D1.1

Report on use case and architecture requirements

Project number:	317930
Project acronym:	HINT
Project title:	Holistic Approaches for Integrity of ICT-Systems
Start date of the project:	1st October, 2012
Duration:	36 months
Programme:	FP7/2007-2013

Deliverable type:	Report
Deliverable reference number:	ICT-317930 / D1.1 / 1.0
Work package contributing to the deliverable:	WP 1
Due date:	Feb 2013 – M05
Actual submission date:	28th February 2013

Responsible organisation:	MOR
Editor:	MOR (Carsten Rust)
Dissemination level:	Public
Revision:	1.0

Abstract:	This document introduces the two application scenarios planned for the HINT project (Unclonable ID Cards and Professional Mobile Radio) and analyses use case requirements for these scenarios, with the focus on security analysis. Moreover, the main building technologies for the R&D work in HINT are described.
Keywords:	Smart ID Cards, Professional Mobile Radio, Use Case Requirements, Security Analysis, Physically Unclonable Functions, Side Channel Analysis, Hardware Trojans, Architecture Requirements

Editor

Carsten Rust (MOR)

Contributors

Thomas Hübner (MOR)

Holger Bock (IFAT)

Marc Mouffron (CCS)

Julien Francq (CCS)

Jacques Fournier (CEA)

Jean-Baptiste Rigaud (ARMINES-ENSMSE)

Dave Singelée (KU Leuven)

Jeroen Delvaux (KU Leuven)

Verena Brunner (TEC)

Martin Deutschmann (TEC)

Disclaimer

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 317930.

Executive Summary

This report on use case and architecture requirements is the first technical deliverable of the HINT project. Together with the second deliverable of the work package on “User Requirements and System Architecture” it shall provide the basis for the research and development work in HINT.

The document starts with a review of the applications that are planned for evaluation and demonstration of HINT results. We give a general overview of the scenarios for unclonable ID Cards and Professional Mobile Radio (PMR) and outline the basic use cases to be considered. Based on this overview, the document describes:

- The functional and security requirements for HINT applications. The focus thereby is on the analysis of security requirements. The document provides a security analysis for the targeted use cases of both application scenarios and derives the requirements for the basic building blocks of the technology to be developed in HINT.
- The main stakeholders in HINT application scenarios and their specific requirements. Stakeholders for HINT scenarios include for instance the end user and the system operator, but also technology providers like semiconductor manufacturers or smart card vendors. Their requirements and potential benefits from the HINT technology will be described.
- The building technologies that are going to be researched and applied by the project. We are going to consider three topics of major relevance for HINT:
    1. Integrity verification & authentication schemes based on Side Channel Analysis (SCA),
    2. Techniques for detection of Hardware Trojans (as a special focus of the above ‘integrity checking’ aspect),
    3. Authentication schemes based on Physically Unclonable Functions (PUFs).

For each of these topics, the deliverable first describes the current state of the art and then derives the architecture requirements for hardware and firmware developments to be further specified and developed within the HINT project.

The current document will be completed in the remaining tasks of WP1. As part of its second deliverable D1.2, the work package will add descriptions of

- a trust architecture based on the building blocks described in this document.
- a security target and assessment.
- the overall architecture for the HINT demonstrators.

Contents

List of acronyms

AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuit
ATEX	Atmosphere Explosive (Explosive Atmosphere)
ATPG	Automatic Test Pattern Generation
CC	Common Criteria
COTS	Commercial Off The Shelf
CPLD	Complex Programmable Logic Device
CRP	Challenge Response Pair
DES	Data Encryption Standard
DPA	Differential Power Analysis
ECC	Elliptic Curve Cryptography
EM	Electromagnetic
EMA	Electromagnetic Analysis
FA	Fault Analysis
FPGA	Field Programmable Gate Array
HoDPA	Higher order Differential Power Analysis
HSM	Hardware Security Module
HT	Hardware Trojan
IC	Integrated Circuit
ID	Identity
IP	Intellectual Property
MIA	Mutual Information Analysis
PIN	Personal Identification Number
PMR	Professional Mobile Radio
PUF	Physically Unclonable Function
RFID	Radio-Frequency IDentification
RSA	Rivest Shamir Adleman public key encryption
SAM	Secure Access Module
SCA	Side Channel Analysis
SCARE	Side Channel Analysis for Reverse Engineering
TC	Trusted Computing
TPM	Trusted Platform Module
WAN	Wide Area Network

List of Figures

List of Tables

Chapter 1 Introduction

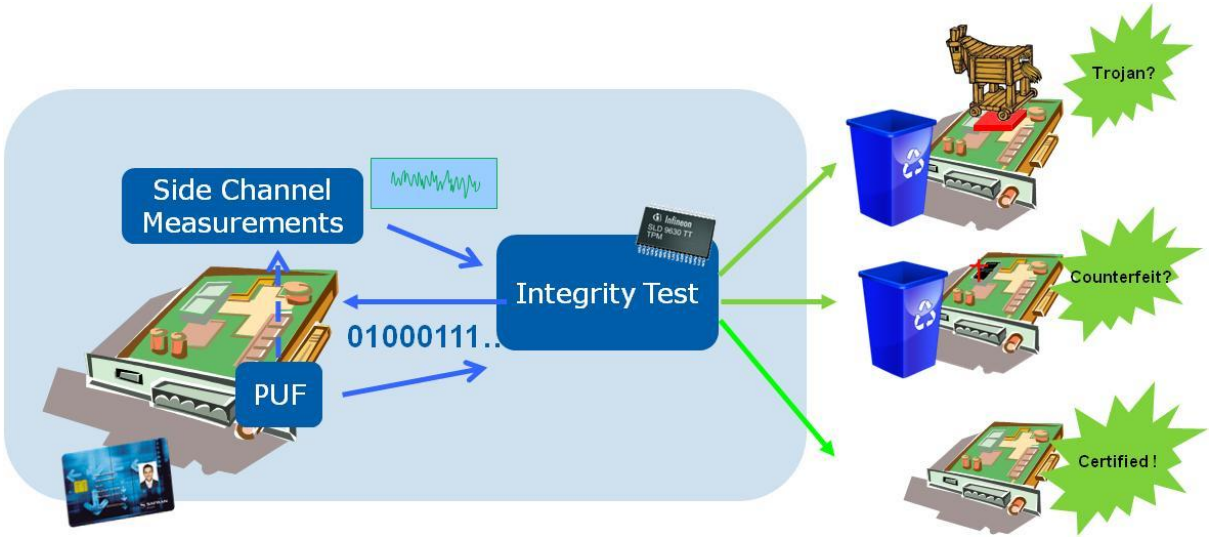

Figure 1: HINT Project Overview

Modern ICT (Information and Communication Technologies) systems involve complex schemes like in homeland security markets (avionics, critical infrastructures, SCADA systems, cyber security, RFIDs…), embedded systems (health, transport, defence, consumer electronics, telecommunication…), smart cards (bank cards, ID cards, Pay TV cards, transportation, (U)SIM…) and personal identity technologies (passports or travel documents…). The security of such systems, which relies on the authenticity and integrity of the hardware components used to implement them, is continuously challenged by improving attacks. Hence new methods for testing and guaranteeing the authenticity & integrity of those hardware devices must be sought.

Physical attacks, based on the passive or active spying of those devices, are 'proven' ways of retrieving secret data out of them. Today's security circuits offer protections against these attacks, but an absolute protection is not possible in practice and the need of extra barriers arises, especially with the growing concerns about the fact that:

- ⇨ counterfeiting of hardware components is dramatically increasing, with approximately 5-20% of counterfeited components on the market.
- ⇨ the threat of "Trojans" or hidden functions in Integrated Circuits (IC) is moving from theory to practice.

The HINT project addresses these new challenges by proposing the development of novel technologies to verify that a system is a genuine and non-modified one. Those technologies shall help to support assurance of authenticity and integrity of the hardware components used in a given system.

Secure architectures and platforms, where secure storage and computations are managed by hardware components, have shown their efficiency for many applications, from user's identification and authentication (e-id, banking cards), to ensuring the security of more complex systems (HSM, SAM, TPM, root of trust). However, even if the security of such tamper proof (or tamper resistant) integrated circuits is better than any other solution, some weaknesses still exist. The HINT project addresses these problems and intends to develop technologies enabling to:

- perform “on board” integrated checking of the global integrity of a system (hardware and embedded software)
- check the “genuineness“ of the secure integrated circuits (detection of functional clones or of counterfeited circuits, using PUF-based authentication schemes) and
- detect the presence of Hardware Trojans.

The mission of the HINT project is to develop a solution to implement a common framework for a system’s integrity checking based on Trusted Computing technologies. The capabilities of the developed technologies will be demonstrated with real-life applications. Moreover, the adaption of the proposed technologies by future Common Criteria evaluation schemes is going to be prepared.

To achieve those goals, the HINT project will focus on some specific technologies like:

- the PUF technology, enabling to authenticate a given hardware component using a physical, intrinsic and unique signature of the device.
- SCA (Side Channel Analysis) based analysis to monitor the behavior of hardware components and to detect changes from their original specifications and implementations.

This document is organized as follows: we first describe the two applications that shall be the focus of the HINT application. For each application we describe specific use cases that shall be used as reference application scenarios for the project. For those use cases, a security analysis is made resulting in security requirements. Stakeholders’ inputs are then described and analyzed resulting in what we call stakeholders’ requirements. The union of the security requirements and the stakeholders’ ones constitute the general requirements for the HINT project. Based on those requirements we then provide an overview of the main technological blocks that shall be investigated in this project in order to meet those requirements.

Chapter 2 HINT-related applications

The issue of measuring the integrity and authenticity of a given hardware or software entity is relevant to a whole range of applications, in other words any time a given system manages critical information or operations, with different levels of importance. In the HINT project, we focus on two applications (supported also by the expertise of the project partners) where those issues of integrity, genuineness and authenticity are of utmost importance given the high levels of security and trust needed by them: smart ID cards and Professional Mobile Radios (PMR).

2.1 Unclonable ID Cards

ID cards constitute one of the most important applications of smart card deployment to date. They comprise electronic passports, the European Citizen Card, national identity cards like the German nPA (“neuer Personalausweis” – new identity card), but also health insurance cards like Health Professional Cards or the EGK (“Elektronische Gesundheitskarte” – electronic health card). Some example cards are shown in Figure 2.

Figure 2: Current ID Cards

Two major categories of security relevant functionalities on such cards are:

- Performing *secure authentication* of the card holder,
- Generating *digital signatures* on behalf of the card holder.

Authentication is the process of verifying one’s own (claimed) identity to another party. Usually this is achieved through cryptographic protocols proving knowledge of a secret without revealing the secret itself to an eavesdropper or even (in case of asymmetric cryptography) towards the verifier. Such protocols are typically based on challenge-response procedures using digital signatures, or Diffie-Hellman variants (avoiding digital signatures).

Digital signatures generated by ID cards are a (legal) substitute for handwritten signatures, and as such one of the enhancements which the new electronic passports or ID cards can provide. As an individual may be held responsible for any assertion he or she has signed with an ID card, the security requirements for such signature cards are accordingly high. This concerns the choice of algorithm, key length, etc. as well as a wide range of security requirements covering hardware attacks (like side-channel analysis, perturbation attacks, and fault intrusion) against ID cards.

Cloning of ID cards (as considered within this chapter) refers to “identity theft”, in the sense of *impersonation* of an individual by an imposter. Cloning of cards with regard to illegal (unlicensed) re-production of HW-/SW- components violating intellectual property rights will be treated in a subsequent chapter.

In a rigid sense, an attacker intends to successfully (but falsely) authenticate as someone else not identical to him; or to generate a signature on some statement/document on behalf of someone else who will then later on be traced to an innocent individual who actually never released the signature at hand.

In a somewhat enlarged perspective such an attacker could even be identical to the legitimate holder himself, aiming at cloning “his own” ID card in order to obtain enlarged privileges for instance. Examples for such a motivation can be to enlarge access rights encoded on an ID card, extend credit-lines stored on a bank card, or manipulate information regarding the holder’s age. In this scenario, instead of manipulating the “genuine” card at hand (which may turn out to be prohibitive for technical reasons), the attacker builds a new card (the clone) which bears the enhanced privileges. It is worth noting in this context that normally even a legitimate card holder does not know (and indeed *must* not know) his own secret private key! (The very best a smart card based system can offer in this context is to generate such sensitive private keys directly on-card and never ever release these keys to anyone, including the manufacturer and the issuer of the card. However, in many cases private keys are generated off-card in a secure environment and subsequently stored on-card.)

The security of both authentication and signature generation basically rests on protecting and maintaining the secrecy (knowledge, access) of the corresponding private key of a legitimate card holder. It is the physical storage of such secret keys – along with other individual data – that defines the card “genuine” with respect to the card holder.

When dealing with authentication or signature generation of an individual card holder indeed *two* processes are involved:

The cryptographic authentication (or signature generation) is actually performed by the smart card and not by the human card holder himself or herself. The verifier can usually not know who is actually physically holding or inserting a card into a card reader.

Therefore, the first link in this chain is the authentication of the card holder towards his card himself, prior to the cryptographic procedure requested from the card. In other words, it must be ensured that only the legitimate card holder is able to activate a card command performing a security relevant operation. This authentication is based on basically two components: (implicitly) on physical possession/control and (explicitly) usually upon a PIN or password check. Both are weak principals in themselves, very much weaker than the cryptographic protocols subsequently processed between the smart card and the verifier (e.g. the network).

