Belgian eID Card Technicalities

Danny De Cock
Danny.DeCock@esat.kuleuven.ac.be
Katholieke Universiteit Leuven/Dept. Elektrotechniek (ESAT)
Computer Security and Industrial Cryptography (COSIC)
Kasteelpark Arenberg 10
B-3001 Heverlee
Belgium
Overview

- eID Visual Aspects & Basics
- Crypto Crash Course
- eID Crypto Content
- Keys and Certificates
- Current Distribution Status
- Typical Use Cases
- Signature Creation & Validation
What do they look like?
Visual Security Mechanisms

- Rainbow and guilloche printing
- Changeable Laser Image (CLI)
- Optical Variable Ink (OVI)
- Alpha gram
- Relief and UV print
- Laser engraving
Visual Aspects

Front:
- Name
- First two names
- First letter of 3rd name
- Title
- Nationality
- Birth place and date
- Gender
- Card number
- Photo of the holder
- Begin and end validity dates of the card
- Hand written signature of the holder

Back side:
- Place of delivery of the card
- National Register identification number
- Hand written signature of the civil servant
- Main residence of the holder (cards produced before 1/1/2004)
- ICAO (cards produced since 1/1/2005)
Who gets an eID card?

- A new eID card is issued to
  - New inhabitants
  - Every youngster at the age of 12
  - People changing from one address to another in the local municipality
  - Replace a lost, stolen, damaged or expired (e)ID card
  - Adjust the citizen’s picture
  - Every citizen who asks to replace his/her old ID card
  - Every citizen who changes his/her name, gender,…

- Specific groups who requested a priority:
  - Medical doctors, lawyers, eID software companies,…
Advantages of an eID Card

- It simplifies the generation of digital signatures
  - No need to
    - Deploy your own closed user group CA
    - Issue key pairs and certificates
  - The eID card does it all for you:
    - Your application calculates a cryptographic hash on the data a citizen should sign
    - The eID card generates a digital signature on the hash

- Generating signatures has become very simple

- But signatures should be verified too…
  - Very complex matter
Public-key Basics

- Key pair = (public key and private key)
- Private key = only known to the user
  - Private decryption key ≠ private signing key
- Public key = available to everybody
  - Public encryption key ≠ public verification key
- Certificate = a means to determine to whom which public key belongs
  - Certificate is issued by a trusted third party aka Certification Authority
- Typical algorithms:
  - RSA, ElGamal
Public-key Signing

- Sender uses his/her private signing key to sign a message
- Receiver uses the sender’s public verification key to verify a digital signature

Terminology:
- The signer signs a document
  - Result: digital signature
- The verifier verifies a digital signature
  - Result: valid/invalid signature
Public-key Encryption

- Sender uses the intended recipient’s public encryption key to encrypt a message
- Recipient uses his/her private decryption key to decrypt a message
- Terminology:
  - The sender encrypts a document
    - Result: cipher text message
  - The recipient decrypts a cipher text message
    - Result: plain text message
Signing vs. En/Decryption

**Signing**
- Alice *signs* data using her **private signing** key
- Bob *validates* Alice’s signature with her **public verification** key
- Single step: Alice pushes her verification certificate to Bob
- Backups of a **signing key** compromise non-repudiation properties

**En/Decryption**
- Alice *encrypts* data using Bob’s **public encryption** key
- Bob *decrypts* the message with his **private decryption** key
- Initial step: Alice pulls Bob’s encryption certificate beforehand
- Backups of a **Decryption key** should be backed up to deal with “emergencies”
eID Card Details
Belgian eID Time line


22 Sept 2000: Council of Ministers approves eID card concept study

19 July 2001: Council of Ministers approves basic concepts (smart card, citizen-certificates, no integration with SIS card, Ministry of Internal Affairs is responsible for RRN’s infrastructure, pilot municipalities, helpdesk, card production, legal framework,… Fedict for certification services

