Common Criteria Certification of a Smartcard: a Technical Overview

CHES 2016 tutorial #1

Victor LOMNE

ANSSI (French Network and Information Security Agency)

Santa Barbara, USA, Thuesday, August 16th, 2016

ANSSI (French Network and Info. Security Agency)

- French governmental agency, created in 2009
- ANSSI core missions:
  - Detect and react to cyber-attacks
    cyber-defense center working 24/7 (CERT-FR)
  - Technical assistance to governmental and private entities
  - Inform companies and the public about threats and related means of protection
  - Prevent threats by supporting trusted IT products through several security labels

Me

- ANSSI Hardware Security Lab.
- Crypto. implementations, Embedded and Hardware Security
- Development of physical attack platforms
- Security expertise on ICs, smartcards, SoCs, ...
- Academic research
- Follow up of technical part of certification projects for the French Certification Center

Disclaimer:

Any resemblance to real smartcard is purely coincidental

Agenda

1 CC Basics
  a. CC Fundamentals
  b. Common Criteria Classes

2 Smartcard and Similar Devices
  a. Products Considered
  b. Specificities for Smartcards and Similar Devices

3 Attack Paths
  a. Hardware Attacks
  b. Software Attacks

4 Attack Rating
  a. How to Compute an Attack?
  b. Attack Factors

5 Attack Rating Examples

6 Evolution

Agenda

1 CC Basics
   a. CC Fundamentals
   b. Common Criteria Classes

2 Smartcard and Similar Devices
   a. Products Considered
   b. Specificities for Smartcards and Similar Devices

3 Attack Paths
   a. Hardware Attacks
   b. Software Attacks

4 Attack Rating
   a. How to Compute an Attack?
   b. Attack Factors

5 Attack Rating Examples

6 Evolution

Common Criteria

- Common Criteria for Information Technology Security Evaluation (abbreviated as Common Criteria or CC)
- International standard (ISO/IEC 15408) for IT products security certification
- Currently in version 3.1 revision 4

Definition

- Framework in which:

  1. Users specify their security requirements
  2. Vendors implement the security requirements in their products
  3. Evaluation laboratories evaluate the security of the products
  4. Certification bodies certify the products security by checking the correctness of all steps

History

- CC originated out of three standards:
  - ITSEC, the European standard, developed in the early 1990s by France, Germany, the Netherlands and the UK
  - CTCPEC, the Canadian standard
  - TCSEC, the United States Department of Defense standard, called the Orange Book

Key Documents

- CC v3.1 Release 4 consists of three parts:
  - Part 1: Introduction and general model
  - Part 2: Security functional requirements
  - Part 3: Security assurance requirements
- CEM v3.1 consists of one part
  - CEM: Common Evaluation Methodology
  - It describes the general evaluation process

Product categories

- CC is a framework allowing to assess the security of all kind of IT products

- For all products, the CEM document gives guidelines for the evaluation process

- But for specific kind of products, dedicated documents refine the CEM:
  - Smartcards and similar devices
  - Hardware Security Boxes

Key Concepts (1/2)

- **Target Of Evaluation (TOE)**: (part of) the product that is the subject of the evaluation
- **Security Target (ST)**: document that identifies the security properties of the TOE (may refer to a PP)
- **Protection Profile (PP)**: document, typically created by a user or user community, which identifies security requirements for a class of security devices
- **Security Functional Requirements (SFRs)**: security functions which have to be provided by the TOE (c.f. CC part 2)

Key Concepts (2/2)

- **Security Assurance Requirements (SARs)**: measures taken during development and evaluation of the product to check claimed security functionalities
(c.f. CC part 3)

- **Evaluation Assurance Level (EAL)**: numerical rating describing depth and rigor of an evaluation
EAL1 (most basic) ... EAL7 (most stringent)

Evaluation Technical Report (ETR): document written by the evaluator summarizing the results of the evaluation, esp. vulnerability analysis and penetration tests

Example: Biometric Passport (1/3)

- TOE: Biometric Passport + Environnment
- ST: refers to PP
- PP: PP for Biometric Passport

Example: Biometric Passport (2/3)

- Example of SFRs on the TOE:
  - TOE must ensure integrity of user data stored in the Passport and exchanged with the terminal
  - TOE must ensure authenticity of Passport data
  - TOE must ensure confidentiality of Passport data
  - TOE must ensure that tracebility data cannot be collected
  - TOE must be protected against information leakage, cloning, DoS, ...

Example: Biometric Passport (3/3)

- Example of SFRs on the TOE environment:
  - Passport emitter must deliver and accept the use of terminal following current laws
  - Passport emitter must ensure that personnalization is correctly done
  - Terminal must follow several rules
    cryptography, protocols, ...
  - Country emitting the Passport must follow several rules
    PKI, authenticity check of the Passport when traveling, ...

