Welcome to the workshop

Building Security Applications on Smart Cards

Dr. Rajesh Kumar Pal

About the speaker

Dr. RAJESH KUMAR PAL
Founder, Mobisec

E-mail: pal@mobisec.in

http://www.linkedin.com/in/rajeshkumarpal

- Education
  - Ph.D., IIT Delhi
  - M.S. Cyber Security, Masaryk University, Czech Republic
  - M.Tech. Computer Science, IIT Kharagpur
  - Executive MBA, IIM Indore
  - B.E. Computer Science, SGGS Nanded

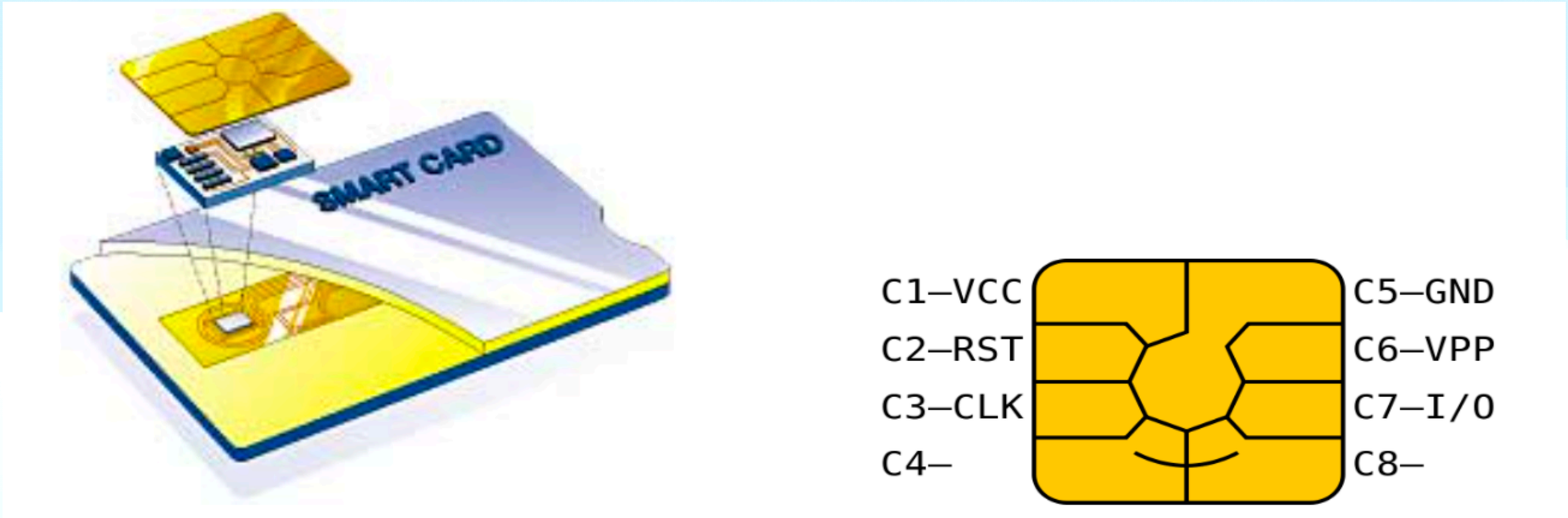
- Work experience
  - Founder and Director at Mobisec Technologies Pvt. Ltd. (https://mobisec.in)
  - Director IAF-CERT (Certificate of Excellence from PM for Remote Audit Tool running at 65K+ nodes in IAF)
  - 22 years of experience in IT and cyber security (Secure File System, Authenticated Encryption Algorithms)
  - Prize winner at Idea Stage in the first Cyber Security Grand Challenge
  - Teaching 1.5 credit course @ IIT Delhi Aug-Nov 22 & building Security Products
  - Publications @ https://scholar.google.co.in/citations?user=yYow2v0AAAAJ&hl=en

Content

- Smart Card Basics
- Fundamentals for developing app on Java Card
- How to Build, Burn & Run apps on Java Card
- Walk-through to the participants
- Conclusion

Smart Card Basics

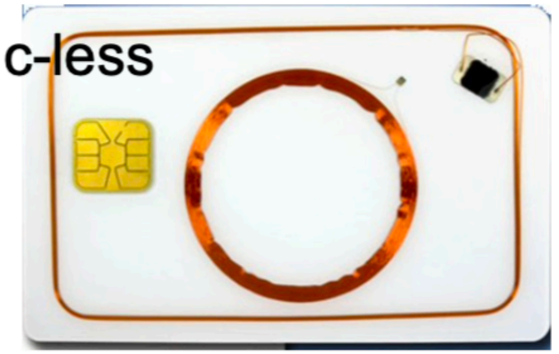
Smart Cards



Smart cards forms

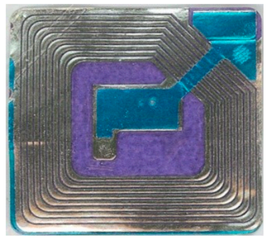
- Many possible forms
  - ISO 7816 standard
  - SIM size, USB dongles, Java rings…
- Contact(-less), hybrid/dual interface

