EUROPEAN UNION AGENCY
FOR CYBERSECURITY

EUCC SCHEME

STATE-OF-THE-ART DOCUMENT

MINIMUM ITSEF REQUIREMENTS FOR SECURITY EVALUATIONS OF SMART CARDS AND SIMILAR DEVICES

Version 1.1, October 2023

Date	Version	Modification	Author's comments
October 2022	1.0	Creation	Version submitted to the ECCG Based on JIL-Minimum-ITSEF-Requirements-SC&SD-v2-1.pdf
October 2023	1.1	Addition of a document history section with a link to the SOG-IS document the state-of-the-art document is based on, where applicable	Based on ECCG comments received Approved for publication on 20/10/2023 by the ECCG and the European Commission

CONTENTS

1 INTRODUCTION

This state-of-the-art document supporting the EUCC scheme provides requirements related to the minimum capabilities an accredited ITSEF shall have in its premises to conduct the different types of attacks present in the state-of-the-art document APPLICATION OF ATTACK POTENTIAL TO SMARTCARDS AND SIMILAR DEVICES. These capabilities include the knowledge and the skills of their evaluators and the necessary equipment, and evaluation methodology description, all necessary to conduct the attacks mentioned above.

The capabilities are intended to cover the minimum requirements to perform the evaluation of an Integrated Circuit (IC), a crypto library, a platform, and Integrated Circuit Card (ICC) with sufficient guarantees.

This document is not intended to provide guidance on how an IC, crypto library, platform (IC + OS) or ICC (IC + OS + App) evaluation has to be performed, but it provides guidance to ensure ITSEFs have the necessary capabilities to conduct such evaluations.

2 REQUIRED CAPABILITIES FOR IC EVALUATIONS

2.1 OVERVIEW FOR AN IC EVALUATION

An IC evaluation requires the development of specific skills and knowledge. The aim of this state-of-the-art document supporting the EUCC scheme is to provide a technical guidance for evaluators running an IC evaluation and to expose the related minimum requirements. To achieve this, the following sections will encompass:

- The understanding of secure IC-based design (such as smartcard, secure element, etc.) and production process in general of the IC design and manufacturing process (refer to section 2.2).
- The understanding of secure IC technology, its underlying principles and the development equipment used by secure IC manufacturers (refer to section 2.3).
- The knowledge and experience in hardware physical attack techniques that could compromise a secure IC and an ability to use the related equipment to stress the hardware layers. This includes the understanding of the IC underlying physical principles (refer to section 2.4).
- The knowledge and experience in physical disruptions that could change the secure IC behaviour, with the aim to subsequently downgrade the security of the IC-based device. The ability to use related equipment to conduct physical disruptions and the understanding of related physical effects on the hardware (refer to section 2.4).
- The knowledge and experience in cryptographic attack techniques and the ability to perform the analysis (including data-capture and signal processing procedures) (refer to section 2.4).

2.2 IC DESIGN AND PRODUCTION PROCESS

IC hardware and software are in general developed by different companies. These components are then integrated and additional security relevant data is injected into the card.

The security objectives for an IC are twofold:

- Ensure a level of security for the card in the field.
- Maintain the level of security throughout the development and production process.

Although many specialists concentrate on security in the field (since the smartcard is delivered into a hostile, unregulated environment and may be subject to tampering), security during the development, production and personalization process is also important. The security objectives that a smartcard component is assessed against depend very much on the application context, which in turn can be dependent upon the production and personalization process. In particular, personalization affects the security functionality to be provided by the smartcard.

The Common Evaluation Methodology (CEM) depicts an ideal development process starting with a definition of requirements followed by the design process, implementation, testing, acceptance, delivery and usage. When looking at the components of a composite product this process must be interpreted and rearranged.

For instance, the chip manufacturer develops the design of the chip hardware and software for testing. He receives the software from the software developer to create the ROM image. Then the mask files are sent to the mask manufacturer. The masks or reticles are returned to the chip manufacturer. After wafer production the chips are tested and initialisation data (transport keys, traceability data) are injected into the EEPROM (or other non-volatile memory). The initialisation data is defined by the card manufacturer. Operational dies are delivered or directly embedded into modules. The protection of die delivery can be complex. The authentication mechanism is realised by the software manufacturer but used by the card manufacturer (or personalisation centre). The keys are generated by the card

manufacturer and injected into the card by the chip manufacturer using a procedure (for diversification etc.) defined by the card manufacturer.

