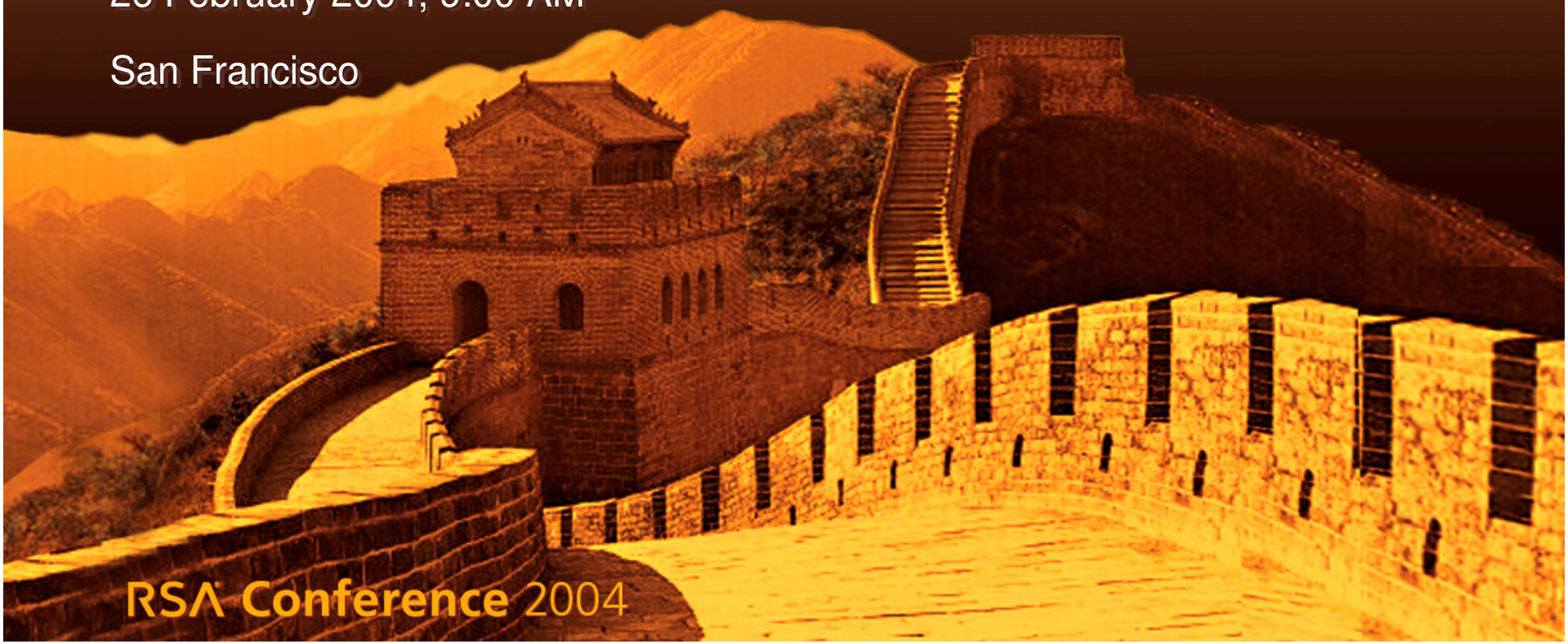
Java Card Security

Marc Witteman, Riscure

witteman@riscure.com

26 February 2004, 9:00 AM

San Francisco



RSA Conference 2004

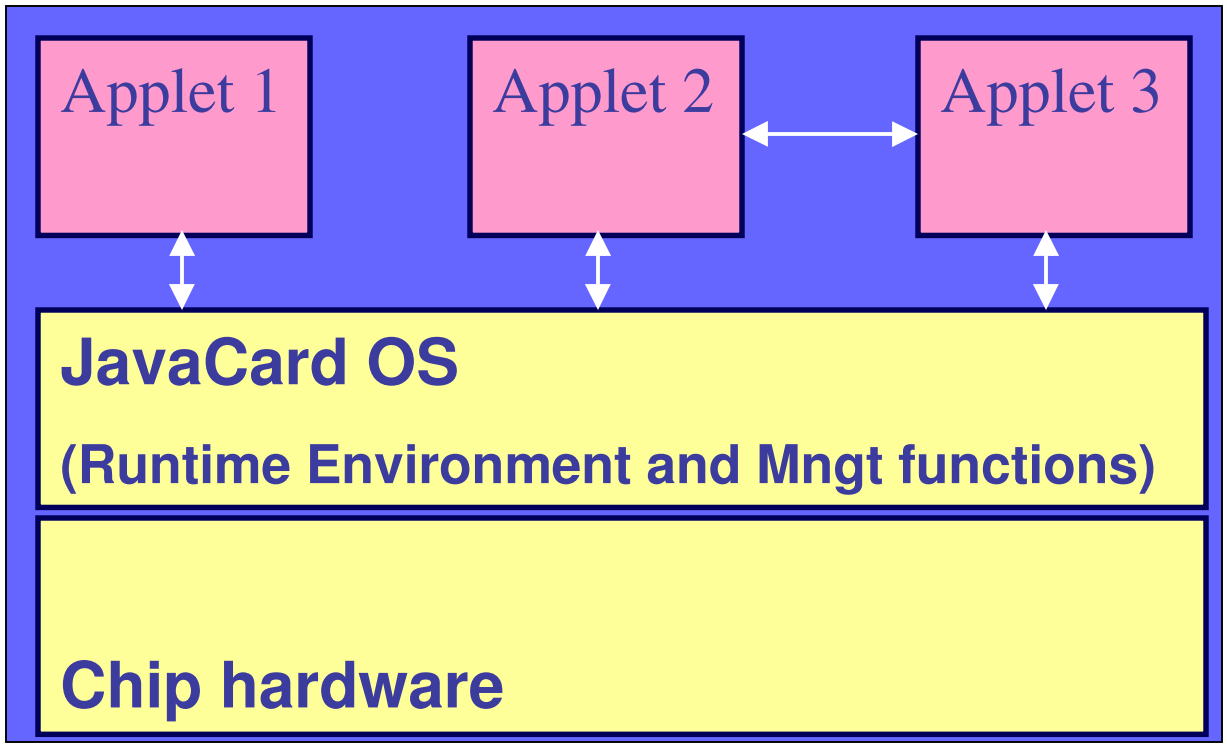
Overview

- What is Java Card?
- Life cycle
- Concepts
- Risk analysis
- Attacks
- Securing Java Card

Riscure

What is Java Card?

A Java Card is a smart card running a small Java based Operating System that can dynamically be upgraded.



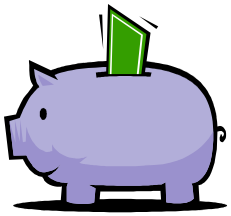
Riscure

Java Card Benefits (according to SUN)

- **Interoperable**
  Applets run on any Java Card
- **Secure**
  Inherent security of the Java language enhanced with new concepts like applet firewall and atomicity
- **Multi-Application Capable**
  Multiple applications can co-exist securely on a single smart card.