Cloning an ID card refers to the process of building a smart card which behaves (at least with respect to selected security operations) indistinguishably from the genuine card. This usually refers to the interaction (cryptographic protocols) between card and terminals/networks, and not to the interaction of the human holder with his card (like PIN-check). In a rather simplified manner, such a cloning procedure may consist in the following two steps:

- Obtain knowledge of a secret key belonging to a card holder and usually embedded in his or her “genuine” card
- Build another smart card and embed the secret key into the clone

We do not consider here attacks aimed at extracting a private key from a genuine ID card. There is a multitude of information in the literature upon this subject, however. Moreover, apart from attacks targeting genuine cards, such a secret key may even have been derived through other channels like corrupted personalisation facilities or leakage from authorities’ database containing such keys.

The second step is rather straightforward – provided that the attacker disposes of sufficient skill, equipment and material. Since authentication of a card is mainly based on the embedded authentication key, once an authentication key is in false hands a cloned card will usually be accepted as authentic.

In order to thwart such cloning attacks, their underlying principle must be defeated: it is not sufficient to declare a smart card’s genuineness merely through embedding confidential data into the card’s memory cells. In other words, there must be a very much tighter “link” between a physical card and individual information defining the card’s genuineness. This link must be much more intrinsic than physical storage of information in memory. Each ID card should be equipped with an individual physical “fingerprint” which cannot be read out or copied, and even if known cannot so easily be duplicated into a clone. Moreover authentication with this “fingerprint” should involve a challenge-response-like protocol so that the embedded fingerprint cannot be replicated (reverse-engineered) from eavesdroppers monitoring the communication exchanged during the authentication process.

This is where the PUF technology could come into play. A PUF embedded into a card makes each individual ID card a “genuine” piece of hardware itself – unlike a conventional ID card which becomes “genuine” merely by individual confidential information stored into its memory.

Several scenarios are conceivable here, a few of which shall be listed below:

- To begin with, an embedded PUF might be considered as an authentication “key” in its own right. Before performing a subsequent security operation with a memory-stored secret key, authenticity of an ID-card may be checked by a “Challenge-Response Pair” involving the PUF. In order to tie a subsequent cryptographic operation to such a successful PUF authentication (i.e. to prevent an imposter to “jump in” with a cloned card immediately following a correct authentication with the genuine one), a session key may be derived from it.
- A secret authentication or signature key may *itself* be embedded into a PUF instead of being stored in a card’s memory. This way not only extraction of an embedded key from a genuine card will be much more difficult. (Since the key is no longer embedded in any non-volatile memory cells any more, most attack scenarios will not be applicable. Dedicated attacks on PUFs need to be considered, of course.) Building a physical clone would then require to embed a functionally identical PUF into the designated copy of the card to be cloned. Moreover, there is a tight and direct link now between each physical sample card and the corresponding embedded secret key, since – simply spoken – the PUF *is* the authentication key itself.

The second proposal puts high requirements on an embedded PUF. It must be complex enough to allow embedding of a secret key with sufficient *key length* as required by current legislation and certification schemes for authentication cards and digital signatures. Moreover, key recovery needs to work reliably even in the presence of all kind of “noise”, i.e. the PUF response needs to be reproducible at least within the correction capacity of error correction procedures. In this context noise can range from the intrinsic noise (generated by variance of the physical switches upon which the PUF is built) to environmental conditions

(like radiation, temperature, etc.). Finally, key recovery should best be completely processed on-card, the extracted key only be temporarily stored in the card's RAM and safely deleted after its usage. On-card extraction of the key poses yet another set of requirements both on the smart card's abilities as well as on the PUF construction itself.

Even biometric information related to the legitimate card holder may be embedded into a template along with a PUF-"fingerprint". This way, the complementary link between the human card holder and his or her ID card can also be strengthened. As mentioned above, apart from physical possession this link currently doesn't consist of more than (simple) password authentication with a comparatively small password (usually a 4-6 digit PIN). To date, biometric authentication of a user towards his smart card (with on-card biometric matching performed) is already an established procedure for a number of years, but here the focus lies on *merging* such biometric features with a cell-based PUF in order to achieve a kind of integral "hybrid" PUF.

2.2 Professional Mobile Radio (PMR) Communication

In this section, we will detail the PMR application. The Public Safety Professional Mobile Radio (PMR) aims at giving Public Safety forces and organisations a mobile communications solution for their day to day law enforcement and incident response job. In section 2.2.1., we will give more details on the main PMR usages. In section 2.2.2, we will present the two main PMR products: PMR handheld and the Low Power Router. Finally, in sections 2.2.3 and 2.2.4., we will detail the PMR architecture and its associated life cycle management.

2.2.1 PMR usages

For decades, radio communication has been the solution for flexible and efficient communication in the field. Radio enables instant communication between two or more people simply by pressing a button.

From a service incident perspective, communities like countries or regions or cities are vulnerable to hazards resulting from the climate, natural disasters in addition to the more conventional incidents resulting from accident or deliberate disruptions. For instance these include:

- Natural Emergencies which are related to naturally occurring elements and conditions including but not limited to severe weather, floods, storms, ice and snow storms;
- Human-Caused Emergencies: those that are accidental and including chemical spills, explosions or leaks, plane crashes, train derailments, public transport or cars crashes, and power outages.
- Technological emergencies are also human caused and can affect critical infrastructure, computer technology, telecommunications and other information technology issues.
- Acts of human based disorder intended to disrupt community services or activities such as terrorist actions.
- Incidents that result from the special events from visits of foreign leaders, to 'big' cultural or sports events or to large scale demonstrations.

For example, Cassidian's PMR solutions have a strong track record in securing large events. Here are just a few of the events where they have been used successfully:

- G8/G20 Summit, Ontario, Canada, 2010
- Football World Cup, South Africa, 2010
- G20 Summit, Cannes, France, 2011
- 22nd Spanish-American Summit, Cadix, Spain, 2012

The PMR system provides enhanced capabilities to enable most effective communications (video/data transfer, voice, etc.), to enhance the situation awareness, collaboration and interoperation among various public security, safety, transportation, critical infrastructures and other public service organizations.

Hence the public security forces do have particular requirements regarding the **reliability** and the **availability** of the system, the available radio coverage and capacity to communicate outside of radio coverage.

Other reasons for adopting PMR solutions are also related to the need for special functionalities such as group calls, instant communications with a sub-second call set-up delay, security and specialised dispatching services. The latter imply managing the organisation's field operations and related communications. For public safety organisations, security is critical and includes **authentication of the users** in the network as well as **encryption of the voice and data** communication itself.

For many organisations, having control of their own network resources is crucial and in many cases PMR services also offer an economical benefit.

A PMR system relies on a trusted dedicated infrastructure: routing and management network, base stations, terminals, antennas, etc. The capacity to have reliable communications without infrastructure that means direct mode communications between the terminals is also a stringent need. It offers the following main services:

- Multi-users radio communications involving subscribers and operators (mission leaders located in control centres), with priority management
- Messages, geo-localisation, applications (database access, etc.)
- Emergency warning system

A PMR network can be shared between different organizations (Police, fire forces, emergency medical services, public transport services, Public work services, critical infrastructures services, etc.). Those organisations represent several 10 tens of thousands users (for a typical nation-wide system). A PMR network usually consists of several regional networks, federated to build a nation-wide network. The PMR network also interoperates with other networks or systems.

The main idea behind the end-user needs is that the PMR device is used in situations where "life is at stake". This can be the life of the user of this PMR device or the life of the people the user is providing help to.

2.2.2 Main PMR Products

We will consider in our study two PMR products:

- a PMR handheld
- a PMR low power vehicular router

They are chosen because they gather the most stringent requirements:

- these devises are on the field and are the most important part of the PMR system for the user.
- these devises have high quality and performances expectations.
- these devices can typically be used in very stringent environments.
- they may be unattended.
- they handle important user and missions information.

2.2.2.1 The PMR handheld

The PMR ATEX handheld is designed to provide high quality voice and data communications during critical missions. ATEX certification, for use in potentially explosive environments, enables users to work in complete safety in places where inflammable substances are produced, processed, transported and stored. Its robust design makes it suitable for use in the harshest conditions.

Figure 3: Handheld products

The internal hardware architecture of this handheld includes 4 boards: the display board, the keypad board, the SIM board and the main board. Those boards are highly integrated and include many COTS or specific ASICs as well as CPLD/FPGA. Its intrinsically safe “ATEX certification” requires a specific design and manufacturing process using reliable components. The origin and quality of the components are absolutely vital for such products.

2.2.2.2 The Low Power Router

The Low Power Router, converged voice and data mobile office solution, combines routing, security and multimedia traffic forwarding into a single cost-effective platform for public safety stakeholders. The Low Power Router delivers fast, secure, reliable, and scalable WAN access, making it ideal for users requiring high-speed IP or Internet access for both in-vehicle mobile and stationary installations.

Figure 4: Low Power Router

The internal hardware architecture includes 2 boards with many COTS or specific ASICs as well as CPLD/FPGA.

2.2.3 PMR Architecture

Figure 5 below illustrates the general architecture of a PMR product. Note that the underlying system is the same whether for the handheld or the low power router with the presence of the display and keypad as the only difference.

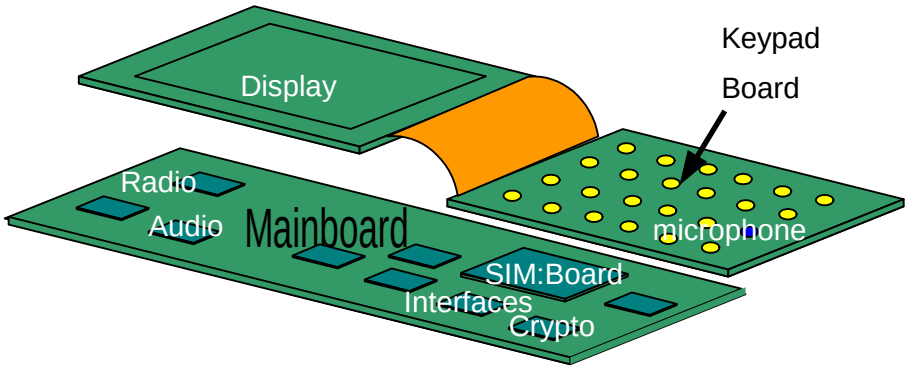

Figure 5: Combination of boards and components on Handheld products

The radio, audio, interfaces, crypto, SIM functions are implemented in one or several ASICs. CPLDs and FPGAs implement some interfaces and management functions.

2.2.4 PMR life cycle management

Before its delivery to an end user the PMR product follows a manufacturing process that is represented hereafter.

The simple mechanical or electronic components (either COTS or specific ASICs) are delivered by suppliers to the sub-part assemblers. These sub-part suppliers build the various boards, covers antennas… in their factories. Then a main manufacturing supplier gathers every needed sub-part and assembles the final product and tests it. At the end PMR-solution providers like Cassidian complete the tests, insert the security settings and pack the product for delivery.

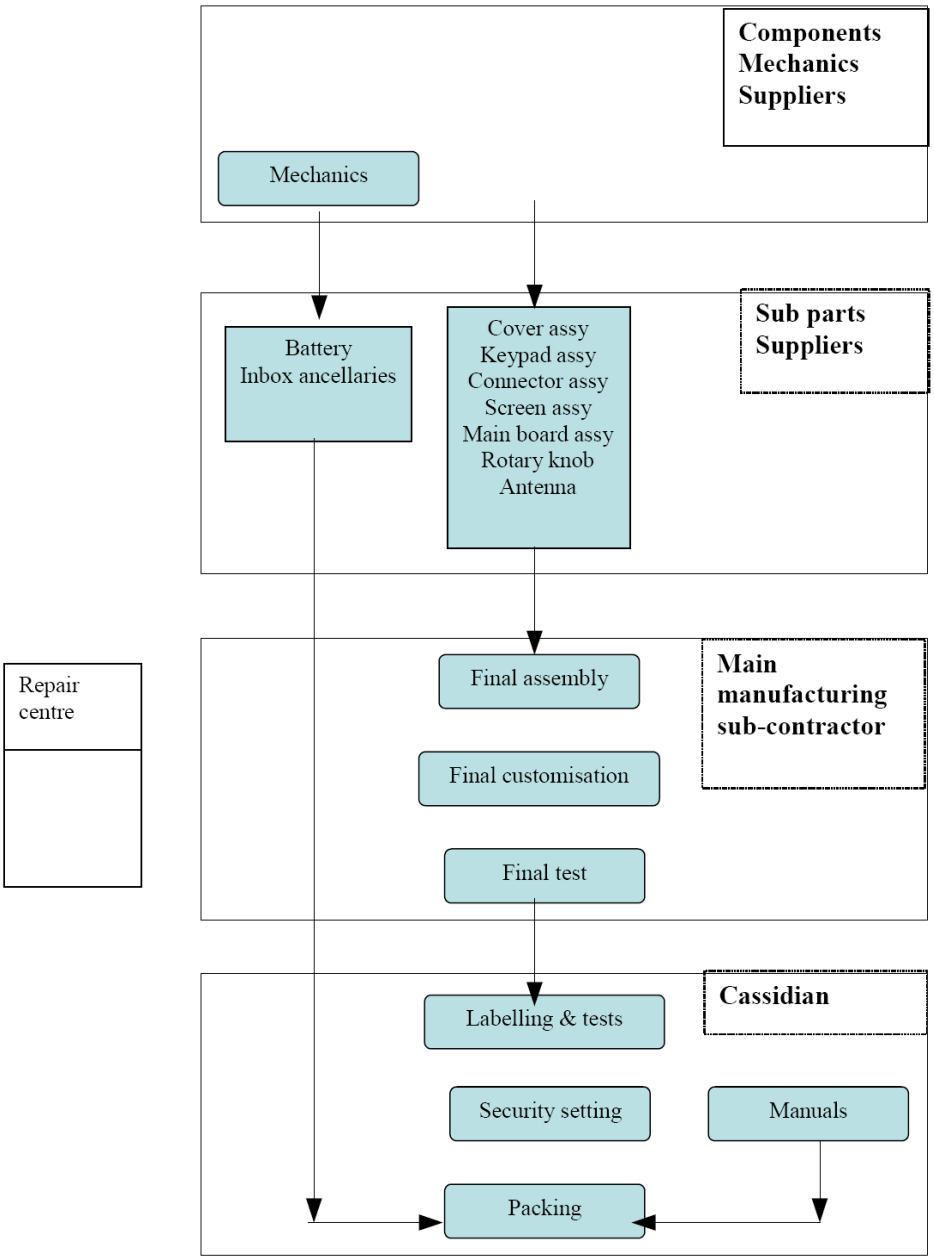

Figure 6: PMR Life Cycle

Chapter 3 HINT use case analysis

In the first section of this chapter, we introduce typical use cases for both ID cards and for the PMR. When these use cases for the two applications addressed by HINT have been described, we focus on the security issues linked to the use cases in order to identify the security requirements derived from them. Section 3.2 provides the security analysis for the ID card use cases and section 3.3 the analysis for the PMR use cases. Both sections are structured following the methodology proposed for Common Criteria evaluations. They differ however regarding the degree of formalization. For the evaluation of PMR equipment, the project aims at preparing a protection profile covering the – in this context – new aspects of integrity and authenticity. Therefore, section 3.3 provides a more extensive formal analysis, whereas section 3.2 is focused on the technical aspects.