Agenda

1 CC Basics
   a. CC Fundamentals
   b. Common Criteria Classes

2 Smartcard and Similar Devices
   a. Products Considered
   b. Specificities for Smartcards and Similar Devices

3 Attack Paths
   a. Hardware Attacks
   b. Software Attacks

4 Attack Rating
   a. How to Compute an Attack?
   b. Attack Factors

5 Attack Rating Examples

6 Evolution

EAL: Evaluation Assurance Level

- Several certification levels exist
  - EAL1: functionally tested
  - EAL2: structurally tested
  - EAL3: methodically tested and checked
  - EAL4: methodically designed, tested and reviewed
  - EAL5: semiformally designed and tested
  - EAL6: semiformally verified design and tested
  - EAL7: formally verified design and tested
- EAL can be seen as a global rating of several classes, where each class has to reach a certain value

Common Criteria Classes

- CC define 6 classes, each one divided in subclasses:
  - class **ASE**: ASE_INT, ASE_CCL, ASE_SPD, ASE_OBJ
  - class **ADV**: ADV_FSP, ADV_ARC, ADV_TDS, ADV_IMP, ADV_INT, ADV_SPM
  - class **ALC**: ALC_LCD, ALC_CMC, ALC_CMS, ALC_DVS, ALC_TAT, ALC_FLR, ALC_DEL
  - class **AGD**: AGD_PRE, AGD_OPE
  - class **ATE**: ATE_COV, ATE_DPT, ATE_FUN, ATE_IND
  - class **AVA**: AVA_VAN

How it works ?

- 5 classes check the TOE conformity
  (ASE, ADV, ALC, AGD, ATE)

- 1 class checks the TOE security
  (AVA)

- Depending of the targeted EAL, each subclass must reach a certain value
  e.g. EAL4, EAL5, ...

- For some products, a symbol + can be added to the EAL meaning that a sublcass has been augmented
  e.g. EAL4+, EAL5+, ...

Example

- EAL4:
  - ADV: ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3
  - ...
  - AVA: AVA_VAN.3
- EAL4+ with AVA_VAN5:
  - ADV: ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3
  - ...
  - AVA: **AVA_VAN.5**

Class ASE: Security Target Evaluation (1/3)

- This class must show that the ST is full and technically coherent
- Goal for the customer / risk manager:
  - ensure that the TOE meets its functional and security needs
- Goal for the evaluator:
  - ensure that the ST is correct

Class ASE: Security Target Evaluation (2/3)

- Subclasses of class ASE (**S**ecurity Target **E**valuation):
  - ASE_**INT**: ST **INT**roduction
  - ASE_**CCL**: **C**onformance **CL**aims
  - ASE_**SPD**: **S**ecurity **P**roblem **D**efinition
  - ASE_**OBJ**: Security **OBJ**ective

Class ASE: Security Target Evaluation (3/3)

- Document(s) belonging to class ASE provided by the developer:
  - Security Target (ST)

Class ADV: Development (1/3)

- This class describes the security functions of the TOE from the specifications defined in the ST until the implementation phase

- Goal for the evaluator:
  - understand the architecture, the implementation and the internal process of the TOE
  - allow the evaluator to perform a vulnerability analysis (white-box approach) and to define the penetration tests to perform

Class ADV: Development (2/3)

- Subclasses of class ADV (**D**evelopment):
  - ADV_**FSP**: **F**unctional **SP**ecification
  - ADV_**ARC**: **ARC**hitecture Description
  - ADV_**TDS**: **T**OE **D**e**S**ign
  - ADV_**IMP**: **IMP**lementation Representation
  - ADV_**INT**: TOE Security Functions **INT**ernals
  - ADV_**SPM**: **S**ecurity **P**olicy **M**odeling

Class ADV: Development (3/3)

- Document(s) belonging to class ADV provided by the developer:
  - Architecture and design documents
  - Source code

Class ALC: Life Cycle (1/3)

- This class describes the life cycle of the product, the development tools and the security of the developer
- Goal for the customer / risk manager:
  - ensure that the developer has taken into account the life cycle of the TOE and the associated risks
- Goal for the evaluator:
  - ensure that measures taken by the developer are sufficient about the potential risk
  - estimate how hard it is for an adversary to get information on the TOE
    confidential documentation, samples, source code

Class ALC: Life Cycle (2/3)