In case of flash based ICs there are even more possibilities. The IC can be delivered either without any content at all, which requires the software developer to use the test interface to initialise the flash with the firmware, boot loader or with bootloader software, hardware drivers or even with an operating system. In any case, proper use of authentication mechanisms must make sure the integrity of the flash content and access to the download functionality of the IC is handled in a secure fashion.

These examples show that a real development process can be more complex than the assumed one by the CEM for conventional software or hardware products, since the complete life-cycle of a smartcard can be quite complex. Inputs and outputs are not always as simple as expected by the CEM. As a result, the corresponding assurance components of the CEM (for instance delivery) must be interpreted, refined, and rearranged as required. In addition, it must ensure the processes of different components (and their description in terms of Common Criteria assurance components) fit together.

The evaluator must understand the smartcard supply chain and its integration into the application context in order to interpret the CEM assurance requirements in an appropriate way. In particular, these assurance requirements are:

- Guidance
- Delivery
- Preparation procedures
- Tools and Techniques
- Life-Cycle Definition
- Development Security

In addition, differences between the evaluation of smartcard ICs and the evaluation of software means the interpretation of the CC assurance components of the classes ASE, ADV, ATE, and AVA are also required.

These interpretations of the CC assurance components and additional guidance are described in several EUCC state-of-the-art documents for smartcards and similar devices that are published on the ENISA website dedicated to cybersecurity certification.

2.3 SMARTCARD INTEGRATED CIRCUIT TECHNOLOGY

The evaluator must understand smartcard integrated circuit technology and the underlying principles to the extent necessary to comprehend the design decisions of the IC manufacturer. Basic knowledge is required of:

- Electron theory of semiconductors (physics) and the electrical behaviour of semiconductors and transistors.
- Physical and electrical behaviour of all standard materials used in integrated circuit manufacturing (for instance silicon, poly-silicon, metal, and isolating and passivation material).
- Production steps and the resulting layer structure on the chip's surface.

In addition, the evaluator must have detailed knowledge of:

- Physical layout (implementation on the semiconductor surface) of standard cells (simple gates), memory cells (E2PROM, RAM, ROM) and memory blocks.
- Layout principles and methods of routing and layering.
- Digital and analogue circuit engineering (digital gates of different complexity and standard analogue circuitry).
- Static and dynamic behaviour of digital and analogue circuitry.
- Microcontroller architecture and functionality.
- Realisation of standard circuitry as used in micro-controllers.

The evaluator must be able to understand the schematics (block diagrams, schematics on gate and transistor level). The functional components can be described in the form of standard schematics or in VHDL sources.

The evaluator must have knowledge of the VLSI design process and must understand the process from the schematics or VHDL sources (logical representation of the chip) to the actual layout and dice/wafers (physical representation). The evaluator must understand the processes of technology qualification, functional testing, characterisation, and reliability testing.

The evaluator must understand the development equipment used by the manufacturers for micro-controller software. This includes simulators, emulators, protocol analysers and special evaluation software masks. The evaluator must be

able to read micro-controller source code and to develop software for penetration testing and other investigations. Therefore, the evaluator must understand the CPU instruction set, the memory map and use of other peripheral units of the micro-controller.

2.4 SMARTCARD SPECIFIC ATTACKS

The following provides an overview about smartcard specific attacks. This is not a complete list but provides some examples. More detailed information about smartcard specific attacks in the context of CC-based evaluation can be found in the state-of-the-art document APPLICATION OF ATTACK POTENTIAL TO SMARTCARDS AND SIMILAR DEVICES.

The evaluator must have knowledge of standard smartcard fraud and attack scenarios and in principle be able to develop new ideas for such attacks. To be more specific, the evaluator must know about attack scenarios for ICs and smartcard software such as physical manipulation and probing, malfunction attacks, inherent and forced leakage attacks, abuse of test features, attacks on the implementation of cryptographic functionality implemented in hardware, software or in a combination of both, cryptographic attacks or software attacks. A multitude of such attack scenarios – along with quotations – is described in the state-of-the-art document cited above.