3.1 Use cases

For each of the applications described in the previous chapter, we hereafter describe specific scenarios which shall be considered as reference application scenarios for the rest of the project. The specification of such use cases is of primary importance for the project in the sense that

- They provide a clear definition of the kind of concrete security and trust problems the HINT project shall focus on.
- They provide a clear reference for the definition of the requirements for the project, and hence a clear justification for the technical orientations adopted by the project.
- They provide reference scenarios that shall be adapted to demonstrate the technologies developed in the project.

3.1.1 ID card use cases

Authentication of an ID-card by a PUF

Instead of using an embedded private key on an ID-card, a PUF embedded into the physical circuitry of an ID card's IC – or an alternative strong hardware-based chip authentication mechanism - shall be used to authenticate the card towards a verifying entity.

It shall be demonstrated that cloned cards (containing all information along with an end-user's private keys) can be identified and rejected based on their PUF responses.

Signature Key recovery from an embedded PUF

Unlike in the first use-case, a PUF shall be used here not to directly authenticate, but to encapsulate an end-user's signature key (i.e. the private key used for digital signature). Extraction of this key shall be performed completely on-card, with the extracted key only stored in volatile memory.

This scenario shall demonstrate that PUFs can be used as secure storages without the need to use a smart card's non-volatile memory.

3.1.2 PMR use cases

In this section, we will study two use cases: a military and a fireman intervention. Each member of a military/fireman team has his own PMR terminal for communicating with its team members.

For the PMR equipment, HINT technologies will be used in order to implement mechanisms or define technical processes checking the integrity and authenticity of either a cryptoASIC (a cryptomodule implemented in an ASIC) or a CPLD/FPGA. They shall enable the detection of Hardware Trojans that attackers would like to insert into the PMR products. Section 3.3 will highlight some prominent problems and in what way they could be solved using HINT technologies.

Military operations

Communication can make the difference between an operation's success or failure. So, military soldiers have a requirement of confidentiality of their communications that can be achieved by a PMR handheld. This implies on the one hand that keys and certificates that shall be kept secret and genuine within the cryptoASIC embedded in the handheld and on the other hand that data stored are protected by these keys and certificates.

This use case is an example where components shall be prevented from information leakage that could result from HTs which HINT solution could detect. This scenario shall demonstrate that HINT technology will detect HTs leaking keys outside an ASIC or FPGA.

Firemen operations

The firemen need more and more communications to operate efficiently in fire fighting or other emergency operations. For this purpose they use PMR equipment.

We highlight here the specific cases were they operate to fight technological threats in chemical factories or oil refineries. Then they use ATEX Handheld.

The firemen are then relying on their handheld to make voice communications, see or send status information or send information from sensors to the command post. They are in this case very cautious on the reliability of these exchanges. In such sensitive environments, the additional need is that the handheld can do these tasks without any sparks to prevent explosion.

To fulfil such missions the handheld's parts and components shall behave correctly and avoid any "out of specification" behaviour. This is another example where component shall be prevented from misbehaviour that could result from HTs which HINT solution could detect. Such scenario shall demonstrate that HINT technology will detect HTs provoking the malfunction of a PMR handheld.

3.2 Security Analysis of the "ID card use case"

In this section, we first analyse the main assets of an ID card, as it is deployed in the above described usage scenarios. Subsequently, possible attack scenarios are outlined.

3.2.1 Assets

With respect to the dedicated use case which is the main purpose of an eID document and its chip, namely to provide a proof of identity of the document holder, we focus within HINT on several clearly defined assets:

- User identity, credentials
- Credentials & privileges associated to the card holder (but "owned" by the card issuer)
- Hardware *integrity*
- Software *integrity*
- Hardware / Software *authenticity*
- Intellectual property rights on hardware and software

1. User identity, credentials

   User Identity is in general proven by the possession (and presentation) of an authorized document fulfilling two requirements:

   First such document must be in a verifiable way connected to the citizen, which is traditionally achieved by inclusion of a photographic picture of the document holder, in modern electronic ID documents with embedded chip such picture data, or other biometric data like fingerprint, hand palm, iris or veins patterns are in addition stored electronically in the data memory section of the integrated circuit.

   On the other hand the authenticity of the document itself has to be made provable by verifiable properties of the issuing institution, traditionally being a seal and signature of the issuer, nowadays supplemented or substituted by enhanced security features of the ID card. Uniqueness of the issued card is achieved by specification of a unique serial number per device.

   In modern eID card systems the binding of the document to the issuing entity is usually supported by cryptographic means, especially by a public key infrastructure (PKI) including hierarchies of signing and verifying keys, organized in certificates and well defined logical data structures.

   Even more complex mechanisms may be provided in the future, if – e.g. due to privacy regulations – methods allowing for anonymous attestation or group signatures will have to be implemented. Such methods are based on credentials that are used to prove certain knowledge about a secret (key) without need of revealing the unique identity of the prover/signer to the verifier. Such credentials must not be copied, stolen or duplicated/cloned since they are representing special roles and associated rights of the ID document holder and are thus another asset of the same class as the user identity itself.

2. Credentials & privileges associated to the Card Holder (but '"owned" by the Card Issuer)

   For ID cards (and not only for them), usually the Card Issuer (e.g. the issuing state) assigns attributes defining the Card Holder's privileges, or confirms the validity of the Card Holder's credentials with his authority (e.g. a Card Holder's age, his nationality, etc.). Even a legitimate Card Holder may have a criminal incentive to alter such data without the

consent of the card issuer. Another less obvious situation is the use of a digital signature: it may only be admissible in contact with approved terminals, a condition which is usually verified by checking certificates ("Role authentication"). A Card Holder himself might wish to circumvent such restrictions. In addition, the card holder must not be able to insert and use an uncertified signature key in order to ensure non-repudiation of a signature.

3. Hardware *integrity*

   In the HINT ecosystem special emphasis is put to the hardware integrity of a system under consideration. State-of-the-art methods to assure integrity of ICT systems are based on secret or private keys on the one hand and on integrity checking of software on the other hand. Pieces of software are checked during boot and some runtime scenarios code sequences are checked, which are about to perform operations and protocols to act as a prover in a challenge-response scenario or as a signer in a signature/certificate scenario. In a trusted computing approach Hash values over certain portions of code are generated and stored so to ensure by hash-and-compare operations – e.g. during the boot process – that the software performing the critical steps in the protocol is integer and potentially authentic.

   Such methods and scenarios do not take into account the option of modified or cloned hardware performing the respective logical functions triggered by the software layer:

   Modified hardware could for example ignore the result of critical compare-operations and allow access to otherwise protected memory-areas and functions, could dump critical information like keys to I/O channels, or it could de-activate countermeasures to support attacks against which the device otherwise would be able to resist.

4. Hardware *authenticity*

   The threat to thwart is copying (including physical cloning) of hardware, which is not authorized by the manufacturer of the hardware, while it does not all damage the integrity of the copy, i.e. the behaviour of the copy is exactly the same as that of the original hardware. In this context, hardware authenticity refers to the property of a specific hardware instance having indeed been manufactured and released with the consent of the manufacturer/distributor who claims to have manufactured it.

   Cloned hardware – which would in case of ROM-based software representations automatically also contain an unauthorized instance of the system software and in case of fuse-based keys potentially also of secret or private keys – could be presented in a security relevant ecosystem and grant rights to the holder of the clone, which should be restricted to the holder of the genuine hardware. As seen from today's point of view for such cloning, expensive methods like reverse engineering must be applied.

   For both of these cases assurance of hardware *integrity* (= hardware being unchanged) and *authenticity* (= hardware being the very instance it is supposed to be) to a well-defined level, could reduce the risk of abuse significantly.

5. Software *integrity*

   Altering software can be an attack target by itself, or (more often) a means to facilitate another attack (for instance disabling checks to subsequently manipulate user data, or introducing SW Trojans).

   Means to detect altered SW are redundancy checks or hash-sums. Means to thwart altering are for instance encrypting SW (making it extremely difficult for an attacker to obtain a meaningful change).

6. Software *authenticity*

   Like for Hardware authenticity, this refers to the copying of (unaltered) software in an unauthorized way. The incentive can be the violation of IP rights (see below), or even ID

theft (in this case credentials should be copied along with the software, which is usually difficult because software and user credentials are separated from each other).

"Controlling" of software authenticity without binding it to specific Hardware authenticity can be done by tying it to specific keys, for instance through memory encryption (which is not very efficient for ROM-stored SW, while for Flash-Devices even device-specific encryption keys can be used). Once software has been leaked to an attacker who is able to store it on HW by himself, the situation becomes much more difficult.

A point worth stressing is the meaning of "copy" here: often minor changes in HW or SW design may be performed by an attacker, while essentially the HW or SW itself is functionally equivalent. So, from a functional point of view, it is still a copy, but rigid checks like hash-summing will no longer detect it as such.

7. Intellectual property rights on Hardware and Software

   Unlike for identity theft, modifying SW and HW, extracting or changing credentials, here the asset to be attacked is the license policy of the legitimate owner of the hardware of software. It is up to the owner's discretion to decide to whom he is willing to grant access to his HW or SW and under what conditions.

3.2.2 Attack scenarios

Focussing on ID cards there are – besides many well known attacks for ID cards in general – four relevant groups of attack scenarios for the HINT project context:

- Extraction of keys (or credentials).
- Manipulation of credentials
- Cloning of devices
- Introduction of Hardware Trojans

Basically the first two approaches are related on a logical abstraction level although their implementation is completely different.

There are invasive, semi-invasive and non-invasive known attacks. Non-invasive attacks like all kinds of variations of side channel analysis may be prevented by logical countermeasures.

1. Extraction of keys

   In today's implementations of ID cards there is a variety of keys on which security mechanisms are built up. Extracting such keys from ID card ICs may enable an attacker to construct an emulation device that could look completely different, but perform as a substitute of an ID card. Especially when using such an emulation device in an online scenario, where it is not needed to show the physical card comprising extra (non-electronic) security features that can be checked, the emulation may be implemented with very powerful devices, or even in software on a fully equipped personal computer.

   Extraction of confidential data may exceed key extraction, however. For instance in a health insurance card, such confidential data may be health records or data related to medication.

2. Manipulation of credentials

   ID card issuers associate a variety of credentials and attributes to the card-holders. It is the Card Issuer however, who is legally in charge of administrating these attributes on behalf of the authority behind him. For cards used in the health sector, the situation can become more complex even when the interplay of the interests of the health insurance company, the health professionals and the patient need to be balanced. Each party

(including the card-user) may have an interest to illegally manipulate such credentials in order to remove restrictions imposed to him or obtain privileges not granted to him.

Such manipulation may be achieved by an attacker through loading/changing unauthorized data (e.g. loading a manipulated health record without being authorized to do so). Usually, such loading operation requires an authentication check, which an attacker may be able to bypass.

Another attack path consists in direct physical change of such data stored in a chip card's memory cells, for instance by bit-flipping through fault attacks. Often, privileges are encoded in concise format, individual bits encode access rights etc. An attacker may be able to identify and change such configuration bits, eventually involving cancelation or manipulation of security checksums used to safeguard configuration data.

3. Cloning devices

A third attack scenario would be the physical cloning of devices. In case of a ROM-based device this would even include the software being executed on a secure micro-controller to be cloned, in case of NVM (EEPROM or Flash) based devices a copy of the memory image would have to be transferred in addition.

Usually reverse-engineering methods are needed to be applied to extract all physical available data necessary to implement a clone. In case of successful espionage layout data might also be copied directly, but to prevent this case strong organizational measures have to be undertaken.

Reverse-engineering is used

- for reading out complete images from an ID card (including confidential information and intellectual property like firmware design, operating system code, proprietary algorithms, also personal data) or
- for reading out dedicated data (e.g. including personal data and secret keys, practiced with illegally acquired and personalized sample cards

Reverse Engineering is part of an attack path which allows cloning a card. Cloning can be a threat to different assets:

- Cloning of ID Cards → Identity theft (relevant for users, public authorities, identity providers and service providers)
- Cloning of hardware → Intellectual property theft (relevant for IC manufacturers like Infineon)
- Cloning of software together with the hardware platform for usage of software not compliant with license agreement → Intellectual property theft (relevant for card manufacturers like Morpho)

4. Hardware Trojan Horse induction

Hardware Trojan horses are implemented by modification of hardware during production process in an untrustworthy environment. This is a threat to functional integrity for instance of an ID card or of a secure element. Trojan Horse could be used e.g.