- **Dynamic**
  Post-issuance applet downloading.
- **Open**
  Developers benefit from object-oriented programming, and off-the-shelf Java development tools.
- **Compatible with Existing Standards**
  ISO7816, EMV, Global Platform and ETSI.

Riscure

Main Java Card application areas today



Financial

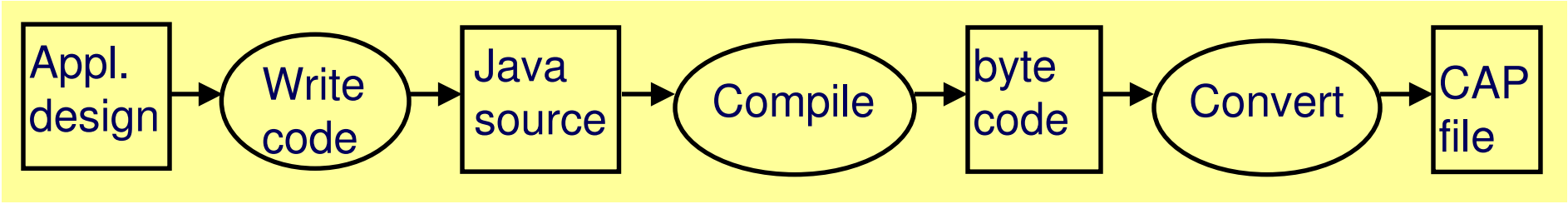
- Smart Credit / Debit
- E-Purses
- Loyalty programs

Mobile Communication

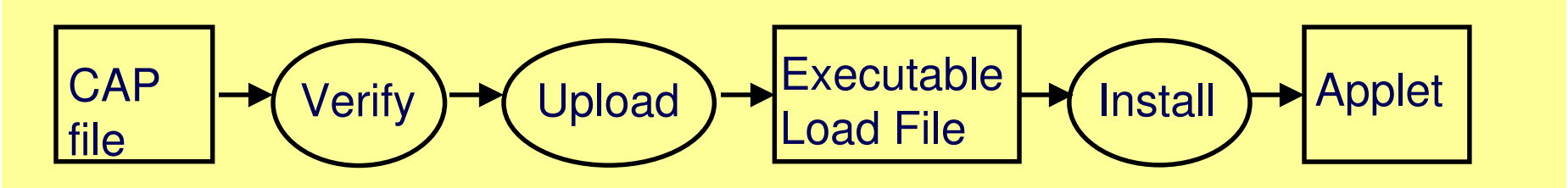
- Infotainment
- Business support
- Network optimizers

Riscure

Java Card Applet Life Cycle



Applet development



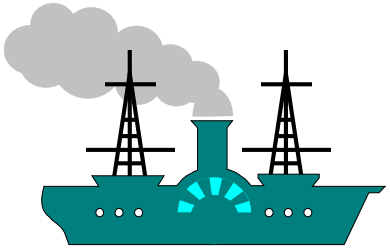
Applet deployment

Riscure

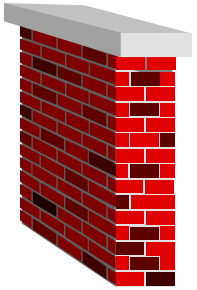
Java Card concepts and innovations

- Verification
  How can mobile code be trusted?