  - contact physical interface
  - contact-less interface
  - hybrid card – separate logics on single card
  - dual interface – same chip accessible contact & c-less



Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

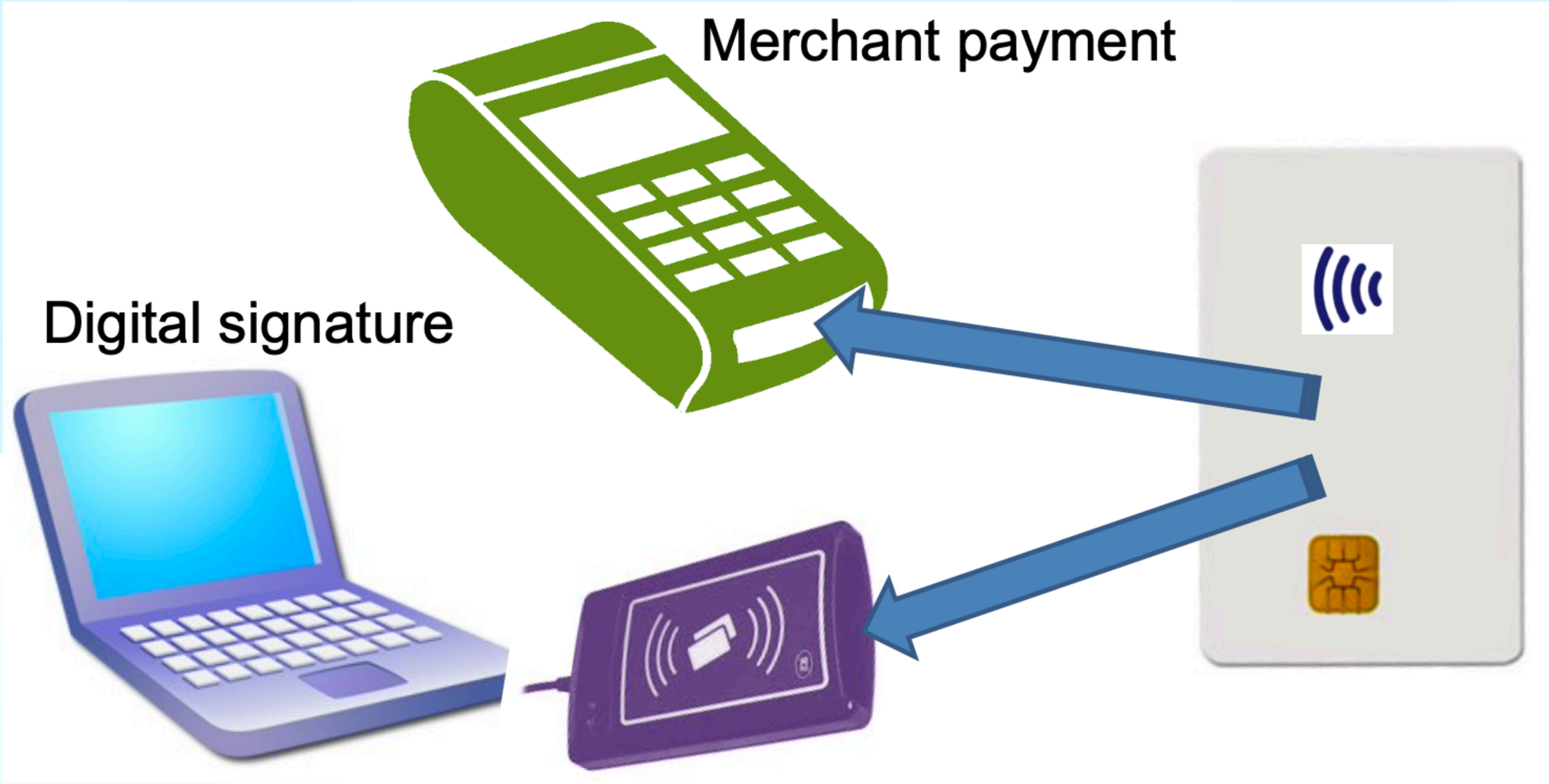
Basic types of (smart) cards



1. Contactless “barcode”
   – Fixed identification string (RFID, < 5 cents)
2. Simple memory cards (magnetic stripe, RFID)
   – Small write memory (< 1KB) for data, (~10 cents)
3. Memory cards with PIN protection
   – Memory (< 5KB), simple protection logic (<$1)
4. Cryptographic smart cards
   – Support for (real) cryptographic algorithms
   – Mifare Classic ($1), Mifare DESFire ($3)
5. User-programmable cryptographic smart cards
   – JavaCard, .NET card, MULTOS cards ($5-$30)