- for Denial of Service attacks
- for getting access to sensitive data

There are other threats to ID document ICs, like e.g. theft of ID cards in un-personalised state, which could be abused as a perfect basis for clone-devices, but such scenarios are not in the immediate focus of HINT, since theft-prevention would have to be undertaken by organisational measures. Nevertheless could tracking of unique properties of every ID document chip allow for detection of stolen instances of genuine products.

3.2.3 Summary of security analysis of the 'ID card use case'

Assets	Threats	Risks	Security Requirements	Security Tools
Card holder's identity	- Card Cloning. - Key extraction by SCA or FA. - Reverse-engineering.	- User identity theft. - Impersonation.	- Unclonability of HW. - Unclonability of SW. - Resistance to SCA & FA. - Secure storage of card holder's data.	- Card holder's authentication. - Key generation using PUFs. - Countermeasures against SCA and FA.
Card holder's data & credentials	- Modification via HW attacks. - Modification via HW Trojans. - Modification via SW Trojans.	- Updating privileges. - Modification of 'credits'. - Modification of info like age etc.	- Countermeasures against invasive attacks. - Verification of HW integrity. - Verification of SW integrity.	- Sensors, shields, memory encryption, address scrambling… - SCA-based integrity analysis tools.
Card holder's privacy	- Eavesdropping of user transactions.	- User privacy violation.	- Anonymity for "privacy-critical" transactions.	- Anonymous attestation. - Group signature schemes.
Secret keys in cards	- Keys extracted by SCA or FA. - Keys extracted from corrupted personalisation facility.	- User identity theft. - Impersonation. - Updating privileges. - Modification of 'credits'. - Modification of info like age etc.	- Resistance to SCA & FA. - Secure storage & manipulation of secret keys. - Security of personalisation facilities.	- Key generation using PUFs. - Countermeasures against SCA and FA.
Card's HW IP	- HW cloning. - Physical Reverse-engineering. - Corrupted production facility.	- IP theft. - User identity theft.	- Circuit's fingerprint which cannot be read out. - Robust HW identification schemes.	- Sensors, shields, memory encryption, address scrambling… - CRP-based HW authentication using PUFs.
Card's SW IP	- SW cloning. - Physical Reverse-engineering. - Corrupted production facility. - HW Trojan.	- IP theft. - User identity theft. - Updating privileges. - Modification of 'credits'. - Modification of info like age etc.	- Methods for detection HW Trojans post-production. - Secure production facilities. - HW/SW binding.	- PUF-based software binding. - SW IP identification schemes.
HW integrity & authenticity	- HW Trojans. - Counterfeiting. - Corrupted production facility.	- Corrupting SW integrity of card (TC-like scheme). - Accessing 'protected' HW parts. - Outputting sensitive data. - User identity theft. - Denial of service. - De-activating countermeasures.	- Methods for detection HW Trojans post-production. - Secure production facilities.	- On-field HW integrity & authenticity verification schemes.
SW integrity & authenticity	- SW Trojans. - Counterfeiting. - Corrupted production facility.	- Outputting sensitive data. - User identity theft. - Updating privileges. - Modification of 'credits'. - Modification of info like age etc. - Denial of service. - Deactivating SW or HW security mechanisms.	- Methods for detecting SW Trojans in production and post-production. - Methods for identifying a SW. - HW/SW binding.	- On-the-field SW integrity & authenticity verification. - PUF-based software binding.

Defintions:
***Asset**: entity (object, data, person) that has a valuable importance.*
***Threat**: Mechanism or procedure that may harm the given asset(s).*
***Risk**: The resulting harm that may be caused to a given asset by a given threat.*
***Security Requirements**: What is needed to mitigate or eliminate the risk associated to a given threat for a given asset.*
***Security Tools**: How (what tools) that may be implemented to mitigate or eliminate the risk associated to a given threat for a given asset.*

Table 1: Security Analysis Summary of the "ID Card Use Case"

3.3 Security Analysis of the "PMR use case"

The following security analysis is presented in a style of a security profile to achieve the full coverage of this use case and to prepare an actual profile.

3.3.1 Assets

The PMR has, by definition, the dual task of managing sensitive information that is useful for its user or its operations and of providing critical service.

A mobile device such as the handheld and even a vehicular router are subjected to theft, borrowing, exchanges, etc. So, various data might be potentially exposed. Here a short list of some of the most common types of sensitive information that might be of interest inside those devices is detailed:

- **Identity**: Any Identity Information that can be used to steal a user's identity.
- **Keys, passwords, and PINs**: Information used by the user or the device to protect local data.
- **User data**: Any valuable Information that the device stores.
- **Intellectual Property (IP)**: Information on which an organization depends for competitive advantage.

A mobile device to operate correctly shall have a **hardware part** and **software** compliant with its usage and shall **perform** according to its specifications.

3.3.2 Threats

This section describes the threats to be averted by the PMR products independently of, or in collaboration with, its IT environment. These threats result from the PMR products method of use in the operational environment and the assets stored in or protected by the PMR products.

In the following, all the threats will be denoted "T.X" where "X" is the threat.

T.Denial-of-Service

A first need for the PMR systems is to have a near-to-perfect reliability and availability, including the radio coverage. An attacker can then want to inject HTs for provoking a **Denial-of-Service** (**DoS**) attack. The most interesting point to attack in a PMR terminal is the ASIC which computes cryptographic operations. This "cryptoASIC" contains a proprietary processor, in which a 2-cluster 32-bit processor can communicate with hardware cryptoprocessors (hash functions, random number generators, *etc.*). So, in order to provoke a DoS (which is very critical for both military and fireman use cases since it can provoke deaths in certain conditions), an attacker can introduce a HT in the cryptoASIC that for example:

- freezes the communication buses between the processor and the cryptoprocessors,
- modifies the finite state machine of a sub-component to go to a hidden state that does infinite loop of NOP operations,
- enables access for a hidden code that shuts off all or part of the chip (kill switch).

These 3 examples of attacks will force the owner of an infected terminal to reset it, and then 2 possibilities can occur:

- the effect of the HT was temporary and so the terminal works normally (until the next payload of the Trojan),
- the effect of the HT is permanent and so the terminal doesn't work anymore.

T.Information_Leakage

Second, PMR systems also bring security features to ensure authentication of the users and confidentiality of the communications. PMR systems produce:

- Mutual authentication between the terminal and the network. It stops "illegal" terminals from registering with the network and enables terminals to ensure they are not signing onto a fake network.
- End-to-end encryption. Voice is encrypted in the transmitting terminal and decrypted in the receiving terminal. So, eavesdropping or tapping is impossible over the air or in the transmission network. The transmission network does not require extra security to prevent tapping.
- Over-The-Air Re-keying (OTAR). It allows encryption keys to be changed remotely (frequent updating enhances security). Clustering between several organizations (different keys and user groups) is also possible.
- Lost or stolen terminals can be disabled remotely.

So, a second way to attack a PMR terminal is to provoke the **leakage of sensitive information** (like cryptographic keys). In this manner, eavesdropping of the communications and then access to sensitive information (data related to the mission) becomes possible. It is more critical for military use case than the firemen since in most cases team members can exchange critical data on the PMR network, and the disclosure of this data can bring security problems if the mission is secret. It can also allow authenticating on PMR networks. There exists many ways to do so:

- A direct leakage: the terminal sends to its I/O, or over the air, the communication key(s). It can also fake crypto-algorithms computations, but directly sent plaintexts in clear (the crypto-algorithm is then skipped).
- An indirect leakage: An attacker may exploit information which is leaked from the PMR terminal during its usage in order to disclose the cryptographic confidential key. The information leakage caused by the inserted HT can be temporary or permanent. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. This leakage may be interpreted as a covert channel transmission but is more closely related to measurement of operating parameters, which may be derived either from measurements of the interface (emanation) or direct measurements (by contact to the chip still available even for a contactless chip) and can then be related to the specific operation being performed. Examples are the Differential Electromagnetic Analysis (DEMA) and the Differential Power Analysis (DPA). Moreover the attacker may try actively to enforce information leakage by fault injection (e.g. Differential Fault Analysis): a HT, by inducing faulty ciphertexts (if the faults are correctly inserted during the cryptographic computation), can allow the mathematical analysis between correct and faulty ciphertexts, which can lead to the key. The state-of-the-art indicates that only one fault is sufficient to break an implementation of AES or the RSA-CRT. It must be also noticed that side-channel or fault attacks can also allow an attacker to leak the secret keys but also the used secret algorithms thanks to a Fault Injection Reverse-Engineering (FIRE) and a Side-Channel Analysis Reverse-Engineering (SCARE).

These kinds of leakage can be easily provoked by modifying the cryptoASIC. For example, at each new key received from an OTAR procedure, the key can be directly sent back on the I/O bidirectional bus if a Hardware Trojan placed on the bus detects the key loading.

Implementing an indirect leakage is not more complicated. For example, [REF 31] proposes to build a TSC to emit permanently the key through CDMA (Code-Division Multiple Access) encoding. For a fault attack provoked by a Hardware Trojan, we can imagine a HT which injects at a rare condition (series of plaintexts with some properties) or at a certain time (e.g. after 1 million successfully encryptions) a fault. This kind of HT adds only several gates in the original circuit and then seems at first sight very difficult to detect thanks to a side-channel analysis.

T. Malfunction

A HT can also provoke the circuit to malfunction. It is not exactly the same threat as “T.Denial-Of-Service” since in this case, the component doesn’t work anymore. In the threat “T.Malfunction”, the circuit works, but badly.

More precisely, we can imagine such kind of related payloads:

- At a certain moment in time, the HT modifies internal nodes or memory contents of the circuit which not provoke information leakage but simply inserts functional errors in the system. In some cases, a reset can allow to restore the initial behaviour of the circuit (the effect of the HT is then temporary), sometimes not (the effect of the HT is then permanent).
- At a certain moment in time, the HT prevents the terminal to enter into an energy saving mode (or “sleep mode”). It then decreases the autonomy of the PMR terminal, and it will become impossible to use such an infected one because it has to be loaded say every 3 hours to a loading battery station (instead of every 24 hours). The user of the terminal will then blame it on the battery, and will not think about an inserted HT in the terminal.
- On the board, an attacker can also nick the wires linking for example the CPLDs and the cryptoASIC. A notch in a few interconnects would be almost impossible to detect but would cause eventual mechanical failure as the wires became overloaded. Another concern is that this notch can induce sparks that can provoke explosion if the fireman is in a chemical environment. These sparks appear with the time and then are not detected in test-time (before deployment) and in ATEX certification process.

T.Phys-Tamper

An attacker can also insert a HT in a circuit for modifying security features of the PMR.

One example can be to enable information leakage through a side channel. To do so, we can imagine that a HT provokes the fact that a Random Number Generator (RNG) outputs always a zero-value, which will contribute to lower security mechanisms which use random data like key generation, masking side-channel protected implementations of cryptographic algorithms or nonces for authentication. Usually, RNGs are associated with an on-line tester to test the quality of the generated random values (e.g., which implements FIPS 140-2 or AIS-31 tests), so at first sight we can think that if the RNG always outputs zero values, the tester will detect it. But in the case of Hardware Trojans, where we have very powerful attackers, we can imagine that this latter will also modify the tester in order to bypass it.

Another example of security features modification of the PMR can be the non-modification of a cryptographic key, even if a new one is sent to the terminal and/or if the key has expired.

3.3.3 Organisational Security Policies

In the following, all the relevant organisational security policies will be denoted “P.X” where “P” is the policy.

P.Safe_Programming

FPGAs contained in PMR products are internally programmed and are fused after programmation. Moreover, the cryptoASIC is also internally programmed.

P.Safe_Personalization

Personalization (implementation of customer’s algorithms) is also performed internally.

P.Sensitive_Data_Protection

All the sensitive data are at least protected in integrity. The keys are protected in both integrity and confidentiality.

P.Key_Function

All the cryptographic routines are designed in such a way that they are protected against probing and do not cause any information leakage that may be used by an attacker.

3.3.4 Assumptions

The assumptions describe the security aspects of the environment in which the PMR products will be used or is intended to be used.

In the following, all the assumptions we make will be denoted “A.X” where “A” is the assumption.

A.PMR_Manufact

It is assumed that appropriate functionality testing of the PMR products (and all the components implemented in it) is used. It is assumed that security procedures are used during all manufacturing and test operations to maintain confidentiality and integrity of the sensitive data contained in PMR products and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

A.PMR_Delivery

Procedures shall guarantee the control of the PMR products delivery and storage process and conformance to its objectives:

- Procedures shall ensure protection of PMR products material/information under delivery and storage.
- Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and storage.
- Procedures shall ensure that people dealing with the procedure for delivery have got the required skill.

A.Pers_Agent

Only the Personalization Agent knows the correct confidentiality and integrity keys used to program cryptoASIC, and the initial keys used by customer's algorithms.

3.3.5 Security Objectives

3.3.5.1 Security Objectives for the PMR products

This section describes the security objectives for the PMR products addressing the aspects of identified threats to be countered by the PMR products and organizational security policies to be met by the PMR products.

In the following, all the security objectives for the PMR products will be denoted "OT.X" where "X" is the objective.

OT.AC_Pers

PMR products must ensure that the software and the personalisation data can be written by authorized Personalization Agents only.

OT.Data_Int

PMR products must ensure the integrity of the software, firmware and the keys stored in the cryptoASIC or CPLD/FPGA against physical manipulation and unauthorized writing.

OT.Data_Conf

PMR products must ensure the confidentiality of the programs and the keys stored in the cryptoASIC.

