A smart card based solution for user-centric identity management

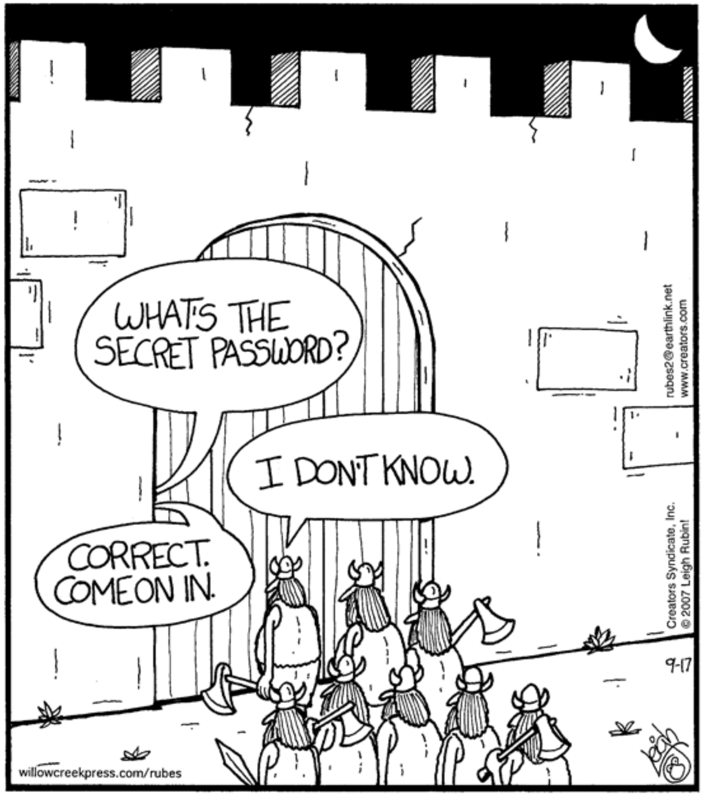
Jan Vossaert
Researcher at KaHo Sint-Lieven
Affiliated Researcher at KULeuven

Overview

- Introduction
- Approach
- Overview of the architecture
- Protocols
- Implementation details
- Evaluation
- Future work

Introduction

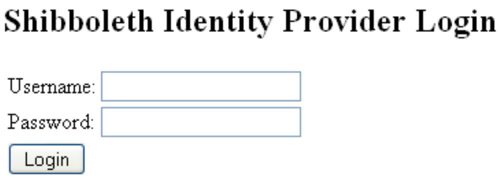
- Traditonal mechanisms for authentication
  - Password based solutions
  - X.509 certificates
- Drawbacks
  - Token management
  - Mobility of tokens
  - Personalized services



Why great care and consideration should be taken when selecting the proper password

Introduction

- Solutions
  - *Federated identity management systems*
    - Increased usability
    - No (or limited) user control
    - Identity provider can profile users
    - Web based
    - One identity provider
    - User impersonization
    - Weak login procedures



Shibboleth Identity Provider Login

Username:

Password:

Introduction

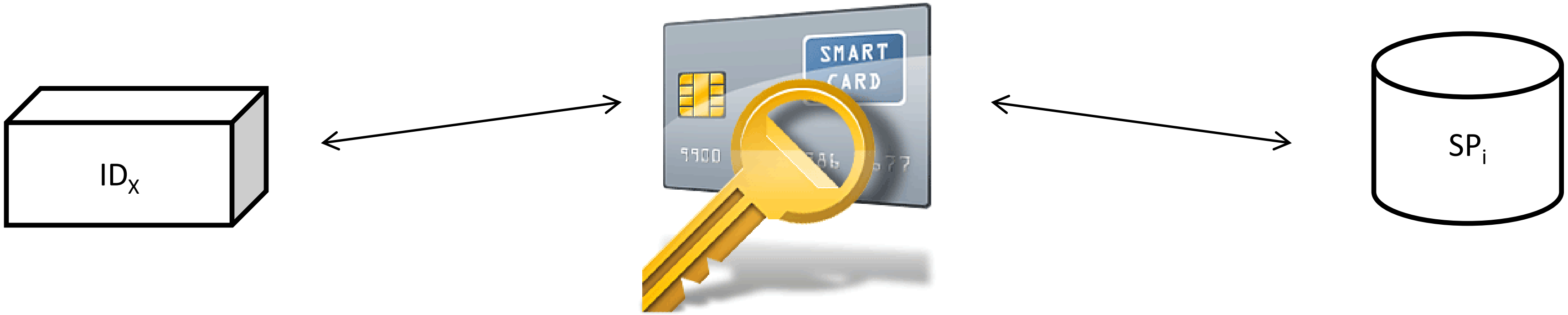
- Solutions
  - *Electronic identity technology*
    - Increased mobility

