



Security and Programming Challenges in Programming Contactless Cards

Eric Vétillard & Alexandre Frey
Trusted Logic

www.trusted-logic.fr

eSMART'04 – Sophia Antipolis – September 2004

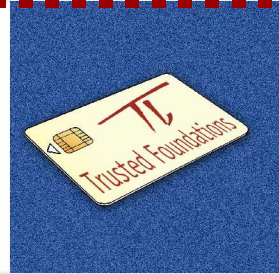


Agenda

- Introduction
- Issues
 - Security
 - Performance
- Practice
 - Java Card 2.2
 - Security constraints
- Coming soon



Introduction



Who programs contactless cards?



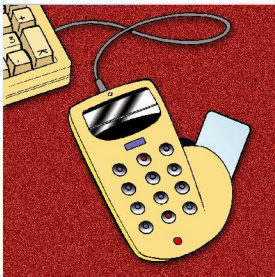
- Experts from card manufacturers
 - Applications must be very small
 - Applications must be lightning-fast
 - Applications are usually sensitive
- Since Java Card 2.2, anybody
 - Or nobody



What is Specific ?



- Contactless cards are wireless
 - They are similar to other wireless technologies
 - They may have similar security problems
 - ✓ Think about WiFi, CDMA, ...
- Contactless cards are powerless
 - They rely on an external power source
 - They have a big reliability problem
 - ✓ Power comes and goes
- Contactless cards are « user-friendly »
 - They are very practical to use
 - But they are not programmer-friendly



Issues



Security Issues

- Smart card issues
- Accessibility
- Improved user experience



Trusted Logic



Smart Card Issues

- A contactless card is a smart card
 - Most traditional attacks should work
 - Only a question of right equipment
 - Traditional countermeasures must be present
- Same defensive programming should be used
 - Protecting data against modification
 - Protecting data against disclosure
 - Protecting processing against fault injections
 - Careful lifecycle management
 - ...



Accessibility

- Turns on whenever there is an appropriate field
 - The user cannot feel it
 - Information comes at the next bank statement
- Sensitive to rogue readers
 - Attacker reading many cards in a crowded subway
 - Fake reader replacing real one
 - Man-in-the-middle attacks are easier to hide
 - Wireless communication makes things easier





Practical Example

Using a PIN code

- Maybe, increase the possible attempts
 - Because of power problems
 - For instance, consider 10 tries
- How long does it take to « try » 1000 cards
 - In a crowded subway ?
 - In a stadium ?
- The PIN code idea is not so good
 - 10.000 combinations are clearly not enough
 - There is a scaling problem to think about





Improved User Experience

Making it easier to use

- User does not have to get the card out
 - No additional authentication possible (hologram, ...)
 - It is possible to use an emulator
- Transaction is fast and simple
 - Complexity of the transaction is limited
 - No online transactions, no checks
 - No PIN entry, no biometrics, no authentication



End-to-end security ?

- Some issues would call for it
 - The accessibility of cards
 - The difficulty to use additional checks
- Some issues make it very difficult
 - End-to-end security difficult to smart cards
 - No authentication, no external control, ...
- What is the right compromise ?



Performance Issues

- General Issue
- Memory Accesses
- Cryptography





General Issue

- The issue originates in two properties
 - Improved user experience
 - External power source
- Only a few milliseconds for a normal transaction
 - Number of operations is limited
 - Expensive operations are prohibited
- Protocols must be very simple and fast
 - They must use the hardware at its best



Memory Accesses

- Applications must have persistent data
- Current persistent technology is slow to update
 - Applications are aware of it
- Applications minimize updates
 - Designers ensure that few updates are required
 - Applications carefully count the updates





Cryptography

- Cryptographic computations are expensive
 - Public-key cryptography is prohibited
 - Secret-key cryptography must be kept under control
- Use of cryptography must be limited
 - From the design of applications
 - The implementation must be efficient





Practice




Trusted Logic

Which Trade-Offs ?

- Design trade-offs are difficult
 - Application must be self-secured
 - Use of cryptography is limited
- Implementation trade-offs are difficult
 - Redundancy is not a good idea
- So far, other trade-offs are made
 - Reducing the value of the assets





Practical Tips

- Change optimization patterns
 - Contact applications are designed for size
 - Contactless applications are designed for speed
- Minimize the overhead
 - Avoid method invocations
 - ✓ Inline short methods (waste space)
 - Do not access EEPROM several times
 - ✓ Use and reuse local variables





Java Card 2.2

- Officially supports contactless cards
- Very limited API
 - A few constants to identify the protocol
 - No specific mechanism
- Actually available on cards





New and Improved Issues

Java Card & Security

- Java Card is accessible
 - Anybody can program cards
 - Anybody can design applications
- Design for security
 - Good principles are well-known ...
 - ... and blown to bits by contactless constraints
- Implement for security
 - Many small tricks
 - Contactless makes them trickier





New and Improved Issues

Java Card & Performance

- Java Card VM decreases performance
 - By a small factor (10 to 30%)
 - Not really significant on contact cards
 - Definitely significant on contactless
- Java Card VM decreases control
 - Especially on persistent memory updates
 - Number of updates difficult to control
 - ✓ Definitely not portable
- Plus many small issues





New and Improved Issues

Java Card Annoyances

- Transaction rollback
 - From the same application
 - ✓ Severely increases the transaction time
 - From another (contact?) application
- API/framework Issues
 - Selecting the right application
 - Unnecessary persistent updates
 - Many small details



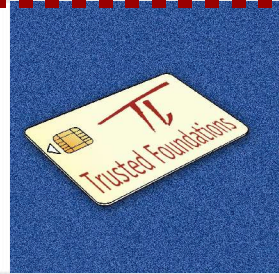


Conclusions

- Java Card on contactless is not « for dummies »
 - Designing applications is very difficult
 - Implementation requires very high skills
- Some issues remain to be addressed
 - In proprietary APIs
 - In evolutions of the standard APIs
- In the meantime, no portability
 - But great opportunities



Coming Soon



TL
Trusted Logic

Combi Cards



- Combination of contact and contactless
 - They are already around
- Collaboration will improve
 - Sharing of information
- Processors will work « together »
 - A multiprocessor card ...



Java Card 2.next

- Evolution of Java Card 2.2
- Contactless is one of the major issues
 - Some low-level improvements





Q & A

Eric.Vetillard@trusted-logic.fr
Alexandre.Frey@trusted-logic.fr



Trusted Logic
