Privacy in Advanced Smart Card Applications

a Challenging Task

Pim Vullers MSc

p.vullers@cs.ru.nl
http://www.cs.ru.nl/~pim/

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

11th Smart Event - e-Smart 2010
23th September 2010
Outline

Context

Privacy and Smart Cards

Java Card and Cryptography

Case Study: e-Ticketing

A Challenging Task
Applet-based e-Ticketing

- Public transport e-ticketing
- Java Card *smart* cards
- Privacy-friendly
- Advanced cryptographic protocols

Why?

- Current situation ('*dumb*' memory cards):
  - *no privacy*
  - *no security*
  - *no intelligence*
- Migration from memory cards to micro-processor cards
Privacy and Smart Cards

Big Brother’s little helper

- Smart cards are easily traceable
- Allows construction of detailed dossiers
- For both legitimate and malicious parties
- Serious privacy concerns!

Example

With OV-chipcard / Oyster / Charlie / . . . , you tell:
who you are, when you get on a bus, metro, train, . . .
Privacy and Smart Cards

Protection against outsiders

- Random UID
- Reader authentication
- Secure messaging
- Issue: Performance

Protection against insiders

- Much harder problem
- Zero-knowledge proofs or blinding identifiable information
- Practical implementations are rare
- Poor performance
Privacy and Smart Cards

Protection against outsiders
- Random UID
- Reader authentication
- Secure messaging
- *Issue*: Performance

Protection against insiders
- Much harder problem
- Zero-knowledge proofs or blinding identifiable information
- Practical implementations are rare
- Poor performance
Java Card and Cryptography

Wikipedia:

- The main design goals of the Java Card technology are portability and security.

- Cryptography. Commonly used encryption algorithms like DES, Triple DES, AES, RSA (including elliptic curve cryptography) are supported. Other cryptographic services like signing, key generation and key exchange are also supported.
Cryptographic coprocessor

- Required to provide security API at acceptable performance
- Provides *hardware implementations* of symmetric algorithms
- Provides *mathematical support* for asymmetric cryptography

Security API

- Implementation is *optional*, may differ per card
- *Restricted* to basic cryptographic functionality
- High-level API prevents access to primitive operations
Privacy-friendly solution

- Attribute-based authorisation
- Anonymous credentials:
  card only says “I’m a first class year pass valid in 2010”
- *Subtle point:* attribute may be non-identifying, 
  but signature may be used for tracing cards/individuals

- Java Card prototype implementation
  joint work with Wojciech Mostowski
Attribute-proving Protocol

Step 1
- show the attribute and a certificate that binds it to the card’s key pair

Step 2
- sign a random value to proof possession of the private key

Anonymisation
blind the actual values of the certificate and key pair
A Challenging Task

Frustration

- Signing is a cryptographic operation (security API)
- Blinding is a mathematical operation (no support)
- Mathematical support of the coprocessor is unaccessible

Solutions

- Java Card implementation of big number arithmetic (slow)
- Abusing the operations provided by the security API (fast)

Example

```java
keyAgreement = KeyAgreement.getInstance(DiffieHellman);
keyAgreement.init(blindingFactor);
keyAgreement.generateSecret(privateKey, blindedKey);
```
Privacy in Advanced Smart Card Applications

- New security objective for smart card applications
- Advanced cryptographic protocols for advanced smart cards
- Requires all possibilities of a modern smart card (even more)
- A challenging task due to limitations of the Security API