3 Jan 2002: Council of Ministers assigns RRN’s infrastructure to NV Steria

27 Sept 2002: Council of Ministers assigns card production to NV Zetes, certificate services to NV Belgacom

25 March 2002: first 4 eID cards issued to civil servants

9 May 2003: first pilot municipality starts issuing eID cards

27 September 2004: start of nation-wide roll-out

End of 2009: all citizens and 5-year non-citizen residents have an eID card

Fall 2005: All 589 municipalities issue eID cards

25 January 2004: start of pilot phase evaluation

25 July 2003: eleventh pilot municipality started

Belgian eID Card Technicalities
Slide 14
24 November, 2005
eID Card Issuing Procedure (1/2)

1. Face to face identification
2. Municipality
3. National Register (RRN)
4. Certification Authority (CA)
5. Card Personalizer (CP)
6. Card Initializer (CI)
7. Citizen
8. Citizen PIN & PUK
9. (10a"
10. (10a’)
11. (10b)
12. Belgian eID Card Technicalities

Slide 15
24 November, 2005
eID Card Content

PKI
- Authentication
- Digital Signature
- RRN, Root CA, CA, ...

Citizen Identity Data
- ID
- ADDRESS

RRN = National Register

Belgian eID Card Technicalities
Slide 16
24 November, 2005
PKI Content – Keys & Certificates

- 2 key pairs for the citizen:
  - Citizen-authentication
    - X.509v3 authentication certificate
  - Advanced electronic (non-repudiation) signature
    - X.509v3 qualified certificate
    - Can be used to produce digital signatures equivalent to handwritten signatures, cfr. European Directive 1999/93/EC

- 1 key pair for the card:
  - eID card authentication (basic key pair)
    - No corresponding certificate: RRN (Rijksregister/Registre National) knows which public key corresponds to which eID card
Certificates for Government web servers, signing citizen files, public information,…

Card Administration:
- update address
- key pair generation
- store certificates,…

Certificates
- 1024-bit RSA
- 2048-bit RSA
- 2048-bit RSA
Citizen Certificate Details

Citizen Qualified certificate (~1000 bytes)

Version: 3 (0x2)
Serial Number:
10:00:00:00:00:00:8d:8a:fa:33:d3:08:f1:7a:35:b2
Signature Algorithm: sha1WithRSAEncryption (1024 bit)
Issuer: C=BE, CN=Citizen CA, SN=200501
Not valid before: Apr 2 22:41:00 2005 GMT
Not valid after: Apr 2 22:41:00 2010 GMT
Subject: C=BE, CN=Sophie Dupont (Signature), SN=Dupont, GN=Sophie Nicole/serialNumber=60050100093
Subject Public Key Info:
RSA Public Key: [Modulus (1024 bit): 4b:e5:7e:6e: ... :86:17, Exponent: 65537 (0x10001)]
X509v3 extensions:
Certificate Policies:
Policy: 2.16.56.1.1.1.2.1
CPS: http://repository.eid.belgium.be
Key Usage: critical, Non Repudiation

Authority Key Identifier: [D1:13: ... :7F:AF:10]
CRL Distribution Points:
URI:http://crl.eid.belgium.be/eidc0002.crl
Netscape Cert Type: S/MIME
Authority Information Access:
CA Issuers - URI:http://certs.eid.belgium.be/belgiumrs.crt
OCSP - URI:http://ocsp.eid.belgium.be
Qualified certificate statements: [00......F..]
Signature: [74:ae:10: ... :e0:91]

Citizen Authentication certificate (~980 bytes)

Version: 3 (0x2)
Serial Number:
10:00:00:00:00:00:0a:5d:9a:91:b1:21:dd:00:a2:7a
Signature Algorithm: sha1WithRSAEncryption (1024 bit)
Issuer: C=BE, CN=Citizen CA, SN=200501
Not valid before: Apr 2 22:40:52 2005 GMT
Not valid after: Apr 2 22:40:52 2010 GMT
Subject: C=BE, CN=Sophie Dupont (Authentication), SN=Dupont, GN=Sophie Nicole/serialNumber=60050100093
Subject Public Key Info:
RSA Public Key: [Modulus (1024 bit): cf:ca:7a:77: ... :5c:c5, Exponent: 65537 (0x10001)]
X509v3 extensions:
Certificate Policies:
Policy: 2.16.56.1.1.1.2.2
CPS: http://repository.eid.belgium.be
Key Usage: critical, Digital Signature

