SECURING BIOMETRIC TEMPLATES ON SMART CARDS

JONATHAN CHESEAUX

11/03/2013

Supervisers

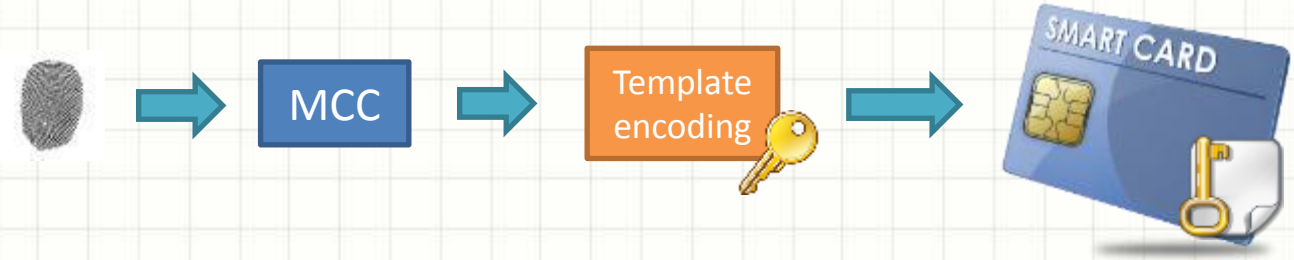
Andrzej Drygajlo - Leila Mirmohamadsadeghi

Presentation plan

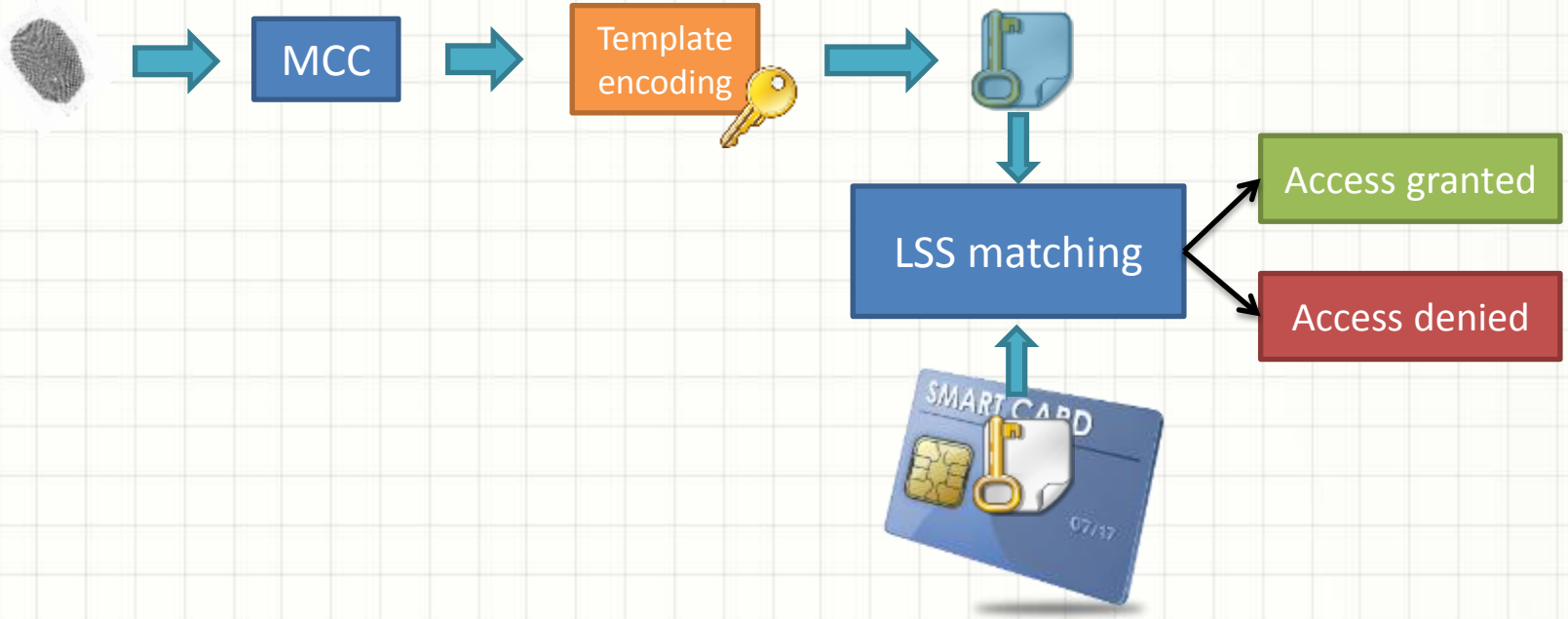
1. Introduction
   - Project presentation, goals
   - Algorithms used (MCC, transformation)
2. Progress
   - Study of given algorithms (collisions, imposters/genuine scores)
   - Programming environment setup and limitations
   - APDU communication protocol
   - Write/read files on the card
3. Future goals
   - Adapt LSS matching algorithm to Java Card API
   - Implement the project on real hardware devices