- Subclasses of class ALC (**L**ife **C**ycle):
  - ALC_**LCD**: **L**ife **C**ycle **D**escription
  - ALC_**CMC**: **C**onfiguration **M**anagement **C**apabilities
  - ALC_**CMS**: **C**onfiguration **M**anagement **S**cope
  - ALC_**DVS**: **D**e**V**elopment **S**ecurity
  - ALC_**TAT**: **T**ools **A**nd **T**echnique
  - ALC_**FLR**: **FL**ow **R**emediation
  - ALC_**DEL**: **DEL**ivery

Class ALC: Life Cycle (3/3)

- Document(s) belonging to class ALC provided by the developer:
  - Documents describing the life cycle management of the TOE
- Document(s) belonging to class ALC provided by the evaluator:
  - Documents summarizing the developer sites audits

Class AGD: Guidance Documents (1/3)

- This class covers the necessary guidance documents mandatory for a secure use of the TOE by administrators and users

- Goal for the customer / risk manager:
  - know how to manage and use the TOE in optimal security conditions

- Goal for the evaluator:
  - ensure that the documentation is clear and full, and that it allows to use the product in conditions described in the ST

Class AGD: Guidance Documents (2/3)

- Subclasses of class AGD (**G**uidance **D**ocuments):
  - AGD_**PRE**: **PRE**parative procedures
  - AGD_**OPE**: **OPE**rational user guidance

Class AGD: Guidance Documents (3/3)

- Document(s) belonging to class AGD provided by the developer:
  - Guidance documents

Class ATE: Tests (1/3)

- This class describes the tests which show the compliance of the TOE against its specifications

- Goal for the customer / risk manager:
  - give proofs about the compliance of the product

- Goal for the evaluator:
  - check test results peformed by the developer and peform if necessary complementary tests

Class ATE: Tests (2/3)

- Subclasses of class ATE (**TE**sts):
  - ATE_**COV**: **COV**erage
  - ATE_**DPT**: **D**e**PT**h
  - ATE_**FUN**: **FUN**ctional tests
  - ATE_**IND**: **IND**ependent testing

Class ATE: Tests (3/3)

- Document(s) belonging to class ATE provided by the developer:
  - Documents summarizing tests performed

Class AVA: Vulnerability Assessment (1/3)

- This class describes the search for vulnerabilities and associated vulnerability tests, and define a rating scale for the attacks depending of the means of the adversary

- Goal for the customer / risk manager:
  - evaluate the risk for assets protected by the TOE to be extracted or modified

- Goal for the evaluator:
  - ensure the robustness of security functions of the TOE against an adversary

Class AVA: Vulnerability Assessment (2/3)

- Subclass of class AVA (**V**ulnerability **A**ssessment):
  - AVA_**VAN**: **V**ulnerability **AN**alysis

Class AVA: Vulnerability Assessment (3/3)

- Document(s) belonging to class ADV provided by the evaluator:
  - Evaluation Technical Report (ETR)

General Process

- Developer sends documents of the TOE to the evaluation laboratory and to the Certification Body (CB)
  ST, design docs., ...

- Developer sends product samples and optionnaly SDK, ... to the evaluation lab

- After reviewing all the documents, performing vulnerability analysis and penetration tests, the evaluation lab writes and sends the ETR to the CB

- The CB checks all the steps through the ETR, and emits the certificate if all is ok

Who pays who ?

- Developer pays the evaluation laboratory for the time spent on the evaluation

- Depending on the country, developer could have to pay the Certification Body for the time spent on the certification

- In France, certification is considered as a public service for both French and foreign companies
  ⇒ Free service

Agenda

1 CC Basics
  a. CC Fundamentals
  b. Common Criteria Classes

2 Smartcard and Similar Devices
  a. Products Considered
  b. Specificities for Smartcards and Similar Devices

3 Attack Paths
  a. Hardware Attacks
  b. Software Attacks

4 Attack Rating
  a. How to Compute an Attack?
  b. Attack Factors

5 Attack Rating Examples

6 Evolution

Smartcards and similar devices

1. Security IC
   - IC w/o O.S.
2. Open Platform w/o application(s)
   - IC + O.S. w/o app.
   - possibility to load post-issuance app(s).
   - e.g. JavaCard, ...
3. Open Platform w/ application(s)
   - IC + O.S. + app(s).
   - possibility to load post-issuance app(s).
   - e.g. USIM, Secure Element, ...
4. Closed Platform
   - IC + O.S. + app(s).
   - non possibility to load post-issuance app(s).
   - e.g. biometric passport, banking card, ...