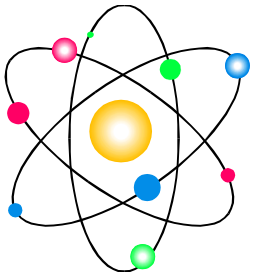
- Loading
  How can the code origin be trusted?



- Firewall
  How to allow secure applet interactions?



- Atomicity
  How to protect data consistency?



Riscure

Concepts: Verification

- What is a CAP file actually?
- What can an ill-formed applet do?
- How important is byte code verification?
- What can the Java Card Off-Card Verifier do?

Catch:
Beware of code changes between verification and running!

Riscure

Concepts: Card Management

- Card management frameworks
  Global Platform / GSM SIM data download
- Management actions
  loading, installation, personalization, deletion
- Secure protocols
  electronic signatures, encryption

Catch:

Side channel attacks on the Java Card platform may retrieve or bypass card management keys!

Riscure

Concepts: Firewall

- Applet firewall allows controlled sharing
- All applet interactions go through the firewall
- Server applets can authenticate client applets
- Object ownership prevents unauthorized access

Catch: An applet becomes vulnerable if the virtual machine does not carefully implement all firewall rules

Riscure

Concepts: Atomicity

- Smart cards operate in an unfriendly environment
- Consistency of data is crucial to reliability and security
- Atomic operations
- Transaction mechanism:
  - beginTransaction
  - commitTransaction
  - abortTransaction

Catch: keeping a reference to a deleted object can break the entire platform security

Riscure

Java Card vs Java

A comparison between Java Card and Java

Conceptual security is better and worse!

Security is better

- No dynamic class loading
- No threading
- Applet firewall
- Applet sources controlled

Security is worse

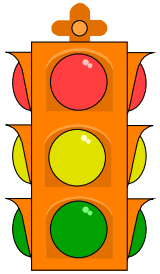
- No on-board byte code verifier
- No sandbox
- Applets are persistent
- Uploading

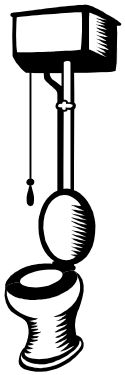
Riscure

Java Card risks

- Annoyance

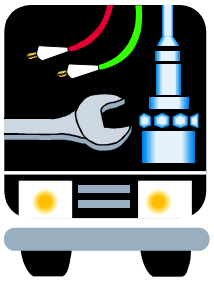
- Denial of service





- Invasion of privacy

- System modification



Riscure

Java card threats

- Verified applet abuses feature

  ⇒**Trojan code example**

- Verified applet exploits bug

  ⇒**Dangling reference demo**

- Ill-formed applet attack

  ⇒**Firewall type confusion demo**

Riscure

Attack example: Trojan code

Applet developer hides small Trojan in useful applet to steal a PIN code of a cell phone

- Applet performs some meaningful action, e.g. Traffic info
- At some stage the applet misleads the user by asking the PIN
- After PIN insertion it is leaked by SMS, and the applet continues

Riscure

Demo: Firewall type confusion

- Two applets communicate through firewall
- Binary incompatibility between server and client
- A reference to a byte array is set to another object
- Runtime allows arbitrary reading & writing!

DEMONSTRATION on Java Card reference implementation

Riscure

Fragments of type confusion code

```
// class that stores a short where arrays may store their length
public class Fake {
  public short size = 0x7FFF;
};
// Server implementation of 'confuse' method
public Fake confuse( Fake fake ) {
   return fake;
}
// Interface definition that client uses is different:
public byte[] confuse ( Fake fake );
// Client binds byte array reference 'array' to object 'fake'
byte[] array = sio.confuse( fake );
// size of 'array' now set to 0x7FFF (32K !!!)
```

Riscure

Demo: Dangling reference

- Applet creates object within transaction
- Transaction is aborted, object deleted
- Reference was not cleared, now dangling!
- New object created, dangling reference confused!
- Runtime allows arbitrary reading & writing!

DEMONSTRATION on recent commercial Java Card!

Riscure

Fragments of dangling reference code

```
// start a transaction
JCSystem.beginTransaction();
// allocate short byte array
array = new byte[2];
// abort transaction, array object must be deleted
JCSystem.abortTransaction();
// create new object of different class to fill the emptied space
Fake fake = new Fake(); // try to reuse the memory
// if ‘array’ not cleared, its size may now be set to 0x7FFF (32K !!!)
```

Riscure

Java Card Security Measures

- Java Card source code review
- CAP file verification and Code signing
- Loading security
- JCRE verification

Riscure

Conclusion

- Java Card is a significant step forward
- Realistic threats exist also for Java Card
- Off-card verification is more risky than it seems
- Java card issuers can and should take specific measures to counter act the threats
- Java Card Security is attainable

Riscure

Thanks!

Want to know more?

Email to: witteman@riscure.com

or visit: www.riscure.com

Articles available on Smart Card and Java Card security

Riscure