Project presentation

Fingerprint enrolment



Access control



Progress

Algorithms analysis

Collision detection

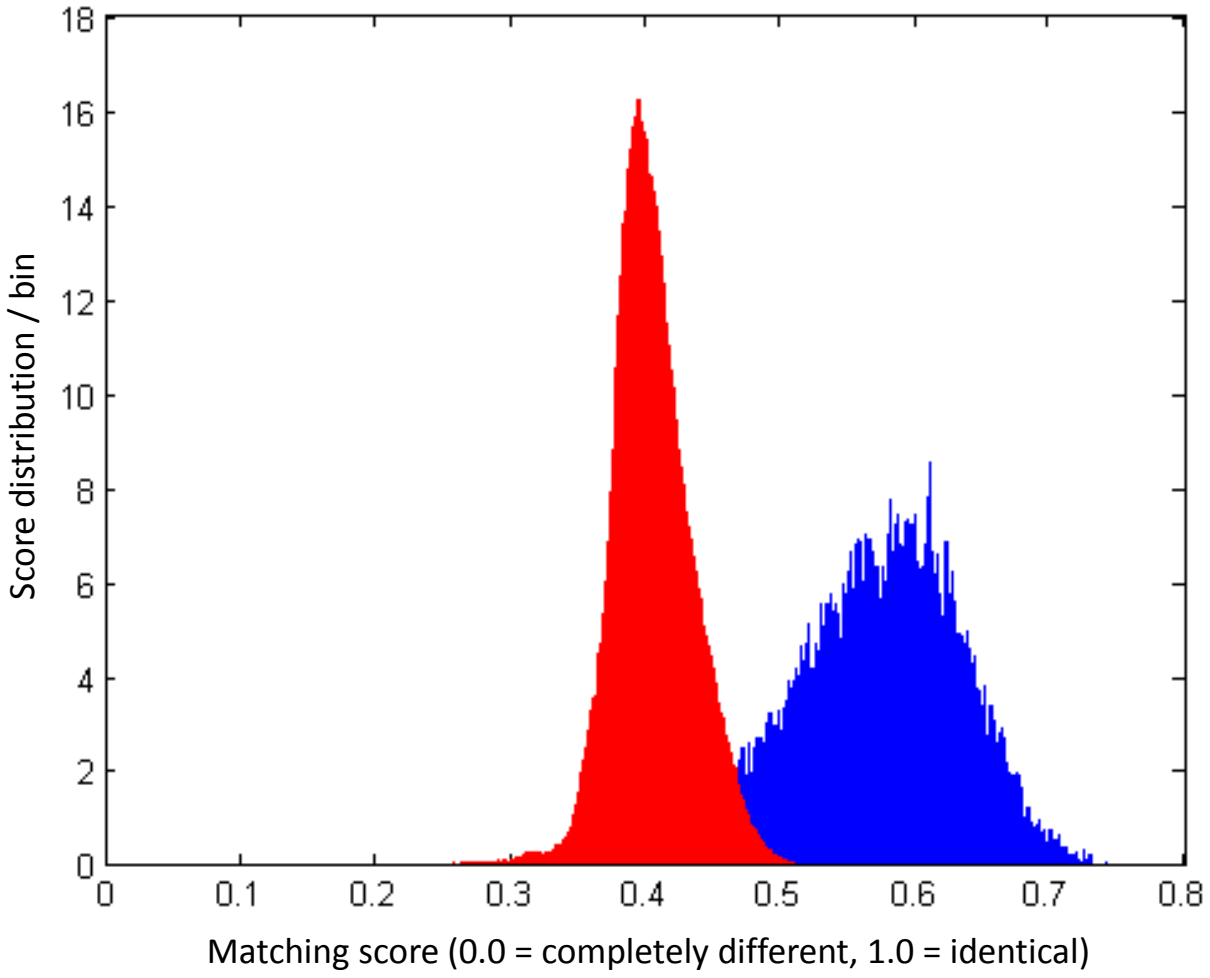
- same transformation key for each fingerprint
- random key for each fingerprint
- different keys to encrypt one fingerprint

Result :

