |
| 6. | $E_{IFD} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(S)$<br>$M_{IFD} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{IFD})$ | $E_{IFD}, M_{IFD}$<br>——> | |
| 7. | ... | ... | ... |
| 8. | | | choose random $K_{\mathrm{ICC}}$ |
| 9. | | | $R := \mathrm{RND}_{\mathrm{IFD}} \| \mathrm{RND}_{\mathrm{ICC}} \| K_{\mathrm{ICC}}$ |
| 10. | | $E_{ICC}, M_{ICC}$<br><—— | $E_{ICC} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(R)$<br>$M_{ICC} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{ICC})$ |
| 11. | ... | ... | ... |
| 12. | $KS_{ENC} \| KS_{MAC} \| \ldots := 3DES(K_{ICC} \text{ XOR } K_{IFD})$ | | $KS_{ENC} \| KS_{MAC} \| \ldots := 3DES(K_{ICC} \text{ XOR } K_{IFD})$ |
| ... | communication | with | $KS_{ENC}$ $KS_{MAC}$ |

# BAC weaknesses

## Attack 1

if you know data from the passport MRZ, you can **understand** the whole communication and even **hijack it**

## Attack 2

if you have a record of some past interactions and data from MRZ of Alice's passport, then you can **fish out** communications with her passport and **understand** it

| | Terminal | | Chip |
|---|---|---|---|
| 1. | optical reading | $D$ <br> $<$—— | $D$ in MRZ |
| 2. | $K_{\mathrm{MAC}} := F_1(D)$ <br> $K_{\mathrm{ENC}} := F_2(D)$ | | |
| 6. | $E_{IFD} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(S)$ <br> $M_{IFD} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{IFD})$ | $E_{IFD}, M_{IFD}$ <br> ——$>$ | |
| 10. | | $E_{ICC}, M_{ICC}$ <br> $<$—— | $E_{ICC} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(R)$ <br> $M_{ICC} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{ICC})$ |
| 12. | $KS_{ENC} \| KS_{MAC} \| \ldots :=$ <br> $3DES(K_{ICC}\ \mathrm{XOR}\ K_{IFD})$ | | $KS_{ENC} \| KS_{MAC} \| \ldots :=$ <br> $3DES(K_{ICC}\ \mathrm{XOR}\ K_{IFD})$ |

# BAC weaknesses

## Attack 3

After just one interaction with a passport it is easy to create a **perfect clone** of the electronic part of the passport

| | Terminal | | Chip |
|---|---|---|---|
| 1. | optical reading | $D$ <br> $<$—— | $D$ in MRZ |
| 2. | $K_{\mathrm{MAC}} := F_1(D)$ <br> $K_{\mathrm{ENC}} := F_2(D)$ | | |
| 6. | $E_{IFD} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(S)$ <br> $M_{IFD} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{IFD})$ | $E_{IFD}, M_{IFD}$ <br> ——$>$ | |
| 10. | | $E_{ICC}, M_{ICC}$ <br> $<$—— | $E_{ICC} := \mathrm{Enc}_{K_{\mathrm{ENC}}}(R)$ <br> $M_{ICC} := \mathrm{MAC}_{K_{\mathrm{MAC}}}(E_{ICC})$ |
| 12. | $KS_{ENC} \| KS_{MAC} \| \ldots :=$ <br> $3DES(K_{ICC}\ \mathrm{XOR}\ K_{IFD})$ | | $KS_{ENC} \| KS_{MAC} \| \ldots :=$ <br> $3DES(K_{ICC}\ \mathrm{XOR}\ K_{IFD})$ |

# BAC advantages

## No PKI infrastructure

- no coordination between countries required (apart from the common standard)
- no lists of public keys etc

## Chip

- if memory not well protected there is no sense to implement any stronger cryptography based on private keys
- only basic symmetric operations

BAC is a pragmatic solution given the tradeoff between security and simplicity

# Active Authentication

## Chip

- we assume that it can keep the secrets securely
- . . . and can use asymmetric cryptography

## Automatic inspection

- automatic border inspection is easy if based on:
  - wireless inspection of electronic part
  - and optionally: biometrics
- ... but then the chip must be **resistant to cloning**

## Active Authentication

**If the chips are tamper resistant, then we inspect a passport on possession of a secret key assigned to this passport.**

# Active Authentication

| Terminal | | Chip |
|---|---|---|
| ... | ... | ... |
| | $E(KP_{u_{AA}}, \text{signature})$ | public key $KP_{u_{AA}}$ |
| | <—— | |
| ... | ... | ... |
| choose random $\text{RND}_{IFD}$ | $E(\text{RND}_{IFD})$ | |
| | ——> | |
| | | choose random nonce $c$ |
| | $\sigma$ | $\sigma := \text{Sign}_{KP_{r_{AA}}}(\text{RND}_{IFD}, c)$ |
| | <—— | |
| verify $\sigma$ with $KP_{u_{AA}}$ | | |

solved: **cloning** requires retrieving signing key $KP_{r_{AA}}$ from secure memory of the passport chip

unsolved: illegal **tracing** still easy

# Extended Access Control (EAC)

# EAC
## goals

### Privacy issues

- **strong concerns of the citizens in Europe** about possible collection of data by the state and/or organized crime:
  - German constitution even forbids state systems that can be used for unnecessary collection of personal data
  - fears of “Big Brother”
- **easy spying** based on electronic artefacts – **high quality undeniable output**

### Goal

- the eID document **talks only with authenticated terminals**
- **identity information not revealed** even to the terminal **until terminal authentication successfully terminated**

Note: French version of EAC fails to fulfill this property

ICAO Standard

# EAC
## overview

### Protocol

- **Terminal Authentication:** the terminal must authenticate itself against the chip
  (in particular the terminal proves that it should get the identity information)
  - strong asymmetric methods
  - so far privacy of the terminal is not a concern
- **Chip Authentication:** the chip must authenticate itself against the terminal
  - strong asymmetric methods
  - personal data must not leak

# Terminal Authentication v. 2

## protocol specification of BSI



| | Terminal | | Chip |
|---|---|---|---|
| 1. | | $cert(PK_{PCD})$ → | Verify $cert(PK_{PCD})$ and extract $PK_{PCD}$ |
| 2. | choose $\widetilde{SK_{PCD}}$ at random<br>$\widetilde{PK_{PCD}} := g^{\widetilde{SK_{PCD}}}$ | | |
| 3. | | $Comp(\widetilde{PK_{PCD}})$ →<br>$r$ ← | choose $r$ at random |
| 4. | $s := Sign_{SK_{PCD}}(ID_{PICC} \| r \| Comp(\widetilde{PK_{PCD}}) \|)$ | $s$ → | |
| 5. | | | Verify $s$ |

# Static Diffie-Hellman Authentication

## preliminaries

### Settings

- all computations in a group with hard DL problem
- e-ID card holds a secret $x$ and a certificate for public key $y = g^x$

| chip | | terminal |
|---|---|---|
| | | generate $a$ at random<br>compute $z = g^a$ |
| | $\xleftarrow{z}$ | |
| compute $K := F(z^x)$ | | compute $K := F(y^a)$ |
| communicate via a channel encrypted with $K$ | | communicate via a channel encrypted with $K$ |

# Static DH authentication

## properties



### 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 a fake transcript of a session – it suffices to write the responses of the chip by himself!

