

Insecurities of smart cards

Željko Vrba
Departement of Informatics
University of Oslo

Why smart cards?

- Two-factor authentication
- Security
 - Tamper-proof
 - Non-exportable keys
 - Certifications
- Standardized programming APIs
 - PKCS#11
 - MS CAPI
- Endorsed in some countries for legally binding signatures

Drawbacks

- Need additional hardware and software
 - Card itself
 - Reader
 - Middleware and (lack of) software to use it
- Often non-trivial installation for the average user

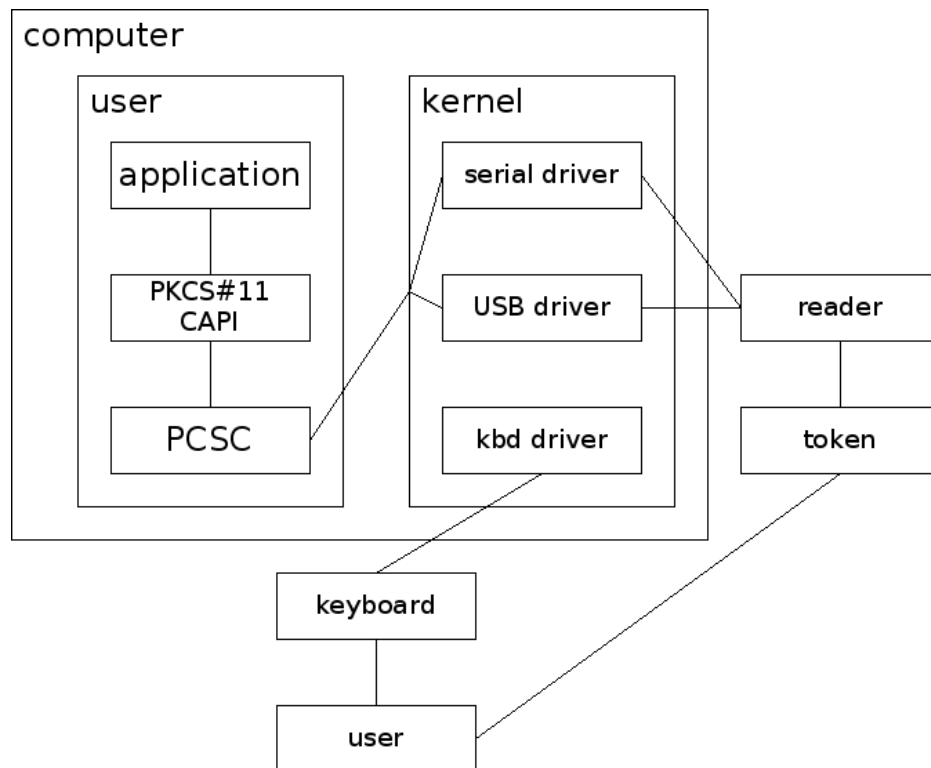
Capabilities

- ISO7816 standards:
http://www.cardwerk.com/smartcards/smartcard_standard_ISO7816.aspx
 1. physical characteristics
 2. dimensions and location of the contacts
 3. transmission protocols
 4. industry commands for interchange
- Asymmetric: RSA 1024 bit; on-card key generation
- 8-64kB EEPROM capacity
 - Static storage
 - Multi application (esp. Java Card)

Certification

- FIPS most commonly used/cited:
<http://csrc.nist.gov/cryptval/vallists.htm>
- 4 security levels
- Security policy
- Currently, no smart card is certified together with its middleware
- Alternative: EAL

Running environment (1)



- Layered software (and hardware!) architecture
- Many components = many weak points
- Effective attacks without breaking cryptographic primitives

Running environment (2)

- OS
 - rich and powerful debugging primitives
 - API hooking
- Helpful to developers, and attackers
- Can be used to completely compromise an improperly configured system
- Generic capabilities: find process, function address in DLL, attach to process, read/write registers and memory, breakpoints

Exploitation (1)

- Library hooking
 - Intercept C_Login, C_Sign, C_Encrypt, C_Decrypt
- Countermeasures:
 - Use pin pad (only protects the PIN)
 - Statically link to PKCS#11
 - Verify the result for correctness

Exploitation (2)

- PCSC layer
 - Intercept APDUs and steal the PIN
 - APDUs depend on the smart card; often undocumented
- Countermeasures:
 - Encrypted communication with the card; ineffective
 - Static linking

Exploitation (3)

- Memory and swap file scanning
 - Take a snapshot of processes memory and inspect it off-line for sensitive data
 - Make it dump core
 - Defeats encrypted communication
- Countermeasures:
 - Proper system configuration, memory locking, zeroing sensitive data when not needed, disable core dumps, ...

Exploitation (4)

- Software keyloggers
 - Records all users' keystrokes
- Countermeasures:
 - Pinpad
 - Disable module loading in kernels (compile-time, securelevel)
 - OS should provide trusted path
 - Proper OS configuration

Exploitation (5)

- Intercept all traffic at the USB level
 - Requires administrative privileges to install “filter” drivers
 - Useful for reverse-engineering the card protocol itself
 - <http://www.hhdsoftware.com/usbmon.html>
 - <http://sourceforge.net/projects/usbsnoop/>
- Countermeasures:
 - Encrypted communication

Other attacks

- Hardware attacks
 - FireWire bus
 - HW keyloggers (readily available on ThinkGeek)
 - Cameras and microphones
- Denial of service attacks
 - Deliberately present wrong PIN several times in a row
 - What if the key is used for encryption and the user doesn't have a backup
 - Non-extractable keys!

Legal questions

- **DISCLAIMER: I'M NOT A LAWYER!**
- Non-repudiation
- Signature: a conscious act of acceptance
- Scenario: a person
 - Wilfully commits fraud and claims that it was “some virus”
 - Prepares the virus in advance
- Who takes the liability in the court case?

Alternatives to smart cards

- Tokens (e.g. Banking)
 - No setup, almost trivial usage
 - Lower “mathematical”, higher “factual” security
 - nonprogrammability
- High-end cryptographic hardware
 - Large memory, multiple keys
 - Cryptographic accelerators
 - Trusted execution engines
 - Even more expensive (~ few kEUR/piece)

Conclusion

- Hard to properly secure OSes
- Fatal security flaws:
 - Programmability
 - Layered architecture
- Currently – expert only technology even if there were no security concerns
- TPM
 - AMD secure execution mode; Intel LaGrande:
<http://www.intel.com/technology/security/>