Security IC

- A Security IC is generally composed of:
  - a CPU
  - one or several RAM(s)
  - one or several NVM(s) (ROM, EEPROM and/or Flash)
  - one or several cryptographic co-processor(s)
    e.g. (3)DES, AES, PKC accelerator, TRNG
  - optionnally a cryptographic library
    (generally relying on the crypto. co-proc. of the IC)
  - one or several security sensors
    e.g. glitch detector, light detector, shield, ...
  - a secure bootloader

Open Platform w/o application(s)

- An Open Platform w/o app(s) is generally composed of:
  - a security IC
  - an Operating System allowing to load post-issuance apps
    e.g. JavaCard OS, ...
  - a secure mechanism for loading post-issuance apps
- Example: JavaCard

Open Platform w/ application(s)

- An Open Platform w/ app(s) is generally composed of:
  - a security IC
  - an Operating System allowing to load post-issuance apps
    e.g. JavaCard OS, ...
  - a secure mechanism for loading post-issuance apps
  - one or several applications
    telecom app, ...
- Example: USIM, Secure Element

Closed Platform

- A Closed Platform is generally composed of:
    - a security IC
    - an Operating System not allowing to load post-issuance apps
      e.g. native OS, JavaCard OS, ...
    - one or several applications
      biometric passport app, banking app, ...
- Example: biometric passport, banking card, ...

Agenda

1 CC Basics
- a. CC Fundamentals
- b. Common Criteria Classes

2 Smartcard and Similar Devices
- a. Products Considered
- b. Specificities for Smartcards and Similar Devices

3 Attack Paths
- a. Hardware Attacks
- b. Software Attacks

4 Attack Rating
- a. How to Compute an Attack?
- b. Attack Factors

5 Attack Rating Examples

6 Evolution

How it works ?

- Depending of the targeted EAL, each subclass must reach a certain value
  e.g. EAL4, EAL5, ...
- For some products, a symbol + can be added to the EAL meaning that a sublcass has been augmented
  e.g. EAL4+, EAL5+, ...
- EAL levels used for smartcards and similar products:
  - Security IC: EAL5+ or EAL6+ (w/ AVA_VAN.5 and ALC_DVS.2)
  - Smartcards: EAL4+ or EAL5+ (w/ AVA_VAN.5 and ALC_DVS.2)
  - TPMs: EAL4+ (w/ AVA_VAN.4)

Concept of Composite Evaluation (1/2)

- A composite Evaluation is an evaluation of a product relying on the certification of a part of the product

- Example 1:

  1. IC manufacturer develops a new security IC
  2. Certification of the security IC
  3. Smartcard vendor develops a new banking card on the IC (e.g. IC + native O.S. + banking app.)
  4. Certification of banking card
     ⇒ use of the certification of the IC
     ⇒ composite evaluation

Concept of Composite Evaluation (2/2)

- Example 2:

1. IC manufacturer develops a new security IC
2. Certification of the security IC
3. Smartcard vendor A develops a new open platform
   (e.g. IC + JavaCard O.S.)
4. Certification of the open platform
   ⇒ use of the certification of the IC
   ⇒ first composite evaluation
5. Smartcard vendor B develops a new app. on the platform
   (e.g. IC + JavaCard O.S. + biometric passport app.)
6. Certification of the biometric passport
   ⇒ use of the certification of the open platform
   ⇒ second composite evaluation

Key Documents for Smartcards and Similar Devices

- Application of Attack Potential to Smartcards
- Protection Profiles (PP):
  - PP for Security IC Platform (PP 035 and 084)
  - PP for Trusted Platform Module (TPM)
  - PP for JavaCard system (closed or open)
  - PP for Biometric Passport (BAC, PACE, EAC, EAC with PACE)
  - PP for Universal SIM card (USIM)
  - PP for Embedded UICC (eUICC) → Secure Element

Agenda

1 CC Basics
   a. CC Fundamentals
   b. Common Criteria Classes

2 Smartcard and Similar Devices
   a. Products Considered
   b. Specificities for Smartcards and Similar Devices

3 Attack Paths
   a. Hardware Attacks
   b. Software Attacks

4 Attack Rating
   a. How to Compute an Attack?
   b. Attack Factors

5 Attack Rating Examples

6 Evolution

1/ Physical Attacks (1/2)

- Goals / equipment:
  - Probe an internal signal of the IC
    FIB, probing station, oscilloscope
  - Force an internal signal of the IC to a particular value
    FIB, probing station, oscilloscope, pattern generator
  - Read the ROM code
    dry and wet chemical tools, optical microscope or SEM
  - Read the Flash memory
    dry and wet chemical tools, AFM or SEM
  - Reverse-engineer a digital block
    dry and wet chemical tools, SEM, HW RE software

1/ Physical Attacks (2/2)

- Requires use of Failure Analysis tools and equipments
- Equipment often very expensive esp. for recent techno. nodes (e.g. SEM, FIB: 100k$ to 1M$)
  (most recent smartcard ICs made in 40-45nm)
- Time consuming if no access to the IC layout
  (several months / years)
- In CC evaluations, the evaluator generally uses the white-box approach to save time
  use of the VHDL / Verilog source code and Layout info.