The evaluator must be able to adapt and combine these attack scenarios for the individual chip or smartcard being subject to evaluation. During the vulnerability analysis, the evaluator must be able to find possible weaknesses (in schematics and their realisation on the chip and the combination thereof) and be able to use the standard techniques to assess them.

The evaluator must have knowledge and experience in IC failure analysis to be used for physical manipulation and probing. The evaluator must at least understand the physical principles of this and be able to operate (as appropriate) the equipment classified as 'standard' and 'specialised'. Moreover, the evaluator must be able to use the 'bespoke' tools with the help of trained operators. The evaluator must know how these tools and techniques can be used during vulnerability analysis in order to assess the IC's security properties and functions. The method and purpose of using the equipment (especially Focused Ion Beam (FIB), Scanning Electron Microscope (SEM), EMMI or E-beam Tester) during the vulnerability assessment need not necessarily correspond to the expectations of the operating personnel. The evaluator should instruct the operating personnel in order to achieve a meaningful and independent evaluation. The evaluator himself shall maintain sufficient technical knowledge (for instance on how to operate IC failure analysis equipment), required for a meaningful instruction.

The evaluator must have sufficient knowledge in probability theory and design principles of RNGs. The evaluator must be able to identify and analyse those characteristics of a system or a process that have significant impact on the distribution of random numbers and to rate the randomness of number generation.

The evaluator must have knowledge and experience of other smartcard attacks (side channel attacks such as Timing Analysis, Differential Power Analysis (DPA), Differential EM radiation Analysis (DEMA), Template Attacks (TA); fault injection attacks such as DFA and related attacks) and possess the equipment (physical and analysis tools) necessary to perform such attacks. The evaluator must be able to operate this equipment (including data-capture procedures) and to perform the analysis (mathematics). Knowledge and experience in cryptography and standard cryptographic attack techniques for all type of algorithms involved is required. The underlying principles of side channel attacks as well as fault injection attacks (such as Differential Fault Analysis (DFA) and other attacks) must be fundamentally understood. In order to fully investigate for potential weaknesses, the evaluator must be able to detect vulnerabilities related to such attacks, encompassing EM emission analysis, single- and multi-laser attacks, etc.

The evaluator must be able to develop software to communicate with the smartcard. Therefore, the evaluator must understand the I/O protocol being supported, the operating conditions and the external command interface if being used or attacked. The evaluator must also understand the security concepts of smartcard software, including file structures, encoding of access rights, etc.

The evaluator must know how to handle chip card readers and be able to modify them in order to use the chips in different packages and to apply non-standard operating conditions. Therefore, the evaluator must be able to use standard equipment such as voltage supply, signal and function generators, oscilloscopes, and soldering irons. In addition, the evaluator shall know how to physically prepare samples (e.g. open package and remove metal layers); for instance to facilitate sophisticated light attacks or EM measurements, provide laser access, enable FIB probing, allow reverse engineering, etc.

The evaluator must be able to combine results of different capabilities described above. This comprises the application of failure analysis methods to localise components on smartcards in order to assess if design data can be substituted or to judge the effectiveness of different attack methods with the same target.

2.5 EQUIPMENT FOR IC EVALUATION

In order to accomplish the vulnerability analysis, physical manipulations and attack scenarios mentioned in section 1.4, the ITSEF must have unlimited access to, and own the majority of the tools necessary to perform those attacks and shall be able to use them efficiently. Categories of this equipment are listed below (the state-of-the-art document APPLICATION OF ATTACK POTENTIAL TO SMARTCARDS AND SIMILAR DEVICES provides further details on necessary equipment with their categorisation):

- Environment control equipment (e.g. to control communication, voltage, clock, and temperature)
- Chemical and mechanical lab equipment (i.e. for sample preparation and analysis)
- Imaging equipment (e.g. cameras, microscopes, SEM)
- Physical manipulation equipment (e.g. probe station, Focused Ion Beam)
- Design analysis tools (e.g. for chip layout analysis, RNG analysis)
- Protocol analysers (e.g., spy devices)
- Logical test tools (e.g. for interface testing, vulnerability scanning)
- Side Channel Analysis equipment (e.g. probes, oscilloscopes, analysis software)
- Perturbation equipment (e.g. pulse generators, lasers, smart triggering)

For the equipment categorised as 'bespoke', the evaluator must have a good understanding of the underlying physical principles and of the capabilities of the tools.