OT.Prot_Abuse_Func

After being delivered to their owners, PMR products must prevent the abuse of test and support functions that may be maliciously used to (i) disclose critical user data, (ii) manipulate critical user data of the embedded software, (iii) manipulate the software itself or (iv) bypass, deactivate, change or explore security features or functions of the PMR products.

OT.Prot_Inf_Leak

PMR products must provide protection against disclosure of confidential data stored and/or processed in the cryptoASIC

- by measurement and analysis of the shape and amplitude of signals or the time between events found by measuring signals on the electromagnetic field, power consumption, clock, or I/O lines and
- by forcing a malfunction of the PMR product and/or
- by a physical manipulation of the PMR product.

OT.Prot_Phys-Tamper

PMR products must provide the confidentiality and integrity of the user data and the customer's algorithms. This includes protection against attacks with high attack potential by means of:

- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or

- measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis)
- manipulation of the hardware and its security features, as well as
- controlled manipulation of memory contents with a prior reverse-engineering to understand the design and its properties and functions.

OT.Prot_Malfunction
PMR products must ensure their correct operation. PMR must prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent errors. The environmental conditions may include external energy (esp. electromagnetic) fields, voltage (on any contacts), clock frequency, temperature, humidity, chemical and explosive atmospheres.

OT.Prot_Availability

The PMR system should be as resilient and available as possible:

- The system shall be able to monitor permanently any area of interest 24/7 with an availability level of 99.999 %.
- The system should operate at any weather or environmental conditions. Temperature or humidity range expected for emergencies situation, like rainfalls or fires.
- The system should continue working even if some components do not work.
- The system shall be able to function in a fall-back mode even if the infrastructure communication network is partially or totally out of order.
- All hardware and software failure should be reported.

OT.Hardware_Integrity
PMR products must provide protection against hardware integrity attacks.

3.3.5.2 Security Objectives for the Operational Environment

PMR terminal Manufacturer implements the following security objectives for the TOE environment.

OE.PMR_Manufact

Appropriate functionality testing of the PMR systems shall be used by the “main manufacturing sub-contractor” (“final test” in figure 5) and by Cassidian (“labelling and test” in figure 5). During all manufacturing and test operations, security procedures shall be used through all the PMR life cycle (see figure 5) to maintain confidentiality and integrity of the PMR products and its manufacturing and test data.

OE.PMR_Delivery
Procedures shall ensure protection of PMR products material/information during delivery including the following objectives:

- non-disclosure of any security relevant information,
- identification of the element under delivery,
- meet confidentiality rules (confidentiality level, transmittal form, reception acknowledgment),
- physical protection to prevent external damage,
- secure storage and handling procedures (including rejected PMR products),
- traceability of PMR products during delivery including the following parameters:
    - origin and shipment details,

- reception, reception acknowledgement,
- equipment location.

Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process (including if applicable any non-conformance to the confidentiality convention) and highlight all non-conformance to this process.
Procedures shall ensure that people (shipping department, carrier, reception department) dealing with the procedure for delivery have got the required skill, training and knowledge to meet the procedure requirements and be able to act fully in accordance with the above expectations.

OE.Personalization
PMR terminal Manufacturer must ensure that the Personalization Agents acting on behalf of company personalize the PMR products for the customers together with the defined physical and logical security measures to protect the confidentiality and integrity of customer's data (secret algorithms and keys).

3.3.6 Attack scenarios

The issue on hardware integrity is a very general and important one, as it is a prerequisite for the secure and trusted execution environment needed by various applications running on the device after its power on. This power on triggers first the hardware and then the boot process and then the various middleware components and applications, so the most critical threats stem from the hardware.

We must prevent an attacker from compromising the device's integrity. This devices' integrity can be subverted in many ways, the most obvious ones are:

Hardware Trojan Horse through modification of hardware during production process in mistrusted environment:

1. Threat to the functional integrity, for instance, of a secure element. Trojan Horses could be used e.g.
   - for Denial of Service attacks
   - for getting access to sensitive data
2. Threat to the quality and performance of the device. Trojan Horses could be used e.g.
   - for Denial of Service attacks
   - for defecting a device
3. Threat to information within the device. Trojan Horses could be used e.g.
   - to steal identity
   - to steal secrets
   - to steal user data
4. Counterfeited COTS IC

   The threat is that the use of a clone of an IC will lead to a degradation of the quality and performance of the product.

3.3.7 Summary of security analysis of the 'PMR use case'

Assets	Threats	Risks	Security Requirements	Security Tools
Device capability (quality, availability, performance, etc.)	- T.Denial-Of-Service - T.Malfunction	- Lifes at stake. - Quality of communications so bad that orders are misunderstood. - Functional errors in the system. - Regular resets and boots. - Limited autonomy. - Sparks and then explosion in a chemical context.	- Post-Production HT Detection Methods.	- On-the-field HW integrity & authenticity verification schemes. - Redundancy of the components (majority voting).
User's identity	- T.Information_Leakage	- User identity theft. - Impersonation.	- Unclonability of HW. - Unclonability of SW. - Resistance to physical attacks. - Secure storage of PMR handheld holder's data.	- PMR handheld holder's authentication. - Key generation using PUFs. - Countermeasures against SCA and FA.
Device's or User's data for protecting local data	- T.Information_Leakage - T.Phys-Tamper	- Eavesdropping of communications. - Authentication on the network by impersonation. - A fraudulent terminal can then be present on a network for a long time and then drag useful discussions. - Passwords, PINs, etc. theft. - User privacy violation. - Privilege escalation (a user that was only able to listen/talk to a certain group can then listen/talk to others).	- Resistance to physical attacks. - Secure storage & manipulation of secret keys. - Security of programming/personalisation facilities.	- As soon as a fraudulent PMR terminal is detected, disable it remotely. - Short cryptoperiods for cryptographic keys, PINs, etc. - Group cryptography.
IP rights on HW, (especially the CryptoASIC)	- T.Information_Leakage	- IP theft. - Programming keys theft. - Attacks become easier for certification labs if they got the information. - Cloning.	- Circuit's fingerprint which cannot be read out. - Robust HW identification schemes.	- Sensors, shields, memory encryption, address scrambling… - CRP-based HW authentication using PUFs.
IP rights on SW	- T.Information_Leakage	- IP theft. - Secret algorithms leakage. - Theft of hidden side-channel countermeasures.	- Organisational security policies. - HW/SW binding.	- Programming in a secure environment. - PUF-based software binding. - SW IP identification schemes.
HW integrity & authenticity	- T.Denial-Of-Service - T.Malfunction - T.Information_Leakage - T.Phys-Tamper	- Deaths. - Functional errors. - Impersonation. - Eavesdropping of communications.	- Methods for detection HW Trojans post-production. - Secure production facilities.	- On-field HW integrity & authenticity verification schemes.
SW integrity & authenticity	- T.Denial-Of-Service - T.Malfunction - T.Information_Leakage - T.Phys-Tamper	- Deaths. - Impersonation. - Deactivating SW or HW security mechanisms. - Replace a side-channel protected implementation of a cryptographic algorithm by a non-protected one.	- Organisational security policies. - Methods for identifying a SW. - HW/SW binding.	- On-the-field SW integrity & authenticity verification. - PUF-based software binding.

Defintions:
***Asset**: entity (object, data, person) that has a valuable importance.*
***Threat**: Mechanism or procedure that may harm the given asset(s).*
***Risk**: The resulting harm that may be caused to a given asset by a given threat.*
***Security Requirements**: What is needed to mitigate or eliminate the risk associated to a given threat for a given asset.*
***Security Tools**: How (what tools) that may be implemented to mitigate or eliminate the risk associated to a given threat for a given asset.*

Table 2: Security Analysis Summary of the "PMR Use Case"

Chapter 4 Stakeholder analysis & requirements

For each of the applications' use-cases described in Chapter 3, there are several parties/users/entities with different levels of implications and vested interests in the scenarios. In this chapter, we focus on those 'stakeholders' and derive the project's requirements based on their expectations. For the sake of this analysis, we have not been able to meet each possible stakeholder but the industrial partners of the project, through their strong presence and mastery of their respective businesses, have shared in this report their experience based on real-life situations.

Those stakeholder requirements, augmented by the security requirements derived from the previous chapter are then summarized into what we shall refer to as being the HINT requirements.

4.1 Stakeholder Analysis

At this stage of the project we identified the following preliminary group of HINT stakeholders:

- Semiconductor manufacturers
- Operating system provider
- Product vendors / Card Issuer (System Operator / System-Integrator)
- End user (Card Holder or PMR product user)

It may be concluded, that integrity of system components is not only the interest of some singular entities but rather that the integrity is vital for all players along the value chain.

4.1.1 Requirements as seen from the semiconductor manufacturers side

Semiconductor manufacturers like Infineon Technologies have increasing demand on assurance of the integrity of the manufactured hardware An important requirement is that all the pieces ordered to be manufactured shall have been produced, but not more than those. Pieces sorted out due to some malfunctions or especially those that are mostly well-functioning but only slightly out of some few specification limits must be prevented from being brought into the market through secondary sales channels. In the environment of highly secure chipcard semiconductor production nowadays these requirements are fulfilled by organizational matters, like separated production lines with dedicated access restrictions and specially trained personnel. Special certification procedures try to document the effectiveness and assurance of these measures. For less security relevant products, tracing through manufacturing processes might become valuable in the future.

For highly secure products tracking of silicon dies throughout the complete value chain may be required to be able to observe leakage through later steps in the manufacturing process, like e.g. packaging or testing. As long as several thousand dies are still mechanically tied together because they are residing on the same wafer, tracking is relatively easy. But after sawing, the number of pieces that have to be tracked is roughly three to four orders of magnitude higher than before. When these chips are further processed, tracking by chip-individual properties might be very useful. Latest when storing chip-individual data in the test flow – like production relevant data for FAR tracking or unique serial numbers or keys – a binding of such properties to individual characteristics of the silicon could be needed.

4.1.2 Requirements necessary for Operating System providers

Smart Card vendors like Morpho first of all demand assurance of the integrity of the hardware platform supplied by semiconductor manufacturers as the basis for smart card products. Any manipulation of the platform e.g. by hardware Trojans must be prevented. Moreover, it shall be possible to guarantee authenticity of an entire Smart Card product, including the individual IC, but also the operating system and application software provided by the Smart Card producer. The general objective is to deliver secure products that effectively thwart impersonation of an attacker under a false identity or any other attack described in the previous section.

With regard to IP of the semiconductor products, embedded software may assist in detecting forged (unlicensed) semiconductors or even deny operation on unlicensed ICs.

From an operating system provider's point of view, it would also be interesting to analyze whether the IPR on its software can be protected by PUF technology, for instance through binding it to reliable hardware with individual "brandings".

4.1.3 Product vendors / card issuer

The Card Issuer in almost all cases is and remains the legal owner of ID cards issued to its end-users. Typical examples for Card Issuers are governmental authorities, health insurance companies, banks, etc. Responsibilities and rights for ID-cards (and their credentials) usually make up a complex system.

The product vendor or the Card Issuer is responsible for *system integration* and *system operation*; either directly (systems are operated under his control) or indirectly by prescribing statutory requirements and orders for the operation. He is also responsible for equipment or card personalization, which is often physically done at the manufacturer's site, however.

On the one hand, the Card Issuer needs assurance that the ID cards issued by him are used only in accordance with the terms and conditions agreed upon (or prescribed by legislation). In particular, there may be a number of restrictions imposed which need to be enforced.

On the other hand, there are credentials belonging to the card holder which even the card issuer has no right to know or access. (To mention one, the card issuer need not be entitled to know a signature key of an end-user, in spite of the card issuer owning the signature card.)

The PMR vendor manufactures and then delivers the product to a customer or user organisation that owns this product.

In the following, we discuss the requirements by product vendors or card issuers respectively regarding the personalization of semiconductor based products.

Personalization of banking cards is done by specialized IT departments of major banks and personalization of eID cards is performed by governmental administrative institutions. Both these types of institutions have usually set up a secured environment for this step. They require that they can track the completeness of a delivery charge and to map an individual card characterized by its individual embedded chip to an individual end user / customer. Usually this has been done by adding a unique asymmetric cryptographic key pair to the device, which in trusted computing related applications is called endorsement key. To avoid duplication of such keys the personalizing entities require an option to connect the unique key pair to a unique property of the piece of silicon on which the key pair is stored, and which cannot be cloned. A cryptographically hard connection of key and product allows for distinction and thus detection of cloned samples of the personalized product.

In their role as system integrators, product vendors and card issuers have to ensure that the right components in the correct configurations and with the intended security parameters are integrated. This is especially challenging in so called systems-of-systems, where sub-

components consist of complex security relevant combinations of hardware, firmware and software.

4.1.4 Requirements from the end user's view

In case of ID cards, these are issued over secure channels to their end-users, and come personalized by the (or in charge of the) issuing authority. Please note that for practically all ID cards, the end-user (card holder) is *not* the legal owner. His or her ID card remains the legal property of the issuing authority (like a state, a health insurance, a bank, etc.).

The user of a PMR device has the same requirement of usage of any communication device, but in the more stringent conditions he expects his device in general to show reliability and trust.

- So the end–user requirement is to have the device to provide the service it was intended to provide.
- On the need for trust the end-user also must be sure that the device does *only* what it was expected to do.

Chapter 5 HINT Use Case Requirements

This Chapter describes the main requirements for HINT applications, summarizing the security requirements identified for both targeted application scenarios in Chapter 3 as well as the stakeholder requirements identified in Chapter 4.