# Chip Authentication v.2

## chip presentation

| | Terminal | | Chip |
|---|---|---|---|
| | | | static key pair $(SK_{PICC}, PK_{PICC})$ |
| 6. | | $PK_{PICC}$ <—— | |
| 7. | | $\widetilde{PK_{PCD}}$ ——> | |
| 8. | $\mathcal{K} := (PK_{PICC})^{\widetilde{SK_{PCD}}}$ | | $\mathcal{K} := (\widetilde{PK_{PCD}})^{SK_{PICC}}$ |
| 9. | | | choose $r'$ at random<br>$\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$<br>$TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ |
| | | $TAG, r'$ <—— | |
| 10. | $\mathcal{K}' := Hash_1(\mathcal{K}, r')$<br>$\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ | | |
| 11. | check<br>$TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ | | |

| | Terminal | | Chip |
|---|---|---|---|
| | | | static key pair $(SK_{PICC}, PK_{PICC})$ |
| 6. | | $PK_{PICC}$ <—— | |
| 7. | | $\widetilde{PK_{PCD}}$ ——> | |
| 8. | $\mathcal{K} := (PK_{PICC})^{\widetilde{SK_{PCD}}}$ | | $\mathcal{K} := (\widetilde{PK_{PCD}})^{SK_{PICC}}$ |
| 9. | | | choose $r'$ at random<br>$\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$<br>$TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ |
| | | $TAG, r'$ <—— | |
| 10. | $\mathcal{K}' := Hash_1(\mathcal{K}, r')$<br>$\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ | | |
| 11. | check<br>$TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ | | |

| | Terminal | | Chip |
|---|---|---|---|
| | | | static key pair $(SK_{PICC}, PK_{PICC})$ |
| 6. | | $PK_{PICC}$ <br> $\longleftarrow$ | |
| 7. | | $\widetilde{PK}_{PCD}$ <br> $\longrightarrow$ | |
| 8. | $\mathcal{K} := (PK_{PICC})^{\widetilde{SK}_{PCD}}$ | | $\mathcal{K} := (\widetilde{PK}_{PCD})^{SK_{PICC}}$ |
| 9. | | | choose $r'$ at random <br> $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ <br> $TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK}_{PCD})$ |
| | | $TAG, r'$ <br> $\longleftarrow$ | |
| 10. | $\mathcal{K}' := Hash_1(\mathcal{K}, r')$ <br> $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ | | |
| 11. | check <br> $TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK}_{PCD})$ | | |

# Proof of Possession of the Key

## Chip Authentication v.2

| | Terminal | | Chip |
|---|---|---|---|
| | | | static key pair $(SK_{PICC}, PK_{PICC})$ |
| 6. | | $PK_{PICC}$ <—— | |
| 7. | | $\widetilde{PK_{PCD}}$ ——> | |
| 8. | $\mathcal{K} := (PK_{PICC})^{\widetilde{SK_{PCD}}}$ | | $\mathcal{K} := (\widetilde{PK_{PCD}})^{SK_{PICC}}$ |
| 9. | | | choose $r'$ at random<br>$\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$<br>$TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ |
| | | $TAG, r'$ <—— | |
| 10. | $\mathcal{K}' := Hash_1(\mathcal{K}, r')$<br>$\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ | | |
| 11. | check<br>$TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ | | |

# Proof of Possession of the Key

## Chip Authentication v.2



### Background

- the chip must not only derive the key based on static DH, but also prove that it has this key
- implicit proof of possession of the key - by sending workload data in a correct form
- EAC chooses explicit proof of possession
- the scheme based on properties of hash functions

### Formal proof

the last steps designed so that a formal proof possible

# Password Authentication

# Purpose of Password Authentication



## How to confirm that the owner is willing to activate eID?

- **contacts:** by insertion to a reader
- **CAN:** in case of wireless communication - a short number to be read by the reader
  replaces *insertion to the reader*
- **PIN:** secret short number entered by the owner to the reader
- **Password:** a longer password entered by the owner to the reader

## Limitations

- **physical access:** a third person holding a eID can easily pass the protocol
- **PIN and password:** entropy limited

  attacks by guessing:
  - the attacker **may guess the correct password**, then nothing can stop him ...
  - ... but failed authentication round should reveal **nothing more** but that the password was wrong

# Architecture

## Components

1. **eID**: holds the secrets but can communicate only with a reader
2. **reader**: communicates directly with the eID, has an input keyboard for introducing the password, communicates with terminal
3. **terminal**: terminal of the system with which the eID wishes to talk

## Password authentication

- between **eID** and **a reader**
- executed **locally** (no lookup etc, since this would mean activity of the eID)

# SPEKE

# Simple Simple Password Exponential Key Exchange - SPEKE



## Properties

1. **US patent**
   many decision makers regard it as a deadly disadvantage when eID are concerned
2. password is not sent in any form

## Parameters

$p = 2q + 1$, $p, q$ are primes, Discrete Logarithm Problem hard in $\mathbb{Z}_p^*$

# SPEKE Protocol

## password dependant random generator

| | Reader | | Chip |
|---|---|---|---|
| | $\pi$ typed in | | $g_\pi = Hash_1(\pi)^2 \bmod p$ stored |
| | $g_\pi = Hash_1(\pi)^2 \bmod p$ | | |
| | choose random $r$ | | |
| | $Y_R := {g_\pi}^r$ | $\xrightarrow{Y_R}$ | |
| | | | choose random $c$ |
| | | | $Y_C := {g_\pi}^c$ |
| | | | $K' := (Y_R^2)^c$ |
| | | $\xleftarrow{Y_C, k}$ | $k = Hash_{2a}(Y_C, Y_R, K', \pi)$ |
| | $K' := (Y_C^2)^r$ | | |
| | $k \stackrel{?}{=} Hash_{2a}(Y_C, Y_R, K', \pi)$ | | |
| | $k' = Hash_{2b}(Y_C, Y_R, K', \pi)$ | $\xrightarrow{k'}$ | |
| | | | $k' \stackrel{?}{=} Hash_{2b}(Y_C, Y_R, K', \pi)$ |
| | $K = Hash_3(Y_C, Y_R, K', \pi)$ | | $K = Hash_3(Y_C, Y_R, K', \pi)$ |
| | communication with $K$ | | |

- Squaring $Hash_1(\pi)$ has to guarantee that the result is of a prime order $q$.
- $Hash(\pi)^2$ is a “random” generator of the group of order $q$.