The tools shall allow flexible usage within their technical limits. The usage shall not be limited to the expectations of the operating personnel as already described in section 2.4. The tools shall enable the evaluator to customise attacks as it can be assumed for experts based on the implementation under assessment.

3 REQUIRED CAPABILITIES FOR COMPOSITE EVALUATIONS

Composite evaluations build upon an earlier certified product. The composite TOE could be the IC supplemented by a crypto library, a platform, or the full ICC including the application. Typically, the TOE concerns software added to the certified underlying product.

3.1 OVERVIEW FOR AN IC CARD OPERATING SYSTEM

3.1.1 Source Code Review

Currently, most smartcard software is written in the programming language C, followed by Java; while manual programming in Assembler language is rather seldom today (except for dedicated core routines). The evaluator needs a thorough understanding of the use of C or Java in the context of the specific hardware architecture and constraints of a smartcard IC; this refers especially to the constraints of Java for Java Card products. (Therefore, section 3.4 below is dedicated to Virtual Machines.)

Moreover, for an in-depth security analysis, an understanding of assembler code and intermediate code (like Java Card byte code) is required. In particular, a variety of security impacts (and defects) cannot be understood on the level of a higher language like C or Java, because they become only apparent in Assembler Code or byte code. Therefore, the importance of understanding Assembler Code produced by a compiler and security impacts of generation tools shall be explicitly emphasized.

3.1.2 Native I/O

Native I/O refers to technologies “at the bottom” of data transfer between a smartcard and a terminal (smartcard reader).

The evaluator needs to understand and be able to interpret different I/O layers ranging from basic interface specification like UART (for sending and receiving single bytes); over the basic command structure of smartcard commands (APDU – Application Protocol Data Unit); up to the level of commonly used data exchange protocols, e.g. (T0 / T1 for contact, and TCL / Single Wire Protocol (SWP) for contactless.

3.1.3 (Security) Protocol I/O

In contrast to Native I/O, Protocol I/O encompasses the security (mostly cryptographic) protocols employed in communication with a smartcard.

In the context of smartcard protocols, Secure Messaging is the term which comprises security features of data transmission between a smartcard and a terminal (or a remote server). Secure Messaging may include mutual or one-sided authentication between a smartcard and a host, message integrity, as well as confidentiality of messages.

The evaluator must understand the various standardized protocols that exist for Secure Messaging, like specified for Open Platform, ECC (European Citizen Card), BAC (Basic Access Control), PACE (Password Authenticated Connection Establishment), EAC (Extended Access Control), etc. Often these standards allow a high degree of flexibility in the configuration of security options, demanding scrutiny when evaluating a specific choice against a set of prerequisite requirements.

3.1.4 Content and Resource Management

The defining task of an operating system is the management of computational resources (like memory, RAM, I/O etc.) and the administration of access (interface) to such resources.

While the previous paragraphs dealt with the communication between a smartcard and the outside world, the focus shall lie here on the resource management inside the smartcard itself.

The evaluator first needs to understand the file structure (e.g. the hierarchy concept of Master Files, Dedicated Files and Elementary Files) and file access rights administration within a smartcard's operating system. Knowledge of the memory types (EE, Flash, ROM, RAM, special dedicated RAM (like Crypto-RAM, Buffer-RAM)) and memory management procedures (e.g. access limitations) are required.

For Java Cards, the concept of Security Domains and Application Isolation (formerly firewalling) needs to be profoundly understood. This is especially relevant for application management, which refers to the secure loading, administration, and deletion of application, as well as the access rights of such applications to the smartcard's resources.

3.2 IC CARD PRODUCTION CYCLE PROCESS

An IC Card is produced by a software developer based on an IC or platform of a (different) vendor. The software for the IC is called the embedded software.

The security objectives for an IC Card are twofold:

- Ensure a level of security for the IC Card in the field.
- Maintain the level of security throughout the development and production process.

Although many specialists concentrate on security in the field (since the IC card is delivered into a hostile, unregulated environment and may be subject to tampering), security during the development, production and personalization process is also important. The security objectives that a smartcard component is assessed against will depend very much on the application context, which can be dependent upon the production and personalization process. In particular, personalization affects the security functionality to be provided by the smartcard.