Authority Key Identifier: [D1:13: ... 7F:AF:10]
CRL Distribution Points:
URI:http://crl.eid.belgium.be/eidc0002.crl
Netscape Cert Type: SSL Client, S/MIME
Authority Information Access:
CA Issuers - URI:http://certs.eid.belgium.be/belgiumrs.crt
OCSP - URI:http://ocsp.eid.belgium.be
Qualified certificate statements: [00......F..]
Signature: [10:ac:04: ... :e9:04]
Belgium issuing eID cards

- 1 Million cards produced and issued in 6 months
- All 589 municipalities issue eID cards

Belgian eID Card Technicalities
A CRL is valid for seven days after it is issued.

A new CRL is issued together with a new Delta CRL.

A Delta CRL refers to a particular Base CRL which is always younger than 7 days.

OCSP queries the database with the most recent certificate status information.

CRLs follow the lifecycle of the eID cards they cover.

The CA stops issuing certificates referring to a particular CRL if it becomes too large.

The graph reflects the evolution of the eID cards following a CRL for which no new certificates are issues.
Typical Use Cases
Example 1 – Using an Authentication Certificate

Case study: Alice visits a website which uses client authentication

1. The web server Alice visits sends a random challenge to her browser
2. Alice confirms she wants to log in on the web site by presenting her PIN to her eID card and authorizes the signature generation
3. The browser sends the hashed challenge to Alice’s eID card to sign it
4. The browser retrieves the signature and Alice’s certificate from her eID card
5. The web server receives Alice’s signature and certificate

The web server learns Alice’s official name, her national number and that her eID card was used to produce the digital signature
Example 2 – Using a Qualified Certificate

Case study: Alice uses her eID card to generate a qualified signature on a file contract.doc for Bob

1. Alice’s computer application asks her whether she wishes to digitally sign the document
2. If she approves, she inserts her eID card in the computer’s smartcard reader
3. She enters her PIN to authorize the generation of a qualified signature
4. Bob receives from Alice:
   - The document contract.doc
   - The digital signature
   - Alice’s qualified certificate

E.g., Contract, Tax Declaration, Certified Mail,…

Bob learns Alice’s official name, her national number and that her eID card was used to produce the digital signature
Signature Generation Steps

Alice’s application
1. Calculates the cryptographic hash on the data to be signed
2. Prepares her eID card to generate an authentication signature or to generate a non-repudiation signature
3. Alice presents her PIN to her eID card
4. Her card generates the digital signature on the cryptographic hash
5. The application collects the digital signature from her eID card

Bob receives an envelope with a digitally signed message and a certificate
Signature Verification Steps

Bob

6. Retrieves the potential sender’s certificate
7. Verifies the certificate’s revocation status
8. Extracts Alice’s public key from her certificate
9. Retrieves the signature from the message
10. Calculates the hash on the received message
11. Verifies the digital signature with the public key and the hash
12. If the verification succeeds, Bob knows that the eID card of Alice was used to produce the digital signature

“The message comes from Alice” is a business decision
Signing Key Pair Properties

- Private signing key only available to the signer
  - **Signer explicitly authorizes** the Signature Creation Engine to generate a digital signature with the signing key, e.g., by **presenting a PIN** (personal identification number, cfr. Bank cards)
  - **Signer protects** the hash of his/her message with his/her signing key
  - **Verifier recovers** this hash correctly only if the right verification key is used

- Private signing key corresponds to the public verification key
  - If the Signature Verification Engine (SVE) outputs ‘valid signature’, **the verification key corresponds to the signing key**
  - If the SVE outputs ‘invalid signature’ the triplet \((\text{message, digital signature, verification key})\) does not match:
    - The message may have been **altered**
    - The verification key may be wrong, i.e., does not correspond to the signing key
    - The certificate of the signer **may have been revoked** (or suspended)

- Private signing key is kept in the smartcard

- Public verification key usually accompanies the digital signature
  - **Integrity of the verification key** is protected through the signer’s certificate
Signature Generation/Verification

1. Compute hash of message
2. Prepare signature
3. Present user PIN
4. SCD generates digital signature
5. Collect digital signature