2/ Overcoming Sensors and Filters (1/2)

- Goals / equipment:
  - Deactivate voltage detector
    FIB or fault injection station
  - Deactivate frequency detector
    FIB or fault injection station
  - Deactivate temperature detector
    FIB or fault injection station
  - Deactivate light detector
    FIB or fault injection station
  - Bypass anti-probing shield
    FIB

2/ Overcoming Sensors and Filters (2/2)

- Same remark than that for Physical Attacks about price of equipment / necessary time / whitebox approach when using a FIB

- Generally this attack path is only a partial attack, meaning that another partial attack has to be led to extract or modify an asset of the TOE

3/ Perturbation Attacks (1/2)

- Goals / equipment:
  - Modify normal execution of software
    fault injection station (glitch, laser, EMFI)
  - Modify normal execution of hardware
    fault injection station (glitch, laser, EMFI)
  - Alter memory reading / writing
    fault injection station (glitch, laser, EMFI)
  - Modify register value(s)
    fault injection station (glitch, laser, EMFI)

3/ Perturbation Attacks (2/2)

- Glitch based fault injection station
  several k$
- EMFI based fault injection station
  several dozen of k$
- Laser based fault injection station
  several dozen of k$
- State-of-the-art attacks use real-time pattern matching module on analogue signals
  (typically on power consumption of IC)
- State-of-the-art attacks also consider multi fault attacks (temporal and / or spatio-temporal)

4/ Retrieving Keys with FA (1/2)

- Goals / equipment:
  - Generate a wrong cryptographic result (ciphertext, signature, ...)
    fault injection station (glitch, laser, EMFI), FA software
  - Generate a correct cryptographic result although injecting a fault with a known effect
    fault injection station (glitch, laser, EMFI), FA software

4/ Retrieving Keys with FA (2/2)

- Several Fault Attack techniques exist
  DFA, CFA, SEA, SFA, ...

- White-box approach allows the evaluator to know the secret
  it allows to guess where the fault has potentially been induced, and then if it is an exploitable fault

5/ Side-Channel Attacks (1/3)

- Goals / equipment:
  - Retrieve a cryptographic secret
    power and EM measurement station, SCA software

5/ Side-Channel Attacks (2/3)

- The first step is to experimentally find a meaningful side-channel signal

- A second step often required is the resynchronization between measurements
  (acquired traces are often misaligned)

- A lot of Side-Channel Attack techniques exist
  SPA, DPA, CPA, MIA, LRA, TA, ...

- White-box approach allows the evaluator to know the secret
  it allows to characterize potential leakages

5/ Side-Channel Attacks (3/3)

- Classical methodology:

  1. By knowing inputs and secret, perform a leakage characterization
  2. If leakage found, redo previous step with available countermeasures activated
  3. If leakage still found, try to perform a key-recovery attack

6/ Exploitation of Test Features (1/2)

- Goals / equipment:
  - Enter IC test mode to dump NVM content
    FIB or fault injection station
  - Enter IC test mode to modify product life cycle
    FIB or fault injection station

6/ Exploitation of Test Features (2/2)

- Same remark than that for Physical Attacks about price of equipment / necessary time / whitebox approach when using a FIB

- Sometimes this attack path is only a partial attack, meaning that another partial attack has to be led to get an asset of the TOE

7/ Attacks on RNG

- Goals / equipment:
  - Stuck at a fixed value one or several bit(s) of the stream
    fault injection station
  - Induce a bias into the stream generated by the RNG
    freezing station, power or EM harmonic injection station

Agenda

1 CC Basics
  a. CC Fundamentals
  b. Common Criteria Classes
2 Smartcard and Similar Devices
  a. Products Considered
  b. Specificities for Smartcards and Similar Devices
3 Attack Paths
  a. Hardware Attacks
  b. Software Attacks
4 Attack Rating
  a. How to Compute an Attack?
  b. Attack Factors
5 Attack Rating Examples
6 Evolution

8/ Protocol Attacks

- Goals / equipment:
  - Editing commands
    (custom) smartcard reader
  - Direct protocol attacks
    (custom) smartcard reader
  - Man-in-the-middle attacks
    (custom) smartcard reader
  - Replay attacks
    (custom) smartcard reader
  - Buffer / stack overflow
    (custom) smartcard reader