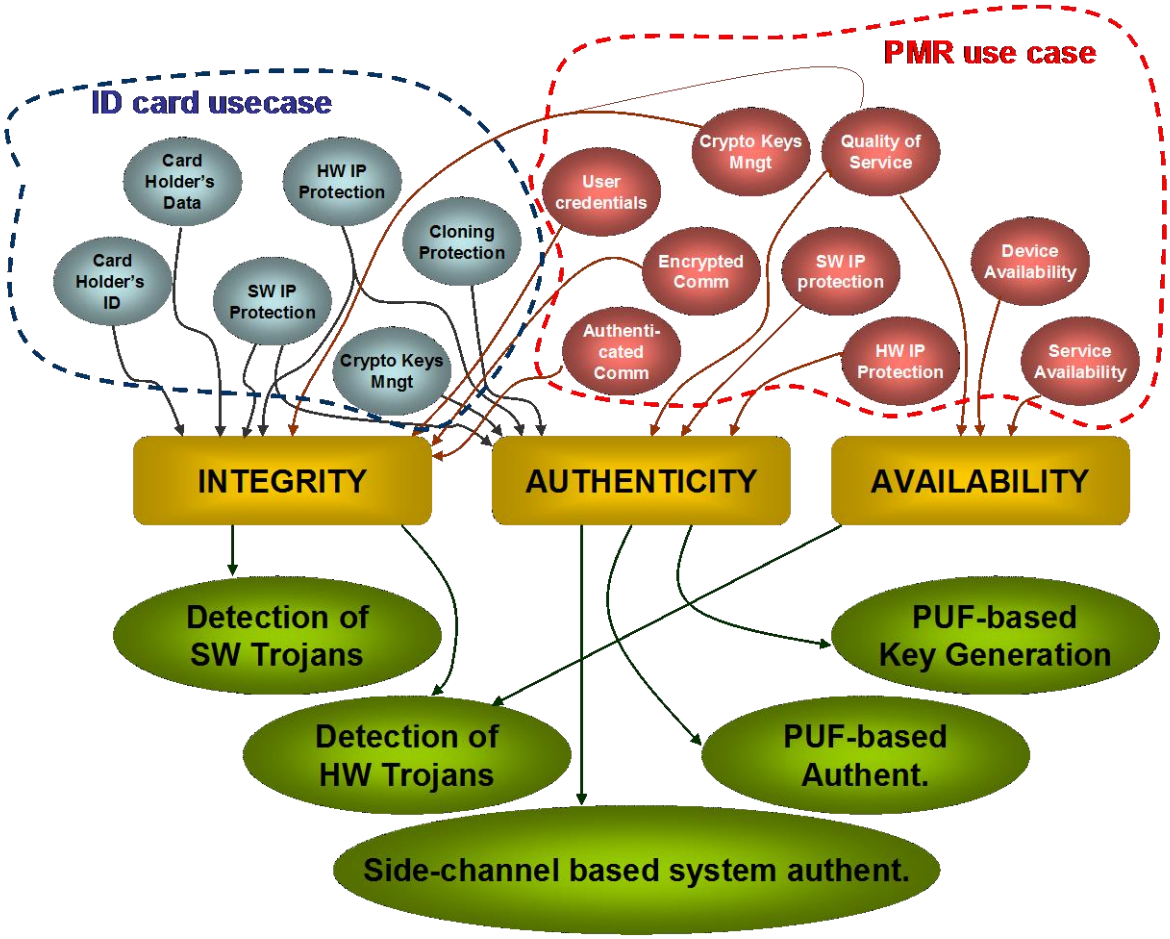

Figure 7: Summary of HINT requirements

Figure 7 gives an overview of the requirements and categorizes them with regard to the three overarching objectives pursued with HINT technologies:

- Integrity
- Authenticity
- Availability

The following requirements have been identified as essential and must be covered by HINT technologies.

- Protection of user identity and credentials

  In the ID card use case, the identity of the user and the corresponding credentials obviously belong to the main assets that are to be protected in the card. Credentials must not be copied, stolen or duplicated/cloned since they are representing special roles and associated rights of the ID document holder. Also in the PMR use case, protection of user identity and credentials is important in order to prevent illegal access to the system. This requirement also applies to credentials & privileges that are associated to but not owned by the user, in the ID card use case for instance to user data owned by the card issuer.

- Hardware IP Protection

  In both application scenarios, HINT technology shall be used to protect IPRs of the device manufacturers including semiconductor manufacturers, ID card providers and PMR device manufacturers. IPR protection includes on the one hand detection of counterfeits. On the other hand, leakage of information about the hard-, firm- and software of the device must be prevented.

- Software IP Protection

  In both application scenarios, methods for detecting hardware counterfeits may also be used to prevent the illegal use of software (like an ID card operating system) on hardware platforms not covered by license agreements.

- Secure cryptographic operations and cryptographic key management

  Cryptographic operations for e.g. authentication, verification of signatures and encrypted communication are core functions of ID cards and likewise essential for the operation of PMR devices. These operations and the underlying cryptographic keys must be protected against attacks.

- Device and service availability

  PMR devices are used in situations where “life is at stake”. Therefore, an availability near to 100% is the overarching requirement for these devices, whereas for ID cards, availability is less important.

- Hardware integrity and authenticity

  These requirements are the basis for most of the aforementioned requirements and hence are keys for all HINT applications.

Chapter 6 HINT Building Technologies

The requirements presented in section Chapter 5 have allowed us to identify several technologies that shall constitute the pillars of the scheme developed in the HINT project. Even though the specification of this scheme shall be detailed in deliverable D1.2, we here provide a brief overview of the technical domains that are going to be investigated in the HINT project: each of those domains has already yielded an extensive body of knowledge which shall provide valuable starting material to meet the complex objectives of the HINT project.

6.1 SCA-based integrity verification & authentication schemes

6.1.1 State-of-the-Art

Side Channel Analysis (SCA) has been an intensively active field of study in the security engineering arena for over a decade now. Since the first publication on the subject by Kocher [REF 1] in 1996, techniques exploiting the power consumption of a given device to extract information about the data manipulated by the given device (for example the cryptographic keys of algorithms like DES, AES, RSA or ECC) or about the processes being executed by the latter have been continuously evolving through different approaches such as Simple Power Analysis (SPA), Differential Power Analysis (DPA), Higher Order DPA (HoDPA) or Mutual Information Analysis (MIA). Such techniques have also been successfully used based on the Electromagnetic (EM) emissions of the devices under test starting with smart-cards [REF 2] to Personal Digital Assistants (PDAs) [REF 3] and more recently Java mobile phones [REF 6]. Given that EM Analysis (EMA) allows more localized and precise measurements, this field has given rise to numerous research works on the experimental set-ups and exploitations [REF 18].

In addition to the initial aim of side channel analysis for secret cryptographic key extractions, recent researches have highlighted other uses of side channels:

- *Side channels for reverse engineering*: Side Channel Analysis for Reverse Engineering (SCARE) is a technique that has been proposed for reverse-engineering “secret” cryptographic algorithms [REF 4] or for the detailed identification of building blocks and execution sequence of a general purpose processor (with or without golden model) [REF 5], [REF 17].
- *Side channels as a watermarking technique*: For managing the authenticity aspects of ICs, in addition to the implementation of strong intrinsic mechanisms based on PUFs (see section 6.2), watermarking has for long been proposed as an interesting approach. One approach can be to deliberately generate a given power signal that can be unique and that can be used to authenticate a given IC, whether in hardware [REF 13] or in software [REF 14].
- *Side channels for detecting circuit modifications*: Several schemes have been proposed for detecting the presence of hardware Trojans (and hence the integrity of a given IC): by measuring the steady state current of a given IC [REF 9]; by measuring the transient current [REF 12]; by measuring the EM radiation emitted [REF 10]; or by determining a characteristic radio frequency signature of a given circuit based on a series of pre-determined sequence of executions [REF 15]. Such techniques have been enhanced through for example the use of ‘self-referencing’ techniques where the measured on chip transient currents are used to reduce the effect of noise due to process variations [REF 11].

- *Side channels for "active checking"*: In the case where in the HINT project some form of "active checking" is sought, using side-channels to convey characteristic circuit-related information might be a path to investigate: for example in [REF 7] the authors hide sensitive information through an intentionally generated side channel, while in [REF 8] the authors propose micro-architectural modifications on general purpose processors for enhancing the leakage of cryptographic keys used by software cryptographic implementations.

6.1.2 HINT Architecture Requirements

In the scope of the HINT project, we shall investigate about the use of side channels, in particular based on EM, to address the (security) requirements summarized in section Chapter 5. To achieve such goals, our research work shall use as starting points the existing state-of-the-art side-channel analysis techniques used in other related fields as described above.

To address the issue of a **system's availability**, side channel techniques used to detect Hardware Trojans shall be further investigated to increase the reliability of detection techniques and to adapt them to the specificities of Hardware Trojans that disrupt a system's availability.

To the issue of availability is also linked the issue of a **system's integrity** which shall be tackled also through Trojan detection techniques, this time both at hardware or software (or a combination of both) levels. In this field, side channel techniques used for reverse engineering (SCARE) could also be a field to investigate for, say, the extraction of the relevant information for verifying the origin or integrity of a targeted system.

A **system's authenticity** (to measure the individual 'identity' of a system, its origin, and at the same time provide some level of trust) might be measured using side channels using either active checking techniques or watermarking ones or even through reverse-engineering.

6.2 PUF-based authentication schemes

6.2.1 State-of-the-Art

A Physically Unclonable Function (PUF) is basically an expression of an inherent instance-specific property of a physical object that is impossible to duplicate [REF 37]. PUFs can be manufactured in a variety of technologies: optical, acoustical, magnetic, electrical, etc. Most relevant for commercial applications are PUFs that can be integrated on chip, particularly in CMOS technology.

PUFs are physical functions and produce a response when queried with a challenge. Responses and challenges are both binary vectors at the highest abstraction level. PUFs are often subdivided in two classes, depending on the number of challenge-response pairs (CRPs). Weak PUFs have few CRPs and are typically used for on-the-fly secret key generation (inherent key storage), as shown below. Their responses are post-processed by so-called fuzzy extractors to ensure that the secret key is reproducible and has a uniform distribution. First, an error correcting code is applied as response bits tend to be noisy. Subsequently a hash function is applied to remove response correlation patterns. PUF challenges are optional; a few of them might be present to select one out-of multiple keys. Weak PUFs can potentially be applied in any cryptographic protocol or algorithm that relies on (secret) keys). An example of a weak PUF structure is an SRAM PUF.

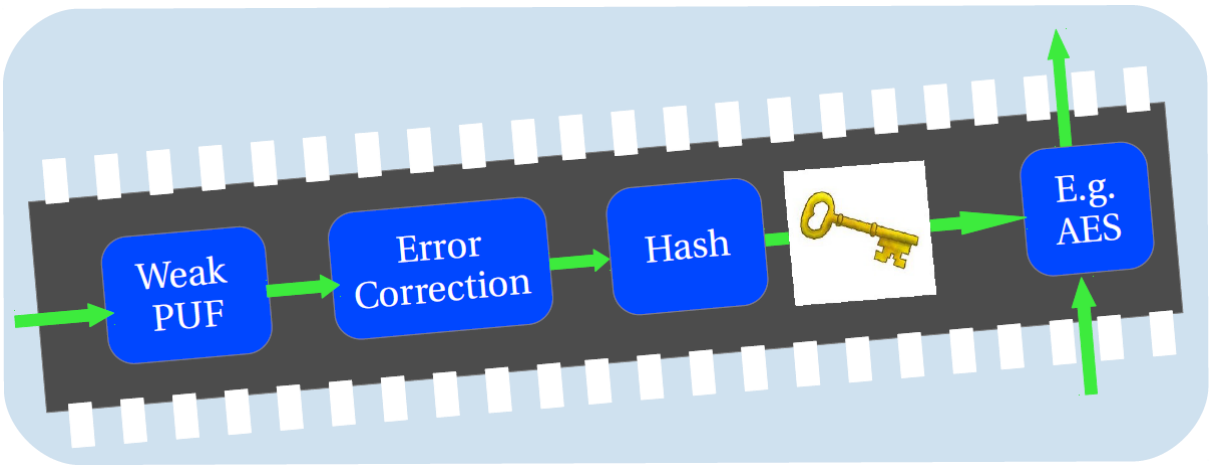

Figure 8: Weak PUF

In contrast to weak PUFs, strong PUFs have many CRPs (e.g., 128 bit challenges). Theoretically, this number increases exponentially with the required chip area. Therefore, it is infeasible to capture all the PUFs CRPs in a reasonable time span. Strong PUFs are a subclass of weak PUFs: every strong PUF can be employed as a weak PUF, but not vice versa. Therefore, they can be used in more applications, the most prominent of them being chip authentication whereby only the verifier has to store secret information. In an enrollment phase, the verifier collects arbitrary CRPs from the chip and stores them securely. In the verification phase, the verifier picks a challenge and requests the PUF response again. The returned response should match the one in the database. A few erroneous bits are typically tolerated to avoid the fuzzy extractor hardware overhead. An example of a strong PUF structure is an Arbiter PUF.