The CEM depicts an ideal development process starting with a definition of the requirements, followed by the design process, implementation, test, acceptance, delivery and usage. When looking at the components of a composite product this process must be interpreted and rearranged.

For instance, the embedded software is developed for a specific IC. The IC has undergone a hardware evaluation and provides security guidance documents in order to make the composite product secure. These guidance documents include information on how the IC must be used to make the IC Card a secure product – usually several items of information are included, ranging from secure use of the cryptographic components, the Random Number Generator and a secure boot procedure. The composite evaluator must therefore understand the importance of the mandatory IC (security) guidance documents. It must be assessed whether the security mechanisms that have been implemented in the embedded software fulfil the requirements mentioned in the (security) guidance documents.

When assembling the IC Card, several entities are involved. For ROM based ICs, the embedded software will be sent to the IC manufacturer, whereas for flash based ICs, software loading could be done by the embedded software developer or even a third party. After assembling the IC Card, it will be made ready for delivery to the final customer or personalization bureau by the software developer. This may involve pre-personalisation of the IC Card and applications. These processes typically involve protection by cryptographic operations. The composite evaluator must understand how all these security mechanisms are implemented to guarantee a secure IC Card production process (including personalization).

The embedded software developer may introduce security mechanisms for changing the behaviour of the IC Card, for example by patching mechanism. The patch mechanism allows loading new (potentially malicious) program code to the IC Card and requires authentication before a patch can be applied. The composite evaluator must be able to assess the security mechanisms involved in such a patch mechanism.

These examples show that a real development process can be more complex than the one assumed by the CEM for conventional software or hardware products, since the complete life-cycle of a smartcard can be quite complex. This life cycle involves several “players” such as the IC manufacturer, the Software Embedder, the Card Issuer (who usually remains the legal card owner even after card issuance), Application Providers, and the End Users (the “card holders”). Inputs and outputs are not always as simple as expected by the CEM, since there is a complex interaction between the aforementioned entities with regard to security relevant procedures such as code exchange, key administration, or applet loading. As a result, the corresponding assurance components of the CEM (for instance delivery) must be interpreted, refined, and rearranged if needed. In addition, it must be ensured that the processes of different components (and their description in terms of the CC Part 3 assurance components) fit together.

The evaluator must understand the smartcard supply chain and its integration into the application context in order to be able to interpret the CC assurance requirements in an appropriate way. In particular, these assurance requirements are:

- Guidance,
- Delivery,
- Preparation procedures,
- Tools and Techniques,
- Life-Cycle Definition,
- Development Security.

In addition, differences between the evaluation of smartcard ICs and the evaluation of software means that the interpretation of the CC assurance components of the classes ASE, ADV, ATE, and AVA is also required.

These interpretations of the CC assurance components and additional guidance are described in several EUCC state-of-the-art documents for Smartcards and similar devices that are published on the ENISA website dedicated to cybersecurity certification.

3.3 CRYPTOGRAPHIC SOFTWARE

Composite products may include (partial) software implementations of cryptographic algorithms. In addition to understanding the algorithms, the evaluator should also understand interaction aspects between software and hardware, and the effect of attacks on a software implementation.

3.3.1 Cryptographic library using a cryptographic coprocessor

This section covers typically asymmetric cryptography using crypto coprocessor such as RSA, ECC, but could also concern symmetric algorithms lying on a cryptographic accelerator.

Such implementations combine a software-based algorithm with a dedicated set of cryptographic features. Both fit closely together because of the nature of the cryptographic accelerator. The evaluator shall be able to identify weaknesses in the interaction between hardware and software.

There is a significant variety of different implementation of hardware-accelerated algorithms, particularly when it comes to big integer operations. As a result, a good knowledge of the different implementations and a strong algebraic and arithmetic mathematical background is necessary.

In addition, a large number of attack paths may compromise the algorithms and many of them are implementation-specific. Therefore, it is of high importance that the evaluator has strong knowledge of attacks and countermeasures to provide an in-depth analysis of the embedded cryptographic library.