6. Retrieve signer certificate
7. Verify the certificate’s revocation status
8. Retrieve public key from signer certificate
9. Retrieve digital signature on the message
    Beware – Bob should validate Alice’s certificate – Beware

10. Compute hash on received message
11. Verify digital signature
12. SVD outputs ‘valid signature’ or ‘invalid signature’

Bob

Alice

Signature Creation Engine

PIN

Hash

Signature Verification Engine

Hash

OCSP

CRL

Email

Beware – Bob should validate Alice’s certificate – Beware
Archiving Signed Data

- Digital signatures *remain valid forever* if one stores:
  - The digitally signed data
  - The digital signature on the data
  - The signer’s certificate
  - A proof of validity of the signer’s certificate
  - The verification timestamp of the signature

- Bottom line:
  - The integrity of this data should be protected!
  - There is no need to retrieve the status of a certificate in the past!
  - Protect your proofs in a digital vault
Certificate Revocation Lists (CRLs)

- **Complete CRL**
  - Enumerates all certificate serial numbers that should not be trusted
  - Typically (very) large, e.g., >500 Kbytes
  - Validity expires 7 days after creation
  - Certificates of new eID cards
    - Appear as on hold
    - Disappear when activated
  - Suspended certificates appear as on hold for up to 7 days
  - Items without reason code remain revoked forever
  - One complete CRL is referred to as the Base CRL

- **Delta CRL**
  - Lists all differences between the current complete CRL and the current Base CRL
  - Typically small, e.g., <25 Kbytes
  - Validity expires 7 days after creation
  - Reason codes:
    - On hold — newly issued eID card certificate is not yet activated, or has been suspended
    - Remove from CRL — eID card certificate has been activated
    - None — eID card certificate has been revoked

Complete CRLs

Delta CRLs vs. Base CRL
OCSP vs. CRLs – “Is the certificate valid?”

- Two options to make this business decision:
  - Do it yourself and use CRLs and Delta-CRLs
  - Trust a third party and use OCSP

- Use the Online Certificate Status Protocol (OCSP) where a trusted OCSP Responder answers the question with either “yes”, “no”, or “I do not know”
  - Remaining issues:
    - An OCSP Responder may use the most recent certificate status information (CSI)
      - An OCSP Responder does not have to use the most recent CSI!
      - The Responder typically uses CRLs to produce its answers
    - How to trust the OCSP Response?
  - Ideal for a few situations:
    - If only a few certificates per time unit must be validated
      - E.g., for citizens who wish to validate a certificate “from time to time”
    - To authenticate high-impact transactions
      - E.g., cash withdrawal, account closure, physical or electronic access control

- Certificate Revocation Lists (CRLs)
  - The digital signature verifier collects the (most recent) CRLs for the certificates in the certificate chain
    - These CRLs may become extremely large (e.g., several megabytes) ⇒ Delta-CRLs
    - Delta-CRLs may be very large (e.g., half a megabyte) ⇒ Delta-Delta CRLs
      - Note: Delta-Delta-CRLs are typically a few kilobytes each, but there is no standard…
Summary on Validity Statuses

- Digital Signature
  - Valid
  - Invalid

- eID Card (Signature Creation Device)
  - Valid
  - Invalid
    - Suspended
    - Revoked
    - Expired

- CRL, OCSP Response
  - Valid
  - Invalid
  - Expired

- Certificate
  - Valid
  - Invalid
    - Suspended
    - Revoked
    - Expired
    - Unknown
eID Card Chip Specifications

- Cryptoflex JavaCard 32K
  - CPU (processor): 16 bit Microcontroller
  - Crypto-processor:
    - 1100 bit Crypto-Engine (RSA computation)
    - 112 bit Crypto-Accelerator (DES computation)
  - ROM (OS): 136 kB (GEOS Java Virtual Machine)
  - EEPROM (Application + Data): 32 KB (Cristal Applet)
  - RAM (memory): 5 KB

- Standard - ISO/IEC 7816
  - Format & Physical Characteristics ↔ Bank Card (ID1)
  - Standard Contacts & Signals ↔ RST, GND, CLK, Vpp, Vcc, I/O
  - Standard Commands & Query Language (APDU)