# SPEKE Protocol

## Diffie-Hellman key exchange with random generator



| | Reader | | Chip |
|---|---|---|---|
| | $\pi$ typed in | | $g_\pi = Hash_1(\pi)^2 \bmod p$ stored |
| | $g_\pi = Hash_1(\pi)^2 \bmod p$ | | |
| | choose random $r$ | | |
| | $Y_R := g_\pi^r$ | $\xrightarrow{Y_R}$ | |
| | | | choose random $c$ |
| | | | $Y_C := g_\pi^c$ |
| | | | $K' := (Y_R^2)^c$ |
| | | $\xleftarrow{Y_C, k}$ | $k = Hash_{2a}(Y_C, Y_R, K', \pi)$ |
| | $K' := (Y_C^2)^r$ | | |
| | $k \stackrel{?}{=} Hash_{2a}(Y_C, Y_R, K', \pi)$ | | |
| | $k' = Hash_{2b}(Y_C, Y_R, K', \pi)$ | $\xrightarrow{k'}$ | |
| | | | $k' \stackrel{?}{=} Hash_{2b}(Y_C, Y_R, K', \pi)$ |
| | $K = Hash_3(Y_C, Y_R, K', \pi)$ | | $K = Hash_3(Y_C, Y_R, K', \pi)$ |
| | communication with $K$ | | |

- The values $Y_C, Y_R$ must be different from $1, -1$ (otherwise $K$ insecure).
- squarings in $K' := (Y_C^2)^r$, $K' := (Y_R^2)^c$ for being in the group of order $q$

# SPEKE Protocol

## security against an eavesdropper

| Reader | | Chip |
|---|---|---|
| $\pi$ typed in | | $g_\pi = Hash_1(\pi)^2$ mod $p$ stored |
| $g_\pi = Hash_1(\pi)^2$ mod $p$ | | |
| choose random $r$ | | |
| $Y_R := g_\pi^r$ | $\xrightarrow{Y_R}$ | |
| | | choose random $c$ |
| | | $Y_C := g_\pi^c$ |
| | | $K' := (Y_R^2)^c$ |
| | $\xleftarrow{Y_C, k}$ | $k = Hash_{2a}(Y_C, Y_R, K', \pi)$ |
| $K' := (Y_C^2)^r$ | | |
| $k \stackrel{?}{=} Hash_{2a}(Y_C, Y_R, K', \pi)$ | | |
| $k' = Hash_{2b}(Y_C, Y_R, K', \pi)$ | $\xrightarrow{k'}$ | |
| | | $k' \stackrel{?}{=} Hash_{2b}(Y_C, Y_R, K', \pi)$ |
| $K = Hash_3(Y_C, Y_R, K', \pi)$ | | $K = Hash_3(Y_C, Y_R, K', \pi)$ |
| communication with $K$ | | |

- $Y_R$ is uniformly distributed in the group of order $q$
- $Y_C$ is uniformly distributed in the group of order $q$

# SPEKE Protocol

## Tags - proving possession of a key

| | Reader | | Chip |
|---|---|---|---|
| | $\pi$ typed in | | $g_\pi = Hash_1(\pi)^2 \bmod p$ stored |
| | $g_\pi = Hash_1(\pi)^2 \bmod p$ | | |
| | choose random $r$ | | |
| | $Y_R := g_\pi^r$ | $\xrightarrow{Y_R}$ | |
| | | | choose random $c$ |
| | | | $Y_C := g_\pi^c$ |
| | | | $K' := (Y_R^2)^c$ |
| | | $\xleftarrow{Y_C, k}$ | $k = Hash_{2a}(Y_C, Y_R, K', \pi)$ |
| | $K' := (Y_C^2)^r$ | | |
| | $k \stackrel{?}{=} Hash_{2a}(Y_C, Y_R, K', \pi)$ | | |
| | $k' = Hash_{2b}(Y_C, Y_R, K', \pi)$ | $\xrightarrow{k'}$ | |
| | | | $k' \stackrel{?}{=} Hash_{2b}(Y_C, Y_R, K', \pi)$ |
| | $K = Hash_3(Y_C, Y_R, K', \pi)$ | | $K = Hash_3(Y_C, Y_R, K', \pi)$ |
| | communication with $K$ | | |

# SPEKE Security

## Formal proof

although intuitively clear, a formal proof was not immediately presented

1. Random Oracle Model
2. based on *Decision Inverted-Additive Diffie-Hellman Problem*:
   distinguish distributions

$$(g^{1/x}, g^{1/y}, g^{1/(x+y)})$$

and

$$(g^{1/x}, g^{1/y}, g^{1/z})$$

# PACE Password Authentication

# PACE
## design outline

**Password Authenticated Connection Establishment**

1. **designed to be patent free**
   - new paradigm in computing: design an algorithm so that it does not resemble any patented one
   - sometimes requires considerable algorithmic and legal experience
2. establishes an authenticated encrypted channel if correct password given
3. main purpose was to secure wireless connections
4. password guessing as hard as possible:
   - passive eavesdropping brings no advantage,
   - a reader interacting with a chip may try one password per session (in case of SPEKE no more than 2 passwords may be checked at once)
5. standard
6. implemented in German personal ID cards, . . .

developed by BSI

# PACE
## parameters

| Card | Reader |
|---|---|
| holds: | holds: |
| $\pi$ - password | $\pi$ - password, input from owner |
| $x_A$ - private key | |
| $X_A = g^{x_A}$ - public key<br>$cert_A$ - certificate for $X_A$<br>$\mathcal{G} = (a, b, p, q, g, k)$ - parameters | |