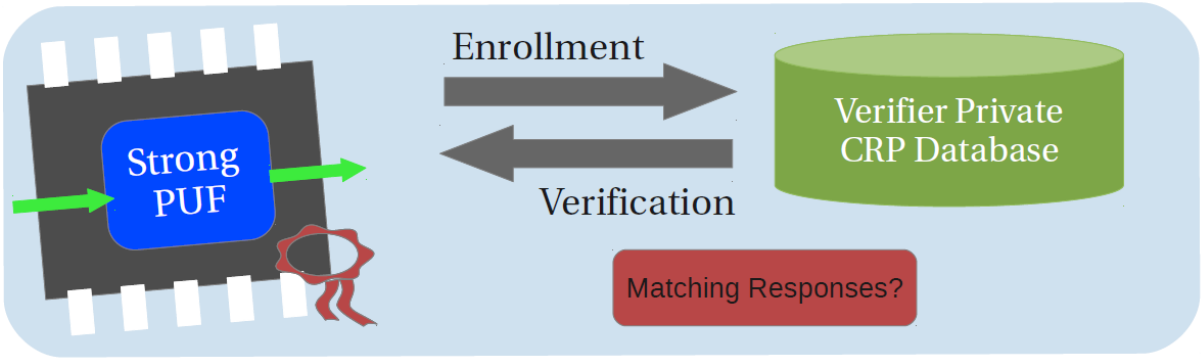

Figure 9: Strong PUF

Weak and strong PUFs have different security requirements. For weak PUFs, it is imperative to keep the responses on chip, as they are post-processed to secret keys. By obtaining the response bits, an attacker can create a clone of the PUF. Hardware attacks (invasive, through side channels and via fault injection) are the main threat. PUFs are often assumed to be resistant against the first category. One can argue that invasion damages the physical structure and hence also the PUF. However, experimental evidence is generally lacking, except for the coating PUF. Furthermore, more research on side channels and fault injection attacks on PUFs is needed. Note that next to the PUF itself, hardware attacks on the fuzzy extractor are a major threat as well.

For strong PUFs, CRPs can be obtained by anyone. The security arises from the unpredictability of the CRPs. It should be infeasible to construct a clone via a mathematical model. Modeling through Machine Learning algorithms, given a training set of CRPs, is a major threat. Hardware attacks on strong PUFs should be considered too, as they can facilitate modeling. More research is needed here as well.

Strong PUFs can be used in more applications, but it is very hard to make them secure and efficient. To enhance strong PUF security, a few schemes have been proposed. The controlled PUF, shown below, is well-known. First, a one-way hash function is applied to the user challenge to avoid chosen-challenge attacks. Second, a fuzzy extractor post-processes the responses to mask/decorrelate them. Note that one should take into account that hardware attacks carried out on these additional building blocks (i.e. bypassing them) could pose a potential threat. Weak PUFs do not require this challenge logic overhead.

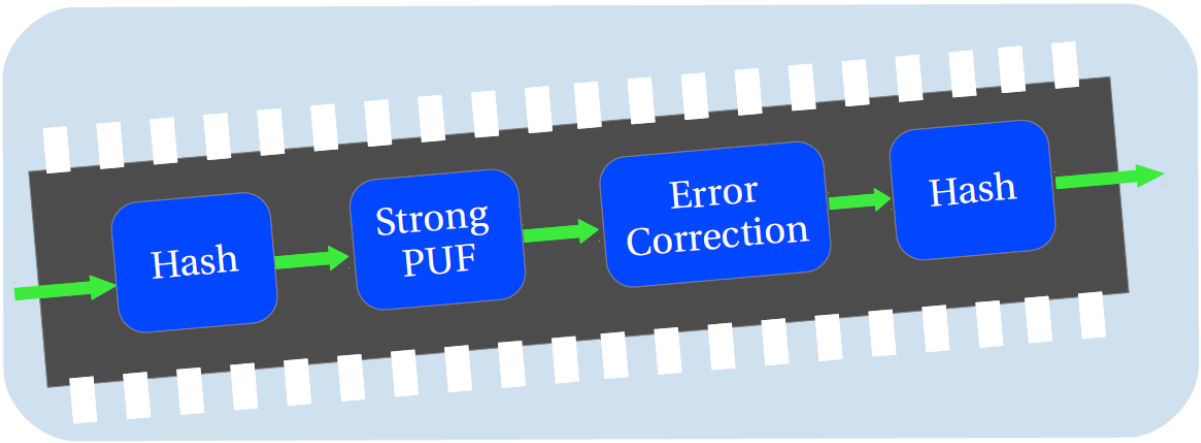

Figure 10: Controlled PUF

The required number of challenge and response bits, which decides if a weak or a strong PUF needs to be used, is application dependent and needs to be specified. However, there are other security requirements, common for weak and strong PUFs, which need to be specified too:

- The PUF should supply enough “entropy” to allow encapsulation of a sufficiently long memory encryption key. Therefore, a sufficient level of manufacturing variability should be harvested. This also means that the key should be retrieved *precisely* (PUF with error-correction) in order to allow subsequent usage for memory decryption.
- The PUF should allow fast enough key recovery for practical applications. Key recovery is not necessary for each encryption block, but at least for each initialization, since the intention is to never store the key in non-volatile memory.
- Short-term reliability (noise + environment robustness). There is typically a trade-off with the error correcting code. The more reliable the PUF is, the smaller the Error Correcting Code hardware footprint, but the larger the PUF, since additional circuitry is required.
- Long-term reliability (aging)
- CMOS compatibility.
- Hardware efficiency (area, power/energy), which is important for lightweight environments.

6.2.2 HINT Architecture Requirements

Two main use cases of PUF technology have been identified in the literature: (1) identification and authentication of devices and messages, and (2) secure key generation and secure key storage. Both use cases are relevant in the scope of the HINT project. We will investigate how PUF technology can be applied in various application areas:

For a clone created through reverse engineering of embedded software, a PUF-based verification of the hardware platform prevents execution of the clone on any IC different from the one the software was created / licensed for.

In this application, PUFs are used as hardware fingerprints preventing product piracy. The main concern is protection of Intellectual Property (IP) like operating systems or proprietary algorithms. The attacker's intention may be to produce unlicensed copies of an IC together with embedded firmware which exhibits the same functionality as the original IC ("Clones", not distinguishable from the "original" ones). PUFs could be used to authenticate the integrity of genuine hardware in order to prohibit non-licensed usage.

Detection can be done *outside* the smart card: the PUFs response is evaluated off-card, and the system can detect a forgery and respond according to the policy in place (for instance, simply deny cooperation, abort a session, or even record a potential forgery). Within limits, detection may even be performed *on-card*, for instance through the embedded software itself. This may make sense in case the embedded software is not under the attacker's control (e.g., a forger trying to sell cloned hardware to be flashed with genuine software).

Beyond checking the integrity of the hard- and firmware in order to protect the IP of the semiconductor manufacturer, we are going to analyse whether it is also possible to ensure that the software of an application software vendors can be executed on specific semiconductors only. That way a vendor could ensure that its software is only used in accordance with a license agreement.

Similarly to the prevention of product piracy, PUFs can also be used to identify cloned products forged for identity theft. Here, a PUF shall guarantee the authenticity of an individual IC and its embedded software and personalization data.

This application aims at preventing impersonation of an attacker under a false identity. The asset to be protected is the authenticity of an individual to whom a smart card, an embedded key, a digital signature etc. have been assigned to.

A possible attack produces a "functional clone" with regard to the purpose. This means that it may not be necessary to completely clone a smart card, or sometimes not even necessary to produce a smart card at all.

A PUF-based countermeasure would be to deliver along with the proof of identity a proof of integrity of the used hardware. This would thwart impersonation with a cloned device or even with a stolen key not running on any genuine device at all (for instance a digital signature pretended to be generated by a smart card, but in fact generated on a PC using a compromised signature key).

The PUFs could in addition be merged with further information related to the (human) user, like biometric features. One could conceive a "super"-PUF merging both hardware elements (like memory cells) with features from a biometric feature (like a fingerprint capture).

Embedding of a key into a PUF based on IC hardware (like memory cells) can effectively prevent reading out a secret key from non-volatile memory, as the key is not physically stored.

Conventional cryptography heavily relies on the ability to store secret information. However, keys are typically stored binary in non-volatile memory (such as EEPROM), which is vulnerable to hardware attacks. These can be prevented by using PUF technology.

One possible application example is the protection of keys for memory encryption (similarly, the private key for an on-card signature generation or other cryptographic operations could be protected). Memory encryption is a commonly used means in smart card technology to protect sensitive data. Such data are considered sensitive since they may constitute the Intellectual Property (IP) of their owner (like an operating system) and hence need protection, or may contain proprietary (and often confidential) algorithms, or even personal data concerning the user (like health records).

Memory encryption is necessary since the non-cryptographic countermeasures are not sufficient to resist current attack potential required for Common Criteria certification for a high security level. The key used for memory encryption is ultimately embedded in the smart card memory itself, and although techniques like splitting the key into distinct components are employed, the safe storage in physical memory is the “last line of defence”. With PUF technology, such a memory encryption key could be “encapsulated” into a PUF construction (e.g., based on memory cells), retrieved whenever the smart card is initialized, and stored in volatile memory only. Attacks on memory encryption targeting the encryption key (i.e. trying to retrieve it from physical memory) would then not have to be considered (unless attacking the PUF itself).

6.3 Innovative techniques for detection of Hardware Trojans

6.3.1 State-of-the-Art

Different methods of classifying hardware Trojans based on various characteristics have been proposed [REF 21]. This section presents a classification of HT based on trigger and payload mechanisms ([REF 24], [REF 19]).

6.3.1.1 Triggering

The Trigger is the mechanism that determines the condition under which the `malicious' effect of the Trojan should start. The Trigger can either be generated externally (external signal or a special external physical condition) or internally (internal state of the IC, special data configurations, etc.). Moreover the trigger can either be combinational where the sought condition is the result of a logical operation among several signals or sequential where the signal is generated by a state machine:

- *Digitally Triggered Trojans (DTTs)*: DTTs are split in two sub classes: combinatorial and sequential. To minimize their detection, a combinatorial DTT (resp. sequential DTT) should only induce a malfunction when a rare value or rare sequence of events is detected. Sequential DTT are also called “time-bombs”. It is a kind of synchronous counter/down counter over n bits which trigs when the counter reaches its final value (id zero in the case of down counter). This methodology can also be asynchronous when we consider internal events on nodes acting like the main clock of the IC. These counters can fire another part of the HT or can be mixed with the synchronous and asynchronous approach. Finally FSMs (Finite State Machines) are more complex ways to trig HT when rare sequence of events appears in the CI.
- *Analogically Triggered Trojans (ATTs)*: ATTs can be designed with sensors on the chip based on resistors and capacitances. They can also use the internal activity of the IC. For example, heat produced by a ring oscillator made of inverters can on one hand produce heat, and on the other hand measure the temperature of a dedicated part of the circuit. In such a case, when the temperature threshold is reached, the HT injects its payload.

6.3.1.2 Payload

The aim of a hardware Trojan is to manipulate the behaviour of a hardware component. The "asset" to be protected is the proper and reliable operation of the component compliant with its specification. Hardware Trojans may be used to disclose or compromise confidential information, or to insert a non-conformant behaviour. Examples for the latter could be the downgrade of a security countermeasure, partial malfunction of cryptographic operations (e.g. inserting computational errors), manipulated hardware random number generators to drastically reduce entropy, etc. In the extreme case even a complete denial of service may be targeted by the attacker. All these malicious effects of HT are called the payloads of the HT.

Recent surveys [REF 29], [REF 30] done in the US revealed that the majority of counterfeit electronic parts were either previously used (70%) or new (16%) microcircuits re-marked as higher grade, used microcircuits being sold as new (5%), fake non-working original component manufacturer products (5%), parts recycled from scrap (2%) or clones made from working copies of original designs (2%). Such components represent a real threat to legitimate companies' brand image but also to public safety and national security when state-grade or public services' products suddenly stop working or contain hidden hardware Trojans.

Recently, HTs interest also more and more the research community dealing with embedded systems security: for example quote the conference CHES (Cryptographic Hardware and Embedded Systems) 2009, whose "hot subject" (Hot Topic) was devoted to HTs [REF 31]. All this shows that HTs are emergent and real threats the application vendors shall counteract. Here a non-exhaustive list of known effects brought by insertion of HTs in ICs. They can:

- Switch off a component thanks to a remote access switch (kill switch) [REF 32]. That can be devastating if this kind of switch is embedded, for example, in launchers of missiles embarked in combat aircrafts;
- Put ICs in abnormal operations by deteriorating internal nodes [REF 33], and for example to involve a premature ageing of the circuit. That can be very expensive: for example, an infected satellite, which is supposed to function 15 years normally cannot work more at the end of 6 months because of a HT;
- To make voluntarily leak a secret (encryption key) through a hidden channel such as electromagnetic radiation [REF 7], [REF 34], [REF 35]. That raises security issues when supposed communications can be deciphered from the leakage of the secret key;
- To assist a software attack (malware) by providing a hardware hidden door (backdoor) which can generate fraudulent operations on a PC such privilege escalation, automatic logins with a system, or theft of passwords [REF 36];
- To prevent IC from re-entering in "energy saving" mode, which results in energy exhaustion of devices [REF 33].

6.3.2 HINT Architecture Requirements

HINT architecture must be able to detect hardware Trojans that can endanger the security objectives for the PMR products described in section 3.3.5.1:

- OT.AC_Pers
- OT.Data_Int
- OT.Data_Conf
- OT.Prot_Abuse_Func

- OT.Prot_Inf_Leak
- OT.Prot_Phys-Tamper
- OT.Prot_Malfunction
- OT.Prot_Availability
- OT.Hardware_Integrity

In the particular case of PMRs, it is not possible to process Hardware Trojan detection analysis in run-time, i.e. during military or fireman operations. Thus, if we analyse the PMR life cycle, the detection can take place at two levels at test-time:

- At the repair centre (see Figure 6), after an incident (compared to smart card, it is worth repairing PMR products since they can cost hundreds of Euros),
- During the loading of the battery of PMR products, between 2 operations.

Here is a list of technological problems that HINT needs to investigate:

- Since we are in a test-time scenario, HINT architecture should be able to test a circuit within some minutes.
- HINT architecture should be able to detect very tiny hardware Trojans (say Trojan area/circuit area < 0.001%) by side-channel analysis.
- HINT architecture should be able to detect all kinds of digital Trojans (section 5.3.1.1.). We will study only digital Trojans and not Analog ones since these latter are more difficult to insert.
- HINT architecture should guarantee a sufficiently high detection rate. We can take a success metric similar to biometric systems: we need < 1 % of false positives (circuits stated as infected, while they are not), and < 0.1 % of false negatives (circuits stated as non-infected, while they are).
- HINT architecture should obtain detection methods which works even with a high process variation noise, because the contribution of a HT on power consumption is very small compared to this noise.
- HINT architecture should be also valid in the future, with the action of the Moore's law.