> 1 million cases tested
0 collision

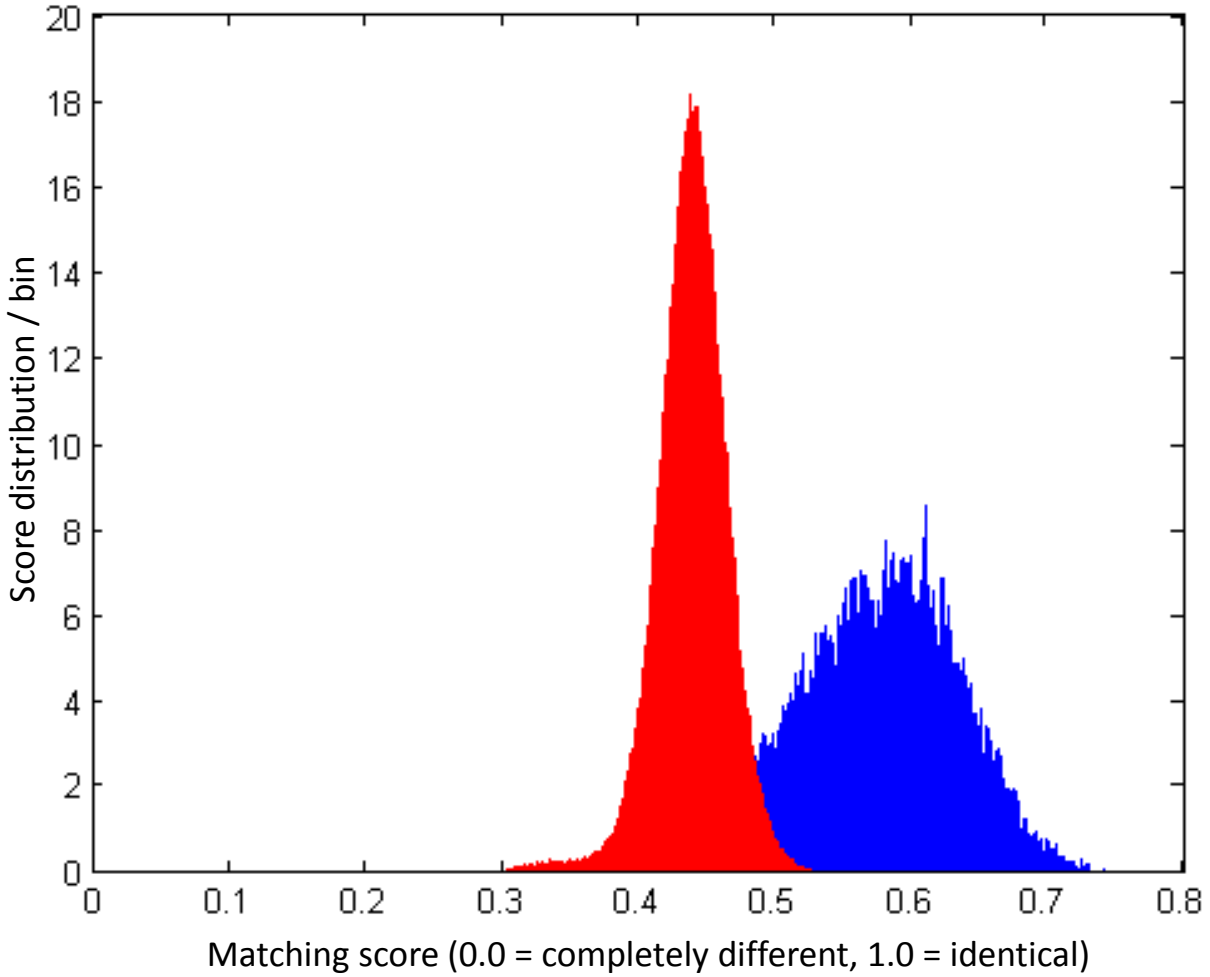
Impostors/Genuine matching results

Unknown encrypting key



Impostors/Genuine matching results

Stolen key scenario

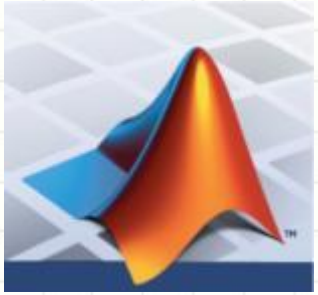


Programming environment setup

Libraries

Java Card 2.2.2

Eclipse Juno
JCWDE (plugin)



Matlab

Programming environment setup

Java Card limitations

Not supported by the API :

- Char, double, float, long
- Multidimensional arrays
- Garbage collection, threads

Hardware limitations :

- Limited storage capacity (<100KB)
- 8- or 16-bit CPU running at 3.7MHz
- Messages/responses size limited (<255 bytes)

Programming environment setup

APDU communication protocol

Table 2.1 Command APDU structure

Mandatory header				**Optional body**		
CLA	INS	P1	P2	Lc	Data field	Le

Table 2.2 Response APDU structure

Optional body	**Mandatory Trailer**	
Data field	SW1	SW2

Zhiqun Chen, *Java Card Technology for Smart Cards – Architecture and Programmer's Guide*

Programming environment setup

Read/Write files on the card

DEMO

Future goals

- Adapt LSS matching algorithm to Java Card API (on-card matching)
- Implement the project on real hardware devices
- Testing

Questions

?