# PACE

## password dependent data

| Card | | Reader |
|---|---|---|
| $\pi$ $x_A$, $X_A = g^{x_A}$ | | $\pi$ |
| $K_\pi := H(0\|\|\pi)$ <br> choose $s \leftarrow \mathbb{Z}_q$ <br> $z := ENC(K_\pi, s)$ | | $K_\pi := H(0\|\|\pi)$ |
| | $\xrightarrow{\mathcal{G}, z}$ | abort if $\mathcal{G}$ incorrect |
| choose $y_A \leftarrow \mathbb{Z}_q^*$ <br> $Y_A := g^{y_A}$ | | $s := DEC(K_\pi, z)$ <br> choose $y_B \leftarrow \mathbb{Z}_q^*$ <br> $Y_B := g^{y_B}$ |
| | $\xleftarrow{Y_B}$ <br> $\xrightarrow{Y_A}$ | |
| abort if $Y_B \notin \langle g \rangle \setminus \{1\}$ <br> $h := Y_B^{y_A}$, $\hat{g} := h \cdot g^s$ <br> choose $y'_A \leftarrow \mathbb{Z}_q^*$ <br> $Y'_A := \hat{g}^{y'_A}$ | | abort if $Y_A \notin \langle g \rangle \setminus \{1\}$ <br> $h := Y_A^{y_B}$, $\hat{g} := h \cdot g^s$ <br> choose $y'_B \leftarrow \mathbb{Z}_q^*$ <br> $Y'_B := \hat{g}^{y'_B}$ |
| | $\xleftarrow{Y'_B}$ <br> $\xrightarrow{Y'_A}$ | |
| check $Y'_B \neq Y_B$ <br> $K := Y_B'^{\,y'_A}$ <br> $K_{...} := H(...\|\|K)$ | | check $Y'_A \neq Y_A$ <br> $K := Y_A'^{\,y'_B}$ <br> $K_{...} := H(...\|\|K)$ |

# PACE

## first DH key exchange - base establishment



| Card | | Reader |
|---|---|---|
| $\pi\ x_A, X_A = g^{x_A}$ | | $\pi$ |
| $K_\pi := H(0\|\pi)$ | | $K_\pi := H(0\|\pi)$ |
| choose $s \leftarrow \mathbb{Z}_q$ | | |
| $z := ENC(K_\pi, s)$ | | |
| | $\xrightarrow{\mathcal{G}, z}$ | abort if $\mathcal{G}$ incorrect |
| | | $s := DEC(K_\pi, z)$ |
| choose $y_A \leftarrow \mathbb{Z}_q^*$ | | choose $y_B \leftarrow \mathbb{Z}_q^*$ |
| $Y_A := g^{y_A}$ | | $Y_B := g^{y_B}$ |
| | $\xleftarrow{Y_B}$ | |
| abort if $Y_B \notin \langle g \rangle \setminus \{1\}$ | $\xrightarrow{Y_A}$ | abort if $Y_A \notin \langle g \rangle \setminus \{1\}$ |
| $h := Y_B^{y_A}$, $\hat{g} := h \cdot g^s$ | | $h := Y_A^{y_B}$, $\hat{g} := h \cdot g^s$ |
| choose $y'_A \leftarrow \mathbb{Z}_q^*$ | | choose $y'_B \leftarrow \mathbb{Z}_q^*$ |
| $Y'_A := \hat{g}^{y'_A}$ | | $Y'_B := \hat{g}^{y'_B}$ |
| | $\xleftarrow{Y'_B}$ | |
| check $Y'_B \neq Y_B$ | $\xrightarrow{Y'_A}$ | check $Y'_A \neq Y_A$ |
| $K := {Y'_B}^{y'_A}$ | | $K := {Y'_A}^{y'_B}$ |
| $K_{\dots} := H(\dots\|K)$ | | $K_{\dots} := H(\dots\|K)$ |

| Card | | Reader |
|---|---|---|
| ... | | ... |
| choose $s \leftarrow \mathbb{Z}_q$ | | |
| $z := ENC(K_\pi, s)$ | | |
| | $\xrightarrow{\mathcal{G}, z}$ | abort if $\mathcal{G}$ incorrect |
| | | $s := DEC(K_\pi, z)$ |
| choose $y_A \leftarrow \mathbb{Z}_q^*$ | | choose $y_B \leftarrow \mathbb{Z}_q^*$ |
| $Y_A := g^{y_A}$ | | $Y_B := g^{y_B}$ |
| | $\xleftarrow{Y_B}$ | |
| abort if $Y_B \notin \langle g \rangle \backslash \{1\}$ | $\xrightarrow{Y_A}$ | abort if $Y_A \notin \langle g \rangle \backslash \{1\}$ |
| $h := Y_B^{y_A}$, $\hat{g} := h \cdot g^s$ | | $h := Y_A^{y_B}$, $\hat{g} := h \cdot g^s$ |
| ... | | ... |

- definition of $\hat{g}$ is so called **Generic Mapping** - PACE v1 Generic Mapping (PACE-GM). according to *ISO/IEC JTC1 SC17 WG3/TF5 for the International Civil Aviation Organization: Supple- mental access control for machine readable travel documents (2011)*
- Integrated Mapping (PACE-IM) from the same standard – specific operations for ECC, partially patented.

# PACE

## the second Diffie-Hellman for key establishment

| Card | | Reader |
|---|---|---|
| $\pi\ x_A,\ X_A = g^{x_A}$ | | $\pi$ |
| $K_\pi := H(0\|\|\pi)$<br>choose $s \leftarrow \mathbb{Z}_q$<br>$z := ENC(K_\pi, s)$ | | $K_\pi := H(0\|\|\pi)$ |
| | $\xrightarrow{\mathcal{G},z}$ | abort if $\mathcal{G}$ incorrect<br>$s := DEC(K_\pi, z)$ |
| choose $y_A \leftarrow \mathbb{Z}_q^*$<br>$Y_A := g^{y_A}$ | | choose $y_B \leftarrow \mathbb{Z}_q^*$<br>$Y_B := g^{y_B}$ |
| | $\xleftarrow{Y_B}$<br>$\xrightarrow{Y_A}$ | |
| abort if $Y_B \notin \langle g\rangle \backslash \{1\}$<br>$h := Y_B^{y_A}$, $\hat{g} := h \cdot g^s$<br>choose $y'_A \leftarrow \mathbb{Z}_q^*$<br>$Y'_A := \hat{g}^{y'_A}$ | | abort if $Y_A \notin \langle g\rangle \backslash \{1\}$<br>$h := Y_A^{y_B}$, $\hat{g} := h \cdot g^s$<br>choose $y'_B \leftarrow \mathbb{Z}_q^*$<br>$Y'_B := \hat{g}^{y'_B}$ |
| | $\xleftarrow{Y'_B}$ | |
| check $Y'_B \neq Y_B$ | $\xrightarrow{Y'_A}$ | check $Y'_A \neq Y_A$ |
| $K := {Y'_B}^{y'_A}$<br>$K_{\ldots} := H(\ldots\|\|K)$ | | $K := {Y'_A}^{y'_B}$<br>$K_{\ldots} := H(\ldots\|\|K)$ |