    - No (or limited) user control
    - Only immutable attributes
    - Security versus scalability

Introduction

- Challenges
  - increased flexibility
    - Mutable attributes
    - Multiple identity providers
  - user control
    - Personalisation
  - online and offline services
    - Feasible revocation strategy

Approach

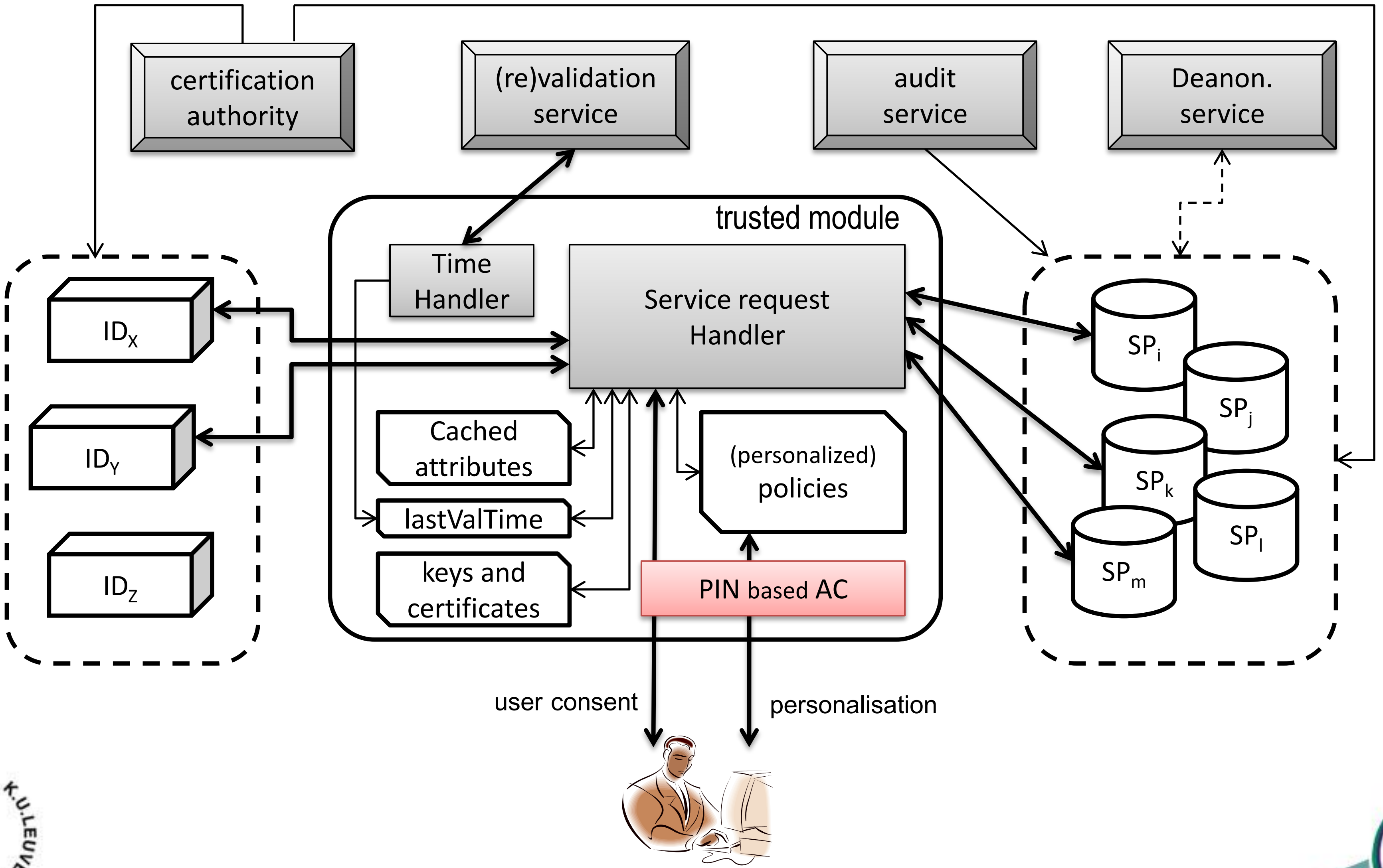


- Secure element is mediator between
  - Identity providers
  - Service providers
- Access to attributes controlled by
  - external authorities: certificates
  - user: personalized policies at the card

Approach

- Privacy properties
  - No profiling
    - by identity providers
    - by collaborating service providers
  - Access control to personal information
    - by audit authorities