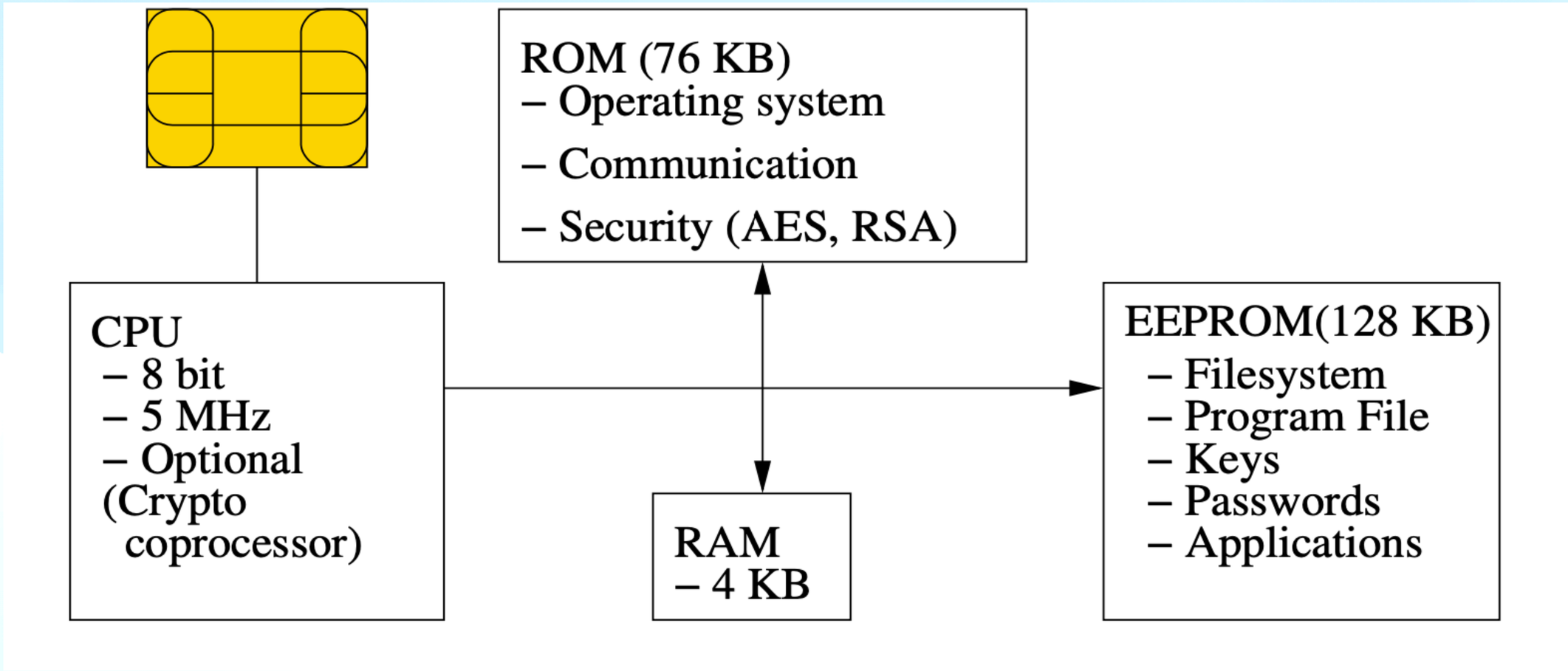
- Chip manufacturers: NXP, Infineon, Gemalto, G&D, Oberthur, STM, Atmel, Samsung...

Credit & Gratitude: Petr Svenda @ Masaryk University, Brno



Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

Smart Card


ROM (76 KB)
– Operating system
– Communication
– Security (AES, RSA)
CPU
– 8 bit
– 5 MHz
– Optional
(Crypto
coprocessor)
RAM
– 4 KB
EEPROM(128 KB)
– Filesystem
– Program File
– Keys
– Passwords
– Applications

Smart cards are used for…

- GSM SIM modules
- Digital signatures
- Bank payment card (EMV standard)
- System authentication
- Operations authorizations
- ePassports
- Multimedia distribution (DRM)
- Secure storage and encryption device
- …

Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

Smart card is highly protected device

- Intended for physically unprotected environment
  - NIST FIPS140-2 standard, security Level 4
  - Common Criteria EAL4+/5+
- Tamper protection
  - Tamper-evidence (visible if physically manipulated)
  - Tamper-resistance (can withstand physical attack)
  - Tamper-response (erase keys…)
- Protection against side-channel attacks (power,EM,fault)
- Periodic tests of TRNG functionality
- Approved crypto algorithms and key management
- Limited interface, smaller trusted computing base
- http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm

Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

Main advantages of crypto smart cards

- High-level of security (CC EAL5+, FIPS 140-2)
- Fast cryptographic coprocessor
- Programmable secure execution environment
- Secure memory and storage
- On-card asymmetric key generation
- High-quality and very fast RNG
- Secure remote card control

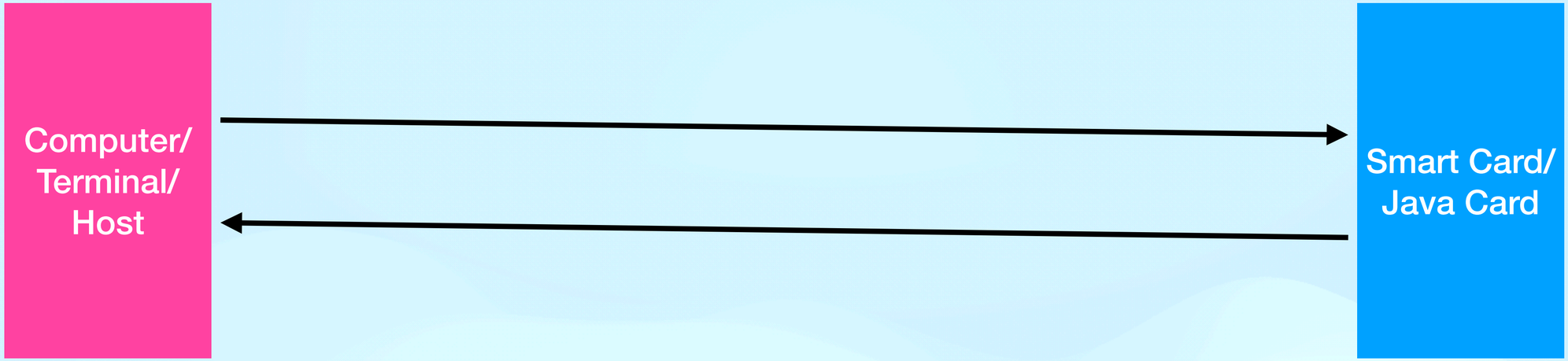
Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

- Basic - cryptographic co-processor
  - Truly random data generator
  - 3DES, AES128/256
  - MD5, SHA1, SHA-2 256/512
  - RSA (up to 2048b common, 4096 possible)
  - ECC (up to 192b common, 384b possible)
  - Diffie-Hellman key exchange (DH/ECDSA)

Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

Fundamentals for developing app on Java Card

APDU

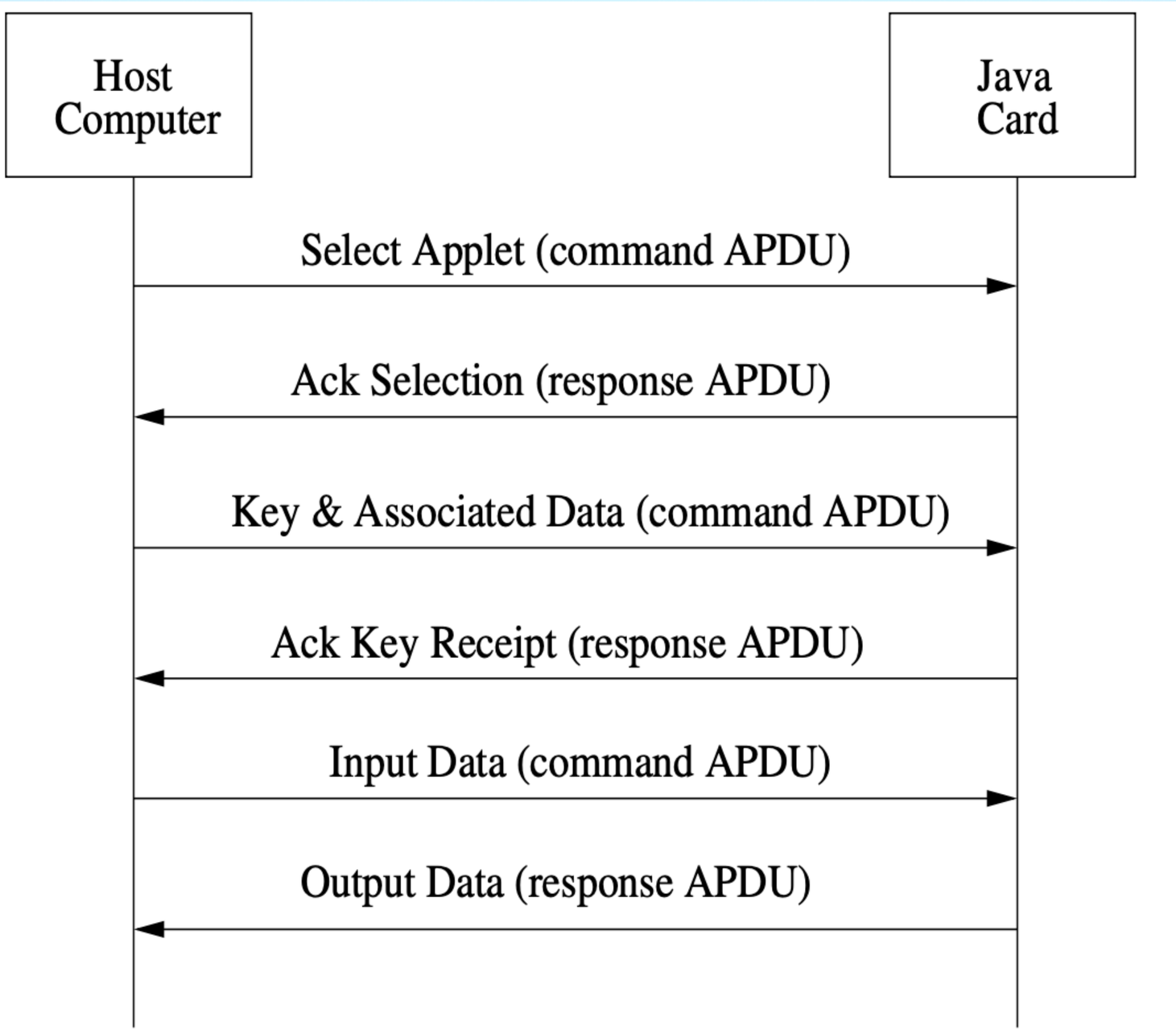


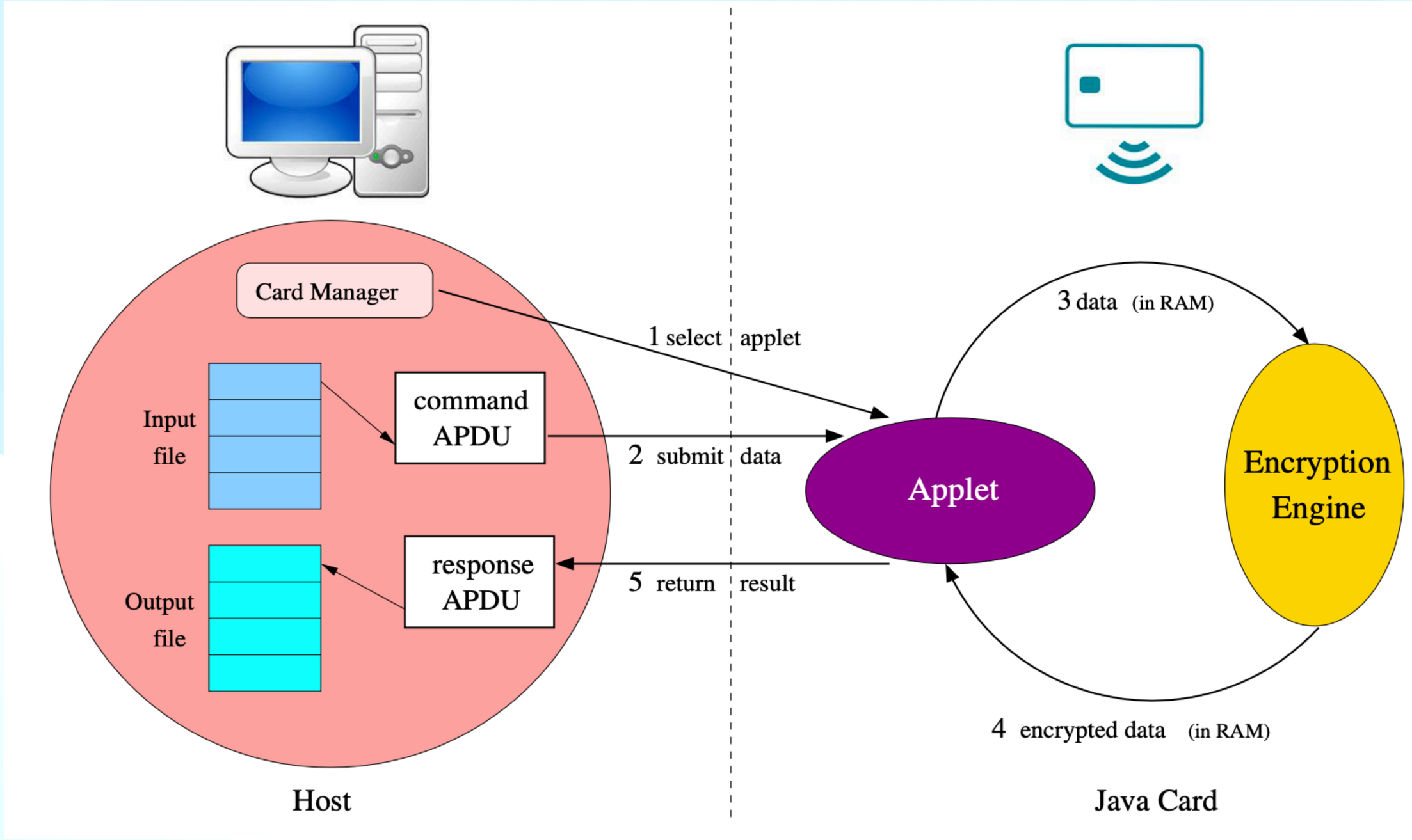
class	instruction	parameter 1	parameter 2	data length	actual data
CLA	INS	P1	P2	LC	DATA
0xB0	0x61	0x00	0x00	0xEE	0xA1 0xB2 ...