# PACE

## final phase - proof of possession and deriving keys

| Card | | Reader |
|---|---|---|
| ... | | ... |
| $K := {Y'_B}^{y'_A}$ | | $K := {Y'_A}^{y'_B}$ |
| $K_{ENC} := H(1\|\|K)$ | | $K_{ENC} := H(1\|\|K)$ |
| $K_{MAC} := H(2\|\|K)$ | | $K_{MAC} := H(2\|\|K)$ |
| $K'_{MAC} := H(3\|\|K)$ | | $K'_{MAC} := H(3\|\|K)$ |
| $T_A :=$ $MAC(K'_{MAC}, (Y'_B, \mathcal{G}))$ | | $T_B :=$ $MAC(K'_{MAC}, (Y'_A, \mathcal{G}))$ |
| | $\xleftarrow{T_B}$ | |
| abort if $T_B$ invalid | | |
| | $\xrightarrow{T_A}$ | |
| | | abort if $T_A$ invalid |

- chip interrupt if it discovers that the tag of the reader is wrong,
- until this moment **all data sent to the reader by the chip have uniform probability distribution for every password** ...
- ... and for **every choice of the reader**

# PACE

## final phase - proof of possession and deriving keys

| Card | | Reader |
|---|---|---|
| ... | | ... |
| $K := {Y'_B}^{y'_A}$ | | $K := {Y'_A}^{y'_B}$ |
| $K_{ENC} := H(1\|\|K)$ | | $K_{ENC} := H(1\|\|K)$ |
| $K_{MAC} := H(2\|\|K)$ | | $K_{MAC} := H(2\|\|K)$ |
| $K'_{MAC} := H(3\|\|K)$ | | $K'_{MAC} := H(3\|\|K)$ |
| $T_A := MAC(K'_{MAC}, (Y'_B, \mathcal{G}))$ | | $T_B := MAC(K'_{MAC}, (Y'_A, \mathcal{G}))$ |
| | $\xleftarrow{T_B}$ | |
| abort if $T_B$ invalid | | |
| | $\xrightarrow{T_A}$ | |
| | | abort if $T_A$ invalid |

- reader interrupt if it discovers that the tag of the chip is wrong (maybe the communication was hijacked by another device?)
- until this moment **the reader sent one message that depends on password**
  security is a more subtle issue

# Restricted Authentication

# Idea of sectors

# Identification
## classical approach

### Standard procedure

1. user identity proved
2. rights of the user determined
3. appropriate access granted

**Prove your identity, then I grant you access to resources**

### Problems

- full disclosure of identity is not really necessary
- unnecessary data flowing in the system is always a **security threat**
- particularly severe problems of **personal data protection rules** as in European Community:
  - high costs of protecting personal data
  - high legal risk of protection violation

# Austrian Concept of Sectors

## Idea of sectors

1. activity areas divided into **independent sectors**
2. **strict separation** between sectors, interaction only if explicitly defined
3. for each sector different authentication, interaction in different sectors **unlinkable**

## Sector examples

- health care system
- citizen-police contacts
- children protection
- psychological hotline
- electronic decision making - voting
- auction services
- discussion forums
- ...

# Citizen-police contacts

## Motivation

1. the witnesses of crime are often afraid to inform police:
   - they fear that policemen and criminals may cooperate
   - they fear that during court procedures they will be forced to act as witnesses

   ... but afterwards the (organized) crime may revenge
2. identity of a person is important during court procedure but not during investigation

## Electronic witness

1. strong authentication that a message comes from a physical person
2. the messages from the same person should be linkable

# Austria

## sketch of the solution





### Details

1. Bürgerkarte computes a password for each sector, the password computed from personal number and sector ID
2. central password verification – just like for PIN numbers of bank cards
3. **given two passwords from different sectors – it is unfeasible to say if they belong to the same person**

### Disadvantages

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

## Symmetric solution - automatic way of deriving sector logins

- ID for each sector computed from the personal ID number $i$, sector ID $s$ and a master key $K_i$ of the user:

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

- $K_i$ is recomputed on the fly by a secure server of a central authority
  solution analogous to ATM PIN mechanism:

$$K_i = F(i, K)$$

  where $K$ is the main secret of the authority

# German Restricted Identification

## Procedure

login in a sector:

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

# German Restricted Identification

## setting up a connection

### Overview

1. activating the card:
   PACE (password ... ) - a DH based protocol in which the reader shows that it knows the owner's password
   - immune against replay attacks
   - as good as it can be regarding small entropy of the password
2. Terminal Authentication:
   a protocol showing that the terminal is trustworthy,
   - system of certificates (CVCA)
   - static DH
3. Chip Authentication:
   ...

## Overview

1. activating the card:

   . . .

2. Terminal Authentication:

   . . .

3. Chip Authentication: the chip has to prove that it is a *Personalausweis*
   - **it is a challenge, since the card cannot show any identification information,**
   - current implementation **based on a *group key* shared by a large group of e-ID cards**

     ok, as long as the cards are really tamper resistant or RI used for non-sensitive areas

## Core RI procedure

| Terminal | | Chip |
|---|---|---|
| $\sigma := ENC_{\mathcal{K}'}(PK_{sector})$ | $\xrightarrow{\sigma}$ | |
| | | $PK_{sector} := DEC_{\mathcal{K}'}(\sigma)$ |
| | | $I_{ID}^{sector} := Hash_2(\,(PK_{sector})^{SK_{ID}})$ |
| | | $\sigma' := ENC_{\mathcal{K}'}(I_{ID}^{sector})$ |
| $I_{ID}^{sector} := DEC_{\mathcal{K}'}(\sigma')$<br>check if $I_{ID}^{sector}$ is on sector's black-list | $\xleftarrow{\sigma'}$ | |

## Assumptions and features

- since the chip of Personalausweis is assumed to be secure, we believe that the card really sends $I_{ID}^{sector} := Hash_2((PK_{sector})^{SK_{ID}})$ using its private RI key $SK_{ID}$
- **a malicious eID might cheat by sending some junk** – it would not be found on the black list with very high probability ...
  – not critical if RI is used for limited importance issues ...