9/ JavaCard Attacks

- Goals / equipment:
  - Bypass applet isolation
    smartcard reader
  - Escape from JavaCard attack
    smartcard reader
  - Combined attack
    fault injection station, smartcard reader

Agenda

1 CC Basics
  a. CC Fundamentals
  b. Common Criteria Classes

2 Smartcard and Similar Devices
  a. Products Considered
  b. Specificities for Smartcards and Similar Devices

3 Attack Paths
  a. Hardware Attacks
  b. Software Attacks

4 Attack Rating
  a. How to Compute an Attack?
  b. Attack Factors

5 Attack Rating Examples

6 Evolution

Overview

- When rating an attack, one considers two steps:
  - Identification: effort required to create and apply the attack to the TOE for the first time
  - Exploitation: effort required to apply the attack to the TOE knowing the techniques developed in the ident. step
- An attack is divided in attack factors, allowing to evaluate the difficulty of the different attack aspects
- The more an attack factor is difficult to apply, the more its rating is high
- The full rating of an attack is obtained by summing the rating of all attack factors of both steps

Overview

VAN level	Range of values	TOE resistant to attackers with attack potential of:
VAN.1	0 - 15	No rating
VAN.2	16 - 20	Basic
VAN.3	21 - 24	Enhanced-Basic
VAN.4	25 - 30	Moderate
VAN.5	$\geq$ 31	High

Examples

- Security IC has to reach VAN.5 security:
  - Either each tested attack is failed or each successful attack must rate 31 points or more
  - TOE resistant to attackers with high attack potential
- TPM has to reach VAN.4 security:
  - Either each tested attack is failed or each successful attack must rate 25 points or more
  - TOE resistant to attackers with moderate attack potential

Agenda

1 CC Basics

a. CC Fundamentals

b. Common Criteria Classes

2 Smartcard and Similar Devices

a. Products Considered

b. Specificities for Smartcards and Similar Devices

3 Attack Paths

a. Hardware Attacks

b. Software Attacks

4 Attack Rating

a. How to Compute an Attack?

b. Attack Factors

5 Attack Rating Examples

6 Evolution

1/ Elapsed time

	Identification	Exploitation
< one hour	0	0
< one day	1	3
< one week	2	4
< one month	3	6
> one month	5	8
not practical	*	*

2/ Expertise

	Identification	Exploitation
Layman	0	0
Proficient	2	2
Expert	5	4
Multiple Expert	7	6

2/ Expertise - Examples

- Side-channel adversary developing a new attack
  ⇒ Expert
- Side-channel adversary applying a known attack
  ⇒ Proficient
- Fault injection adversary developing a new attack
  ⇒ Expert
- Fault injection adversary applying a known attack
  ⇒ Proficient or Expert
- Adversary performing a Combined attack
  (JavaCard + Laser injection)
  ⇒ Multiple Expert

3/ Knowledge of the TOE

	Identification	Exploitation
Public	0	0
Restricted	2	2
Sensitive	4	3
Critical	6	5
Very critical HW design	9	NA

3/ Knowledge of the TOE - Examples

- Only public information
  ⇒ Public
- Access to non-public datasheet
  ⇒ Restricted
- Access to design description
  ⇒ Sensitive
- Access to source code
  ⇒ Critical
- Access to hardware source code
  ⇒ Very Critical HW design

4/ Access to the TOE

	Identification	Exploitation
< 10 samples	0	0
< 30 samples	1	2
< 100 samples	2	4
> 100 samples	3	6
not practical	*	*

5/ Open Samples / Samples with Known Secrets

	Identification	Exploitation
Public	0	NA
Restricted	2	NA
Sensitive	4	NA
Critical	6	NA

5/ Open Samples / Samples with Known Secrets - Examples

- Samples from the field
  ⇒ Public
- Samples with known key (e.g. TA)
  ⇒ Restricted
- Samples with known key and mask(s)
  ⇒ Sensitive
- Samples with FIB preparation
  ⇒ Critical

6/ Equipment

	Identification	Exploitation
None	0	0
Standard	1	2
Specialized	3	4
Bespoke	5	6
Multi Bespoke	7	8

6/ Equipment - Examples

- Online service for brute-force attack
  ⇒ None

- Computer + smartcard reader
  ⇒ Standard

- Side-channel / fault injection platform
  ⇒ Specialized

- FIB / Probing station
  ⇒ Bespoke

Agenda

1 CC Basics

a. CC Fundamentals

b. Common Criteria Classes

2 Smartcard and Similar Devices

a. Products Considered

b. Specificities for Smartcards and Similar Devices

3 Attack Paths

a. Hardware Attacks

b. Software Attacks

4 Attack Rating

a. How to Compute an Attack?