    - by user
  - No user impersonization

Overview of the architecture



Overview of the architecture

- Service provider certificate
  - Keeps a list of access rights approved by audit authority
  - Keeps a list of trusted identity provider (groups)
- Identity provider certificate
  - Keeps a list of access rights
- Public keys of root CAs are placed at the card

Protocols

- **Card issuance**
  - – Common secret keypair
    - Prevents profiling
  - – Card specific pseudonym
    - Used to generate service specific pseudonyms
- **Card revalidation**
  - – Mutual authentication
  - – Card releases chip number

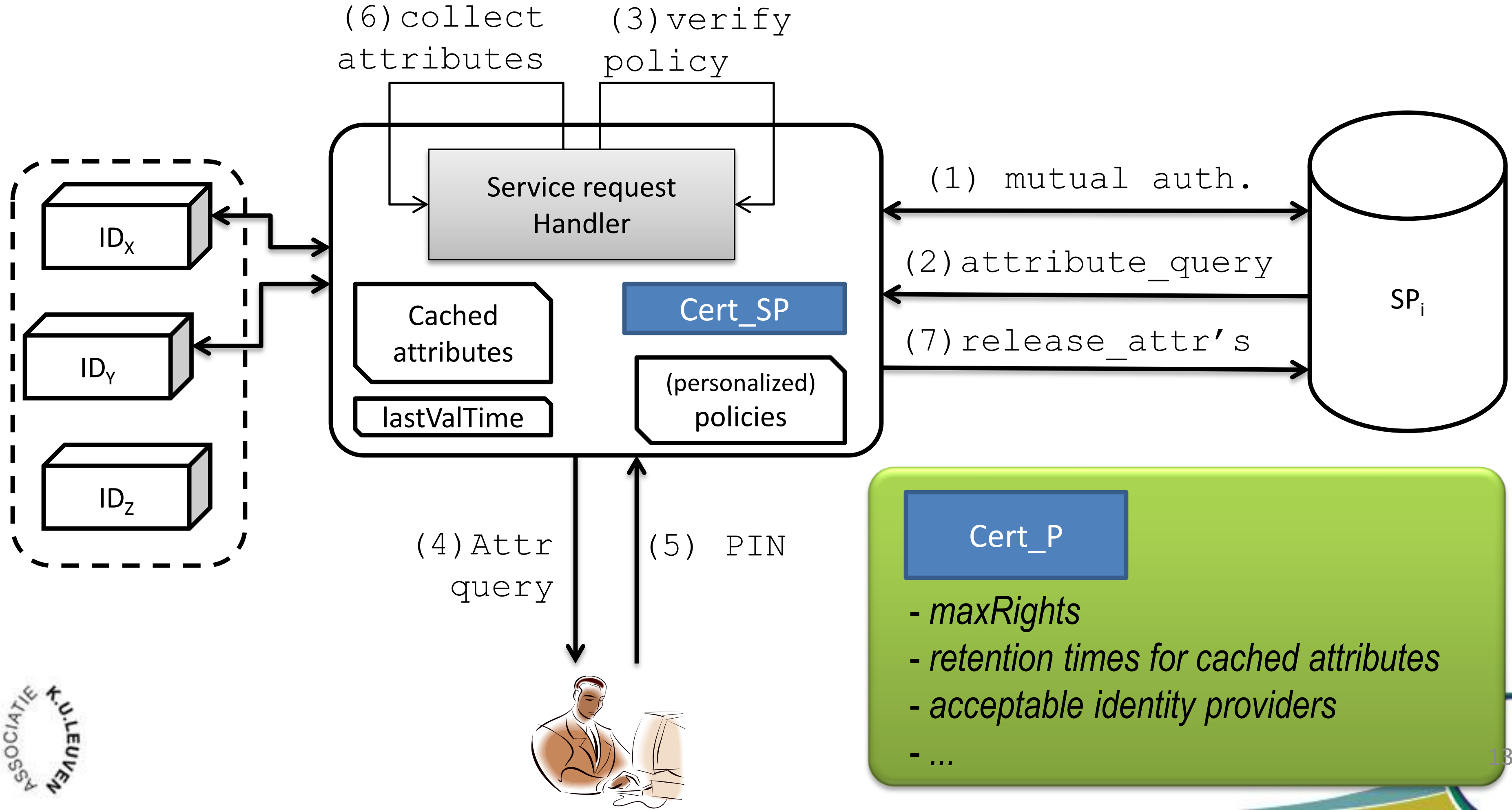
    -
      ```
      IF stillValid THEN update lastValTime
      ELSE block_card
      ```