## Blacklist

a list of values $Hash_2((PK_{sector})^x)$, where $x$ belongs to a banned person

## Excluding a user from a sector

- the password of a user in the sector computed in a two-party protocol by e-ID Authority issuing personal identity cards and a sector.
- a simple protocol based on DH mechanism

# German Restricted Identification

## blacklisting

### Computing public key of sectors

e.g. Diffie-Hellman Key agreement with

- **CVCA**:
  - private key $SK_{Revocation}$
  - public key $PK_{Revocation} = g^{SK_{Revocation}}$
- **Sector**: private key $SK_{Sector}$
- **Sector public key**: $PK_{Revocation} = g^{SK_{Sector}}$

### Revoking a user with public key $PK_{ID}$

1. CVCA computes $PK_{ID,Revocation} := PK_{ID}^{SK_{Revocation}}$
2. Sector computes $Hash_2((PK_{ID,Revocation})^{SK_{Sector}})$ and puts in the blacklist

$$(PK_{ID,Revocation})^{SK_{Sector}} = PK_{ID}^{SK_{Revocation} \cdot SK_{Sector}} = (g^{SK_{ID}})^{SK_{Revocation} \cdot SK_{Sector}}$$

$$= (g^{SK_{Revocation} \cdot SK_{Sector}})^{SK_{ID}} = PK_{Sector}^{SK_{ID}}$$

# White-list RI

# White-list approach

## PKI concept

a modification of a German scheme such that

1. management of users in a sector with
   - white-lists (list legitimate users) and/or . . .
   - . . . blacklists (list of excluded users)
2. each time a different password –
   the terminals need not to be trusted

## Intended primary application areas

access to medical data

Miroslaw Kutylowski, Lukasz Krzywiecki, Przemyslaw Kubiak, Michal Koza: Restricted Identification Scheme and Diffie-Hellman Linking Problem. INTRUST 2011: 221-238

E-ID

M.Kutyłowski

Introduction
Privacy issues
E-passport
BAC & AA
EAC
PAKE||PACE
SPEKE
PACE
RI
sectors
German RI
White-list RI
CHARI
Group key
CHARI
PACE||AA
PACE|AA
Domain sign
Proofs

## Keys in a sector

1. each e-ID card holds a single secret key $x$ for many sectors,
2. a sector $S_i$ holds a base key $PK_i = g^{\sigma_i}$, for $\sigma_i = r_i + R_i$, where $r_i$ is known to ID Authority, $R_i$ is a secret of $S_i$
3. the public keys of users in the sector with the base key $PK_i$ are

$$y_1^{\sigma_i}, y_2^{\sigma_i}, \ldots$$

where

$$y_1 = g^{x_1}, y_2 = g^{x_2}, \ldots$$

are the main public keys of the users

# Authentication
## White-list RI



| Terminal | | Chip |
|---|---|---|
| | $\xrightarrow{PK_i}$ | |
| | | $a := PK_i^x$ |
| | $\xleftarrow{a}$ | |
| check if $a$ is on white-list<br>choose random $t$<br>$c := (PK_i)^t$ | | |
| | $\xrightarrow{c}$ | |
| $K = Hash((a^t)$ | | $K = Hash(c^x)$ |
| tags for $K$ | | exchanged |

note that

$$a = PK_i^x = (g^{\sigma_i})^x = (g^x)^{\sigma_i} = y^{\sigma_i}$$

## Omitted details

Some additional mechanisms is the protocol:

1. the e-ID card must know that it talks with a terminal of a given sector
2. some additional mechanisms to allow a full equivalence between impersonation and computational Diffie-Hellman Problem

# Unlinkability
## White-list RI

### Unlinkability issues

- 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 $y_i^r$?**
- this turns to be as hard as DDH despite possible advantage of the adversary

# Anonymous Chip Authentication

# Extended Access Control (EAC) for RI



## RI protocol stack

**Terminal Authentication:** Terminal proves that it has the right to talk with Chip.

**Chip Authentication:** Chip proves that it is genuine – it proves to hold a secret key given by the document issuer.

**Restricted Identification:** Chip identifies and authenticates itself against Terminal using its identity specific to Terminal.

## Problems

- **Chip Authentication:** the chip has to prove that it is a genuine eID issued by appropriate authorities
  - **it is a challenge, since the card cannot show any identification information,**
  - current implementation **based on a *group key* shared by a large group of e-ID cards**
    ok, as long as the cards are really tamper resistant or RI used for non-sensitive areas

# Problems

## ChA Dilemma

If the Chip is using some special pair of keys for ChA, or any unique certificate, serial number, ... then it leaks the unique fingerprint, and unlinkability is gone!

## EAC solution and problem

- A **group key** is used by a set of Chips.
- Once a group key is leaked, it is easy to produce fake cards that authenticate via TA+ChA+RI.
- It is impossible to revoke a fake card with a random key used for RI and a genuine group key.

# Chip Authentication with Group Key



| Terminal | | Chip |
|---|---|---|
| | | group key pair $(SK_{gr}, PK_{gr})$ |
| | $PK_{gr}$ <br> $\longleftarrow$ | |
| | $\widetilde{PK_{PCD}}$ <br> $\longrightarrow$ | |
| $\mathcal{K} := (PK_{gr})^{\widetilde{SK_{PCD}}}$ | | $\mathcal{K} := (\widetilde{PK_{PCD}})^{SK_{gr}}$ <br> choose $r'$ at random <br> $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ <br> $TAG := MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ |
| | $TAG, r'$ <br> $\longleftarrow$ | |
| $\mathcal{K}' := Hash_1(\mathcal{K}, r')$ <br> $\mathcal{K}_{MAC} := Hash_3(\mathcal{K}, r')$ <br> check <br> $TAG \stackrel{?}{=} MAC_{\mathcal{K}_{MAC}}(\widetilde{PK_{PCD}})$ | | |

# ChARI Protocol
## properties

### Goal
eliminate group keys

### Crucial Properties
- RI key used instead of group key for ChA
- identity hidden until communication established
- Terminal Authentication unchanged

Lucjan Hanzlik, Kamil Kluczniak, Przemysław Kubiak, Mirosław Kutyłowski: Restricted Identification without Group Keys. TrustCom 2012: 1194-1199

# ChARI Protocol

## Chip Authentication + Restricted Identification – Part 1

| | Terminal | | Chip |
|---|---|---|---|
| 6. | | $\widetilde{I^{sector}_{ID}}$ <——— | choose $b$ at random<br>$\widetilde{I^{sector}_{ID}} := (PK_{sector})^{b \cdot SK_{ID}}$ |
| 7. | $\mathcal{K} := (\widetilde{I^{sector}_{ID}})^{\widetilde{SK_{PCD}}}$<br>choose $r'$ at random,<br>$\mathcal{K}_{MAC} := Hash_1(\mathcal{K}, r')$<br>$\mathcal{K}_{ENC} := Hash_2(\mathcal{K}, r')$ | | $\mathcal{K} := (\widetilde{PK_{PCD}})^{b \cdot SK_{ID}}$ |
| 8. | $TAG := MAC(\mathcal{K}_{MAC}, \widetilde{I^{sector}_{ID}})$ | $TAG, r'$ ———> | |
| 9. | | | $\mathcal{K}_{MAC} := Hash_1(\mathcal{K}, r')$<br>$\mathcal{K}_{ENC} := Hash_2(\mathcal{K}, r')$<br>check $\widetilde{TAG} \stackrel{?}{=} MAC(\mathcal{K}_{MAC}, I^{sector}_{ID})$ |