---

Belgian eID Card Technicalities
24 November, 2005
eID Card Middleware

- PKCS#15 file system for ID applications
  - All eID-related data (certificates, photo, address, identity files,...)
  - No key management
- PKCS#11 standard interface to crypto tokens
  - Abstraction of signing functions (authentication, digital signatures)
  - Access to certificates
  - Available for Unix, Windows, MacOSX,...
- CSP for Microsoft Platforms
  - Only keys & certificates available via MSCrypto API
  - Allows authentication (& signature)
  - For Microsoft Explorer, Outlook,...
That’s it…
Questions?

Belgian eID card information on the Internet
http://www.rijksregister.fgov.be
http://www.fedict.be
http://www.belgium.be
http://eid.belgium.be
http://www.cardreaders.be

Test cards can be ordered at
http://www.eid-shop.be

Myself
Danny.DeCock@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/~decockd

Yourself
http://www.mijndossier.rrn.fgov.be
http://www.mondossier.rrn.fgov.be
http://www.meindossier.rrn.fgov.be
Backup Slides
Various Authentication Interfaces

- Authentication of a transaction, client authentication, digital signature,… requires a PIN to be presented to reflect the cardholder’s consent
Comparing eID and Bank Card Functionalities

- Citizen Identification
- Data Capture
- Strong Authentication
  - Authentication
  - Digital Signatures
  - eID Card
- Access Control
  - Container Park, Swimming Pool, Library,…

- Customer Identification
- Data Capture
- Authentication
  - Electronic Transactions
  - ATM Transactions
  - Electronic Purse
- Access Control
  - Self-Bank
eID & Bank Cards Crypto

- 2 Citizen Key Pairs
  - Citizen-authentication
    - X.509v3 *authentication certificate*
  - Advanced electronic (non-repudiation) signature
    - X.509v3 *qualified certificate*
    - Can be used to produce digital signatures equivalent to handwritten signatures, cfr. European Directive 1999/93/EC

- 1 eID Card-specific Key Pair
  - eID card authentication (basic key pair)
    - No corresponding certificate: RRN (Rijksregister/Registre National) knows which public key corresponds to which eID card

- Transactions with vending machines, ATMs, phone booths, parking meters,…
  - MAC-based use chip card

- Home banking
  - MAC-based
    - Family of secret master keys
    - Uses chip card or Digipass
    - MAC authenticates login, transaction

- PKI-based
  - Closed user group PKI
  - Key pair stored in key file or smart card
  - Banking organization issues certificate
  - Digital signature authenticates login, transaction
### CA Certificate Details

#### Root CA certificate (920 bytes)

- **Version:** 3 (0x2)
- **Signature Algorithm:** sha1WithRSAEncryption (2048 bit)
- **Issuer:** C=BE, CN=Belgium Root CA
- **Not valid before:** Jan 26 23:00:00 2003 GMT
- **Not valid after:** Jan 26 23:00:00 2014 GMT
- **Subject:** C=BE, CN=Belgium Root CA

**Subject Public Key Info:**
- **RSA Public Key:** (Modulus: 00:c8:a1:71: ... :b0:6f, Exponent: 65537 (0x10001))

**X509v3 extensions:**
- **Certificate Policies:** Policy: 2.16.56.1.1.1
  - CPS: http://repository.eid.belgium.be
- **Key Usage:** critical, Certificate Sign, CRL Sign
- **Subject Key Identifier:** [10:F0: ... :7E:DB:E6]
- **Authority Key Identifier:** [10:F0: ... :8E:DB:E6]

**Netscape Cert Type:** SSL CA, S/MIME CA, Object Signing CA

**Basic Constraints:** critical, CA:TRUE

**Signature:** [c8:6d:22: ... :43:2a]

#### CA certificate (975 bytes)

- **Version:** 3 (0x2)
- **Signature Algorithm:** sha1WithRSAEncryption (2048 bit)
- **Issuer:** C=BE, CN=Belgium Root CA
- **Not valid before:** Apr 10 12:00:00 2003 GMT
- **Not valid after:** Jun 26 23:00:00 2009 GMT
- **Subject:** C=BE, CN=Citizen CA