Command APDU

Response Data	Response Status
0x25 0x10 0xA4 ...	0x00 0x90

Response APDU


Host Computer
Java Card
Select Applet (command APDU)
Ack Selection (response APDU)
Key & Associated Data (command APDU)
Ack Key Receipt (response APDU)
Input Data (command APDU)
Output Data (response APDU)


Card Manager
1 select applet
Input file
command APDU
2 submit data
Applet
3 data (in RAM)
Encryption Engine
Output file
response APDU
5 return result
4 encrypted data (in RAM)
Host
Java Card

JC development process

1. Extends javacard.framework.Applet

2. Compile Java→*.class (Java 1.3 binary format)

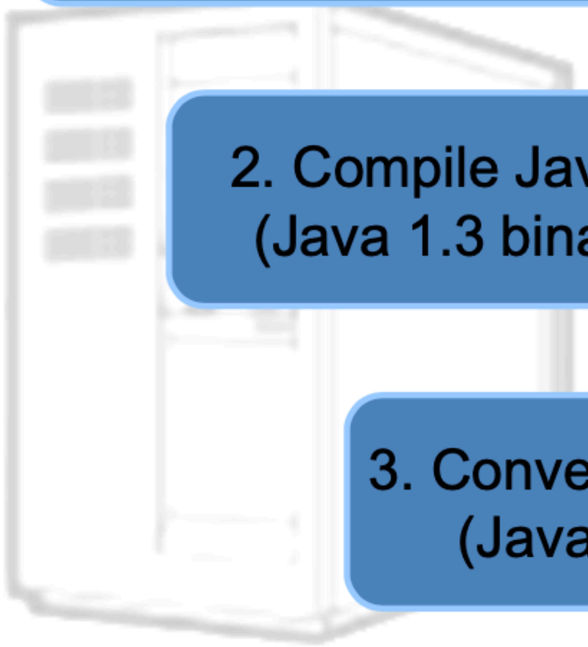
3. Convert *.class→*.jar/cap (JavaCard Convertor)

4. Upload *.jar/cap → smart card (GPPro/GPShell)

5. Install applet (GPPro/GPShell)

7. Use applet on smart card (APDU)

6. Write user Java app (javax.smartcardio.*)



Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

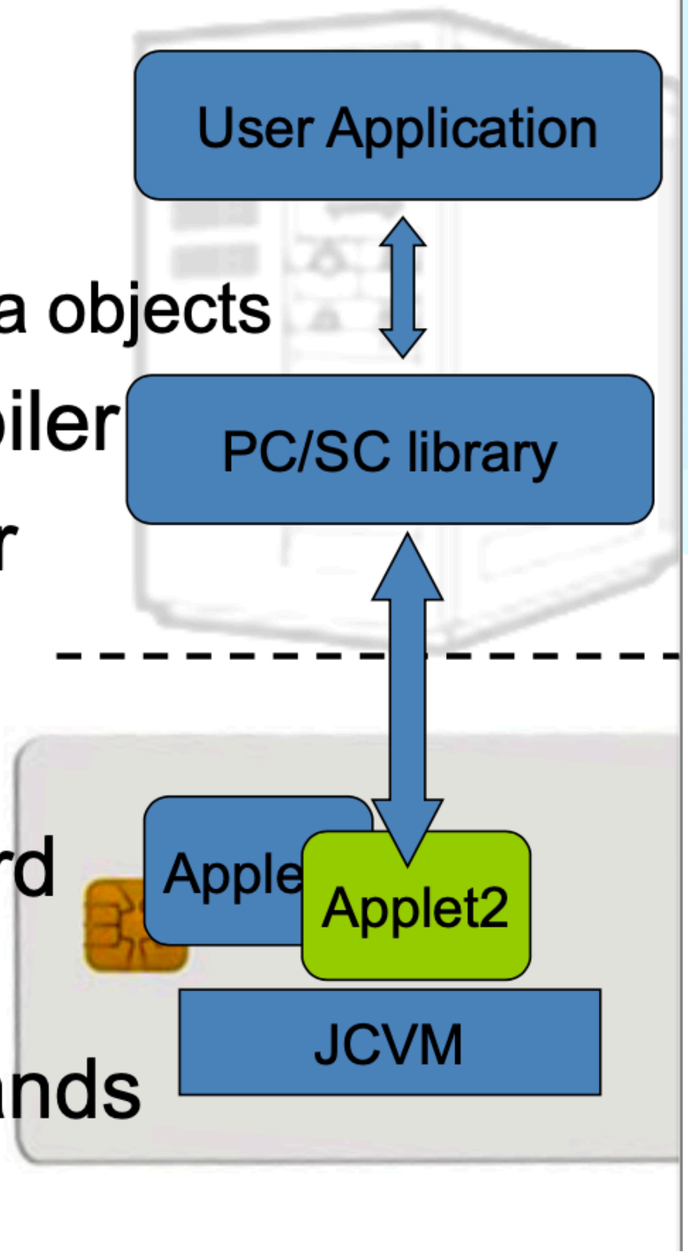
Simple JavaCard applet - code

1. Subclass javacard.framework.Applet
2. Allocate all necessary resources in constructor
3. Select suitable CLA and INS for your method
4. Parse incoming APDU in Applet::process() method
5. Call your method when your CLA and INS are set
6. Get incoming data from APDU object (getBuffer(), setIncomingAndReceive())
7. Use/modify data
8. Send response (setOutgoingAndSend())

Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

JavaCard applets

- Written in restricted Java syntax
  - byte/short (int) only, missing most of Java objects
- Compiled using standard Java compiler
- Converted using JavaCard converter
  - check bytecode for restrictions
  - can be signed, encrypted…
- Uploaded and installed into smartcard
  - executed in JC Virtual Machine
- Communication using APDU commands
  - small packets with header



Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

Sending and receiving data

- `javacard.framework.APDU`
  - incoming and outgoing data in APDU object
- Obtaining just apdu header
  - APDU::getBuffer()
- Receive data from terminal
  - APDU::setIncomingAndReceive()
- Send outgoing data
  - APDU::setOutgoingAndSend()

```
package example;
import javacard.framework.*;

public class HelloWorld extends Applet {
   protected HelloWorld()  {
     register();
   }
   public static void install(byte[] bArray, short bOffset, byte bLength) {
      new HelloWorld();
   }
   public boolean select() {
      return true;
   }
   public void process(APDU apdu) {
      // get the APDU buffer
      byte[] apduBuffer = apdu.getBuffer();
      // ignore the applet select command dispached to the process
      if (selectingApplet()) return;
      // APDU instruction parser
      if (apduBuffer[ISO7816.OFFSET_CLA] == CLA_MYCLASS) &&
         apduBuffer[ISO7816.OFFSET_INS] == INS_MYINS)) {
         MyMethod(apdu);
      }
      else ISOException.throwIt( ISO7816.SW_INS_NOT_SUPPOR
   }
   public void MyMethod(APDU apdu) { /* ... */ }
}
```

include packages from javacard.*

extends Applet

Called only once, do all allocations&init HERE

Called repeatedly on application select, do all temporaries preparation HERE

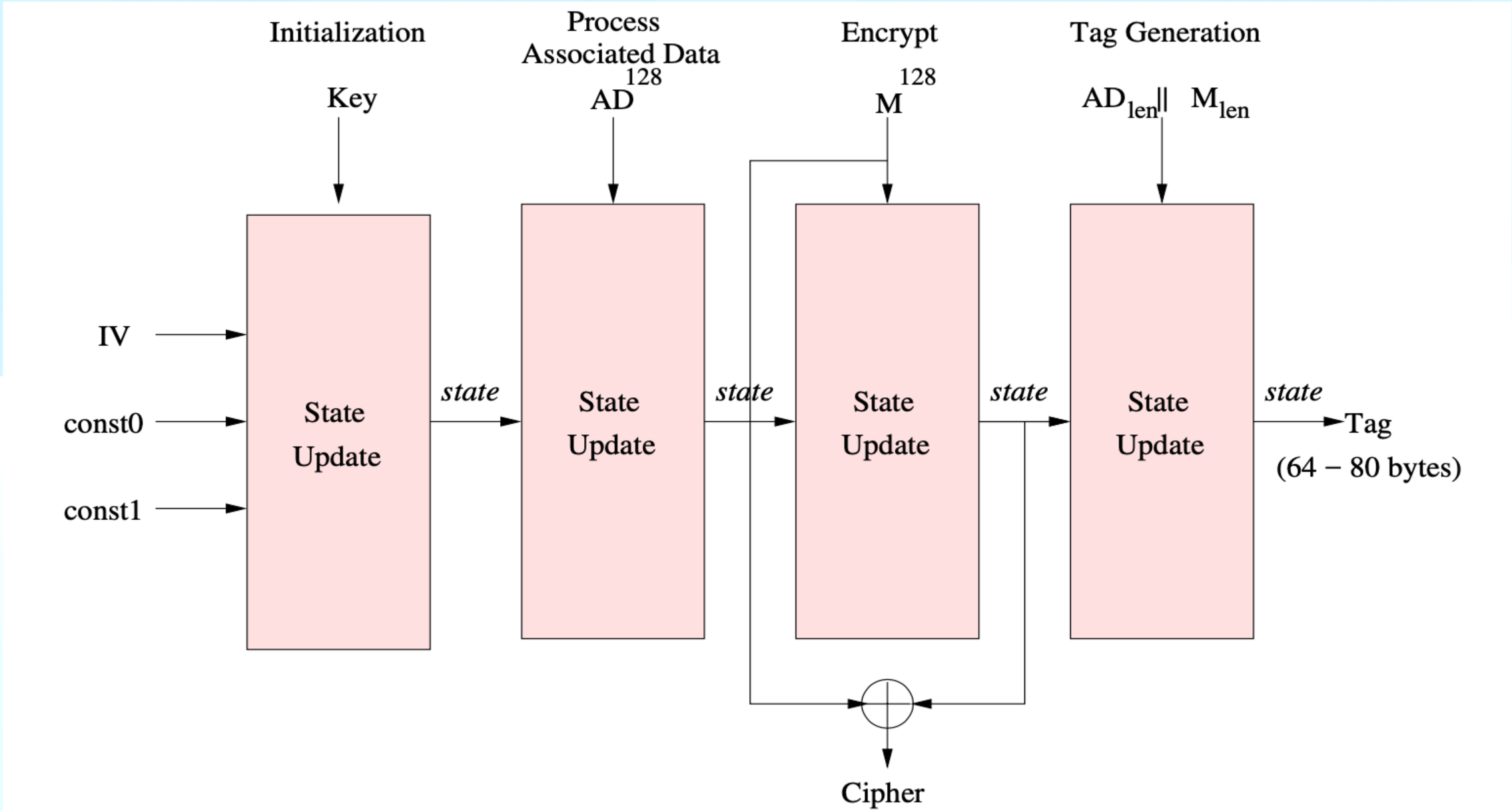
Called repeatedly for every incoming APDU, parse and call your code HERE

Credit & Gratitude: Petr Svenda @ Masaryk University, Brno

Constrained Environment

- Memory constraints
- Processing speed constraints
- Only byte & char
- No multidimension arrays
- No structure

AEGIS: Authenticated Encryption



How to Build, Burn & Run Apps on Java Card

Development Tools

- JDK 1.8 and Netbeans 15: Dev environment and JDK
- Java Card Development Kit 2.2.2: Java Card Dev Kit/ libraries/ api
- Apache-ant 1.10: Build .cap files
- Global Platform Pro: Burn app on smart Card
- PCSC lite: Communication drivers at host for communicating with smart Card

Playing with Java Card

Walk-through to the participants

References

- Gemalto. Java Card™ & STK Applet Development Guidelines. *http://developer.gemalto.com*
- PV204 course at Masaryk University by Petr Svenda
  - Credit & Gratitude: https://crocs.fi.muni.cz/people/svenda
- Sun Microsystems. Java Card Applet Developer's Guide. ftp://ftp.icm.edu.pl/packages/javasoft-docs/javacard/JCADG.pdf
- Martin Paljak. Applet Playground. https://github.com/martinpaljak/AppletPlayground
- Ludovic Rousseau. PCSC lite project. https://pcsclite.apdu.fr/
- Rajesh Kumar Pal. Authenticated Encryption on Java Card. *https://github.com/palkrajesh/AEonJC*

Thank You