Protocols

- **Mutual authentication**
  - Mutual key agreement protocol
  - SP → CARD
    - `lastValTime` used to check validity of SP Certificate
    - Short-lived server certificates
  - CARD → SP
    - proves to be genuine
    - `lastValTime > accValTime`

Protocols

- **Access to (personalized) services**



(6) collect attributes

(3) verify policy

Service request Handler

Cached attributes

lastValTime

Cert_SP

(personalized) policies

ID_X

ID_Y

ID_Z

(1) mutual auth.

(2) attribute_query

(7) release_attr's

SP_i

(4) Attr query

(5) PIN

Cert_P

- *maxRights*
- *retention times for cached attributes*
- *acceptable identity providers*
- ...

Protocols

- **Access to personalized services**
  - Special attribute → service specific pseudonym
    - `nym`$_{IP}$ `= Hash(secret||Cert`$_{SP}$`.subject)`
- **Deanonymization**
  - Releasing encrypted attributes
  - Can be decrypted by TTP

Implementation details

- Prototype on Gemalto TOP IM GX4 smart card
  - Java Card 2.2.1
  - Performance constraints
  - No clock
  - Authorisation
    - PIN based

Implementation details

- **Certificates**
  - Standard X509 certificates
    - Authentication towards providers
    - Obtain derived card verifiable certificates
  - Custom card verifiable certificates
    - Trusted providers
    - Attribute ID list/Level of assurance

Implementation details

- **Memory management**
  - No garbage collection
  - Cached attributes
    - `Value/retention time/LOA/last time of use/identity provider/…`
    - `Fixed set of byte arrays with variable length`
    - `Least recently used update policy`
  - Static memory configuration

Implementation details

- **Release attributes**
  - Cached attributes
  - Attribute ←→ identity provider
- **Personalization policies**
  - Update policy based on PIN
  - Select cached attributes (persistent attributes)
  - Assign trust level to service providers
  - Assign sensitivity level to attributes

Evaluation

- Trust properties
  - Card issuer knows common key pair

    BUT card-specific secret is not known by card issuer
  - Trust in workstation for user interaction

    BUT implementation in SIM possible
- Scalability & flexibility
  - Clear separation of duties
  - Representatives for set of identity providers
  - Flexible revocation strategy

Evaluation

- Controlled release of attributes
  - Access control at multiple levels
    - certificates, user policies, user consent
  - Limited value of attributes to SP
  - Proving properties of attributes
  - Encrypted attributes → accountability measures
- Performance
  - 2 identity providers: 3461 ms
  - 1 identity providers: 2287 ms
  - 0 identity providers: 1110 ms

Future work

- Building concrete services and identity providers
- Integration in Web applications
- Fine-grained access policies
- From smart card to SIM, dedicated module, ...
- Accurate performance results