**Subject Public Key Info:**
- **RSA Public Key:** (Modulus: 00:c9:ae:05: ... :cb:71, Exponent: 65537 (0x10001))

**X509v3 extensions:**
- **Certificate Policies:** Policy: 2.16.56.1.1.1.2
  - CPS: http://repository.eid.belgium.be
- **Key Usage:** critical, Certificate Sign, CRL Sign
- **Subject Key Identifier:** [D1:13: ... :7F:AF:10]
- **Authority Key Identifier:** [10:F0: ... :8E:DB:E6]
- **CRL Distribution Points:**
  - URI: http://crl.eid.belgium.be/belgium.crl
- **Netscape Cert Type:** SSL CA, S/MIME CA, Object Signing CA
- **Basic Constraints:** critical, CA:TRUE, pathlen:0

**Signature:** [b2:0c:30: ... :18:6e]
Government Certificate Details

**Government CA certificate (~979 bytes)**

- **Version**: 3 (0x2)
- **Serial Number**: 
- **Signature Algorithm**: sha1WithRSAEncryption (2048 bit)
- **Issuer**: C=BE, CN=Belgium Root CA
- **Not valid before**: Jan 27 00:00:00 2003 GMT
- **Not valid after**: Jan 27 00:00:00 2009 GMT
- **Subject**: C=BE, CN=Government CA

**Subject Public Key Info:**
- **RSA Public Key**: 
  - Modulus (2048 bit): 00:ac:c9:a0: ... :89:13
  - Exponent: 65537 (0x10001)

**X509v3 extensions:**
- **Certificate Policies**: Policy: 2.16.56.1.1.1.3
- **CPS**: http://repository.eid.belgium.be
- **Key Usage**: critical, Certificate Sign, CRL Sign
- **Subject Key Identifier**: [F5:DB: ... :D1:8B:D6]
- **Authority Key Identifier**: [10:F0: ... :8E:DB:E6]
- **CRL Distribution Points**: 
  - URI: http://crl.eid.belgium.be/belgium.crl
- **Netscape Cert Type**: SSL CA, S/MIME CA, Object Signing CA
- **Basic Constraints**: critical, CA:TRUE, pathlen:0

**Signature**: [a0:53:21: ... :1d:c9]

**RRN certificate (~808 bytes)**

- **Version**: 3 (0x2)
- **Serial Number**: 
  01:00:00:00:00:00:f8:20:18:9e:17
- **Signature Algorithm**: sha1WithRSAEncryption (1024 bit)
- **Issuer**: C=BE, CN=Government CA
- **Not valid before**: Oct 9 09:06:09 2003 GMT
- **Not valid after**: Jan 26 09:06:09 2009 GMT
- **Subject**: C=BE, CN=RRN, O=RRN

**Subject Public Key Info:**
- **RSA Public Key**: 
  - Modulus (1024 bit): 00:db:72:4d: ... :80:0d
  - Exponent: 65537 (0x10001)

**X509v3 extensions:**
- **Certificate Policies**: Policy: 2.16.56.1.1.1.3.1
- **CPS**: http://repository.eid.belgium.be
- **Key Usage**: critical, Digital Signature, Non Repudiation
- **Subject Key Identifier**: [09:22: ... :30:01:37]
- **Authority Key Identifier**: [F5:DB: ... :D1:8B:D6]
- **CRL Distribution Points**: 

**Signature**: [12:89:cd: ... :ca:2a]
**Certificate Revocation List details**

### Citizen CRL (+500 Kbyte)

**Version 2 (0x1)**  
Signature Algorithm: sha1WithRSAEncryption (2048 bit)  
Issuer: C=BE, CN=Citizen CA  
Creation date: Apr 6 15:19:23 2004 GMT  
Next update: Apr 13 15:19:23 2004 GMT  
CRL extensions:  
Authority Key Identifier: [D1:13: ... :7F:AF:10]  
CRL Number: 4294995040