# ChARI Protocol

## Chip Authentication + Restricted Identification – Part 1



| | Terminal | | Chip |
|---|---|---|---|
| 6. | | | choose $b$ at random |
| | | $\widetilde{I_{ID}^{sector}}$ <br> $\longleftarrow$ | $\widetilde{I_{ID}^{sector}} := (PK_{sector})^{b \cdot SK_{ID}}$ |
| 7. | $\mathcal{K} := (\widetilde{I_{ID}^{sector}})^{\widetilde{SK_{PCD}}}$ | | $\mathcal{K} := (\widetilde{PK_{PCD}})^{b \cdot SK_{ID}}$ |
| | choose $r'$ at random, $\mathcal{K}_{MAC} := Hash_1(\mathcal{K}, r')$ $\mathcal{K}_{ENC} := Hash_2(\mathcal{K}, r')$ | | |
| 8. | $TAG := MAC(\mathcal{K}_{MAC}, \widetilde{I_{ID}^{sector}})$ | $TAG, r'$ <br> $\longrightarrow$ | |
| 9. | | | $\mathcal{K}_{MAC} := Hash_1(\mathcal{K}, r')$ $\mathcal{K}_{ENC} := Hash_2(\mathcal{K}, r')$ |
| | | | check $TAG \stackrel{?}{=} MAC(\mathcal{K}_{MAC}, \widetilde{I_{ID}^{sector}})$ |

# ChARI Protocol

## Chip Authentication + Restricted Identification – Part 2

| | Terminal(PCD) | | MRDT Chip |
|---|---|---|---|
| 10. | | $\sigma, \sigma'$ <br> $\longleftarrow$ | $\sigma := ENC_{\mathcal{K}_{ENC}}(cert(I_{ID}^{sector}))$ <br> or <br> $\sigma := ENC_{\mathcal{K}_{ENC}}(r)$ if white/black-list used <br> $\sigma' := ENC_{\mathcal{K}_{ENC}}(b)$ |
| 11. | $z := DEC_{\mathcal{K}_{ENC}}(\sigma)$ <br> $b := DEC_{\mathcal{K}_{ENC}}(\sigma')$ <br> $I_{ID}^{sector} := (\widetilde{I_{ID}^{sector}})^{b^{-1}}$ <br> verify that $I_{ID}^{sector}$ on white/black list or <br> verify the certificate | | |

# PACE and Active Authentication

# PACE|AA

## PACE and Active Authentication

1. PACE proves to the Chip that the correct password has been presented to the reader –presumably by the card owner
2. PACE does not prove to the terminal that the chip is genuine, **any chip knowing the password would succeed to establish communication**
3. standard solution: Chip Authentication running **after** PACE
   this is an **Active Authentication** – the chip proves to hold a secret that is stored (presumably) only on the chip

# PACE|AA
## design outline

### PACE|AA

- PACE and active authentication merged into one protocol
- Active Authentication **reusing exponentiations** from PACE

FC'2012, Jens Bender, Özgür Dagdelen, Marc Fischlin, Dennis Kügler: *The PACE|AA Protocol for Machine Readable Travel Documents, and Its Security*.

# PACE with AA



| Card | | Reader |
|---|---|---|
| $\pi$ $x_A$, $X_A = g^{x_A}$ | | $\pi$ |
| random $s$ chosen | $\xrightarrow{ENC(K_\pi, s)}$ | retrieve $s$ |
| choose $y_A \leftarrow \mathbb{Z}_q^*$ | | choose $y_B \leftarrow \mathbb{Z}_q^*$ |
| | $\xleftarrow{Y_B}$ | $Y_B := g^{y_B}$ |
| $Y_A := g^{y_A}$ | | |
| abort if ... | $\xrightarrow{Y_A}$ | abort if ... |
| $h := Y_B^{y_A}$, $\hat{g} := h \cdot g^s$ | | $h := Y_A^{y_B}$, $\hat{g} := h \cdot g^s$ |
| choose $y'_A \leftarrow \mathbb{Z}_q^*$ | | choose $y'_B \leftarrow \mathbb{Z}_q^*$ |
| | $\xleftarrow{Y'_B}$ | $Y'_B := \hat{g}^{y'_B}$ |
| $Y'_A := \hat{g}^{y'_A}$ | | |
| check ... | $\xrightarrow{Y'_A}$ | check ... |
| $K_{\dots} := H(\dots \vert\vert {Y'_B}^{y'_A})$ | | $K_{\dots} := H(\dots \vert\vert {Y'_A}^{y'_B})$ |
| ...tags checked | ... | ...tags checked |
| $\sigma := y_A +$ | $\xrightarrow{E_{K'_{SC}}(\sigma, cert_A))}$ | decrypt with $K'_{SC}$ |
| $H(5 \vert\vert Y_A, Y'_A) \cdot x_A$ | | check certificate $cert_A$ |
| | | $w := \sigma^{-1}$, $r := Y_A$ |
| | | $Y_A \stackrel{?}{=} g^{wH(5 \vert\vert Y_A, Y'_A)} X_A^{rw}$ |

# PACE|AA





## Protocol features

1. the last part is a Schnorr signature
2. exponentiation $Y_a := g^{y_A}$ used both for PACE and for signature creation

## Deniability

1. protocol data should not enable the terminal to prove that authentication between the card and the terminal took place
2. **faking** a transcript:
   - change the internal PACE computation on the card:

$$Y'_A := g^{y'_A}, \quad , c := H(Y'_A), \quad , Y_A := X_A^{-c} g^{y_A},$$

   - derive the signature: $s := y_A$
   - all values have exactly the same probability distribution as before

# Domain Signatures

# Signatures in different sectors

## Goal

1. use a different electronic signature in each sector
2. for signatures designated for sectors $A$ and $B$ it should be unfeasible to say if they come from the same person

## A trivial solution?

for each sector a different key pair

**wrong! we cannot afford it:** the memory space on a smart card is very limited, only a limited number of sectors possible (just a few)

## Detailed goal

design a signature scheme such that **one private key can be used for an arbitrary number of sectors**
but the signatures created for different sectors remain unlinkable

this solves the problem since the public keys and their certificates may be stored outside the smart card.