6.3.3 Trojans detection

Most research works have indeed focused on the detection of Trojans. Detecting Trojans is a complex, multi-dimensional problem which depends on the type of the Trojan functional or parametric, its size, its distribution over the IC's surface and its structure. Taxonomy for detecting Trojans is proposed in [REF 21]. Apart from classical IC analysis techniques either based on Failure Analysis or on ATPG (Automatic Test Pattern Generation) which are limited in terms of coverage of the IC, novel techniques have been proposed:

- *Destructive methods*: are mainly based on reverse-engineering. We try to find the IC skeleton and its functions with advanced techniques of microscopy. Once we have recovered the chip, a comparison with a reference circuit is done. This method has some drawbacks: the reversed circuit cannot work after this step. The process is very expensive (thousands Euros) and time consuming (several months). Reverse-engineering becomes very complex due to high integration density in the design of IC. Finally an attacker can infect only a few die on a wafer then with this approach we can miss some infected circuits. We won't focus on this detection category in HINT.

- *Non-destructive methods*: There are invasive methods and non-invasive ones:
  - *Invasive non-destructive methods*:
    - Preventive: Some invasive methods can be preventive ones: they are used to make the insertion of malicious circuit very difficult. Obfuscation is an example [REF 25]. This methodology protects circuit from the reverse-engineering and thus the insertion of a malicious circuit. Modifications in the IC transition diagram hide rare important events, the same targeted by the attacker. In this case the hacker has more difficulties to find out about the circuit architecture. As a result, obfuscation leads to a better detection of HT (+24%) for a reasonable cost area (10%).
    - Helping methods: split IC operating mode in two: a normal mode and a "transparent" one [REF 26]. This mode offers the capability in controlling rare events; creating dedicated signatures that will be memorized and then compared with the ones computed during conception phase. This helps in detecting HT insertion during the production phase.
      In [REF 27] another helping method called VITAMIN inverts some logic values inside the circuit during testing phase. Here we increase the probability of trigger HT and detect them.

      A last method uses shadow latches ([REF 28], [REF 23]) in the delay calculation of numerous internal paths. Back from foundry, differences between design phase and testing phase will detects HT insertion. The advantages here are a low cost area, evaluation phase is very short and this technique can be used before or after the deployment of the genuine circuit.
  - *Non-invasive methods*: In this use case all the tested circuits are compared to a reference circuit (golden circuit). There are on-line (normal mode) or off-line (used during testing phase) methods:
    - On-line: In [REF 23] a reconfigurable logic block named DEFENCE (DEsign-For-ENabling-Security) is added. It implements a kind of real time monitor for security. It is inserted at RTL level of the conception flow. This method offers very few documentation and results for the cost area and the detection level of HT.
    - Off-line: Detection methods based on side channel analysis monitor the physical parameters of an IC such as the power consumption, electromagnetic emissions or the combinatorial delays. Here we can use SCA to detect HT even it has not been triggered yet. This method lies on the comparison of power consumption of the suspected circuit with a golden circuit. Side-channel analysis is one of the most 'documented' non-invasive techniques. By changing the design's parametric characteristics, the power/delay characteristics of the circuit are directly impacted. Several researches are oriented on the analysis of the circuit power consumption [REF 10] and current consumption (Hardware Trojan detection and isolation using current integration and localized current analysis). However such techniques are limited by the size of the Trojan with respect to the IC under analysis. More sophisticated region-based analysis have been proposed in [REF 22]. Timing based analysis can represent a more elaborate detection technique. In [REF 20], the authors propose a method of detecting Trojans by measuring the delay paths of a DES circuit.

Instead of detecting HT, some researches focus on the activation of these malicious modifications. The goal is to minimize the circuit activity [REF 22]. The study of the nets' design and the observability of various part of this net could permit to find the payload and the trigger [REF 30]. Moreover, there is a recent methodology consisting of the improvement of the Trojan detection thanks to the design flow modification of [REF 23].

However, most of the techniques proposed in the literature rely on the availability of a “golden reference”. The “golden reference” is an implementation of the IC which is Trojan-free and onto which the reference side-channel or delay path measurements are made and later used as a reference when implementing the Trojan detection scheme onto other instantiations of the IC. Such a “golden” is only available once the circuit has been fabricated and totally reverse-engineered (which is an expensive and time consuming process) to make sure that it contained no Trojan.

When the question is to detect an additional circuit with low activity by SCA, as it is the case for a HT or for a cryptographic circuit protected by a logical circuit SCA-resistant, a long acquisition campaign is required. This means the recording of billion traces to be dealt with later on. One objectives of HINT project is to carry out the traces processing on the fly in order to increase the traces number significantly and thus to allow detection of low secret activity like a HT.

Bibliography

[REF 1] P.C. Kocher, *Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems*, in proceedings of Crypto'96, LNCS 1109, pp. 104-113, 1996.

[REF 2] K. Gandolfi, C. Mourtel and F. Olivier, Electromagnetic analysis: concrete result, in the proceedings of CHES 2001, LNCS 2162 , pp. 251-261, Springer-Verlag, 2001.

[REF 3] C. Gebotys, S. Ho and A. Tiu. EM Analysis of Rijndael and ECC on a PDA, Technical Report:CACR 2005-13, Dept of Electrical and Computer Engineering, 2005.

[REF 4] Christophe Clavier, Side Channel Analysis for Reverse Engineering (SCARE) - An Improved Attack Against a Secret A3/A8 GSM Algorithm, eprint Cryptology archive, 2004.

[REF 5] Xinmu Wang, Seetharam Narasimhan, Aswin Krishna, Swarup Bhunia, SCARE: Side-Channel Analysis Based Reverse Engineering for Post-Silicon Validation, VLSI Design, International Conference on, pp. 304-309, 2012 25th International Conference on VLSI Design, 2012

[REF 6] D. Aboulkassimi, L. Freund, J. Fournier, M. Agoyan, B. Robisson & A. Tria, ElectroMagnetic Analysis (EMA) of software AES on Java mobile phones, in the proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS'11), November 2011

[REF 7] L. Lin, W. P. Burleson, and C. Paar. MOLES: Malicious on-chip leakage enabled by side-channels. In Proceedings of the 27th IEEE/ACM International Conference on Computer-Aided Design (ICCAD 2009), pp. 117-122. ACM Press, 2009.

[REF 8] Jean-François Gallais, Johann Großschädl, Neil Hanley, Markus Kasper, Marcel Medwed, Francesco Regazzoni, Jörn-Marc Schmidt, Stefan Tillich, and Marcin Wójcik. Hardware trojans for inducing or amplifying side-channel leakage of cryptographic software. In Proceedings of the Second international conference on Trusted Systems (INTRUST'10), Liqun Chen and Moti Yung (Eds.). Springer-Verlag, Berlin, Heidelberg, pp 253-270, 2010.

[REF 9] Jim Aarestad, Dhruva Acharyya, Reza M. Rad, Jim Plusquellic: Detecting Trojans Through Leakage Current Analysis Using Multiple Supply Pad IDDQ s. IEEE Transactions on Information Forensics and Security 5(4): 893-904 (2010).

[REF 10] Dakshi Agrawal, Selcuk Baktir, Deniz Karakoyunlu, Pankaj Rohatgi, and Berk Sunar. Trojan Detection using IC Fingerprinting. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP '07). IEEE Computer Society, Washington, DC, USA, pp 296-310, 2007.

[REF 11] Dongdong Du, Seetharam Narasimhan, Rajat Subhra Chakraborty, and Swarup Bhunia: Self-referencing: a scalable side-channel approach for hardware Trojan detection. In Proceedings of the 12th international conference on Cryptographic hardware and embedded systems (CHES'10), Stefan Mangard and François-Xavier Standaert (Eds.). Springer-Verlag, Berlin, Heidelberg, 173-187., 2010.

[REF 12] Reza Rad, Jim Plusquellic, and Mohammad Tehranipoor. Sensitivity analysis to hardware Trojans using power supply transient signals. In Proceedings of the 2008 IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08). IEEE Computer Society, Washington, DC, USA, 3-7. 2008.

[REF 13] Daniel Ziener and Jürgen Teich. Power Signature Watermarking of IP Cores for FPGAs. J. Signal Process. Syst. 51, 1 (April 2008), 123-136, 2008.

[REF 14] Georg T. Becker, Wayne Burleson and Christof Paar. Side-channel Watermarks for Embedded Software. In 9th IEEE NEWCAS Conference (NEWCAS 2011), Bordeaux, France, June 2011.

[REF 15] William E. Cobb, Eric D. Laspe, Rusty O. Baldwin, Michael A. Temple, and Yong C. Kim. Intrinsic Physical-Layer Authentication of Integrated Circuits. Trans. Info. For. Sec. 7, 1 (February 2012), pp 14-24, 2012.

[REF 16] Farinaz Koushanfar. Integrated circuits metering for piracy protection and digital rights management: an overview. In Proceedings of the 21st edition of the great lakes symposium on Great lakes symposium on VLSI (GLSVLSI '11). ACM, New York, NY, USA, 449-454, 2011.

[REF 17] Martin Goldack. Side-Channel based reverse engineering for microcontrollers, Thesis, Ruhr-University Bochum, January 2008.

[REF 18] B. Mounier, A-L. Ribotta, J. Fournier, M. Agoyan & A. Tria, EM probes characterisation for security analysis, in 'Cryptography and Security: From Theory to Applications', Quisquater Festschrift, pp 248-264, D. Naccache editor, Springer LNCS 6805, March 2012.

[REF 19] Y. Jin, N. Kupp, and Y. Makris, Experiences in hardware trojan design and implementation, in Hardware-Oriented Security and Trust, 2009, HOST '09. IEEE International Workshop on, july 2009, pp. 50 –57.

[REF 20] Y. Jin and Y. Makris, Hardware trojan detection using path delay fingerprint, in Proceedings of HOST'08, 2008.

[REF 21] X. Wang, M. Tehranipoor, and J. Plusquellic, Detecting malicious inclusions in secure hardware: Challenges and solutions, IEEE International Workshop on, Hardware-Oriented Security and Trust, vol. 0.

[REF 22] Mainak Banga, Michael S. Hsiao : A Novel Sustained Vector Technique for the Detection of Hardware Trojans, VLSI Design , pp. 327-332, 2009

[REF 23] Hassan Salmani, Mohammad Tehranipoor, Jim Plusquellic. New design strategy for improving hardware Trojan detection and reducing Trojan activation time, San Francisco, CA, USA, July 27-July 27; M. Abramovici and P. Bradley, "Integrated Circuit Security - New Threats and Solutions", CSIIR Workshop, 2009

[REF 24] R. S. Chakraborty, S. Narasimhan and S. Bhunia. "Hardware Trojan: Threats and Emerging Solutions". In High-Level Design Validation and Test Workshop (HLDVT), IEEE. Pages 166-171. 2009.

[REF 25] R. S. Chakraborty and S. Bhunia. "Security against Hardware Trojan through a Novel Application of Design Obfuscation". In International Conference on Computer-Aided Design Digest of Technical Papers (ICCAD), IEEE. Pages 113-116. 2009.

[REF 26] R. S. Chakraborty, S. Paul and S. Bhunia. "On-Demand Transparency for Improving Hardware Trojan Detectability". In International Workshop on Hardware-Oriented Security and Trust (HOST), IEEE. Pages 48-50. 2008.

[REF 27] M. Banga and M. S. Hsiao. "VITAMIN: Voltage Inversion Technique to Ascertain Malicious Insertions in ICs". In International Workshop on Hardware-Oriented Security and Trust (HOST), IEEE. Pages 104-107. 2009.

[REF 28] J. Li and J. Lach. "At-Speed Delay Characterization for IC Authentication and Trojan Horse Detection". In International Workshop on Hardware-Oriented Security and Trust (HOST), IEEE. Pages 8-14. 2008.

[REF 29] American Defense Science Board. "Task Force on High Performance Microchip Supply". 2005.

[REF 30] F. Wolff, C. Papachristou, S. Bhunia and R. S. Chakraborty. “Towards Trojan-Free Trusted ICs – Problem Analysis and Detection Scheme”. In Design, Automation and Test in Europe (DATE), IEEE. Pages 1362 – 1365. 2008.

[REF 31] Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2009. http://www.iacr.org/workshops/ches/ches2009/ches09_hottopic_cfp.pdf

[REF 32] S. Adee. “The Hunt for the Kill Switch”. In IEEE Spectrum, vol. 45(5). Pages 34-39. 2008.

[REF 33] F. Wolff, C. Papachristou, S. Bhunia and R. S. Chakraborty. “Towards Trojan-Free Trusted ICs – Problem Analysis and Detection Scheme”. In Design, Automation and Test in Europe (DATE), IEEE. Pages 1362 – 1365. 2008

[REF 34] L. Lin, M. Kasper, T. Güneysu, C. Paar and W. Burleson. “Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering”. In Workshop on Cryptographic Hardware and Embedded Systems (CHES), LNCS. Volume 5747. Pages 382-395. 2009.

[REF 35] Y. Jin and Y. Makris. “Hardware Trojans in Wireless Cryptographic ICs”. In IEEE Design a Test on Computers. Pages 26-35. 2010.

[REF 36] S. T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang and Y Zhou. “Designing and Implementing Malicious Hardware”. In Workshop on Large-Scale Exploits and Emergent Threats (LEET), USENIX. 2008.

[REF 37] "Physical one-way functions", Ravikanth S. Pappu, PhD Thesis, MIT, 2001.