Furthermore, the evaluator will not be able to assume a specific usage of the algorithm at this stage of the assessment. For instance, the format of the input data must remain agnostic. Therefore, the evaluator needs to take into account various scenarios encompassing the most representative cryptographic protocols potentially relying on the cryptographic algorithms.

3.3.2 Cryptographic software without dedicated hardware (HW) support

Different secret key implementation without any HW support or with a partial HW support can be found in several products. Such software implementations can involve several countermeasures like random permutations, dummy operations or random masking as depicted in various publications to protect the product against first and higher order side-channel attacks. It is also very important to analyse the key bit (bytes) manipulations that must protect against other statistical attacks like template attacks.

It is very important the evaluator has strong knowledge in the different side-channel and fault attack techniques that can defeat all these countermeasures if they are not strong enough nor properly implemented.

The evaluator shall understand the algorithms that fall in the evaluation scope of the TOE. We can list the following cases of software implemented algorithms that are frequently met in products:

- "stand-alone" implementation of AES, DES (in spite of existing HW support still SW implementations are used) can be used. The whole implementation is done in software which relies on the set of instructions provided by the IC core (CPU).
- Mixed software/hardware implementation of DES and AES where additional software and countermeasures are required to the accelerations offered by the HW.
- Implementation of algorithms for which usually no HW support on smartcard IC exists, like:
    - Hash algorithms: Sha1, Sha2, Sha3, Ripemd160, Md5, etc…
    - Various authentication algorithms for mobile networks (Milenage, TUAK; moreover, a multitude of proprietary algorithms).
    - Other secret key algorithms from different NIST or national scheme standards.

3.4 VIRTUAL MACHINE

A virtual machine, by definition, is a software implementation of a computing environment in which an operating system or program can be installed and run. An evaluator must understand how the virtual machine and the run time environment works and protects the security assets relying on the platform. Different basic knowledge and skills are required:

- Generic knowledge and experience on interpreted languages, such as Java Card, with specific knowledge on the virtual machine architecture and parts, supported instruction set and data types and structures.
- Knowledge on the different programming languages used for the native parts and interpreter implementation (lower layers) and also for the applications (upper layers).
- Knowledge and experience with the development process and involved tools for the different platform parts are required. Compilers, converters and simulators, as well as, their associated intermediate and final file types and configurations, are required to be known.

3.4.1 Runtime environment

Evaluators must understand how the Runtime Environment (RE) ensures the security model of the virtual platform is upheld. This comprehends a deep knowledge on the relationship and interactions between RE, operating system, applications and hardware, the RE lifetime and transaction mechanisms, how the RE allows application isolation and data sharing mechanisms and how the applications are loaded and managed are part of the RE core knowledge that is required to be deeply comprehended.

3.4.2 Application Programming Interface

An Application Programming Interface (API) defines a set of services which are available for the application developers and provide system services, such as application management, transaction management, communications or cryptographic functionality. Evaluators must know the scope of the API services and how applications access RE services and their security implications. API services for Card Holder Verification, card content management or cryptographic operations are examples of critical services which evaluators must have very specific understanding. It is also required to be able to develop and use the different provided API's in order to develop security testing applications.

3.5 ATTACKS

In the following a short overview of typical attacks that need to be considered for composite evaluations will be given. A more complete list is provided by the state-of-the-art document APPLICATION OF ATTACK POTENTIAL TO SMARTCARDS AND SIMILAR DEVICES. There is a large overlap with section 2.4 dedicated to IC evaluations but the focus is now on the embedded software to be added by the composite evaluation and the interplay between the already certified part(s) and the new software.

Typically, some additional hardware attacks need to be performed, although the hardware belongs to the already certified part: The evaluator will need to ascertain through side channel measurements and fault injection attacks that the software correctly utilises hardware protection features and adds additional protection when necessary. For example, it could be required to configure registers in a particular way, interpret attack attempts reported by the hardware properly or implement software counter-measures for increased side channel or fault injection resistance. It must be ensured that the TOE as a whole maintains the required security level and the evaluator must also consider the purpose, use cases and frequency of use of the cryptographic keys, algorithms, and secret data stored in the

TOE. The knowledge required to perform these attacks is identical to what is described in the corresponding paragraph in section 2.4.