# Solution
## sector setup



### System parameters

- a group $G$ of a prime order, where Decisional Diffie-Hellman Problem is hard,
- a generator $g$ of $G$,
- a secure hash function $H_G : \{0,1\}^* \to G$

### Parameters for a sector $A$

- public key

$$g(A) := H_G(A)$$

where $A$ stands for the legal name of sector $A$

- (no private key)

# Person
setup



## Electronic personal identity card

Person $B$ holds an ID card obtained by ID-Authority:

1. the ID card generates and stores $x_B$, the private key of $B$
2. $y_B := g^{x_B}$ is the public key for $B$
3. the ID card holds a **certificate** for $y_B$ issued by ID-Authority

## Person $B$ registering to sector $A$

$B$ appears at ID-Authority

1. the ID card generates $p(A)_B := g(A)^{x_B}$
2. the ID card presents $p(A)_B$ to ID-Authority and **proves in a zero-knowledge way** that its discrete logarithm with respect to $g(A)$ is the same as discrete logarithm of $p_B$ with respect to $g$,
3. ID-Authority issues a **certificate** for $p(A)_B$ for sector $B$
   the certificate contains only a restricted subset of personal data of $B$

# Signatures of $B$ for sector $A$



## Creating a signature of $m$ by $B$

1. choose $r \in [1, q-1]$ uniformly at random, compute
$$R := (g(A))^r$$
2. $$S := H_q(g(A), p(A)_B, R, m) \cdot x_B + r \bmod q$$

$(R, S)$ is the signature of $m$, it comes together with the certificate of $p(A)_B$

## Signature verification

1. public key $p(A)_B$ retrieved from the certificate
2. verification test:

$$g(A)^S \stackrel{?}{=} (p(A)_B)^{H_q(g(A), p(A)_B, R, m)} \cdot R$$

# Security features

## Unforgeability

reduction to Discrete Logarithm Problem in ROM

## Privacy

Public keys $P(C)$, $P'(D)$ from sectors $C$ and $D$, and some signatures).
**Question: Are $P(C)$, $P'(D)$ are assigned to the same person?**

reduction to Decisional Diffie-Hellman Problem in the random oracle model

## Unlinkability

Given the public keys of Alice and Bob, and two public keys $X$ and $Y$ for sector $A$. We know that they belong to Alice and Bob.
**Question: which of them belongs to Alice and which to Bob?**

reduction to Decisional Diffie-Hellman Problem in the random oracle model

# German Problem of Certificates



## Grundgesetz - German Constitution

- $\approx$ the State must not keep centralized databases with personal data of citizens
- legal problems with solutions based on CRL, OCSP, certificates

## Target

design a solution so that verification does depend on external central database

# Solution

## Keys

main parameters :

- private: $z, x$
- public: $g, g_2 = g^z, y = g^x$

sector :

- private: $r$
- public: $R = g^r$

user :

- private: $x_2, x_1 = x - z \cdot x_2$

user in sector :

- private: $x_2, x_1 = x - z \cdot x_2$
- public: $nym = R^{x_1}$

# Domain-Specific Pseudonymous Signatures

## Keys

**main parameters:**

- private: $z$, $x$
- public: $g$, $g_2 = g^z$, $y = g^x$

**sector:**

- private: $r$
- public: $R = g^r$

**user:**

- private: $x_2$, $x_1 = x - z \cdot x_2$

**user in sector:**

- private: $x_2$, $x_1 = x - z \cdot x_2$
- public: $nym = R^{x_1}$

---

- $x_1 + z \cdot x_2 = x$, so $x_1$ and $x_2$ depend on $x$ and $z$ and must be derived during smart card personalization process
- still $x_1$ has random distribution
- deriving sector public key (pseudonym) $nym = R^{x_1}$ executed as before, the pseudonyms in different sectors are unlinkable

# Domain-Specific Pseudonymous Signatures

## Signature creation of message $M$

1. choose $t_1$ and $t_2$ at random
2. $a_1 := g^{t_1} g_2^{t_2}$, $a_2 := R^{t_1}$
3. $c := Hash(R, R^{x_1}, a_1, a_2, M)$
4. $s_1 = t_1 - cx_1$ $s_2 = t_2 - cx_2$

signature $(c, s_1, s_2)$, the sector name $R$, the user pseudonym $R^{x_1}$

## Verification

given signature $(c, s_1, s_2)$, the sector name $R$, the user pseudonym $R^{x_1}$, system parameters $(g, g_2, y)$, and message $M$

1. $\alpha_1 := y^c g^{s_1} g_2^{s_2}$
2. $\alpha_2 := (R^{x_1})^c R^{s_1}$
3. $c \stackrel{?}{=} Hash(R, R^{x_1}, \alpha_1, \alpha_2, m)$

# Domain-Specific Pseudonymous Signatures

1. designed by: Jens Bender, Özgür Dagdelen, Marc Fischlin, Dennis Kügler: Domain-Specific Pseudonymous Signatures for the German Identity Card. ISC 2012: 104-119
2. essentially Schnorr signature with non-interactive version of Okamoto proof of knowledge
3. works as long as the chips are safe: **once two chips broken we have two equalities with unknowns $x$ and $z$ :**

$$x_1 + z \cdot x_2 = x$$
$$x_1' + z \cdot x_2' = x$$

# Security Proofs

# Formal security proofs



## Current situation

- **publication driven**: for a paper to be accepted a security proof is almost necessary condition,

  let's get PhD/position/grant money/...

  ⇒ algorithm

  ⇒ formulating security proof

  ⇒ formulating model for this proof

- **business driven**:

  what is to be sold?

  ⇒ standards

  ⇒ certification

- **government driven**

# Formal security proofs



## Messy abundance of models

- a big fraction of papers come with new models
- hard to compare
- differences frequently very subtle
- even specialists may easily loose track

## Attacks

- come for schemes that has been proven to be secure – flaws in models and not in schemes
- frequent overlooking some practical issues
- basic problem: **designing a scheme is a great adventure**, proving the most challenging security reductions is fascinating, but proving all details is boring, non publishable, time costly . . .

# Formal security proofs

## Design criteria

1. . . . (usual stuff)
2. **just a few line of pseudo-code** – otherwise complete security proof may become infeasible

# Example

## Transferability of a proof

### Problem

- authentication protocol has to convince Alice that she is talking with Bob
- ...but we may overshoot the target if the protocol:
  - enables Alice to convince Eve that she has been talking with Bob
  - enables Bob to convince Eve that he has authenticated himself against Alice
  - enables Alice and Bob to convince Bad Guys that they have been communicating
  - enables Alice and Bob to convince Bad Guys that the transcript of a conversation does not belong to them
  - ...

# Simultability concept

## Faking protocol transcript -simultability

1. if Alice (respectively: Bob, Alice and Bob, eavesdropper) can **create transcripts of the protocol that have the same probability distribution**, then any transcript has no value for the Bad Guys
2. this should hold **even if the Bad Guys request Alice to behave in a certain way**

## Examples

- **EAC fails:** due to Terminal Authentication – a chip can prove its contact with any terminal by getting signatures of the terminal for strings delivered by Bad Guys
- **PACE|AA and SPACE|AA succeed:** tricky for PACE|AA and straightforward for SPACE|AA

**Acknowledgments**

to SKLOIS, BSI, and Foundation for Polish Science for cooperation and support