b. Attack Factors

5 Attack Rating Examples

6 Evolution

Explanations

- In the rest of this section, 5 cases are considered
- For each case, a security mechanism is described
- Then the identification and exploitation phases of the attack are detailled
- Finally the rating of the attack is given
- These 5 cases can be used as training

1/ SCA on AES co-processor w/o CM (1/2)

- Target:
  - straightforward AES co-processor without countermeasure
- Attack steps for identification phase:
  1. 10000 measurements
     (3 hours)
  2. No resynchronization step
  3. Classical CPA
     (10 minute)
- Attack steps for exploitation phase:
  1. 1000 measurements
     (20 minutes)
  2. No resynchronization step
  3. Classical CPA
     (1 minute)

1/ SCA on AES co-processor w/o CM (2/2)

- Rating table:

Factor	Comments	Ident.	Exploit.
Elapsed Time	In Ident. phase phase, signal search, data collection and DPA analysis will take less than one day. In Exploit. phase it will take less than one hour.	1	0
Expertise	For Ident. phase, an expert is required. Only a proficient is required for Exploit. phase	5	2
Knowledge of the TOE	No knowledge of the TOE is required for both phases	0	0
Open Samples / Known Key	No Open Sample is required for both phases	0	NA
Access to TOE	One sample is enough to mount the attack	0	0
Equipement	A SCA station is required for both phases	3	4
Sub Total		9	6
Total		15	

2/ SCA on AES co-processor w/ CM (1/2)

- Target:
  - AES co-processor with jittered clock and 1^{st} order masking scheme
- Attack steps for identification phase:
  1. 1000000 measurements with masks set at known values for charac.
     (7 days)
  2. Resynchronization step
     (2 days)
  3. 1^{st}-order leakage charac. + 2^{nd}-order CPA
     (2 days)
- Attack steps for exploitation phase:
  1. 500000 measurements with masks set at unknown value
     (3 days)
  2. Resynchronization step
     (several hours)
  3. second-order CPA
     (several hours)

2/ SCA on AES co-processor w/ CM (2/2)

- Rating table:

Factor	Comments	Ident.	Exploit.
Elapsed Time	In Ident. phase, understand and RE the masking scheme will take less than one month. In Exploit. phase, data collection and DPA analysis will take less than one week	3	4
Expertise	For Ident. phase, an expert is required. Only a proficient is required for Exploit. phase	5	2
Knowledge of the TOE	The datasheet is necessary for the Ident. phase. Nothing is required for the Exploit. phase	2	0
Open Samples / Known Key	An Open Sample is required for the Ident. phase	4	NA
Access to TOE	One sample is enough to mount the attack	0	0
Equipement	A SCA station is required for both phases	3	4
Sub Total		17	10
Total		27	

3/ FA on Internal Autenticate (1/2)

- Target:
  - Banking card Internal Autenticate command
  - Generate a faulty RSA signature for Bellcore Attack (RSA CRT implem.)
- Attack steps for identification phase:
  1. Spatial laser cartography #1 to find PKC co-processor weak spot
     Performed on IC open sample (1 week)
  2. Spatial laser cartography #2 to find spot for code flow modif.
     Performed on IC open sample (1 week)
  3. Temporal laser cartography for bypass of signature check
     Performed on TOE (1 week)
- Attack steps for exploitation phase:
  1. Short spatial laser cartography #1 to find PKC co-processor weak spot
     Performed on IC open sample (1 day)
  2. Short spatial laser cartography #2 to find spot for code flow modif.
     Performed on IC open sample (1 day)
  3. Temporal laser cartography for bypass of signature check
     Performed on TOE (3 days)

3/ FA on Internal Autenticate (2/2)

- Rating table:

Factor	Comments	Ident.	Exploit.
Elapsed Time	In Ident. phase, laser cartographies will take less than one month. In Exploit. phase, full attack will take less than one week	3	4
Expertise	For Ident. phase, an expert is required. Only a proficient is required for Exploit. phase	5	2
Knowledge of the TOE	The IC datasheet is necessary for the Ident. phase. Nothing is required for the Exploit. phase	2	0
Open Samples / Known Key	An Open Sample is required for the Ident. phase	4	NA
Access to TOE	Less than 10 samples are enough to mount the attack	0	0
Equipement	A double laser injection station is required for both phases, as well as a real-time pattern matching module for sync.	3	4

Sub Total		17	10
Total		27	

4/ FIB + Probing on Security IC (1/2)

- Target:
  - Security IC with anti-probing shield
  - Probing the 8-bit instruction bus inside CPU glue logic
- Attack steps for identification phase:
  1. Shield bypass with FIB and use of VHDL/layout (1.5 week)
  2. Making of custom pads with FIB and use of VHDL/layout (1.5 week)
  3. Bus probing and decoding of instructions (1 week)
- Attack steps for exploitation phase:
  1. Shield bypass with FIB (1 week)
  2. Making of custom pads with FIB (1 week)
  3. Probing of the bus and decoding of instructions (1 day)

4/ FIB + Probing on Security IC (2/2)

- Rating table:

Factor	Comments	Ident.	Exploit.
Elapsed Time	Both phases will take less than one month	3	6
Expertise	An expert is required for both phases	5	4
Knowledge of the TOE	In Ident. phase hardware source code is required. Nothing is required in Exploit. phase	9	0
Open Samples / Known Key	No open sample is required for both phases	0	NA
Access to TOE	In both phase less than 10 samples are required	0	0
Equipement	A FIB and a probing station are required for both phases	5	6
Sub Total		22	16
Total		38	

5/ Template Attack on RSA (1/2)

- Target:
  - RSA Square and Multiply Always with message and exponent blinding
- Attack steps for identification phase:
  1. 1000000 measurements with randoms set at known values for charac.
     (7 days)
  2. Resynchronization step
     (2 days)
  3. Leakage charac. + Template building and matching phases
     (5 days)
- Attack steps for exploitation phase:
  1. 1 measurement with randoms set at unknown value
     (1 hour)
  2. Resynchronization step
     (several hours)
  3. Template Attack
     (several hours)

5/ Template Attack on RSA (2/2)

- Rating table:

Factor	Comments	Ident.	Exploit.
Elapsed Time	Ident. phase will take less than one month. Exploit. phase will take less than one day	3	3
Expertise	For Ident. phase, an expert is required. Only a proficient is required for Exploit. phase	5	2
Knowledge of the TOE	The datasheet is necessary for the Ident. phase. Nothing is required for the Exploit. phase	2	0
Open Samples / Known Key	An Open Sample is required for the Ident. phase	4	NA
Access to TOE	One sample is enough to mount the attack	0	0
Equipement	A SCA station is required for both phases	3	4

Sub Total		17	9
Total		26	

Agenda

1 CC Basics
   a. CC Fundamentals
   b. Common Criteria Classes

2 Smartcard and Similar Devices
   a. Products Considered
   b. Specificities for Smartcards and Similar Devices

3 Attack Paths
   a. Hardware Attacks
   b. Software Attacks

4 Attack Rating
   a. How to Compute an Attack?
   b. Attack Factors

5 Attack Rating Examples

6 Evolution

Common Criteria Recognition Agreements

- CCRA (Common Criteria Recognition Arrangement):
  - Worldwide recognition arrangement allowing the certificate of a product certified in a country A to be recognized in a country B (27 members)
  - Periodic audits between Certification Bodies (only about procedures)
  - Limitation about the maximum recognized AVA_VAN level ⇒ maximum AVA_VAN.2

SOG-IS (1/2)

- SOG-IS (Senior Official Group Info. Systems Security):
  - European recognition arrangement allowing the certificate of a product certified in a country A to be recognized in a country B (10 members)
  - Periodic audits between Certification Bodies (procedures and technical skills)
  - No limitation about the maximum recognized AVA_VAN level (for smartcards and similar devices)
    $\Rightarrow$ maximum AVA_VAN.5

SOG-IS (2/2)

- SOG-IS organisation:
  - SOG-IS MC (Management Committee)
  - JIWG (Joint Interpretations Working group)
    - JHAS (JIWG Hardware Attack Subgroup)
    - ISCI (International Smartcard Certification Initiative)
    - SOG-IS Crypto group

JHAS

- JHAS (JIWG Hardware Attacks Subgroup):
  - Update of attacks list
    Attack Methods for smartcards (confidential)
  - Update of attack rating rules
    Application of Attack Potential to smartcards (public)
  - Members are European CBs, IC manufacturers, smartcard vendors, evaluation laboratories
  - Strict rules to become member

Other Schemes (1/2)

- EMVCo
  - Specifications for worldwide interoperability of payment transactions
  - Private certification scheme
    (lab licensing, own certification process, ...)
  - Regular exchanges with JHAS for consistency of attacks list and rating rules

Other Schemes (2/2)

- Global Platform (JavaCard, TEE)
  - Specifications for JavaCard platforms and recently for TEE (Trusted Execution Environnment)
  - New private certification scheme for TEE (lab licensing, certification process based on CC, ...)
  - Regular exchanges with other schemes
    PP for TEE has been certified by ANSSI

Questions ?

- contact: victor.lomne@ssi.gouv.fr