**Revoked Certificates:**
- **Serial Number:** 1000000000000004B823FAE7B1BB44B1  
  **Revocation Date:** Jan 14 12:56:50 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 1000000000000062F6A1BB1431902D4  
  **Revocation Date:** Oct 23 23:15:11 2003 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 10000000000000125DC2DF2031534033  
  **Revocation Date:** Sep 5 09:49:44 2003 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 100000000000007E5B11506303959320  
  **Revocation Date:** Apr 8 16:33:23 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 100000000000091ACC84FC377F8A6ECE  
  **Revocation Date:** Apr 8 16:55:14 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 100000000000127BE2DA18842E8A7BAC  
  **Revocation Date:** Apr 8 15:20:13 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 1000000000001902ECF11657FE2813A5  
  **Revocation Date:** Apr 8 16:29:54 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 100000000000FE6A4ACD4ECF04233442  
  **Revocation Date:** Apr 8 17:33:31 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 100000000000FDFF72C4E59AD46AFC21  
  **Revocation Date:** Apr 8 15:32:38 2004 GMT  
  **CRL Reason Code:** Certificate Hold

**Signature:** [95:19:b2: ... :21:31]  

### Citizen Delta CRL (~15 Kbyte)

**Version 2 (0x1)**  
Signature Algorithm: sha1WithRSAEncryption (2048 bit)  
Issuer: C=BE, CN=Citizen CA  
Creation date: Apr 8 17:43:14 2004 GMT  
Next update: Apr 15 17:43:14 2004 GMT  
CRL extensions:  
Authority Key Identifier: [D1:13: ... :7F:AF:10]  
CRL Number: 4294995072  
Delta CRL Indicator: critical, 4294995040

**Revoked Certificates:**
- **Serial Number:** 100000000000007E5B11506303959320  
  **Revocation Date:** Apr 8 16:33:23 2004 GMT  
  **CRL Reason Code:** Certificate Hold
- **Serial Number:** 1000000000000091ACC84FC377F8A6ECE  
  **Revocation Date:** Apr 8 16:55:14 2004 GMT  
  **CRL Reason Code:** Remove From CRL
- **Serial Number:** 100000000000127BE2DA18842E8A7BAC  
  **Revocation Date:** Apr 8 15:20:13 2004 GMT  
  **CRL Reason Code:** Remove From CRL
- **Serial Number:** 1000000000001902ECF11657FE2813A5  
  **Revocation Date:** Apr 8 16:29:54 2004 GMT  
  **CRL Reason Code:** Remove From CRL
- **Serial Number:** 100000000000FE6A4ACD4ECF04233442  
  **Revocation Date:** Apr 8 17:33:31 2004 GMT  
  **CRL Reason Code:** Remove From CRL
- **Serial Number:** 100000000000FDFF72C4E59AD46AFC21  
  **Revocation Date:** Apr 8 15:32:38 2004 GMT  
  **CRL Reason Code:** Remove From CRL

**Signature:** [64:20:22: ... :c3:5e]
Decryption vs Signing

- Alice encrypts data for Bob using his encryption key
- Bob decrypts the message with his **private decryption** key
- Initial step: Alice must fetch Bob’s encryption certificate beforehand
- Decryption key should be backed up to deal with “emergencies”
- Alice signs data using her **private signing** key
- Bob validates Alice’s signature with her **public verification** key
- Single step: Alice pushes her certificate to Bob
- Backups of a signing key compromise non-repudiation properties
No need for encryption certificates!

Alice sends a digitally signed message to Bob

Alice receives an encrypted message from Bob

Give me Alice’s authentic encryption key

Bob

Alice
eID Card Issuing Procedure (2/2)

0: Citizen receives a convocation letter or takes the initiative
1: Visit municipality with photo
2: Formal eID request is signed
3,4: CP receives eID request via RRN
5: CP prints new eID card, CI starts on-card key pairs generation
6: RRN receives part of the eID card activation code PUK1
7: CA receives certificate requests
8: CA issues two new certificates and issues new CRLs
9: CI stores these certificates on the eID card
10a: CI writes citizen data (ID, address,…) to the card, deactivates the card
10b: CI sends invitation letter with citizen’s PIN and activation code PUK2
11: Citizen receives invitation letter
12: Civil servant starts eID card activation procedure
13: eID card computes a signature with each private key, CA removes certificates from CRL