Another topic that is concerned with the correct interplay of hardware and software is attacks on the random number generator. Again, correct use of a hardware TRNG and additional software measures such as post processing and on-line testing of random numbers must be verified. Attack methods include hardware attacks such as fault injection and software tools such as statistical analysis.

Primarily, the evaluator concentrates on the embedded software implemented on top of the already certified part. The embedded software can be very complex and the evaluator must develop a good understanding of its architecture, the interfaces, and protocols used for external communication, as well as the assets it is intended to protect. The evaluator must be able to review code, while tracing the use of assets and identifying vulnerabilities.

When attacking the software implementation from external interfaces, it is crucial the evaluator is able to communicate with the TOE, send arbitrary commands and exercise all life-cycle states. The evaluator will have knowledge of software debugging tools and in order to operate them efficiently, the evaluator must have knowledge of the programming language used for implementation, the assembly instructions available on the CPU and the functionality of a debugger (breakpoints, memory inspection). A supporting technique is to apply automated tools to the source code, which perform a static analysis. The evaluator must be able to interpret and judge the results.

Additionally, some TOEs such as Java Cards may allow the installation of additional software such as applets. If that is the case, the evaluator must be able to load additional applets onto the card. Good programming skills are required in this case and precise knowledge of the internal separation mechanisms of the TOE such as firewalls, memory management, and bytecode verification.

Based on his knowledge about the TOE and the technical abilities described in the previous paragraphs, the evaluator must develop attack scenarios aiming at revealing sensitive assets or circumventing the intended security functionality of the TOE. These can be logical attacks (e.g. side effects or unintended effects of legal commands and API functions, malformed commands, parameters, or confusion of the internal state of the TOE) on the available interfaces or a combination of logical and hardware attacks (such as fault injection). A broad range of attack ideas is given in the state-of-the-art document APPLICATION OF ATTACK POTENTIAL TO SMARTCARDS AND SIMILAR DEVICES.

In addition, the evaluator must develop new attacks or modify and adapt standard attacks to assess the specific implementation of the current TOE. In order to be successful, the evaluator must perform a careful vulnerability analysis and have good knowledge of all technologies described in section 3.1 to 3.4 of this chapter and possible attacks against them.

3.6 EQUIPMENT FOR A COMPOSITE EVALUATION

For the composite evaluation, use bespoke failure analysis equipment is not expected since the intrinsic resistance of the TOE against physical attacks has already been investigated during the IC evaluation and these kind of attacks are not influenced by the embedded software. On the other hand, most of the IC exhibits some remaining leakages or fault sensitivities that could be exploited by an attacker if the embedded software does not implement additional countermeasures. Finally, software attacks and combined attacks can only be investigated during composite evaluation since they are fully linked to the embedded software.

So in order to be able to evaluate the resistance of the final product the ITSEF must have unlimited access to equipment and tools that can be used to operate the above mentioned class of attacks. The categories of required equipment include:

- Environment control equipment (e.g. to control communication, voltage, clock, and temperature)
- Chemical and mechanical lab equipment (i.e. for sample preparation and analysis)
- Imaging equipment (e.g. cameras, microscopes)
- Logical test tools (e.g. for interface testing, vulnerability scanning, operating system testing, randomness analysis)
- Protocol analysers (e.g. spy devices)
- Side Channel Analysis equipment (e.g. probes, oscilloscopes, analysis software)
- Perturbation equipment (e.g. pulse generators, lasers, smart triggering)

For in depth analysis, it appears necessary to have tools with enough flexibility to customise the attacks in line with the implementation under assessment. This includes the combination of tools (test benches) described above.

ABOUT ENISA

The mission of the European Union Agency for Cybersecurity (ENISA) is to achieve a high common level of cybersecurity across the Union, by actively supporting Member States, Union institutions, bodies, offices and agencies in improving cybersecurity. We contribute to policy development and implementation, support capacity building and preparedness, facilitate operational cooperation at Union level, enhance the trustworthiness of ICT products, services and processes by rolling out cybersecurity certification schemes, enable knowledge sharing, research, innovation and awareness building, whilst developing cross-border communities. Our goal is to strengthen trust in the connected economy, boost resilience of the Union's infrastructure and services and keep our society cyber secure. More information about ENISA and its work can be found at www.enisa.europa.eu.

ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

Heraklion office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu