HECTOR

D4.3

Security Evaluation of the HECTOR Demonstrators

Project number:	644052
Project acronym:	HECTOR
Project title:	HECTOR: Hardware enabled crypto and randomness
Start date of the project:	1st March, 2015
Duration:	41 months
Programme:	H2020-ICT-2014-1

Deliverable type:	Report
Deliverable reference number:	ICT-644052 / D4.3 / V1.0
Work package contributing to the deliverable:	WP 4
Due date:	February 2018 – M36
Actual submission date:	28th February 2018

Responsible organisation:	BRT
Editor:	Gerard van Battum
Dissemination level:	PU
Revision:	1.0

Abstract:	This report describes the security assessment of the three HECTOR demonstrators. For each demonstrator it consists of a description, evaluation scoping, a vulnerability analysis and – when applicable – a description of tests that have been done to get sufficient assurance on the security of the devices. The approach and results of selected tests are described.
Keywords:	Security evaluation, vulnerability analysis, demonstrators, physical attacks, side channel analysis, perturbation attacks, attack potential.

The project HECTOR has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644052.

Editor

Gerard van Battum (BRT)

Contributors

Martin Deutschmann (TEC)

Dave Singelee (KUL)

Marek Laban (MIC)

Viktor Fischer (UJM)

Bernard Kasser (STR)

Emeline Hufschmitt (TCS)

Executive Summary

This report describes the security evaluations of the three HECTOR demonstrators.
For any evaluation it is necessary to know the intended use case of the product, to define the assets, the environment in which the product will be operated and the anticipated attack potential of the adversaries that may attack the device. When this landscape is drawn, a vulnerability analysis is started that tries to envision all attack scenarios in which assets might be compromised, within the limitations of the environment and the estimated attacker's attack potential.
In case execution of the attack scenario cannot be executed successfully due to the above limitations, it is discarded as unrealistic. In case the execution of the scenario seems feasible, but its success depends on behaviour of the device under attack, a penetration test will be proposed to determine the actual threat.
The vulnerability analysis will thus provide a list of potential attack scenarios and a list of penetration tests to verify their practical feasibility. The resulting test plans aim to demonstrate the presence (or absence) of specific behaviour.

The outcome of the vulnerability analysis showed the following:

- Demonstrator 1: A twin-TRNG design to be used as stand-alone, high speed and high-entropy source of random numbers: The intended use case and environment cause that there are no exploitable attack scenarios.
- Demonstrator 2: A high-security USB storage device demonstrating the use of Authenticated Encryption and a PUF: The high entropy offered by the pass-phrase and the PUF does not allow feasible attack scenarios on the secured data at rest. Attacking a live device is impractical because it requires chip-level attacks while the user is entering the pass-phrase and the PUF response is being reconstructed.
- Demonstrator 3: A secure messaging device for communication over insecure channels: In Demonstrator 3 the pass-phrase and PUF are used for authentication. Demonstrator 2 Attacking live communication requires chip-level attacks while the user pass-phrase is being entered and the PUF response is being reconstructed.

Although no practically feasible attack scenarios could be devised for the three demonstrators, and thus no associated penetration tests were defined, it was important for the HECTOR partners to research the security strength of protection mechanisms. Besides the robustness testing in work packages 2 and 3, a number of additional penetration tests were devised that support such research. These tests are described in the annexes of this report.
It must be specifically noted that the development of physical security is not a HECTOR objective. At the same time it is an important aspect for commercialization of the HECTOR achievements by the industrial partners (MICRONIC, THALES, ST). Therefore a separate physical security analysis was done on the platforms of Demonstrators 2 and 3 to provide more insights and determine vulnerabilities related to the hardware enclosure. Protection provided by the enclosure is less relevant for Demonstrator 1 because this device will only be used in a controlled environment.
The additional penetration tests showed weak and strong aspects in the implementation of the demonstrators. Some weaknesses are inherent and difficult to avoid (keyboard and display emanation). Strong aspects are the robustness of Demonstrator 1 TRNGs against frequency injection (WP2 research) and the resistance of the pass-phrase retry-mechanism against laser perturbation for Demonstrators 2 and 3.
In addition also verification testing was done to determine the temperature effect on the entropy generation of the Demonstrator 1 TRNGs.
Demonstrator 1 is also the most likely product to be formally certified for commercial use. Therefore a trial evaluation was done for HECTOR partner UJM according to the Common Criteria methodology using TRNG standard AIS20/31. An example section of this (confidential) work is enclosed in this report.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Three Demonstrators are conceived in work package 4 of the HECTOR project:

- Demonstrator 1: An USB card with two HECTOR True Random Number Generators (TRNGs)
- Demonstrator 2: A secured memory storage device that can connect to a computer through USB
- Demonstrator 3: Communication devices that can be used to securely exchange messages between two parties

All Demonstrators contain primitives that are developed during the HECTOR project. The purpose of the Demonstrators is to show how these primitives could be used in commercial applications. The HECTOR primitives fulfil the requirements for use in industrial commercialization, but are not commercial products.

In order to obtain the information to do the evaluation, it has to be found out if the product is sensitive for certain attacks. In most cases this is determined from the design documentation: if the design inherently protects against an attack step, the associated attack scenario will not be feasible and thus discarded. In other cases it cannot be determined from the design that the product has sufficient or effective protection. Actual testing needs to be done to obtain information for a verdict. This way the vulnerability analysis will eventually lead to a list of tests that need to be done to get assurance. This is called the test plan.

A vulnerability analysis aims to show weaknesses in a design that might be exploitable by attackers. This is always done on products that will be applied within a pre-defined context. ‘Scoping’ is necessary to prevent that unrealistic or exotic attack scenarios need to be taken into consideration. Usually a range of use-cases is assumed for the product, while the user is obliged to follow the operational guidance that comes with the product (like user-manuals). At the other end assumptions need to be made on the capabilities of the envisioned attacker. It makes a large difference if the product has to protect assets against hobbyists or state-funded intelligence services. The capabilities of envisioned attackers (attack potential) depend on the use-case and the value of the assets that the product is intended to protect.

The vulnerability analysis will provide so-called attack scenarios, which describe all steps an attacker has to overcome to get access to assets. A vulnerability analysis will result in potential attack scenarios. If the effort for exploitation of a scenario is within reasonable boundaries, it makes sense to estimate the difficulty in terms of required *time, knowledge, equipment and number of samples* that an attacker needs to successfully execute the attack scenario. During the evaluation it will be verified if the Target Of Evaluation (TOE) provides sufficient protection to withstand (parts of) the attack. The Target Of Evaluation is a commonly used expression in security evaluations to describe the product that will be subject for certification.

The resistance against attacks depends on the design and its implementation. The *assurance* that this resistance is adequate and effective is obtained by assessment of design information and/or by testing. Two types of testing can be used to get assurance on the resistance of attacks, which are:

- Penetration testing: (parts of) attacks are executed in order to get a metric on the difficulty of exploitation of an attack scenario
- Verification testing: verify if the TOE behaves in practice as designed.

Both types of tests are used to obtain a final verdict on the resistance against attacks.

It is also important to notice the difference between evaluation (valuing the attack difficulty) and testing (pass/fail, digital).

The use-cases for the HECTOR demonstrators can be diverse and depend on commercialization by the industrial partners. For the security evaluations we assumed that the demonstrators should withstand attacks by attackers with 'High attack potential'. This is a deliberate choice as we would like to show-case the HECTOR achievements. For reference: the level 'High attack potential' is the level used in high-end financial applications. The level 'Attack potential beyond high' is rare and aims to protect against state-level organizations with virtually unlimited resources (military, national security agencies), which was not the HECTOR design target.

The HECTOR project uses three different demonstrators to show-case the capabilities of the building blocks developed during the project. These demonstrators have specific characteristics and use cases, and therefore separate vulnerability analyses were done.

Demonstrator 2 and Demonstrator 3 share the same hardware platform, which – in commercialized situations – should provide additional physical protection against access to assets of the applications. It must be clearly noted that the development of physical security is NOT a HECTOR objective. At the same time it is an important aspect for commercialization of the HECTOR achievements by the industrial partners (MICRONIC, THALES) that a suitable hardware platform is present. Therefore a separate analysis was done on the physical platforms for Demonstrators 2 and 3 to determine vulnerabilities that are related to the hardware. This is valuable input for the HECTOR partners in creating future products.

Chapter 2 Demonstrator 1

2.1 Introduction

Demonstrator 1 is a plug-in device with the form factor of a 2,5” Hard Disk Drive which can be mounted in a host PC. Communication is done via USB, using an internal USB port of the host PC. The device is powered by the internal host PC power supply and contains two individual TRNG primitives:

- The first TRNG is a Phase Lock Loop-based generator developed by HECTOR partner Université Jean Monnet (UJM)
- The second generator is based on Delay Chains and is developed by partner Katholieke Universiteit Leuven (KUL).

A more detailed description of Demonstrator 1 is provided in the demonstrator accompanying report D4.2 [2].

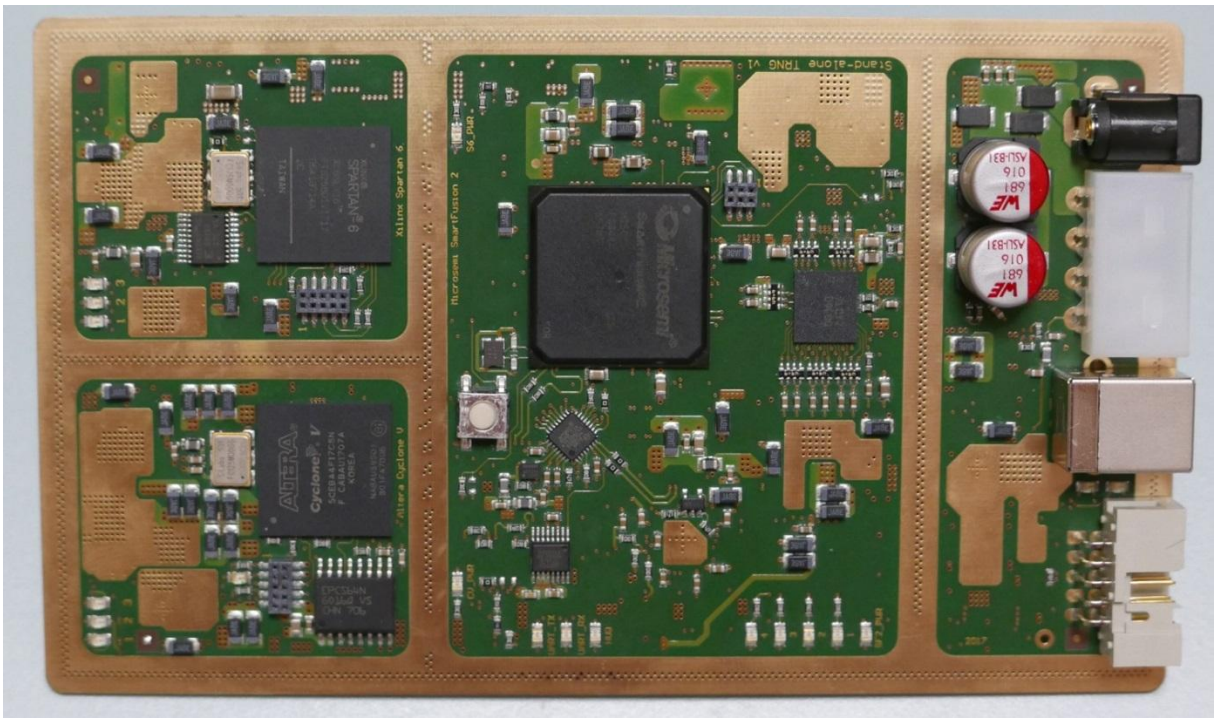

Figure 1: Demonstrator 1, dual TRNG with USB interface in 2,5”enclosure.

2.2 Scope of the evaluation

High-speed TRNGs like Demonstrator 1 are typically used for key generation during personalization in production environments, or as independent source for random numbers for user authentication in e.g. data centres. Due to the required confidentiality these operations take place in controlled environments. This means that physical access to the TOE is restricted by means of access control and procedures. In other words, it is assumed

that attackers cannot physically reach (or get near) the device to mount an attack and thus it does not need protection against such attacks.

The host PC that powers and drives Demonstrator 1 is not in scope of the evaluation. A standard PC is assumed that offers no special protection, except for what you may expect of a standard PC (normal metal enclosure, power supply, filters on interfaces).

The secure environment is not in scope of the evaluation. A standard secure room is assumed offering protection against physical intrusion and a suitable power provision, as well as sufficient damping of electro-magnetic radiation by structural elements and/or physical distance between the sensitive components and unprotected area.

Depending on the application access protection might be required to the TRNG output stream. This is not a design criterion for Demonstrator 1 and – if required – should be enforced by the environment (e.g. by the host PC or a firewall).

All guidance on secure operation of the demonstrator is applied (see [2]).

2.3 Vulnerability analysis

2.3.1 Attacking Demonstrator 1

The TRNGs are located at a plug-in device inside a host PC. Influencing the TRNGs by direct physical methods is excluded because the device resides inside a secured area. An attacker could have advantage when he/she is able to:

1. Remotely influence the generation process of random numbers in such way that entropy is degraded. It depends on the application that uses the random output if degraded random numbers are exploitable in an attack scenario. However, in general it can be said that it is not tolerable that random numbers are degraded below the requirement level.
2. Eavesdrop on the random number output stream in order to obtain the random output values. It depends on the application of the random numbers if this is a threat to the system. When random numbers are used as a challenge in communication protocols they might be publicly known. However, random numbers should remain secret when used as cryptographic key or input for countermeasures against e.g. perturbation or side channel attacks. For this evaluation it is assumed that the generated random numbers should remain secret.

2.3.1.1 Remote influence on TRNG entropy

The operation of any random number generator can be brought down in essence to be dependent on analogue physical processes. Examples are molecular vibrations by thermal input (Brown movements), Poisson noise and noise caused by electro-magnetic forces. Due to their nature such processes can be influenced by physical effects. By itself an attacker is not interested in creating more noise, but to reduce it or to control it. The final aim is to reduce the entropy in the random number output stream in a controlled way.

Attacks that aim on removing all entropy (e.g. by shutting down the source of randomness) might appeal favourable, but all TRNGs that fulfil the BSI AIS20/31 and NIST 800-90B requirements are equipped with Total failure tests that detect such attack. Also the HECTOR TRNGs in Demonstrator 1 are compliant to the aforementioned BSI and NIST standards, so they will detect shutdown of the entropy source and stop their services. Combination attacks can be devised in which both the entropy source is affected *and* the Total failure test is bypassed. In practice such combination attack becomes infeasible within the given scope because their complexity requires direct physical access to the device.

Potentially practical attacks might be to remotely degrade the TRNG entropy using radiation (electro-magnetic, light) or by direct coupling through interfaces such as the power supply.

Literature [1] and our experiments done for Work Package 2 show that the TRNG entropy can be influenced by injecting energy directly at the power supply. However, the amount of energy and the (in) effectiveness of the coupling makes that this will not be feasible on devices that are located at some distance in a secured environment. Even in case an adversary is capable of injecting high-power, high-frequency energy on the electrical power grid that powers the PC in the secure room, damping and filtering by network impedance and PC power supply capacitors will absorb the energy to ineffective levels.

Frequency injection by electro-magnetic fields is theoretically feasible as well. However, experiments using EM injection in WP2 show that high energy levels are required at short distance (centimetres) to obtain any noticeable effects. If this is to be done remotely – targeting the TRNG and host PC in a secure room – unrealistic equipment is needed to obtain the required energy levels.

In case the remaining energy level would still be high, then the host-PC will likely suffer from the energy injection before the TRNG will noticeably be influenced.

The research in HECTOR Work Package 2 also shows that the entropy reduction by frequency injection is not trivial. Moreover, it was demonstrated that entropy degradations will be detected by the integrated on-line tests.

The experiments show that it is very difficult for an attacker to obtain useful results from degradation of TRNG entropy by remote influence, when the TRNG is operated within a secured room and the guidance is respected.

2.3.1.2 Eavesdrop on random output stream

Since the use-case excludes direct physical access to the demonstrator, only side channels might be used to disclose the TRNG output stream. This requires:

1. *Useful side channel analysis methods*
   In this case only template attacks are candidate to yield potentially interesting results since these provide a means to predict output data based on properties of the emanation. Template attacks model processed data (e.g. bytes) with side channel signals and noise to search for matches between the data and the signal. For a successful template attack a learning phase is required which is used to create the model. For this situation this means that the attacker collects side channel information while knowing the output values. In case of Demonstrator 1 the learning phase cannot be done on the device itself, since it is assumed to reside permanently inside the secured environment. However, an identical demonstrator device could be used to do the template modelling phase. As long as the source of emanation is nearly identical to that of the target source, templates might be applied cross-platform (with some efficiency loss). Once templates are created, a real device can be attacked by matching the templates with the challenge traces.

2. *Suitable side channel signals to be obtained from the target at a distance*
   Side channel emanation can be obtained over-the-air as electro-magnetic radiation, or by direct galvanic coupling, e.g. through the power supply.
   Practical research during Work Package 2 showed that template attacks were not successful in extracting information from similar TRNGs. These tests required electro-magnetic signals to be collected at the target source itself, i.e. at very close proximity (<1 cm, e.g. on top of the FPGA). Unlike the test targets of the template attacks, Demonstrator 1 is shielded with a thick aluminium housing, which further reduces emanated radiation. In addition, Demonstrator 1 is designed to be positioned inside a PC. All electro-magnetic radiation of the PC will also be included in the signal in case electro-magnetic emanation has to be collected from this target at larger distance (several meters, outside the secure room) . Because of the large distance, shielding of the demonstrator and the environmental noise of the PC it is unlikely that the

electro-magnetic radiation will contain sufficient information to do a successful template challenge phase.
Side channel leakage by galvanic coupling is also unlikely to work. The assumption of the source of leakage is that the TRNG random output data will cause some modulation of the power consumption of the PC, which is connected to the electrical power grid of the secure room. The power consumption of the appliances inside the secure room might be measurable outside the secure room, e.g. at a switch board. Not only will such signals be extremely weak due to power supply filtering (capacitors, inductances, switching-mode power supplies, grid impedance), in addition creating the side channel templates would require the operational environment to be taken into account (PC + grid). This is not practical since it would require the training traces to be done on the device in its targeted environment.

Based on reasoning it can be concluded that it is very difficult for an attacker to obtain useful results from TRNG eavesdropping, when the TRNG is operated from within a secured room and the guidance is respected.

2.3.2 Influence of temperature

In principle the quality of random numbers is guaranteed within the specified operating range of 0 ºC to 85 ºC. Outside that range the output quality may be influenced negatively. On-line tests monitor the output quality and will stop generation when the quality drops below defined thresholds.

Demonstrator 1 is not equipped with temperature sensors, which is advisable for real commercial products. As soon as the temperature is outside specified limits, an alarm will be generated. Guidance will explain to the software developers how the application has to react when such alarm is triggered.

An attacker has little possibilities to have controlled influence on the temperature inside the PC in the secure room. Besides that, Demonstrator 1 will not output random numbers once the quality is outside specification, so the attacker has no practical benefit of any degradation by temperature. In WP2 extensive tests were done by HECTOR partners and Brightsight to verify robustness of TRNG designs at different temperatures. To demonstrate that this behaviour is identical in the demonstrator, temperature tests will be done as part of the verification tests.

It was concluded that there are no attack scenarios for Demonstrator 1 for the envisioned use case that can be exploited by adversaries.

2.4 Testing

A distinction is made between evaluation of resistance against attacks and verification testing of behaviour of the demonstrator.

2.4.1 Testing as a result of potential vulnerabilities for attacks

From the vulnerability analysis it was concluded that there are no attacks that require testing.

2.4.2 AIS20/31 verification testing

Verification testing is not to be confused with penetration testing. Verification testing is primarily done as a robustness test to verify if the design behaves as intended over e.g. a certain temperature range. It is far-fetched that an attacker is able to influence the temperature of the Demonstrator 1 host PC with the aim to influence the quality of the entropy generation.

2.4.2.1 Introduction

During Work Package 2 several robustness tests were done on both HECTOR TRNG designs. For this final demonstrator design verification testing is done using statistical tests according to AIS20/31on TRNG output data at different temperatures.

2.4.2.2 Test description

For both TRNG designs – the PLL-TRNG of UJM and the DC-TRNG of KUL – high (+80 ºC) and low temperature tests (-40 ºC) are executed.

The TRNG output quality is verified by running the AIS20/31 test suite. This test suite consists of eight individual statistical tests T1 to T8 that each covers a statistical aspect of the data. A test will pass when complete runs of the AIS20/31 statistical test suite passes (three times). A test fails when two out of three runs of the statistical tests contain failing tests. 'A proposal for functionality classes for random number generators' paragraph 210 describes the procedure for the statistical tests. It comes down to: The test suite is in principle performed once and all basic tests must pass in order to pass 'Test procedure A'. If one of the 1,285 basic tests (e.g. a single mono-bit or poker test) fails, then the test procedure can be repeated once on new random data. That second time all 1,285 tests must have the verdict 'pass' to formally pass 'Test procedure A'. In all other cases 'Test procedure A' fails. Note that – due to the nature of randomness – one or more of the individual tests of the test suite may fail. Depending on the severity of the fail and the importance of the test, this may or may not cause the whole run of the test suite to fail. It can also happen that online tests detect output anomalies, which cause the TRNG to stop operating. In such cases the TRNG will not provide output and the test will fail.

The expected outcome of the test is that all runs of the AIS20/31 statistical test suite pass over the temperature range of -40 ºC to +80 ºC.

2.4.2.3 Test results

The temperature robustness tests on the two TRNGs of Demonstrator 1 are described in Appendix A.

2.4.3 TRNG Common Criteria evaluation of AIS20/31claim

2.4.3.1 Introduction

Industrial application of True Random Number Generators usually requires certification against standards. The TRNG can be part of a system that has well-defined claims on its secure behaviour. A useful claim for TRNGs in a Common Criteria evaluation in Europe is that it conforms to a protection class of AIS20/31. In order to verify if the HECTOR TRNG developments will withstand such Common Criteria evaluation, a trial evaluation was done on one of the HECTOR TRNGs. The PLL-based TRNG – developed by UJM – was selected for this evaluation because HECTORs industrial partners are interested in the commercial application of this design. The results of the evaluation should show that the design is easy to evaluate and will pass relevant work units of the CC evaluation methodology.

The UJM PLL-based TRNG design is developed to conform to TRNG Class PTG3.

The design parameters - which are essential in the entropy generation - are UJM's trade secret. For this reason the trial evaluation was done using a separate non-Disclosure Agreement between UJM and Brightsight. This NDA also covers the evaluation report (see [3]) that was produced as part of this effort. In order to show what such evaluation report contains, UJM gave permission to extract a section of this report. This is presented below.

2.4.3.2 Example of Common Criteria AIS20/31 evaluation

Brightsight has performed a Common Criteria evaluation of the AIS20/31 claims on the TRNG designed by UJM. AIS20/31 is the most widely used standard for evaluation of TRNG's and is developed, maintained and published by BSI (the German Common Criteria (CC) certification body). It describes how a physical or deterministic RNG could be claimed by the developer in the so-called Security Target (ST) document. It also describes the developer evidence contents required and the developer actions needed.

The claim for the HECTOR PLL-based TRNG in Common Criteria style looks like:

*FCS_RNG.1.1 The TSF provides a **hybrid physical** random number generator that implements:*

- *(PTG.3.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.*
- *(PTG.3.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG **prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source**.*
- *(PTG.3.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.*
- *(PTG.3.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.*
- *(PTG.3.5) The online test procedure checks the quality of the raw random number sequence. It is triggered **continuously**. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.*
- *(PTG.3.6) The algorithmic post-processing algorithm belongs to Class DRG.3 with separate cryptographic state transition function and cryptographic output function, and the output data rate of the post-processing algorithm, which does not exceed its input data rate (input and output data rates are the same).*

*FCS_RNG.1.2 The TSF provides **random bits** that meet:*

- *(PTG.3.7) Statistical test suites cannot practically distinguish the internal random numbers from output sequences of an ideal RNG. The internal random numbers pass test procedure A.*
- *(PTG.3.8) The internal random numbers use PTRNG of class PTG.2 as random source for the post-processing. The average Shannon entropy per internal random bit exceeds 0.997.*

All bold face text is being chosen by UJM to make the claim more specific for their design. The evaluator of Brightsight has performed the so-called work-units and given a verdict per work-unit. Work-units PTRNG.3-1 up to PTRNG.3-5 are relevant for the claim. Work-unit PTRNG.3-2 contains PTRNG.2-2 up to PTRNG.2-9 that are defined for a PTG.2 class TRNG. Some examples of work-units are given below:

PTRNG.3-1 Examine the description of the intended use of the RNG in the developer evidence document, the ST, and the guidance documents, and check whether the descriptions are complete and internally consistent.

The verdict is ‘pass’ and the full analysis is given as an example:

1. The evaluator notes that no Security Target is written for the RNG under consideration. All claims are contained in [RNG_Design]. The evaluator determined that [RNG_Design] is referring to one type of PTRNG namely the PTG.3.
2. The evaluator has examined [RNG_Design] and determined that the intended usage is as security service for the user. This is as specific as the purpose of the project allows.
3. The evaluator has examined the operations of SFR FCS_RNG.1 class evaluation and determined that no operations are left open in the SFR.
4. The evaluator has examined the developer evidence and the claim FCS_RNG.1 and determined that they are consistent, as is demonstrated in appendix B[1] of the evaluation report.

PTRNG.2-2 Examine the developer description of the PTRNG module and check for internal consistence.

The verdict is ‘pass’ and is supported by a point-wise check of design choices in 10 different categories.

PTRNG.2-3 Evaluate that the implementation of the RNG is according to the developer's description of the PTG module.

The verdict is ‘inconclusive’ as the activities within the HECTOR project did not include a full implementation review at the detail that is required within Common Criteria.

PTRNG.2-4 Examine the developer's evidence that the internal random number sequence contains at least a minimum amount of entropy, which is identified in the element FCS_RNG.1.2 clause (PTG.3.8) under all intended environmental conditions.

The verdict is ‘pass’ and is supported by an analysis of the stochastic model provided by the developer, explaining how the sampling of the random process can account for the amount of entropy as claimed.

PTRNG.2-7 Examine the developer's demonstration that the online test detects non-tolerable statistical weaknesses of the raw random signals sufficiently soon.

The verdict is ‘pass’. A theoretical model demonstrates how error patterns or a drop of entropy would trigger the online test. From the evidence it can be seen that it is guaranteed that an error vector is faster than the post processing result being outputted. The control I/F module that signals that random data is available, is also the module that triggers an interrupt request in case of an error bit being asserted.

As a summary, most work-units show a ‘pass’ verdict, however some of the work-units are not yet in the ‘pass’ state. This is caused by the rigorous implementation representation review required in a Common Criteria evaluation and also by the way the embedding of a TRNG into a certified product is subject to criteria that cannot be met by Demonstrator 1. However the design principles as such are suitable for being part of a product design that can be CC certified.

[1] Note: This is appendix B of confidential evaluation report [3].

In particular the online tests, that give statistical confidence on the amount of entropy present in the TRNG output, represent very suitable and novel solution for an AIS20/31 compliant TRNG. All work-units related to this online tests gave a ‘pass’ result, see for example PTRNG.2-7.

2.5 Conclusions

Several evaluation work items were done on Demonstrator 1.

- The vulnerability analysis did not show any feasible attack scenarios. This is mainly due to the fact that this demonstrator is intended to be used in a protected environment, does not contain permanent assets other than temporarily during generation of random numbers and has on-line testing that prevents output manipulation.
- Verification tests show that the TRNGs in the device provide sufficient entropy in the raw random numbers over a large temperature range. A slight bias in the raw output streams cause the AIS20/31 statistical tests to fail in occasions. Additional post processing will remove such bias to make the results compliant for passing AIS20/31.
- The Common Criteria evaluation of the AIS20/31 claim ‘PTG3’ of the PLL-based TRNG developed by UJM shows that the design is likely to pass. Some work units could not be verified because Demonstrator 1 is not designed as a commercial product to be subjected to a real CC evaluation.

Chapter 3 Demonstrator 2

3.1 Introduction

Demonstrator 2 is a secure storage device that can be connected to a PC through an USB connection. The main security characteristics are:

- The data is stored at an SD card and is encrypted with an internal Data Encryption Key.
- At rest this key is encrypted by a pass-phrase and a PUF response.
- The pass-phrase is generated during enrolment and has high entropy.
- Pass-phrase-entry is protected by a retry-counter.
- The PUF response binds the secured data to the device.

A more detailed description of Demonstrator 2 is provided in the demonstrator accompanying report D4.2 [2].

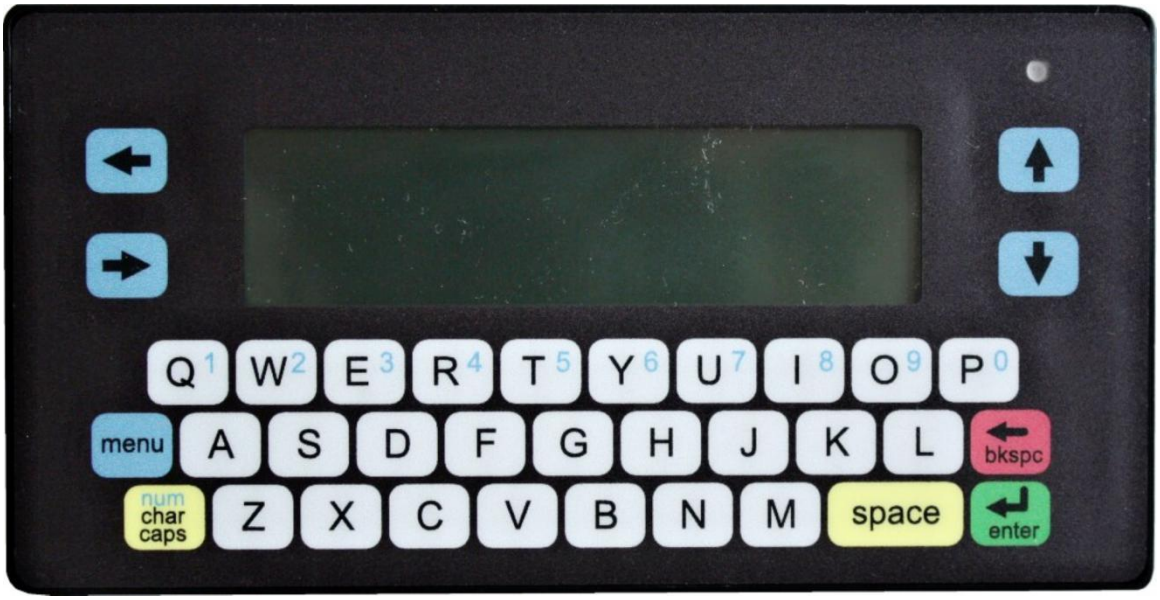

Figure 2: Demonstrator 2, Secure storage device.

3.2 Scope of the evaluation

The secure storage device is intended to be used ‘on the road’. This means that chances exist that the device gets stolen or will be accidentally lost. This is the envisioned attack scenario where a device is “Lost and Found”, after which the attacker tries to obtain the stored data at rest. This is the most common attack scenario that industry uses and where secure storage devices have to be resistant against.

However, one could also envision an attack scenario where a device is selectively targeted, for example when it is owned by a high official and adversaries deliberately try to collect information from this particular device. Attackers in this category are usually highly organised, have large resources and capabilities and can be found in government organizations or well-organised crime. In such situations the physical protection of the device (tamper evidence, tamper resistance and even tamper response) are of importance.

Developing tamper resistant hardware was not an objective of HECTOR, and therefore less emphasis was put on this design aspect. Therefore attacks at 'government level', including device modification, bug insertion and chip-level attacks, are out of scope for this demonstrator.

However, it is helpful for the industrial HECTOR partners to have information on the physical security strength of the current design. This is covered in Appendix B.

To mimic a realistic use-scenario all user guidance is applied (see [2]) on how to operate the demonstrator in a secure manner.

3.3 Vulnerability analysis

The user data – stored in the SD card – is encrypted by a Data Encryption Key (DEK). At rest this key is stored inside the device, encrypted by the Key Encryption Key (KEK). To be able to decrypt and read the data in the SD card, the DEK needs to be decrypted first using the KEK. The KEK is composed of the PUF response plus the user-passphrase.

The pass-phrase consists of eight words with entropy of 13 bits per word. The initialization method and user guidance guarantees that the complete pass-phrase has total entropy of at least 96 bits. The PUF will add an additional 32 bits, making the total entropy of the KEK 128 bits. Attacking the access mechanism by brute forcing the KEK is currently infeasible due to this high entropy.
In addition a retry counter limits the number of incorrect pass-phrase entries. The value of this retry-counter can be set by the user during enrolment from 1 to 20 attempts. In case this value is exceeded, the device will automatically erase the Data Encryption Key, so all data on the SD card will be permanently lost.

The use of a pass-phrase as (part of) an encryption key prevents attacks in the Lost and Found scenario. The attacker does not have any means to derive the pass-phrase because the legitimate user is not involved, while a brute-force attack is ruled-out due to the large key-space. Given the minimum entropy of 96 bits for the pass-phrase, an attacker has a search space of 2^{96}. This makes attacks at a device at rest practically infeasible.

In that perspective the retry-mechanism seems redundant and may even be annoying for the user-experience, especially because the demanding pass-phrase entry process may cause false rejections by typing errors. Nevertheless, the retry mechanism is present as a back-up in case parts of the pass-phrase would become exposed. In such cases the pass-phrase retry-mechanism may have its relevance.

Experience shows that the effectiveness of the retry mechanism is strongly dependent on proper implementation. Although the retry-mechanism is not strictly necessary for protection of the device, this mechanism is analysed in more detail to provide HECTOR partners with information that can be used in product commercialization.

The retry mechanism will verify if the tag of the ASCON-128 encryption of PUF helper data is correct by comparing it with the tag value stored in NVM during enrolment. The length of the tag is 16 bytes. The comparison is done in the FPGA fabric. Verification of the tag is more secure than a direct comparison of the pass-phrase.

The retry mechanism consists of:

- A verification that compares the calculated tag with the ASCON tag stored in NVM. The result of the comparison is a bit set in the FPGA crypto module status word.
- An update mechanism of the retry counter, including a reset procedure in case a correct password is entered after incorrect ones. The value of the retry-counter is stored in FPGA NVM.

Two categories of attacks are applicable:

1. Side channel analysis, e.g. as method to distinguish differences in program flow due to correct versus incorrect comparisons
2. Perturbation attacks, e.g. aiming to skip updating of the retry counter.

When trying to defeat a retry counter the first concern of an attacker is to prevent the counter from updating. A classic method is to detect the NVM write operation that increments (or decrements) the counter value. NVM writes require more power and have a relatively long duration, which makes them generally easy to detect. If detected, the device is immediately switched-off (or reset). This attack belongs to the family of 'card-tearing' attacks. Repetitive entries of pass-phrases can be done using this principle in order to do a brute force attack.

The tag will be verified in one clock period inside the FPGA. The limited bus width of the microcontroller requires the 128-bit tag to be sent to the FPGA in four blocks of 32 bits for verification. It is therefore crucial that no differences are visible in side channel signals between correct and incorrect verifications.

When the retry-counter passes the threshold as set during enrolment it activates the mechanism that erases all content and blocks further use of the device.

Although out of scope of a Lost and Found scenario, an attacker could obtain a copy of the encrypted user data stored on the internal SD card. In the special case that also the pass-phrase would be available to the attacker, the data on the SD card might become exposed by a brute force attack. This is required to derive the 32-bit PUF response that composes $1/4^{th}$ of the entropy of the Key Encryption Key.

It was concluded from the vulnerability analysis that no obvious attack scenarios are present that can lead to a successful exploitation.

3.4 Perturbation testing on the retry mechanism

The vulnerability analysis shows that no attacks can be devised within the scope of the Lost and Found scenario which have a reasonable chance of success. However, it might be that – for better user-experience in future versions of the device – more entropy is extracted from the PUF, so the requirements on the pass-phrase security can be relaxed. In such cases the retry-mechanism becomes more important for protection of the user data. Therefore a test is devised that verifies this mechanism.

3.4.1.1 Test description

This test intends to change the behaviour of the demonstrator when incorrect pass-phrases are being input. First a normal enrolment is done. This loads the demonstrator with all keys and data required for normal operation. The pass-phrase retry counter is set to 20. Then sequences of incorrect pass-phrases are sent to the demonstrator, which reacts by decreasing the retry-counter value. During the verification and the counter-update process, a laser perturbation is done at varying moments in time at varying locations of the chip. The settings of the laser parameters (intensity, wavelength) are determined upfront by testing the sensitivity of the target for light pulses.

The reaction of the demonstrator on the perturbation pulses is determined by the replies that it provides after each operation. The replies can be used to separate normal behaviour from unexpected behaviour. The latter ones could indicate a successful attack and need further explanation.

3.4.1.2 Test results

The light manipulation tests on the pass-phrase retry mechanism are described in Appendix E.

3.5 Conclusions

The vulnerability analysis of Demonstrator 2 – a high-security USB storage device demonstrating the use of Authenticated Encryption and a PUF – showed that the high entropy offered by the pass-phrase does not allow feasible attack scenarios on the secured data at rest. Attacking a live device is impractical because it requires chip-level attacks while the user is entering the pass-phrase and the PUF response is being reconstructed. The effort of chip-level attacks is considered to be at the level of state-funded attacks, which is beyond the scope of the HECTOR designs.

Physical attacks are out-of-scope for the evaluation, but were done to provide knowledge to the HECTOR partners for making commercial secure products based on HECTOR building blocks. The keyboard and display PCB of the demonstrator can be removed without tamper evidence. This may allow bugs to be inserted. Getting access to the FPGA for a chip-level attack will cause substantial damage.

The pass-phrase retry mechanism cannot be manipulated using laser perturbation attacks.

The electro-magnetic emanation of the demonstrator keyboard and display can be used to successfully extract information on keystrokes and displayed characters respectively. This requires a side channel template attack, which consists of a training phase and a challenge (attack) phase. Although this attack is demonstrated to be feasible, in practice it is too complicated. Attackers can obtain better results using a spy-camera.

Chapter 4 Demonstrator 3

4.1 Introduction

Demonstrator 3 is a secure communication device that can be connected to a PC through an USB connection. The main security characteristics are:

- At rest this key is encrypted by a pass-phrase and a PUF response.
- The pass-phrase is generated during enrolment and has high entropy.
- Pass-phrase-entry is protected by a retry-counter.
- The PUF response binds the secured data to the device.

A more detailed description of Demonstrator 3 is provided in the demonstrator accompanying report D4.2 [2].

Figure 3: Demonstrator 3, Secure messaging device.

4.2 Scope of the evaluation

Demonstrator 3 is a communication device that is intended to be used ‘on the road’. As with the Demonstrator 2, the communication device can be lost (and found) or deliberately stolen. The attacker can then try to extract stored secrets from the device. In case of a deliberate attack the goal of the attacker can be to tap-in on the communication or to modify it.

Like with Demonstrator 2, an attack scenario can be foreseen where a device is selectively targeted, for example when it is owned by a high official and adversaries deliberately try to collect information from this particular device. Attackers in this category are usually highly organised, have large resources and capabilities and can be found in government organizations or well-organised crime. In such situations the physical protection of the device (tamper evidence, tamper resistance and even tamper response) are important. Developing tamper resistant hardware was not an objective of HECTOR, and therefore less emphasis was put on this design aspect. Therefore attacks at this ‘government level’, including device modification, bug insertion and chip-level attacks, are out of scope for this demonstrator.

However, it is helpful for the industrial HECTOR partners to have information on the physical security strength of the current design, which is therefore covered in Appendix B.

To mimic a realistic use-scenario all user guidance is applied (see [2]) on how to operate the demonstrator in a secure manner.

4.3 Vulnerability analysis

Similar to the case for Demonstrator 2, the data protection relies on the ASCON-128 algorithm with a key composed of the externally-input pass-phrase and the internally-generated PUF response. It also uses the same retry-mechanism to prevent brute force attacks.

At rest, the device does not contain any keys. As a result, within the scope of the evaluation, no vulnerabilities exist that allow attackers to successfully attack the messaging device at rest.

Attacking the device while operated by the legitimate user requires unobtrusive methods that do not raise suspicion. Therefore only side channel analysis using electro-magnetic radiation is a potential method. A potential attack scenario is to obtain the pass-phrase while the user operates the device and steal it in a later stage. Having both the device and the pass-phrase the attacker can pretend to be the legitimate user and send fake messages (as long as the device is not reported as stolen and locked-out for further use).

Even when the pass-phrase of targeted devices becomes exposed a Man-In-the-Middle attack (MIM) is not possible without knowing the PUF response. Obtaining the on-chip PUF response requires a chip-level attack, which is out of scope of the evaluation.

Demonstrator 2 and Demonstrator 3 share the same hardware. A description of hardware-related vulnerabilities is provided in Appendix B.

4.4 Testing

The protection mechanisms of Demonstrator 3 are identical to those of Demonstrator 2. The different use-case of Demonstrator 3 does not give rise to new threats. No additional tests could be identified.

4.5 Conclusions

The vulnerability analysis of Demonstrator 3 – a high-security messaging device demonstrating the use of Authenticated Encryption and a PUF – showed that the high entropy offered by the pass-phrase does not allow feasible attack scenarios. Attacking a live device is impractical because it requires chip-level attacks while the user is entering the pass-phrase and the PUF response is being reconstructed. The effort of chip-level attacks is considered to be at the level of state-funded attacks, which is beyond the scope of the HECTOR designs.

The additional tests that were done on the physical design of Demonstrator 2 are also applicable on Demonstrator 3. The same conclusions therefore apply.

Chapter 5 Conclusion

The evaluation of the HECTOR demonstrators started with an analysis of the designs and a vulnerability analysis. The outcome of the vulnerability analysis showed the following:

- Demonstrator 1: The intended use case and environment makes that there are no exploitable attack scenarios.

- Demonstrator 2: The high entropy offered by the pass-phrase does not allow feasible attack scenarios on the secured data at rest. Attacking a live device is impractical because it requires chip-level attacks while the user is entering the pass-phrase and the PUF response is being reconstructed. Chip-level attacks are beyond the attack potential of the envisioned attackers.

- Demonstrator 3: Similar as for Demonstrator 2. Attacking live communication requires chip-level attacks while the user pass-phrase is being entered and the PUF response is being reconstructed.

Additional research tests were done to provide information to the partners for commercialization of secure products based on HECTOR building blocks.

For Demonstrator 1 these tests were:

- Entropy verification on both TRNGs on different temperatures:
  Verification tests show that both TRNGs in Demonstrator 1 provide sufficient entropy in the raw random numbers over a large temperature range. A slight bias in the raw output streams cause the AIS20/31 statistical tests to fail in occasions. Additional post processing – foreseen in both designs – will remove such bias to make the results compliant to pass AIS20/31.
- A Common Criteria evaluation of the AIS20/31 ‘PTG3’ claim:
  The Common Criteria evaluation of the AIS20/31 claim ‘PTG3’ of the PLL-based TRNG developed by UJM shows that the design is likely to pass. Some work units could not be verified because Demonstrator 1 is not designed as a commercial product to be subjected to a real CC evaluation.

For Demonstrator 2 and 3 these research tests were:

- Physical attacks were attempted to open the enclosure of Demonstrators 2 and 3. The keyboard and display PCB of the demonstrators can be removed without tamper evidence. This may allow bugs to be inserted. Getting access to the FPGA for a chip-level attack will cause substantial damage.
- The pass-phrase retry mechanism cannot be bypassed using laser perturbation attacks. Although the protection of the assets in the demonstrator does not rely on this mechanism, this is relevant information for the HECTOR partners for future designs.
- The electro-magnetic emanation of the demonstrator keyboard and display can be used to successfully extract information on keystrokes and displayed characters respectively. This requires a side channel template attack, which consists of a training phase and a challenge (attack) phase. Although this attack is demonstrated to be feasible, in practice it is too complicated. Attackers can obtain better results using a spy-camera.

Chapter 6 List of Abbreviations

AIS	Anweisungen und Interpretationen im Schema
BGA	Ball Grid Array
BSI	Bundesamt für Sicherheit in der Informationstechnik
CC	Common Criteria
CCS	Combined Classification Success
DC	Delay-Chain
DEK	Data Encryption Key
DEMA	Differential Electro Magnetic Analysis
DPA	Differential Power Analysis
DRNG	Deterministic Random Number Generator
EEPROM	Electrically Erasable Programmable Read Only Memory
FPGA	Field Programmable Gate Array
FPGA	Field Programmable Gate Array
JTAG	Joint Test Action Group
KEK	Key Encryption Key
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LM	Light Manipulation
NDA	Non-Disclosure Agreement
NVM	Non-Volatile Memory (e.g. EEPROM or Flash)
OCCS	Overall Combined Classification Success
PCB	Printed Circuit Board
PLL	Phase Locked Loop
PUF	Physically Unclonable Function

SCA	Side Channel Analysis
SD	Secure Digital (storage card)
SEMA	Simple Electro Magnetic Analysis
SPA	Simple Power Analysis
ST	Security Target
TA	Template Attack
TRNG	True Random Number Generator
VA	Vulnerability Analysis

Chapter 7 Bibliography

[1] Contactless Electro-magnetic Active Attack on Ring Oscillator Based True Random Number Generator, Bayon, Bossouet, Aubert, Fischer. HAL archives-ouvertes, HAL Id: ujm-00699618, https://hal-ujm.archives-ouvertes.fr/ujm-00699618, May 2012

[2] HECTOR report D4.2, Demonstrator platforms accompanying report, reference ICT-644052 / D4.2 /1.0, December 2017

[3] Report on PTG.3 requirement for Hector TRNG UJM, Brightsight report 17-RPT-081 version 3.0, 24 July 2017 (confidential)

[4] J.-S. Coron: On the Security of Random Sources, Gemplus' Corporate Product R&D Division, Technical Report IT02-1998

Appendix A Temperature robustness tests on Demonstrator 1

A.1 Test summary

The following tests are performed in order to get assurance about the quality of the random numbers generated at varying temperatures. Two different generators were tested,

- The first one - called DC TRNG - was tested without post processing (Raw_DC_TRNG) and with post processing (Pp_DC_TRNG)
- The second one - called PLL TRNG - was tested without post processing (Raw_PLL_TRNG).

A.2 Test details

Table 1 shows the details of the performed experiments.

Test details	
Sample details	Hector demonstrator : Xilinx Spartan 6 (DC TRNG) Altera Cyclone V (PLL TRNG)
Evaluator	STIR
Reviewer	MWAK
Hardware	SHMOO setup
Software	Matrix v3.6.1 AIS31 test suite

Table 1: Test details.

The structure of the test is as follows:

- The temperature of the device is varied between -40 ºC and 80 ºC in steps of 10 ºC
- Once the temperature is set and stabilized, three sets of 2 MB of random data are acquired at the three TRNG outputs
- The data sets are processed by the Brightsight AIS31 statistical analysis tool, developed according to the specifications of BSI.

The AIS31 test suite includes the following statistical tests:

Test T0 *(disjointness test)*

A sequence of groups of six bytes passes the disjointness test if the subsequent members are pairwise different.

Test T1 *(monobit test)*

The bit sequence $b_1, \ldots, b_{20,000}$ passes the monobit test if X, the number of values 1, satisfies: $9{,}654 < X < 10{,}346$.

Test T2 *(poker test)*

The bit sequence $b_1, \ldots, b_{20,000}$ passes the poker test if groups of four bits do not repeat significantly more than statistically expected.

***Test T3** (run test)*

A run is a maximum sub-sequence of consecutive zeroes or ones. The bit sequence b_1, …, $b_{20,000}$ passes the run test if the number of occurring run lengths lies within the permitted intervals, as specified below. The runs of zeroes and ones are evaluated separately.

Run length	Permitted interval
1	2,267-2,733
2	1,079-1,421
3	502-748
4	233-402
5	90-223
≥ 6	90-233

***Test T4** (long run test)*

A run of length ≥ 34 is called a long run. The bit sequence $b_1,\ldots,b_{20,000}$ passes the long run test if no long run occurs.

***Test T5** (autocorrelation test)*

For $\tau \in \{1, \ldots, 5{,}000\}$, $Z\tau$ is the of number of times a bit from {1, .., 5,000} differs from the bit τ positions further. The bit sequence $b_1, \ldots, b_{20,000}$ passes the autocorrelation test (with shift τ) if $2{,}326 < Z\tau < 2{,}674$. (Note that the sub-sequence $b_{10,001}, \ldots, b_{20,000}$ is not used in the test variable.)

***Test T6** (uniform distribution test)*

The sequence $w_1, \ldots, w_n \in \{0, 1\}^k$ passes the uniform distribution test with parameters (k, n, a) if none of the $x \in \{0, 1\}^k$ occurs more than $n(2^{-k} + a)$ or less than $n(2^{-k} - a)$. Comment: for k = 1, n = 20,000 and a = 0.0173, the uniform distribution test corresponds to the monobit test T1.

***Test T7** (comparative test for multinomial distributions)*

This test checks that the occurrence of a specific value for elements of a sequence is approximately χ^2-distibuted over different samples.

***Test T8** (entropy test)*

The entropy test is performed in accordance with Coron [4]. The bit sequence $b_1,\ldots,b_{(Q+K)L}$ is segmented into non-overlapping output words $w_1,\ldots,w_{Q+K}$ of length L. A_n is the distance from w_n to its predecessor with the same value, which is used for the Coron test.

Tests T0 to T5 are applied on the internal numbers, while tests T6 to T8 are applied on the raw output data.

A.3 Test results

All AIS 31 tests were executed. The monobit test, poker test, run test, long run test, and autocorrelation test were performed 257 times, by applying them on different parts of the data collected. The disjointness test, uniform distribution test, comparative test for multinomial distributions, and entropy test were performed once on the whole data.

Table 2 shows the results of the entropy test. When the entropy is above 7.976 bits entropy per byte (meaning that the entropy of 256 bit seed is at least 255 bits), it is above the

threshold required by the standard. It can be seen that all entropy tests passed for all three TRNG output streams for all temperatures.

The entropy value above eight can be explained by rounding errors in the calculation (summation of p.log(p) many times). In fact the real entropy values are indistinguishably close to eight in the precision used.

Temperature	Raw_DC_TRNG	Pp_DC_TRNG	Raw_PLL_TRNG
-40 ºC	7.9955741691613	8.00375236696662	7.99988913136029
-30 ºC	8.00138732168935	8.002850615391	7.99741875756392
-20 ºC	7.99527355588504	8.00113067669824	7.99172237514431
-10 ºC	7.99527355588504	8.00106280444145	7.99565238029673
0 ºC	7.99826560662571	7.99752439254334	8.00063637536556
10 ºC	7.99506689742722	7.99794764915984	7.99648417893198
20 ºC	7.9929556472311	7.99784943803931	8.00056649160738
30 ºC	7.99173964387156	8.00251816823819	7.99469349355319
40 ºC	7.99417876130061	7.99854314926949	8.00068235333365
50 ºC	7.9823122108646	8.0033354846321	7.99927082734219
60 ºC	7.99911989431192	7.99818322409914	7.99898011060039
70 ºC	7.98977813518	7.99234891795867	7.99455525085946
80 ºC	7.99052874776427	8.00033414943942	8.0017179377577

Table 2: Entropy test results at different temperatures.

Table 3 shows the tests that failed for each output stream and at which temperatures. It can be observed that, in case of the post-processed DC TRNG output, all tests passed. For the raw outputs of the two TRNGs it can be seen that some monobit tests fail. However, the bias is rather minor and quite stable and the number of ones or zeros does not exceed much the allowed interval. When the difference between the number of zeros and ones gets higher, other tests are also being impacted. When this is the case, the poker and run tests start failing also.

When the monobit test fails on the RAW PLL TRNG, it is always due to the number of ones being higher than the allowed maximum. The number of zeros and ones should be between 9,654 and 10,346 in order to pass the test. When the test fails, the number of ones is above 10,346. On the acquired data the highest number of ones for any of the failing tests was ~10,500. Therefore, on the acquired data, the number of ones was always below 101.5% of the allowed maximum, which can be considered as a small bias.

When the monobit test fails on the RAW DC TRNG, it is most of the time because of the number of ones being lower than the allowed minimum (9,654). On the acquired data, the lowest number of ones for failing tests was ~9,000. Therefore, on the acquired data, the number of ones was always above 93.2% of the allowed minimum.

Temperature	Raw_DC_TRNG	Pp_DC_TRNG	Raw_PLL_TRNG
-40 ºC	3 Monobit tests	Pass	5 Monobit tests 1 Poker test
-30 ºC	7 Monobit test 1 Poker test	Pass	53 Monobit tests 4 Poker tests 1 Run test
-20 ºC	1 Monobit test	Pass	118 Monobit tests 20 Poker tests 1 Run test
-10 ºC	1 Monobit test	Pass	57 Monobit tests 6 Poker test
0 ºC	3 Monobit tests 1 Poker test	Pass	Pass
10 ºC	5 Monobit tests 1 Poker test	Pass	Pass
20 ºC	Pass	Pass	Pass
30 ºC	31 Monobit tests 7 Poker tests 1 Run test	Pass	Pass
40 ºC	30 Monobit tests 21 Poker tests 8 Run tests	Pass	Pass
50 ºC	80 Monobit tests 75 Poker tests 25 Run tests	Pass	1 Monobit test
60 ºC	Pass	Pass	Pass
70 ºC	58 Monobit tests 36 Poker tests 8 Run tests	Pass	Pass
80 ºC	1 Monobit test	Pass	Pass

Table 3: Test failure for different temperatures (number of failures out of the 257 tests).

For the DC TRNG no obvious correlation can be found between the monobit test failures and temperature.

For the PLL TRNG, Table 3 seems to indicate that more failures occur at low temperatures. In order to get assurance, the acquisition of data was repeated for the PLL TRNG, and the tests were launched again. Table 4 shows the results for the second acquisition. It can be observed that the number of failures can vary a lot even in the same conditions (same temperature). The second acquisition confirms that more failures occur at low temperatures.

Temperature	Raw_PLL_TRNG
-40 ºC	50 Monobit tests 3 Poker tests 2 Run tests
-30 ºC	2 Monobit tests
-20 ºC	16 Monobit tests
-10 ºC	139 Monobit tests 32 Poker tests 1 Run test
0 ºC	179 Monobit tests 35 Poker tests 1 Run test
10 ºC	4 Monobit tests
20 ºC	Pass
30 ºC	Pass
40 ºC	Pass
50 ºC	4 Monobit tests
60 ºC	21 Monobit tests 1 Poker test
70 ºC	Pass
80 ºC	Pass

Table 4: Test failure on the second set of data for the PLL TRNG at different temperatures (number of failures out of the 257 tests).

A.4 Test conclusion

Tests were performed changing the operation temperature of the two HECTOR TRNG designs of Demonstrator 1 in order to get assurance on the quality of the random numbers generated over a large temperature range. The results show that for the DC TRNG no obvious impact of the temperature on the quality of the random numbers could be observed. For the PLL TRNG, more failures could be observed at low temperatures.

Failures of the tests on the raw data of the two TRNGs could be observed; some monobit, poker and run tests did not pass, making the whole test not pass. This is caused by a small bias in the raw output streams.

Despite the small bias, it is to notice that the quality of the raw random data is high, as the obtained entropy values are high.

For a formal certification all AIS20/31 tests must pass over the temperature range as claimed in the Security Target. In practice a commercial product will be further enhanced by trimming of parameters related to the hardware platform and use-cases. A TOE commonly has a narrower temperature range than applied in the presented tests (especially excluding low temperatures). The small bias is therefore no significant issue in the demonstrator design.

Appendix B Physical analysis of Demonstrator 2 & 3

B.1 Introduction

Demonstrators 2 and 3 share the same physical design. Although the vulnerability analysis does not show particular attack scenarios that rely on physical attacks to the enclosure, it makes sense to investigate the resistance of the enclosure against physical attacks for attack scenarios that go beyond the foreseen protection level of the demonstrators. In particular attackers that deliberately target a device and have beyond high resources could devise an attack that aims to capture and modify a device in such way that user secrets – such as the pass-phrase and PUF response – become exposed. This section describes the physical analysis.

The physical design of Demonstrators 2 and 3 is such that it can serve as a basis for commercial products. It is intended to have reasonable physical protection of the internal assets, but the HECTOR project is not about development of secure hardware. For this reason the demonstrators provide tamper evidence and tamper resistance, but not tamper responsiveness. This means that it is made difficult to disassemble the devices (tamper resistance) without causing damage (tamper evidence), but that there are no active circuits that monitor the physical status of the device (tamper responsiveness).

B.2 Attack scenario

The strength of both Demonstrator 2 and 3 comes from the fact that the secrets that are used to encrypt the user data – respectively protect the message communication – are not present in the devices in rest: The PUF response is generated at power-up and the pass-phrase is entered by the user after each start-up. An attacker needs to obtain both in order to be able to reconstruct the stored information.

Since the PUF generates its output response when the device is powered-on, the PUF response is physically present inside the device at a certain moment in time. Also the pass-phrase – typed-in by the user – will be present only when the device is actually operated. So, because both data elements are available only when the device is actually operated by the legitimate user, the attacker has the challenge to collect this information while the device is in the hands of the user. Practically this could be done by using bugs that are hidden inside the device and store the captured data or transmit it to the adversary (comparable to a key logger). It is clear that the user should not be alarmed by any obvious damage to the device by the installation of the bug, nor shall the user-experience with the device be affected in a noticeable manner. Such damage or changed user-experience is called *tamper evidence*.

Two attack scenarios can be discerned:

1. Without notice of the legitimate user, the attacker steals the device, modifies it by installing a key-logger bug for user pass-phrase collection and returns it to the user. The user will operate the device by typing-in the pass-phrase, which is then captured and stored inside the device. The attacker then steals the device for a second time and then applies the obtained pass-phrase on the device (and its PUF secret) to decrypt the content of the SD card.
2. The attacker steals the device, obtains device internal data (encrypted data storage key, the actual encrypted data on the SD card, helper data and the PUF response) and then installs a key-logger bug to capture and transmit the user pass-phrase over a radio link. The attacker then uses this pass-phrase – together with the previously collected PUF response and encrypted Data Encryption Key to decrypt the stored data from the SD card.

Case 1 is fairly straightforward and does not require special knowledge or tools. It relies on getting the device in possession for modification without the legitimate user noticing it. When

assuming that a storage device (Demonstrator 2) or communication device (Demonstrator 3) is regularly used, then the time available between 'interventions' for installing any bug is short.

In case 2 the attacker needs more resources, because obtaining PUF information from a 'live' device requires chip-level attacks. These are significantly more complicated than e.g. installing a key-logging bug. The advantage is that the device does not need to be stolen for a second time. To avoid chip-level attacks, alternatively the attacker could try to obtain the PUF key by brute forcing. This requires exploration of an attack space of 2^{32}, which translates to ~25,000 hours on average at an optimistic test rate of one second per attempt.

Scenarios like this are not common and will usually only be applicable for 'high-value targets', such as use by top-government officials or the military. Stealing and returning of such device requires nearby access to the victim and thus a high degree of coordination by the adversaries. This could be in range of e.g. national secret services (three-letter agencies) with attack potential 'Beyond high'). The design of the demonstrators is not developed to withstand attacks to that level.

The security properties of the FPGA are not taken into consideration, because this goes beyond the purpose of HECTOR as a whole and of the demonstrators specifically. It is assumed that the FPGAs are protected and that re-programming is blocked.

B.3 Construction of the demonstrators

The enclosure of both Demonstrators 2 and 3 is completely sealed. It is not meant to be serviceable, so there are no possibilities to disassemble it without damage. The demonstrators are composed of three main components, which are the keyboard assembly, the main PCB and the aluminium case. Referring to Figure 4, the keyboard assembly consists of a stack of Keyboard label, the capacitive keyboard, the adhesive foil and the interlayer. The main PCB is referred to as 'Printed circuit board'.

The keyboard assembly is attached to the main PCB by another layer of 3M double-side sticking film that is cut to the exact contours of the main PCB. Openings are cut-out for the display, the LED and interconnections. After sticking the two parts together, the seven electrical connections of from keyboard assembly and main PCB are soldered. This way the keyboard and main PCB cannot be separated without cutting or de-soldering the interconnections. The individual components are shown in Figure 4.

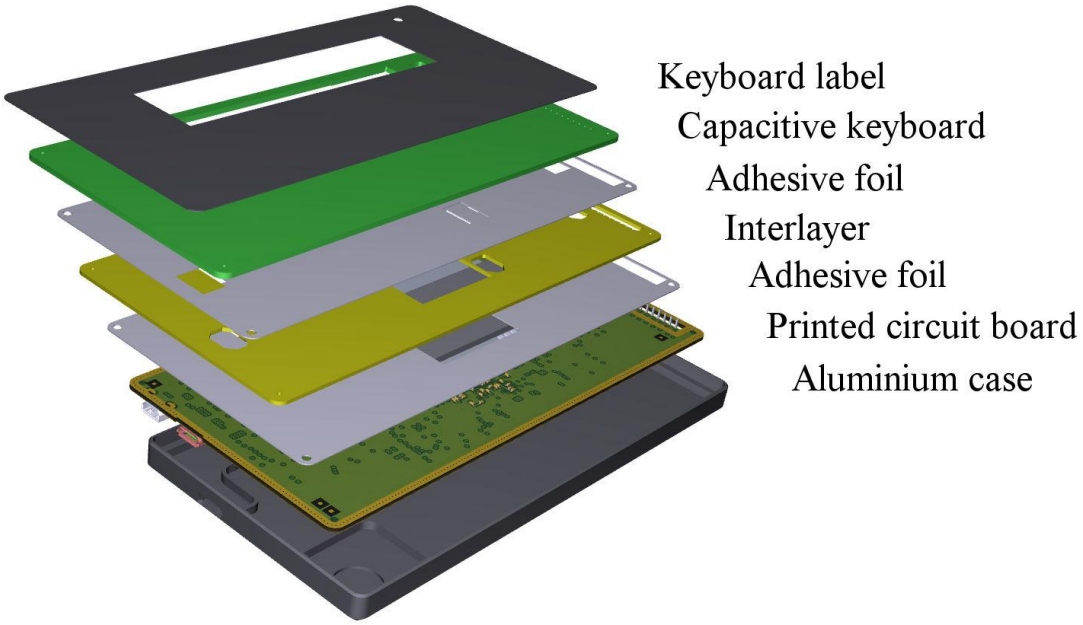

Figure 4: The main PCB, The keyboard assembly and the double-side sticky foil.

The stack of keyboard and main PCB is mounted inside the aluminium case using epoxy potting compound. During manufacturing a well-defined amount of liquid epoxy is poured inside the aluminium case, after which the stacked combination of keyboard assembly and main PCB is inserted. When the resin is cured, all parts are fixed together to the back cover and form a single rigid assembly.

The demonstrator includes a mechanical vibration motor and needs external openings for the SD card and USB connector. Therefore not all surface area of the aluminium case should be filled with liquid epoxy. ‘Walled zones’ are implemented to prevent the epoxy from flowing into these restricted areas. The ‘walls’ separating the zones are clearly visible in Figure 5.

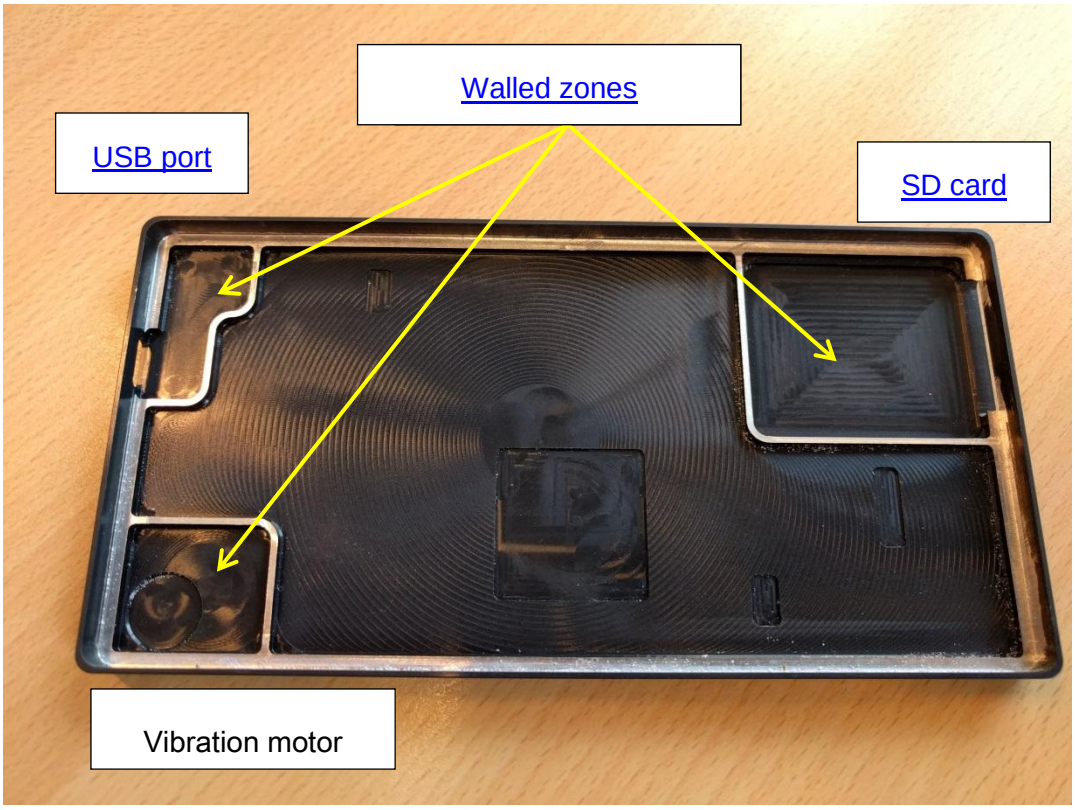

Figure 5: Internal 'walled zones' inside the demonstrator back cover.

B.4 Removal of the keyboard PCB

To get access to the internals of the demonstrators the most obvious way is to disassemble the parts in the reverse order as they were assembled. This means that the attacker may attempt to 'lift-out' one or more PCBs from the aluminium case.

A challenge for the attacker is to get physical grip of the PCB stack in such way to pull it out of the case, without damaging both PCB stack and case. Quite some force is needed because the epoxy bond is very strong. In addition, because the surface area of the bond is large, a vertical pull-off is not feasible. In practice an attacker will start pulling by first creating room in a corner or edge and then gradually 'tear-off' the PCB assembly from the aluminium case.

A first practical experiment to disassemble the device was done by searching for possibilities to put force on the PCB stack with respect to the aluminium back cover without leaving traces. A possibility was found by using the opening of the SD card at the right sidewall. A metal part can be inserted and used as a lever to push the PCB stack upwards inside the case. The area of the SD card is not filled with epoxy resin, which is the reason why this can be done. Gentle force will bend the main PCB – with keyboard assembly on top – until the latter is pushed above the edges of the enclosure (Figure 6).

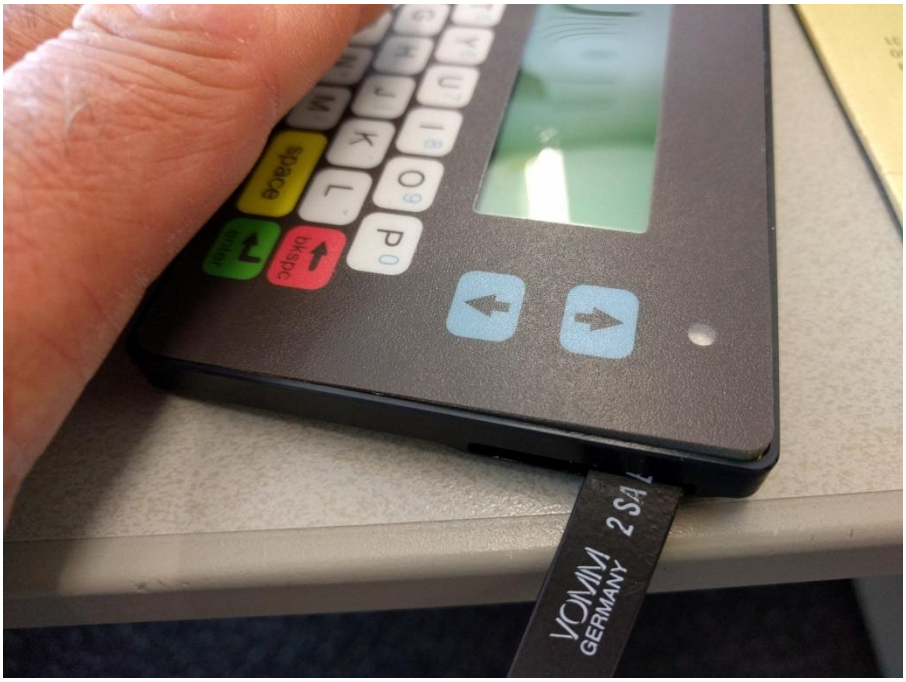

Figure 6: Lifting the PCB until it is above the edge of the aluminium enclosure.

Then – by inserting sharp objects – the keyboard assembly can be prevented from 'bending back' (Figure 7).

Figure 7: Prevent the keyboard PCB from bending back.

As a next step the complete keyboard assembly can be worked-loose from the main PCB and case by moving the knife around the enclosure in a counter-clock direction. When the keyboard PCB assembly is loosened in this manner at the top, left and bottom edges, it can be lifted upward at the left side.

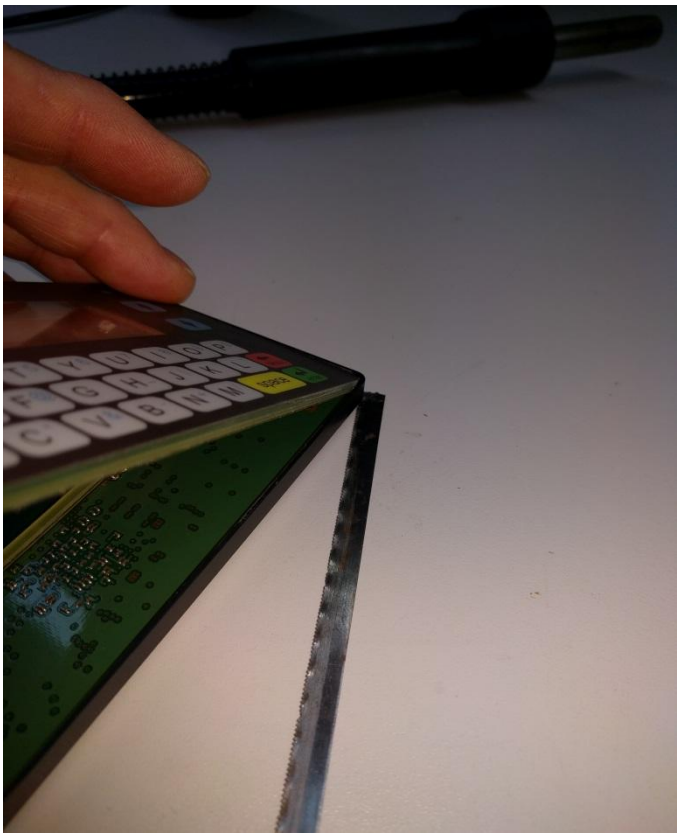

Figure 8: Keyboard PCB assembly lifted at left side.

The right side of the keyboard PCB assembly will remain stuck, because of the interface wires by which it is soldered to the main PCB. Through the narrow opening between main PCB and keyboard PCB assembly the interface wires can be cut using a sharp knife or a thin blade saw. This will separate the keyboard assembly (Figure 9).

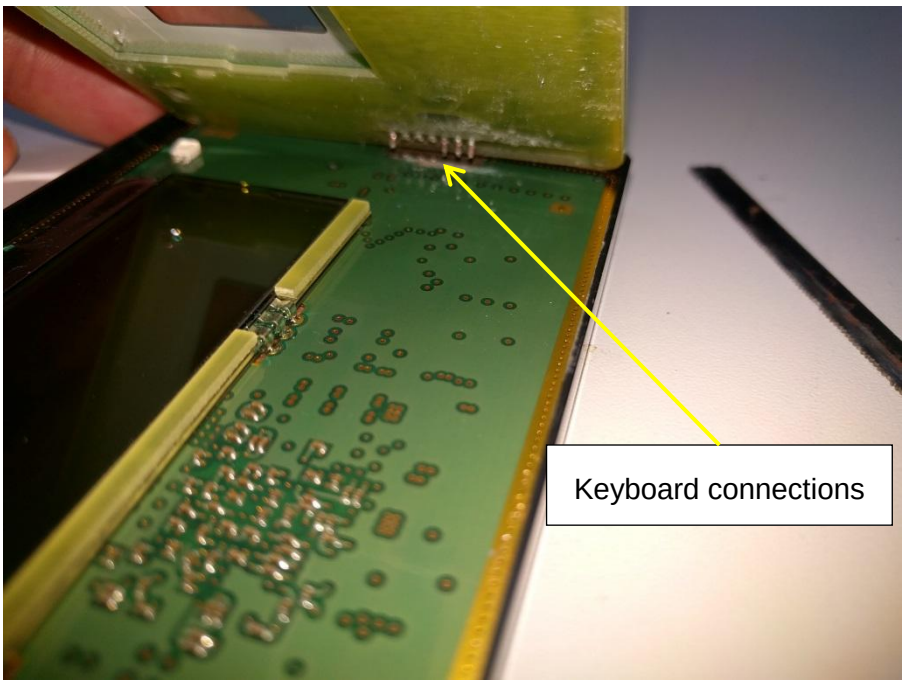

Figure 9: Cutting the keyboard interface wires using a blade saw.

The now separated keyboard assembly can be re-connected by wires with the main PCB and is still operational, without any damage to the demonstrator. The opened and operational version of the demonstrator can be seen in Figure 10.

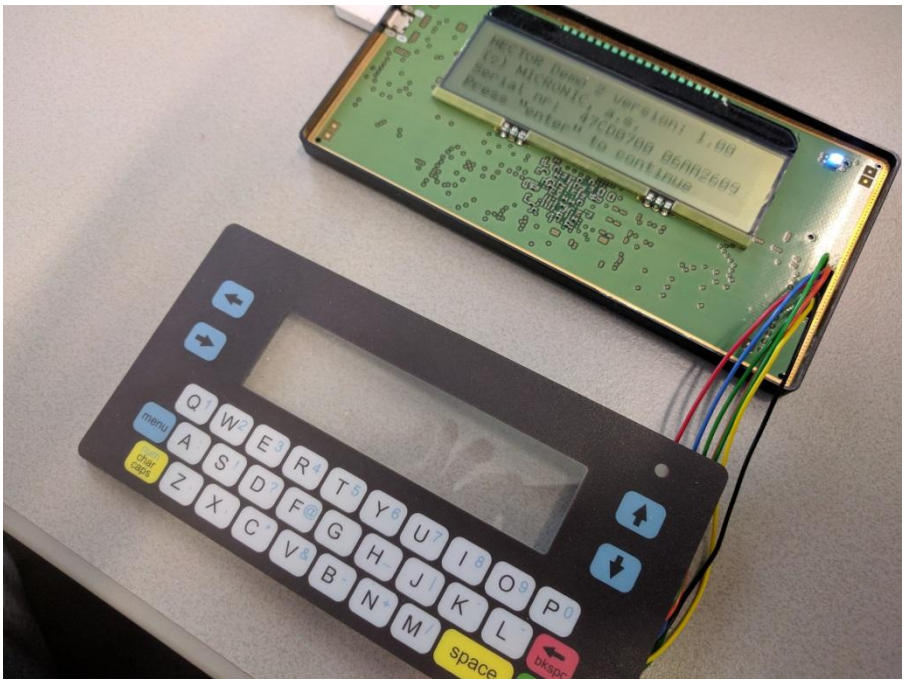

Figure 10: Interface wires restored. Device is opened, fully operational and undamaged.

At this point an attacker might be able to add e.g. a keyboard or display tapping bug. Both the keyboard and display interfaces are directly accessible and carry the plaintext pass-phrase.

Once the keyboard is separated, also the top layer of the main PCB is accessible. It could be interesting for an attacker to access e.g. the JTAG test interface of the FPGA. However, analysis of the schematics and the PCB layout showed that no interesting nodes (such as e.g. JTAG -select at FPGA pin AA22) can be accessed. The four-layer PCB applies buried vias and is well-designed from a security point of view.

B.5 Separation of main PCB and enclosure

With the keyboard removed the attacker could attempt to also remove the main PCB from the case. This requires breaking the epoxy bond between the main PCB and the aluminium and pulling it out. It is known that epoxy resin loses much of its strength at elevated temperatures. An experiment was started to heat the back of the aluminium enclosure to ~150 ºC in order to weaken the bond between epoxy and aluminium. The demonstrator enclosure was therefore placed on a pre-heated hot plate. Aluminium is a very good heat conductor, so the heat is quickly transferred to the epoxy resin. This works well without damaging the electronics because the epoxy serves as a barrier for the most temperature-sensitive components like the Liquid Crystal Display. Obviously the temperature of the electronics should not become too high. The setup for the heating experiments can be seen in Figure 11.

Figure 11: Setup for heating experiments: demonstrator on top of hot plate.

As was the case with removing the keyboard PCB from the enclosure, it is also difficult to remove the main PCB because there is no ‘grip’ that allows an attacker to exert the required amount of force. The fit of the PCB in the enclosure is tight, which means that no mechanical tools can be used to wedge in between the PCB and the aluminium side walls.

Using the same ‘lever’ technique (see Figure 6) to lift the main PCB out of the aluminium case is more prone to failure, since it has the LCD directly soldered on top. This all-glass module is easily damaged or even broken when bending the main PCB in the long direction, starting from the SD-slot. This limits the amount of force that the attacker may assert to the PCB. In case the LCD gets damaged beyond repair, an attacker could choose to replace it with a similar off-the-shelve component. If done carefully, this will not be noticeable by the legitimate user.

The first experiments with a hot-plate temperature around 150 ºC failed, because the bond remained too strong to resist the force of the lever. Then the hot plate temperature was increased to 300 ºC. This temperature can easily damage components and especially the LCD, so exposure time must be as short as possible. The demonstrator was placed briefly on top of the hot plate until some smoke developed. At this point the LCD starts getting black. Exerting force with the lever now releases the main PCB from the aluminium case. The result can be seen in Figure 12.

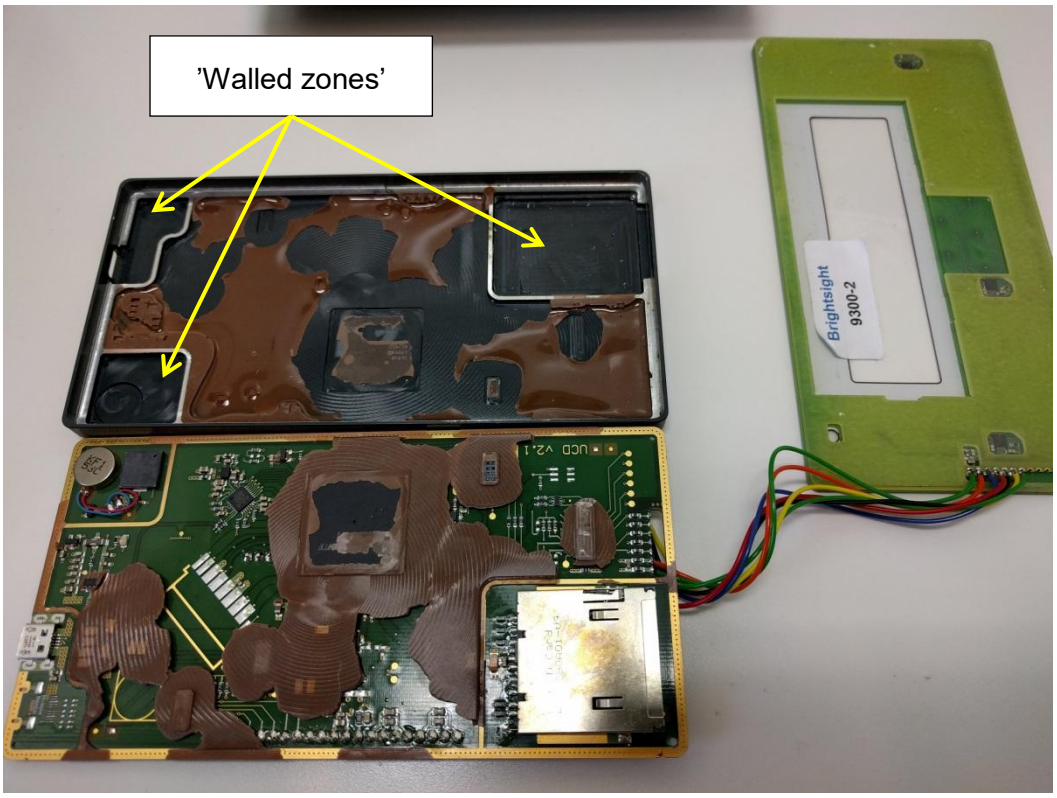

Figure 12: Main PCB separated from the aluminium enclosure by heat.

Figure 13 shows close-ups of the remaining epoxy resin on PCB and case. Clearly visible are the 'walled sections', which do not contain epoxy resin by design. It can also be seen that the remaining volume between the main PCB and the aluminium case appears to be incompletely filled with resin during manufacturing. As visible in Figure 12, approximately half of the surface area is actually filled with resin from 'PCB-to-aluminium'. The resulting voids weaken the bond between the two parts, which contributes to the fairly easy separation.

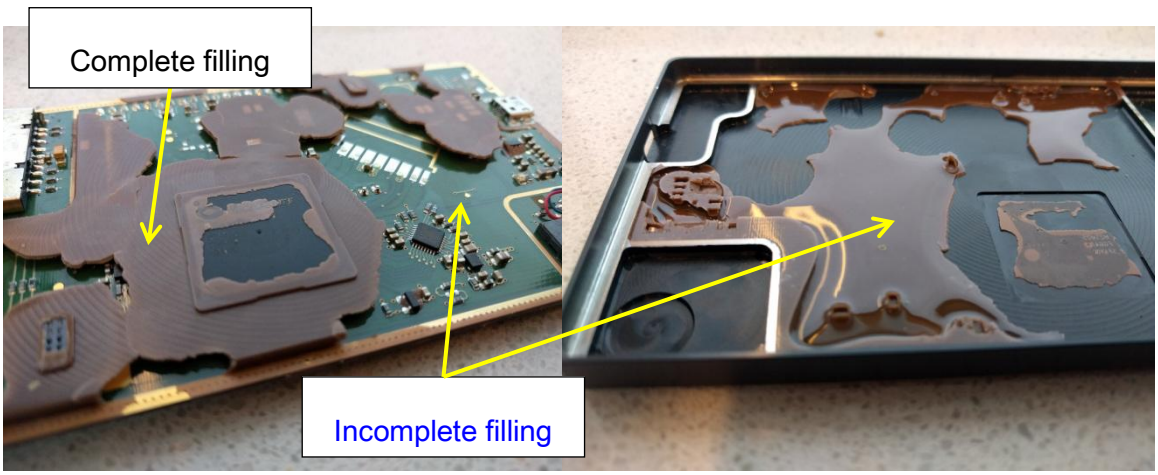

Figure 13: Details of voids in the epoxy resin.

It is expected however that the two parts can be separated even when the full volume would be filled perfectly with epoxy resin. As said, most epoxies weaken severely when heated, which would allow the parts to be separated anyway. This is visible in the sections where filling was reasonably fine, such as around the FPGA: Despite the complete filling, the mechanical bond between the aluminium and epoxy was broken by slight force. This is expected, because the aluminium-epoxy bond heats-up the fastest. Separation is also

helped by the fact that the main PCB is lifted from the short edge. This concentrates the pulling force to a relatively short separation front, travelling from right to left.

The electronics do not sustain permanent damage due to the heat resistance of the resin, the air voids and the short heating period. After the heating experiment the demonstrator remained fully operational, despite the high heating temperature. Although the LCD display turned black during heating it returned to its original state and still worked fine when cooled down. This is visible in Figure 14.

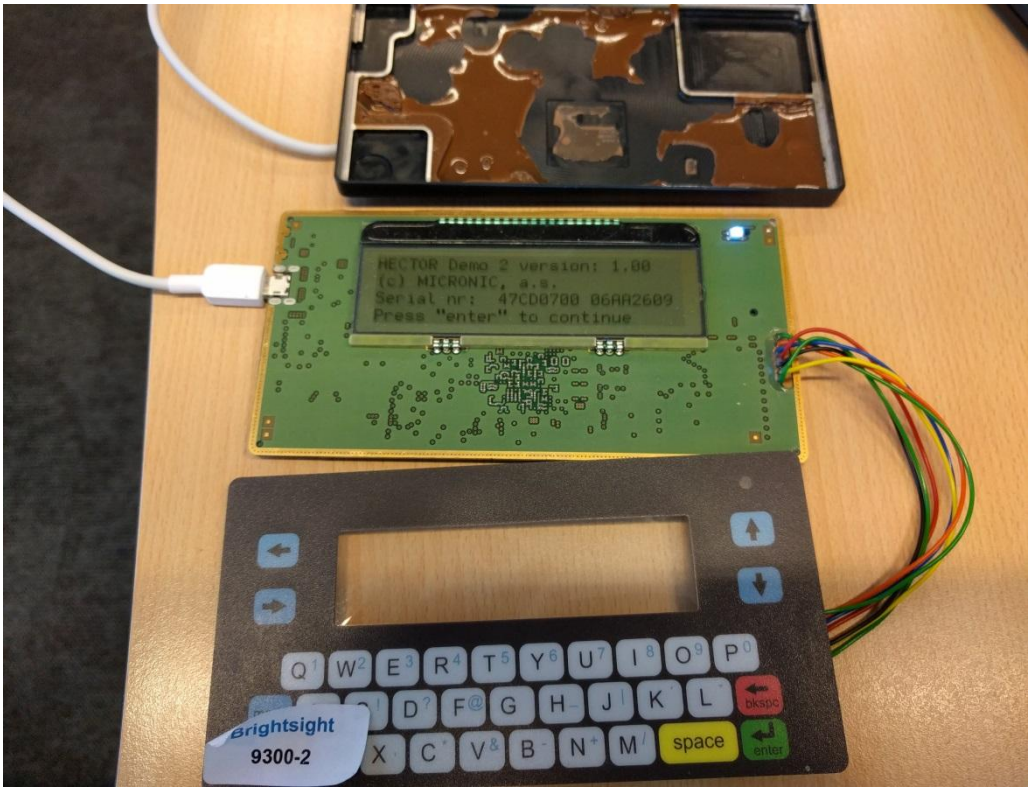

Figure 14: Demonstrator hardware operational after disassembly.

In case an attacker aims to attack the hardware, he/she can remove remaining epoxy fragments from the main PCB by using a hot air gun (such as used to repair PCBs). These can be used to heat the resin locally so it can be removed bit by bit by mechanical tools.

Physical attacks like this are difficult to prevent for devices that are powered-down in rest. More complicated protection concepts can be applied using permanently active monitoring of the device's environment. This allows for active erasure of sensitive information when a (potential) attack is detected. This is called 'Tamper responsiveness' and requires an energy source (battery) to power the monitoring and erasure circuitry. The sensors, detectors and built-in batteries have a negative impact on the reliability of a product. Finally it is up to the manufacturer to find the right balance between protection measures and functionality that fits the intended purpose of the product.

As said, the HECTOR project was not about designing tamper responsive hardware. The experiments delivered useful knowledge for the HECTOR partners for future design and manufacturing of secure hardware.

B.6 Conclusions

It was possible to remove the keyboard PCB without tamper evidence, which could allow an attacker to insert bugs inside the enclosure for tapping display or keyboard data. With careful work this can be done without causing tamper evidence. With keyboard PCB removed there

are no other interesting nodes – besides keyboard and display connections – that an attacker can exploit.

Further disassembly showed that – when using temperatures around 300 ºC – that the epoxy bond between the main PCB and the aluminium case can be broken by mechanical force. With careful work this can be done without causing further tamper evidence. This allows attackers to remove or add components and is a pre-requisite for a chip-level attack.

Appendix C Attacking Demonstrator 2 and 3 keyboard entry

C.1 Introduction

As concluded from the vulnerability analyses the security of both Demonstrators 2 and 3 hinge mainly on the secrecy of the pass-phrase. It is therefore crucial that information as typed-in by the legitimate user does not become exposed to adversaries by any means. In the previous section physical attacks are described that could insert a bug that collects pass-phrase information. Such attacks require direct physical access inside the device, which makes them difficult to exercise in practice. Side channel attacks however might be done without intrusion; they may require preparation of the user-environment in such way that side channel signals can be obtained without raising suspicion by the legitimate user.

C.2 Test sample

Testing the keyboard properties on a real device is difficult because it requires repetitive and automatic entry of single digits. Therefore a dedicated FPGA design is developed by MICRONIC – based on specifications of Brightsight – providing interfaces to the keystroke-entry mechanism that allows for automated testing. The test function is implemented on actual Demonstrator 2/3 hardware in order to include the influence of the hardware design and enclosure. The hardware of the demonstrators is modified to provide trigger output signals to make testing more efficient.

The test sample only handles the keyboard entry; the characters that are sent to the test target are not displayed.

For the penetration testing using DEMA the USB cable was modified. With this modification is possible to power the TOE automatically off and on again.

The keyboard is of the capacitive scanning type. Dedicated keyboard controller ICs are used to sense changes in the environment of the keys by human fingers and transfer those into characters. Due to the operating principle of capacitive keyboards it is not possible to use any object to ‘press’ a key; it should have similar characteristics as a human finger.

The keyboard data is then sent over a serial interface to the FPGA for further processing.

C.3 Test description

This test aims to obtain side channel information on which key is pressed by the legitimate user, in order to obtain the pass-phrase.

The two candidates for side channel signals are:

- Power consumption of the device
- Its electro-magnetic radiation.

Electro-magnetic radiation is the most likely candidate for a successful attack in terms of signal content (Signal-to-Noise) and practical applicability in the given demonstrator use-cases.

The tests are done by tapping electro-magnetic information from the front-side of the demonstrators. Tapping electro-magnetic signals at the back of the demonstrator is much less effective due to shielding by the aluminium enclosure.

The actual test is done by collecting ‘training’ traces while known keyboard keys are pressed. These training traces will be used to compare with ‘challenge’ traces of unknown key-presses. An attack is successful when an attacker is able to reduce the entropy of the protection keys to a level where a brute force attack becomes feasible.

In practice an attacker would need to measure electro-magnetic information in an inconspicuous way, thus not at the front. Nor would he/she possess an attack target with additional trigger outputs. In practical settings such attack will be much harder to perform. Therefore the presented tests are 'worst-case' situations, just to demonstrate the operating principles and to get a feeling of the feasibility of attacks.

C.4 Test details and test results

The following table shows the details of the performed experiments. Detailed descriptions about the measurement set-up and related components can be found in Appendix F

Test details	
Evaluator	JLUH, JPAP
Reviewer	JGAL
Hardware	SCA5
Software	Matrix v3.6.1 Sideways v3.21/22 Template Attack v1.4/1.5
Equipment parameters	Vcc = 5 V Sampling rate = 1 GS/s (TA) / 5 GS/s (DPA)

Table 5: Test details.

In this test the emanation of the keyboard was investigated. A manual scan was done using an electro-magnetic pick-up coil over the whole keyboard surface, while random keys were pressed. It was possible to observe a signal that appeared only after a key was pressed (and again after a key was released). An example can be seen in Figure 15. The signal was found roughly on top of one of the three microcontrollers, which are handling the keyboard. Also this signal appears close to the USB connector. To exclude the possibility that this signal refers to the send data via the USB cable, the coil was placed directly above the USB connector. It was not possible to observe a similar signal. Additional the coil was placed directly above the positions of the other two microcontrollers, which are marked by the red circles in Figure 16. Here the signal could be observed again. So it is most likely that this signal refers to the activity of the microcontroller.

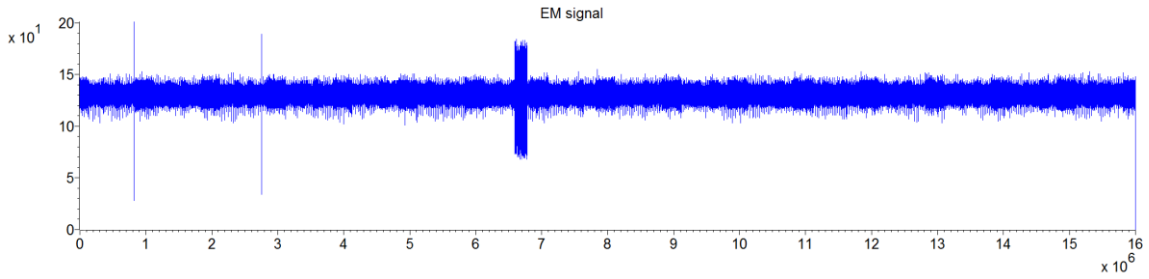

Figure 15: Observed pattern after a key of the keyboard was pressed.

A set of 1,000 traces at a sampling rate of 1GS/s was acquired, while random keys were pressed. The corresponding characters of the keys were stored as ASCII-encoded values together with the traces.

Three locations were found with high signal strength, which all correspond to presence of a capacitive keyboard controller IC. The coil position that provided the highest signal strength for this measurement can be seen near the 'Q' in Figure 16. The red circles mark the other two positions where a similar signal could be observed.

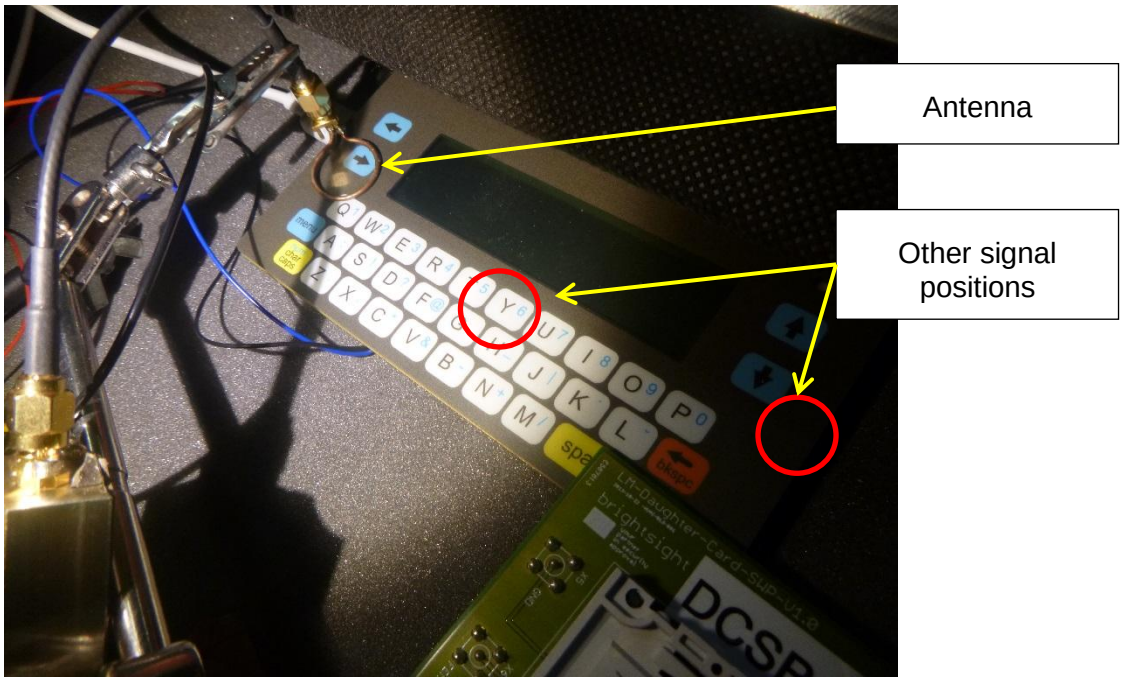

Figure 16: Final coil position for the measurement on the keyboard.

Due to practical reasons the acquired traces were aligned at the end of the observed pattern. The point of alignment is indicated by the arrow in Figure 17. A successful correlation analyses was performed on the traces. The correlation was calculated for all eight bits of the stored ASCII encoded value. The results can be seen in Figure 17.

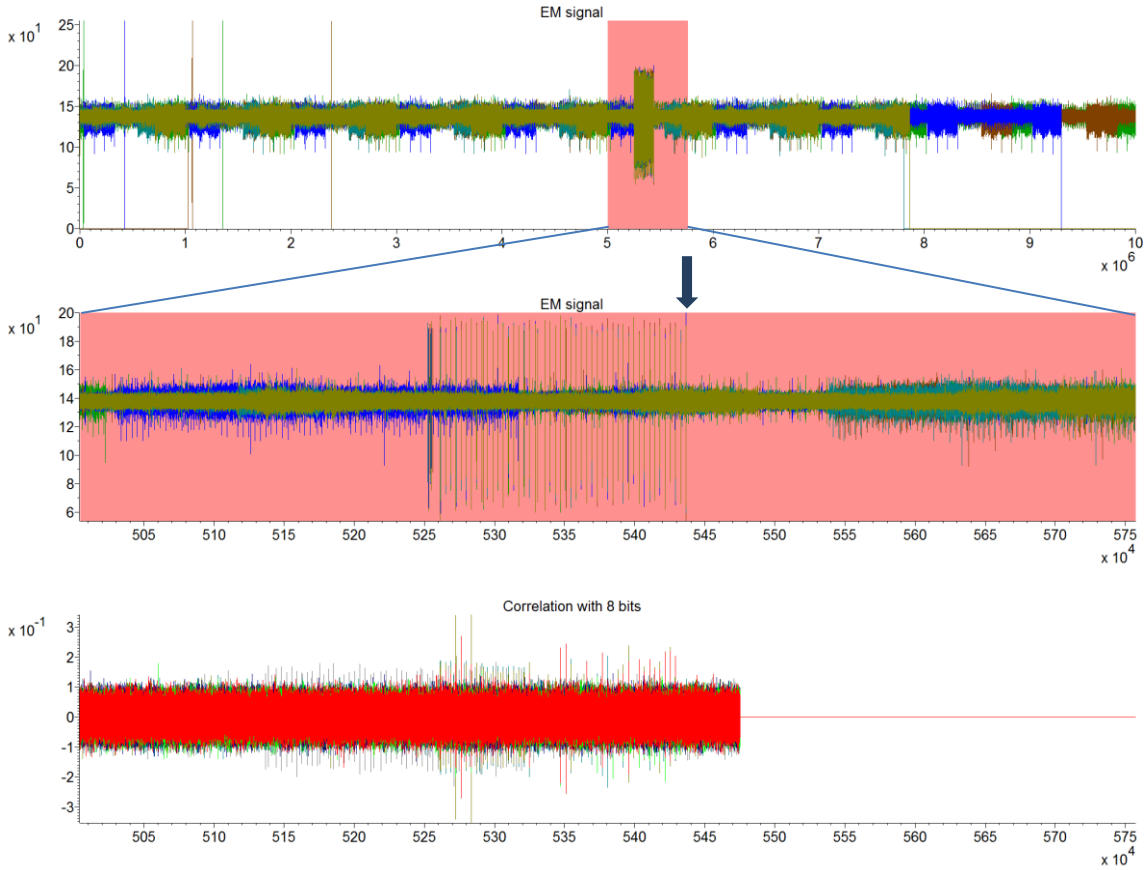

Figure 17: Five overlaid and aligned traces can be seen in the top. The middle trace shows a zoomed in view of the pattern. The arrow indicates the point of alignment. The bottom trace shows the correlation results for all eight bits.

Because of the presence of this correlation it was decided to perform a template attack to attempt to recover the value of a pressed key. To make this test feasible a pneumatic finger was used with a pen designed for touch screens. A major drawback of this setup is that the order of pressed keys cannot be randomized. A picture of the setup can be seen in Figure 19.

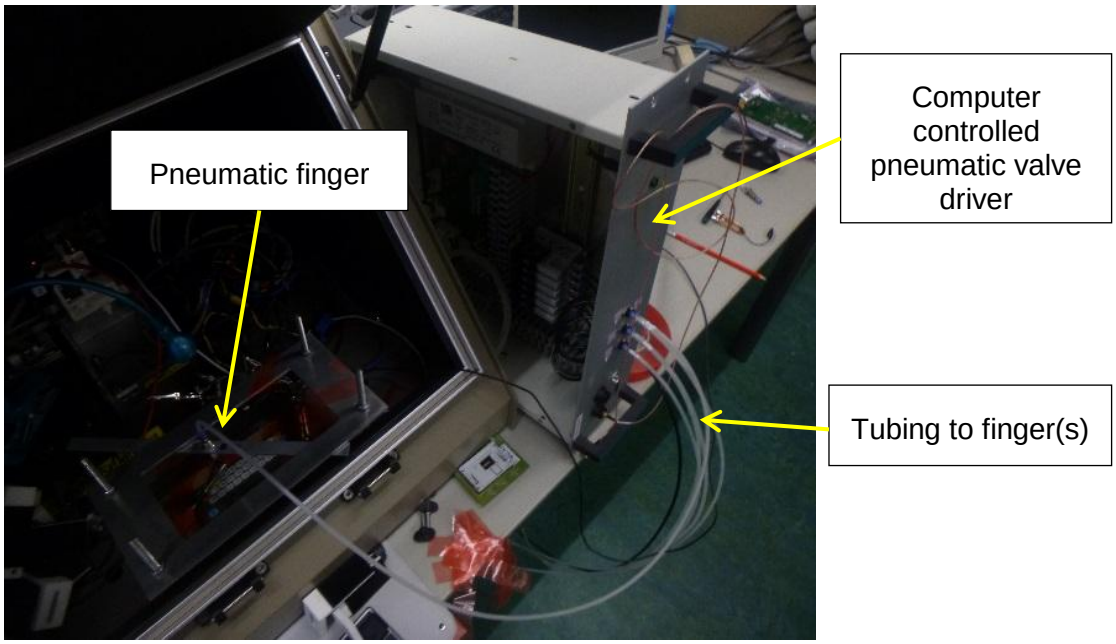

Figure 18: Pneumatic test setup for automatic key presses.

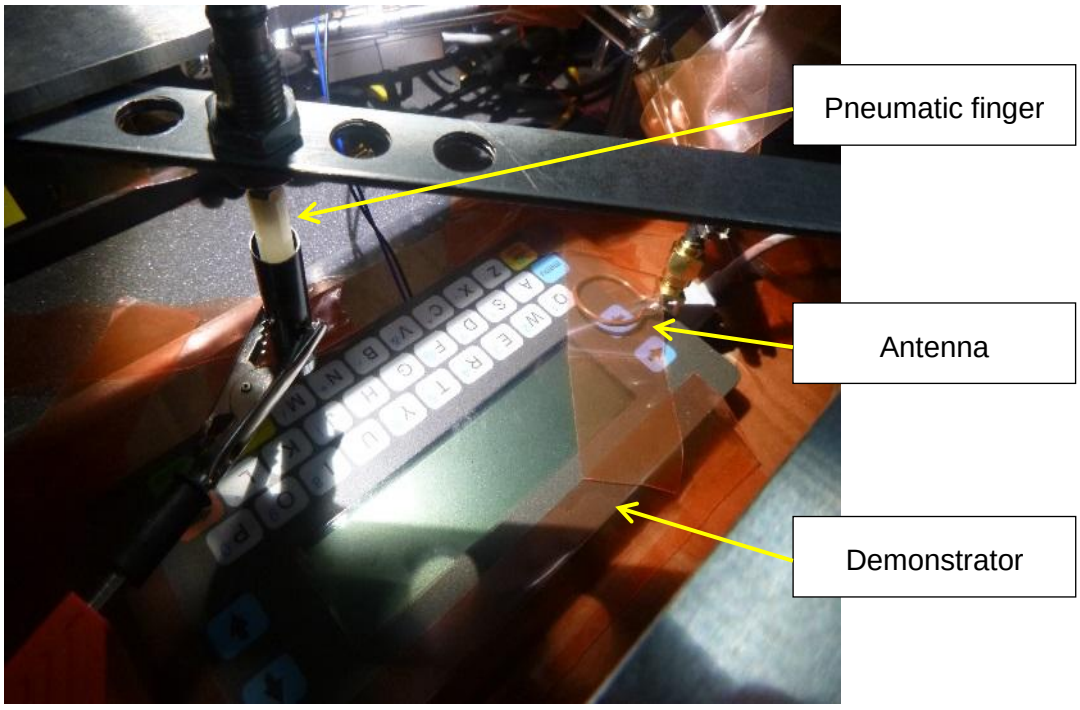

Figure 19: Test setup for acquiring traces on the keyboard.

A set of 57,500 traces at a sampling rate of 1 GS/s was acquired. The main goal of the attack was to find out if different keys can be distinguished by building templates. Therefore the number of investigated keys should not matter. For practical reasons the attack targets 25 different keys. This contains all ‘real’ characters, except for the “Q”, which was too close to the coil. Also the special characters were not included. Nevertheless it should be mentioned that all key presses could be recognized in the electro-magnetic traces, including ‘space’, ‘backspace’, ‘enter’, ‘caps’ and ‘menu’. There is no fundamental difference in operation between ‘real’ keys and special characters.

For each key 2,300 traces were acquired, of which 60% was used for training and 40% used as challenge. The traces were merged and aligned together afterwards.

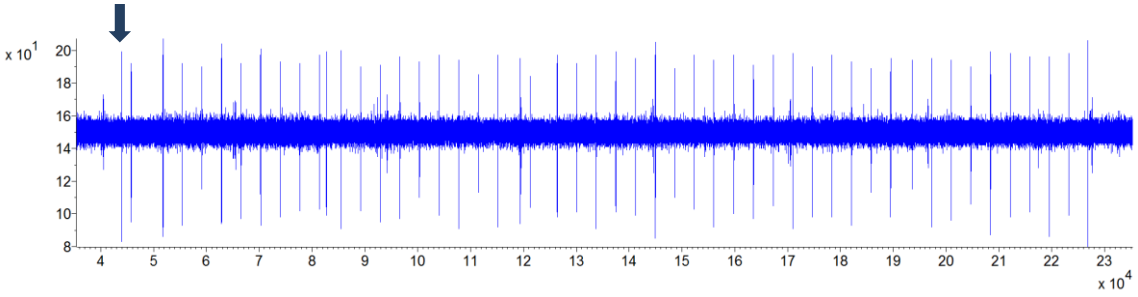

Figure 20: Example of a trace used for the template attack. The arrow indicates the peak used for the alignment that gave the best results.

Figure 21 shows the best template attack results that were obtained with the alignment on the peak indicated by the arrow in Figure 20. For this graph a single covariance matrix was used while the minimum distance between points of interest equals two. The results lead to a maximum success rate of 1.0. This corresponds to the recovery of 25 classes out of 25. This means that all key presses can be recognized from their EM traces after the analysis tool is sufficiently trained.

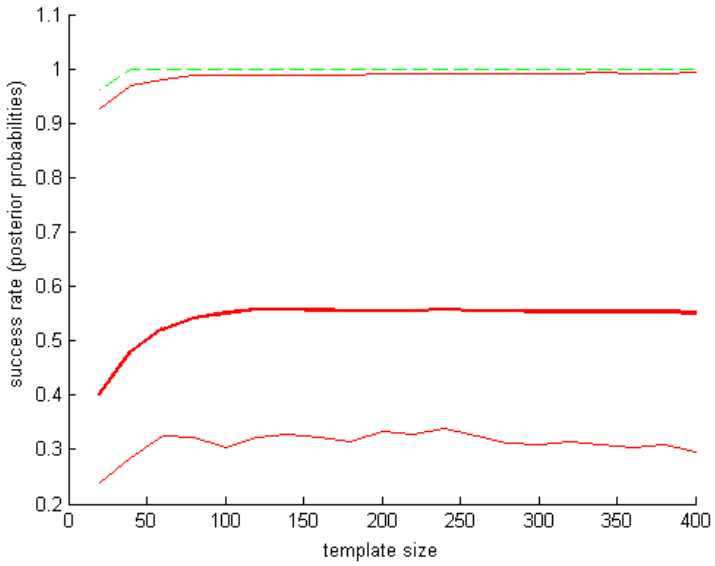

Figure 21: Success rate of the template attack on EM traces of keyboard emanation on 25 values as a function of the template size.

It should be mentioned that the missing randomisation during the acquisition could lead to a false positive result. It is possible that the acquired EM signal is influenced by external sources during the acquisition. If these sources have different influence on the signal from one acquisition run to another, it is possible that only these differences are matched in the attack. Nevertheless data dependency could be observed in the signal. So it is likely that the recovery of the classes not only depends on external influences on the signal. Maybe a larger set is necessary in practice, which will make the attack more difficult.

C.5 Test conclusion on keyboard emanation

This test was intended to determine if the demonstrator platform leaks information through electro-magnetic side channels while interacting with the legitimate user through the keyboard interface.

It was possible to identify data-dependency in the EM signal related to the keys pressed.

The template attacks showed that it is possible to recover electro-magnetic side channel information from the keyboard operations. Together with the known structure of the pass-phrases, this information can reduce the search space for a pass-phrase attack significantly.

It should be mentioned again that this investigation was done on a modified test target and hence corresponds to a worst-case test scenario. The modifications make identification of the interesting attack significantly easier.

In practice an attack using a small spy-camera is much easier than a side channel analysis template attack. A real exploitation using a side channel analysis attack will therefore be highly unlikely. The emanation test was done for research purposes and helps the industrial partners of HECTOR in their path to commercialization.

Appendix D Attacking Demonstrator 2 and 3 display

D.1 Introduction

This test is intended to determine if demonstrator platforms D2 and D3 radiate electro-magnetic radiation that can be used to recover sensitive information during interfacing with the legitimate user through the keyboard. It is investigated if it is possible to recover (parts of) a passphrase entered by the legitimate user by attacking the display.

The display will show the pass-phrase characters in clear text, which are typed-in by the legitimate user. This process may leak additional information which can lead to disclosure of the secret user-pass-phrase. An attacker may combine information leaking from the display with information leaking from keyboard-entry (see Appendix C), thus creating a more effective attack.

A test was developed to get insight in the display leakage, separate from the keyboard leakage.

D.2 Test description

Characters to be displayed are sent to the test target while measuring electro-magnetic radiation above the display. Similar as for analysis of the keyboard signals, correlation is searched between challenge traces and training traces.

D.3 Test details and test results

The following table shows the details of the performed experiments. Detailed descriptions about the measurement set-up and related components can be found in Appendix F.

Test details	
Evaluator	JLUH, JPAP
Reviewer	JGAL
Hardware	SCA5
Software	Matrix v3.6.1 Sideways v3.21/22 Template Attack v1.4/1.5
Equipment parameters	Vcc = 5 V Sampling rate = 1 GS/s (TA) / 5 GS/s (DPA)

Table 6: Test details.

D.4 Test sample

The TOE was prepared by MICRONIC in a way that it is possible to turn on only the display. This reduces the possible noise during the measurement. Additionally a trigger signal was used, that indicates when characters are displayed. This means that it will be more difficult to attack a real device, where this preparation and the trigger are not present. The keyboard is not used during this test.

The display is a commercial-off-the-shelf product with a standard interface. It contains its own controller that generates the column and row voltages to drive the liquid crystals. Commands and data can be sent by the FPGA in sequence to set the operating mode of the display and to show characters respectively.

D.5 Test results

First the behaviour of the display was investigated. Randomly generated, ASCII-encoded characters were sent to the display of the TOE. A manual scan above the whole surface of the TOE was performed, in order to find the best position of an EM signal that can be related to the sent input. In the area around the coil position shown in Figure 24 it was possible to observe an interesting pattern in the signal. An example trace can be seen in Figure 22.

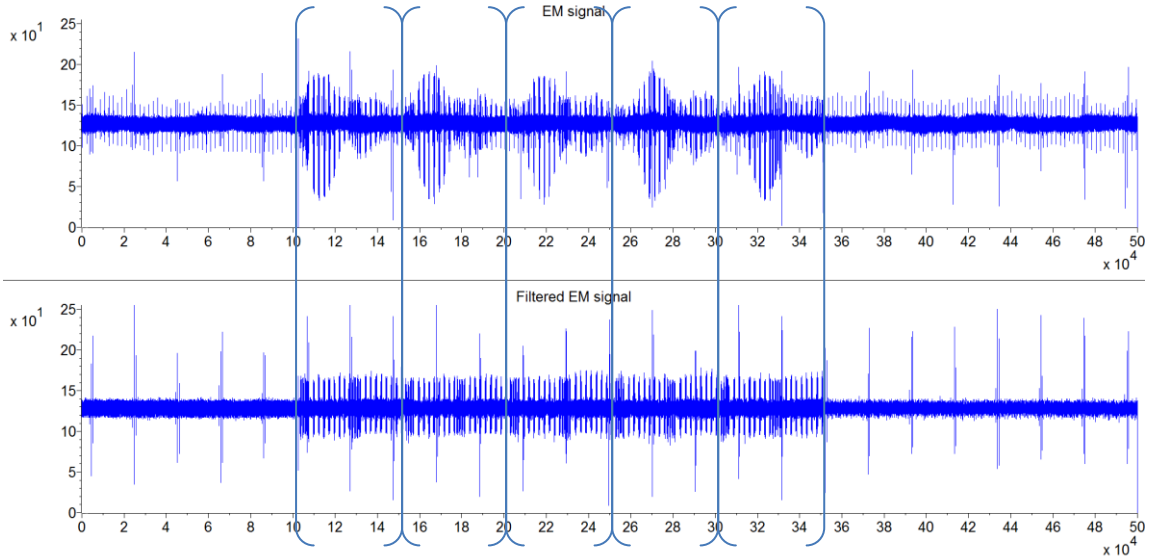

Figure 22: Recorded trace for five bytes of input. There are five repeated patterns visible in the trace (indicated by the brackets).

To confirm that this signal is related to the input, six bytes were sent to the display. The trace can be seen in Figure 23, where six repeated pattern can be observed. Within the pattern of each sent byte similar sequences of 4 + 8 peaks can be distinguished.

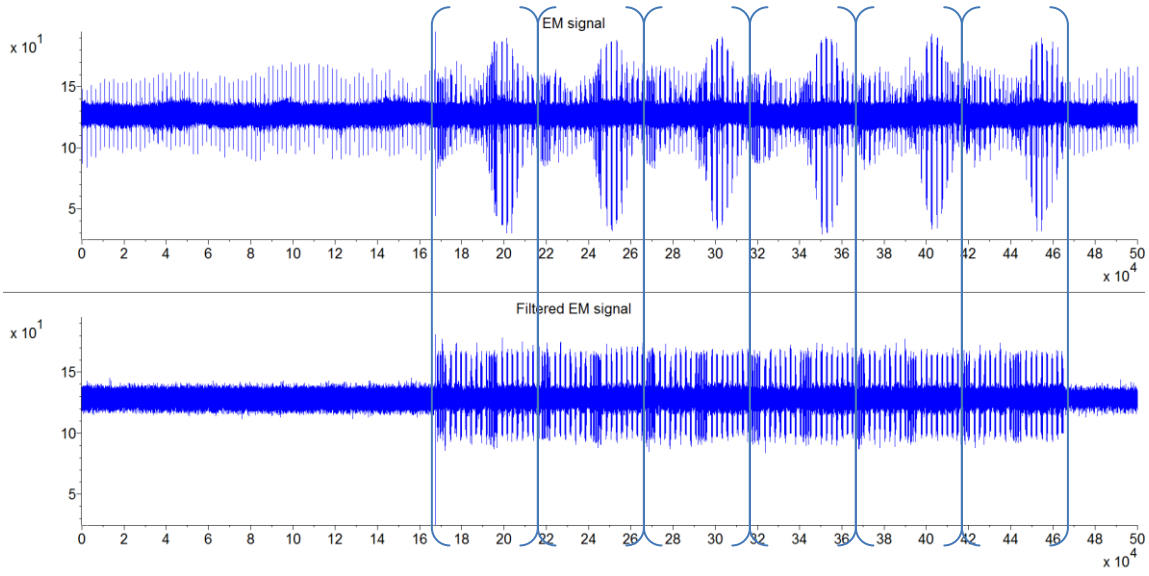

Figure 23: Recorded trace for six bytes of input. The patterns are clearly visible in the trace (indicated by the brackets).

Now the area around the point, where this signal was observed was again scanned manually in order to find the spot with the strongest and clearest signal. This position is used for acquisition and called final coil position. It can be seen in Figure 24.

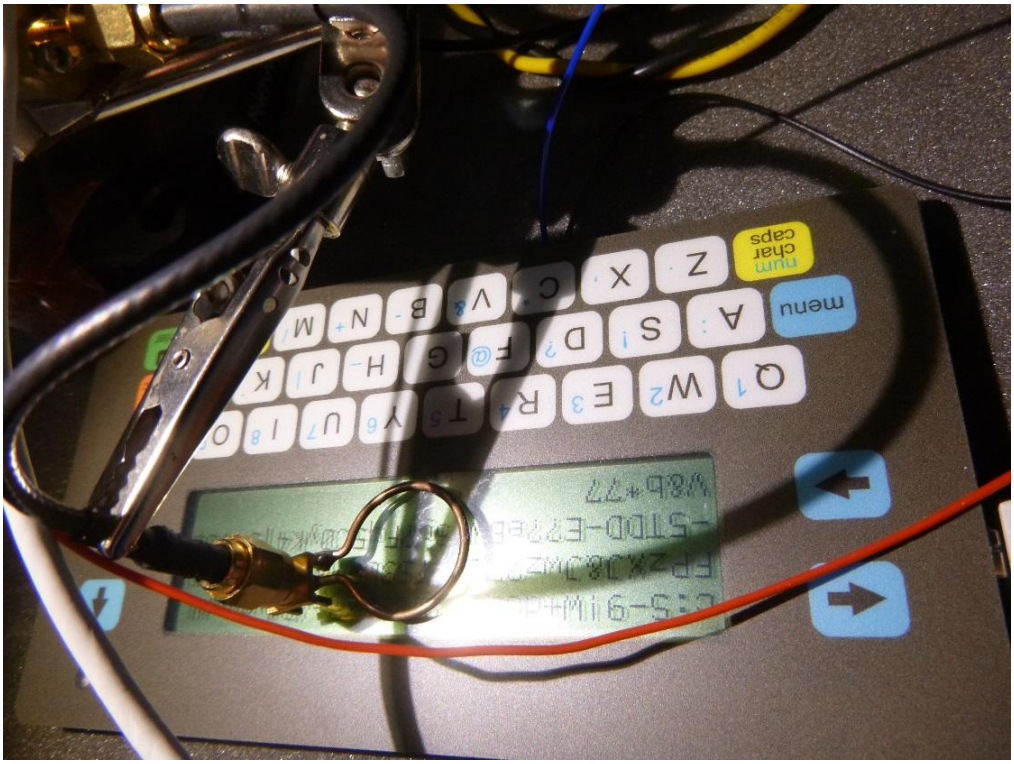

Figure 24: Final position of the coil for the test on the display.

A set of 1,000 traces was recorded at a sampling rate of 5 GS/s in order to perform correlation analysis with the displayed byte values. To speed up the acquisition only two randomly generated ASCII-encoded characters were sent to the display and stored with the traces. An example trace can be seen in Figure 25.

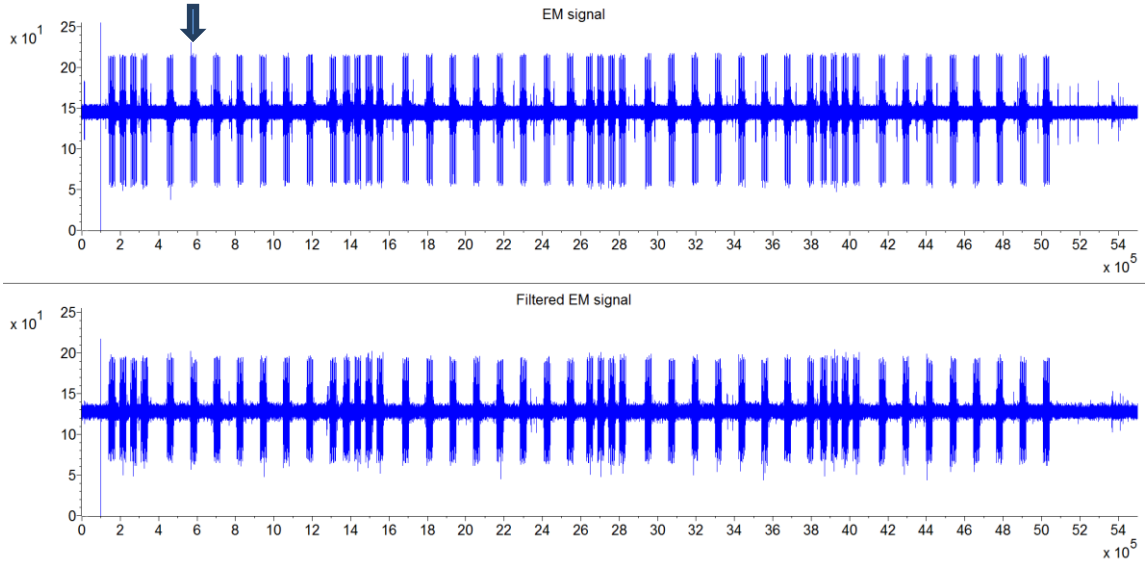

Figure 25: Example of traces acquired at the final coil position. The arrow indicates the peaks which were used for alignment.

The traces were aligned at the peaks indicated by the blue arrow in Figure 25. It was now possible to find correlation with all bits of the first byte sent to the display. Figure 26 shows the aligned traces and the observed correlation with the raw EM signal. Figure 27 shows the correlation with the filtered EM signal.

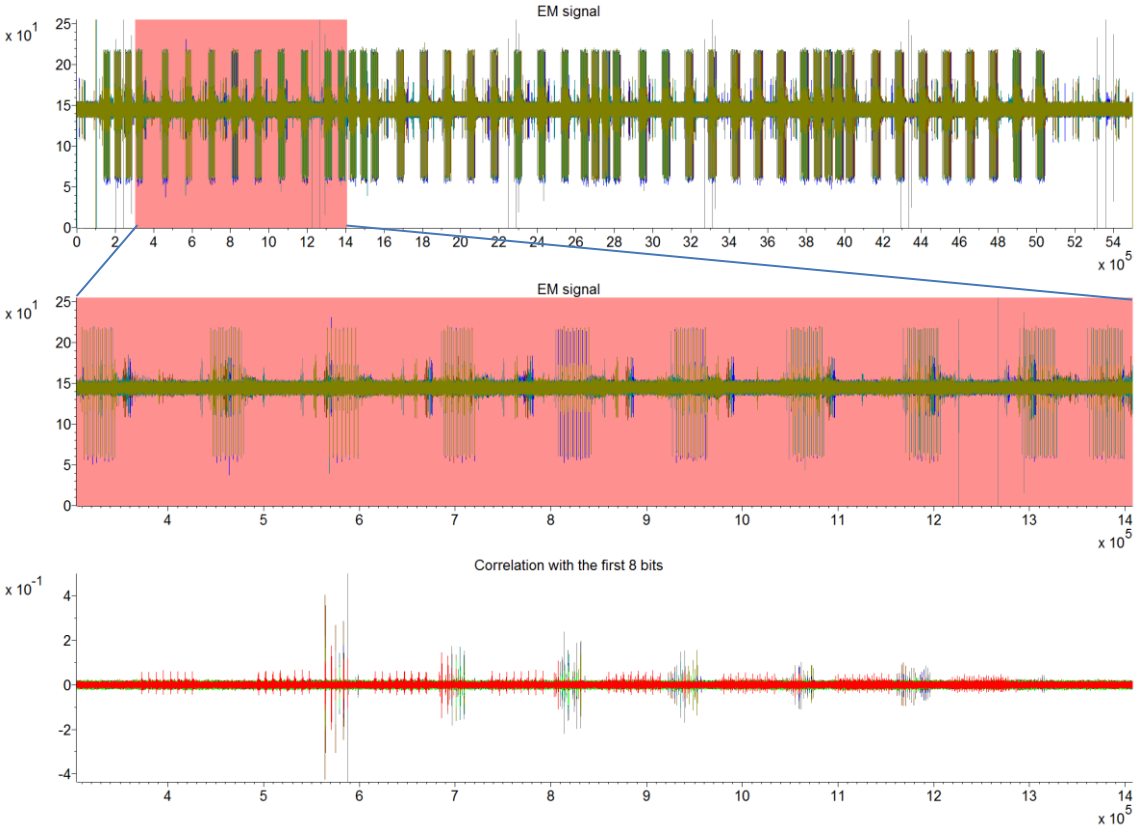

Figure 26: Five overlaid raw EM traces (top) and a zoomed-in view at the aligned part (middle). The bottom trace shows the correlation results with the eight bits of the first sent byte.

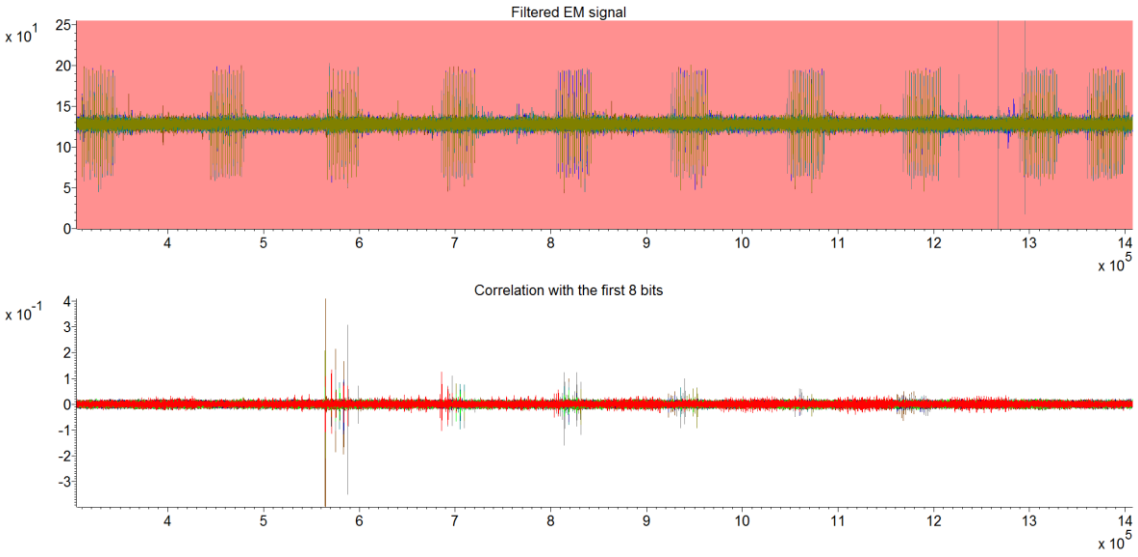

Figure 27: Zoomed-in view on the aligned part of the filtered EM traces and the corresponding correlation with the eight bits of the first sent byte.

The raw EM signal seems to show better correlation results. Therefore, further investigations are done on this signal.

Since correlation can be found, it was now decided to perform a template attack. The goal of this attack was to attempt to retrieve a byte that was sent to the display. A set of 190,000 EM traces was acquired at a sampling rate of 1 GS/s, comprising 114,000 training traces and 76,000 challenge traces. The attack was targeting 75 ASCII encoded characters, which were chosen randomly to reduce the influence of the distribution of the values. The set of ASCII characters that was used contains all 26 small and capital characters, as well as all numbers from 0-9 and the most common used special characters. This leads to the restriction that the high nibble of the ASCII encoded byte can only be the hexadecimal values 0x2 (0b0010), 0x3 (0b0011), 0x6 (0b0110) and 0x7 (0b0111). That leads to b7 = 0 and b5 = 1. Therefore, there are two fixed bits, which has to be taken into account in the later analysis.

Detailed information on the metrics for measuring the success of template attacks and on interpretation of the results can be found in Appendix 0

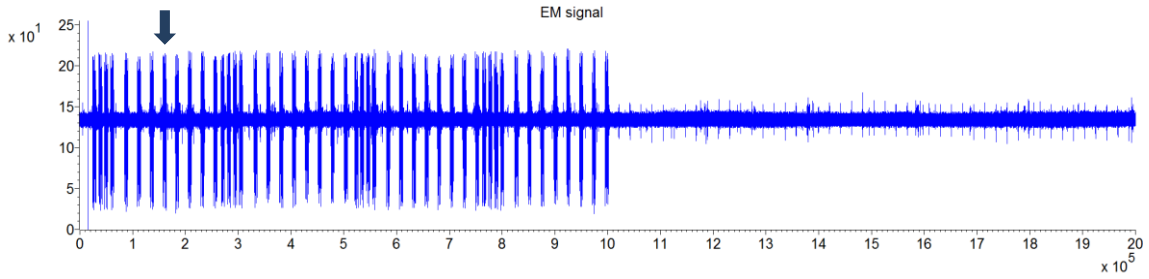

Figure 28: Acquired trace for the template attack. Only the first character pattern was used for the template attack.

Figure 28 shows one of the traces that were acquired. The traces were aligned at different positions. Figure 29 shows the best template attack results that were obtained with the alignment on the pattern indicated by the arrow in Figure 28. They lead to a maximum success rate of 0.92. This corresponds to the recovery of 70 classes out of 75. The keyboard characters do not cover the full 256 different values: two bits are fixed. The remaining search space consists of up to three unknown bits per byte.

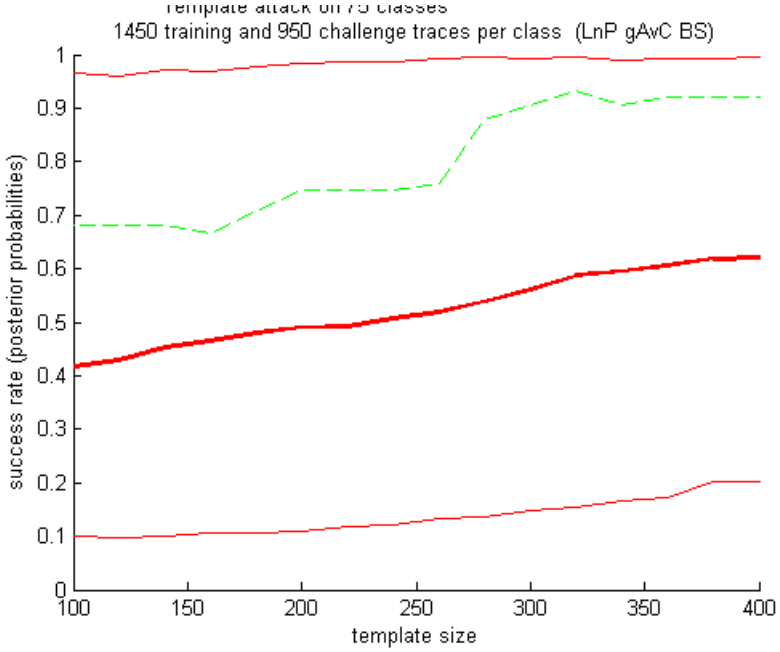

Figure 29: Success rate of the template attack on EM traces, on 75 ASCII values on the first sent byte as a function of the template size (minimum distance between points of interest equals two and a single covariance matrix was used).

The passphrase typed-in by the legitimate user is displayed on the keyboard in plaintext. A passphrase consists of eight words randomly chosen out of a dictionary of 8,192 words. One word has a length between one and six characters. In the best case the passphrase consists of 8 × 6 = 48 characters, in the worst case[2] the passphrase consists of 8 × 1 = 8 characters. Ignoring the fact that longer words occur more often than short ones, this results in an average length of 28 characters per passphrase and thus in an average remaining search space of $(2^3)^{28} = 2^{84}$ and an average brute force effort of 2^{83}, which can be considered as secure nowadays. This calculation ignores the position of the unknown bits. Due to the structure of the ASCII encoding it is possible that the partial knowledge of a byte reduces the search space for the unknown part drastically. This happens in the test set for example if the first 5 bits are known. The remaining search space can then be as less as 3 possibilities for the last 3 bits (this depends on the value of the first four bits).

It is to mention that this calculation is based on the assumption that the passphrase is sent character by character to the display and it is thus not possible to recognize the length of the different words. If this would be possible the number of possibilities for each word is decreased and therefore the remaining brute force effort can be feasible. Also the structure of the dictionary is not taken into account. The knowledge of several bytes leads to a reduction of the possible values for the next byte(s), because the words in the dictionary are public. This can make the brute force effort feasible. Furthermore, the property of the random number which is used to select the words for the passphrase during enrolment is not taken into account. This number should make sure that the dictionary is used with a uniform distribution.

An additional test was done with a set of 240,000 traces acquired at a sampling rate of 5 GS/s. For this test a slightly larger coil was used. The results were less significant with a maximum success rate of 0.7467. Nevertheless, it cannot be excluded that it is possible to obtain a success rate up to 1.0 during the classification. This could be achieved either by

[2] A word in the pass-phrase can consist of a single character. It is however unlikely that all eight words consist of a single character.

acquiring more traces, which is always a thread for side-channel leakage, or by further improvements of the attack methods in the future.

D.6 Shielding effect of back cover

All tests were performed on the front side of the TOE, which seems not very likely in practice because presence of a pick-up coil causes suspicion to the legitimate user. A larger threat could be if an electro-magnetic measurement on the backside of the TOE provides sufficient signal level for a practical attack. In such case an adversary could place a pick-up coil in a table-top to collect the electro-magnetic emanation of the demonstrator.

For testing its effectiveness the shielding back cover was removed. The pick-up coil was placed roughly one centimetre away from the TOE's backside to see how much the device is leaking. Indeed signals could be identified at selected locations. The EM traces can be seen in Figure 30.

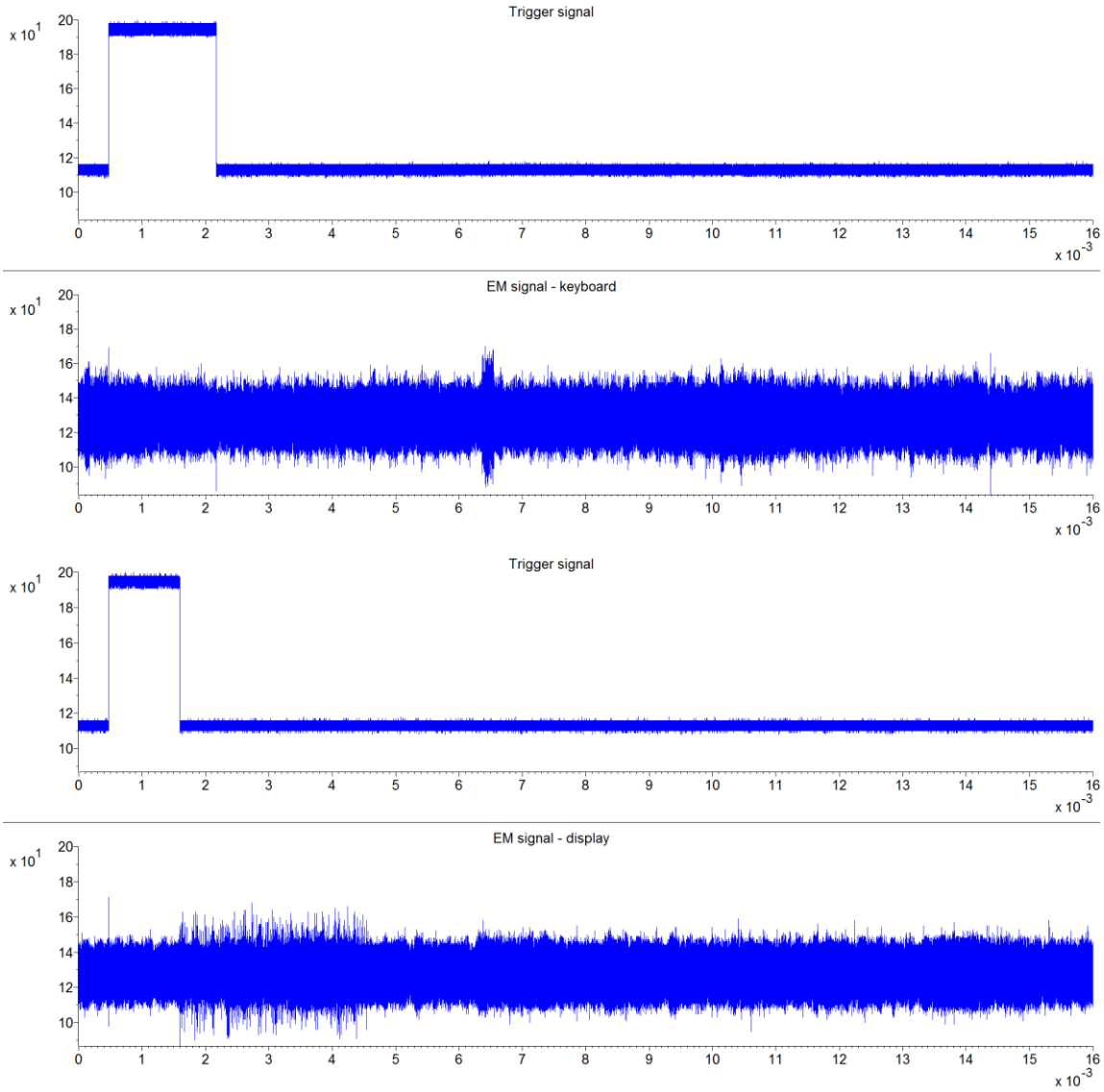

Figure 30: Measurements on the backside of the TOE without shielding. The top traces are related to key press emanation, the bottom traces show the display emanation.

In the following test the shielding was re-applied and signals were measured at the same locations. The signals were nearly not detectable.

The traces measured with shielding present can be seen in Figure 31.

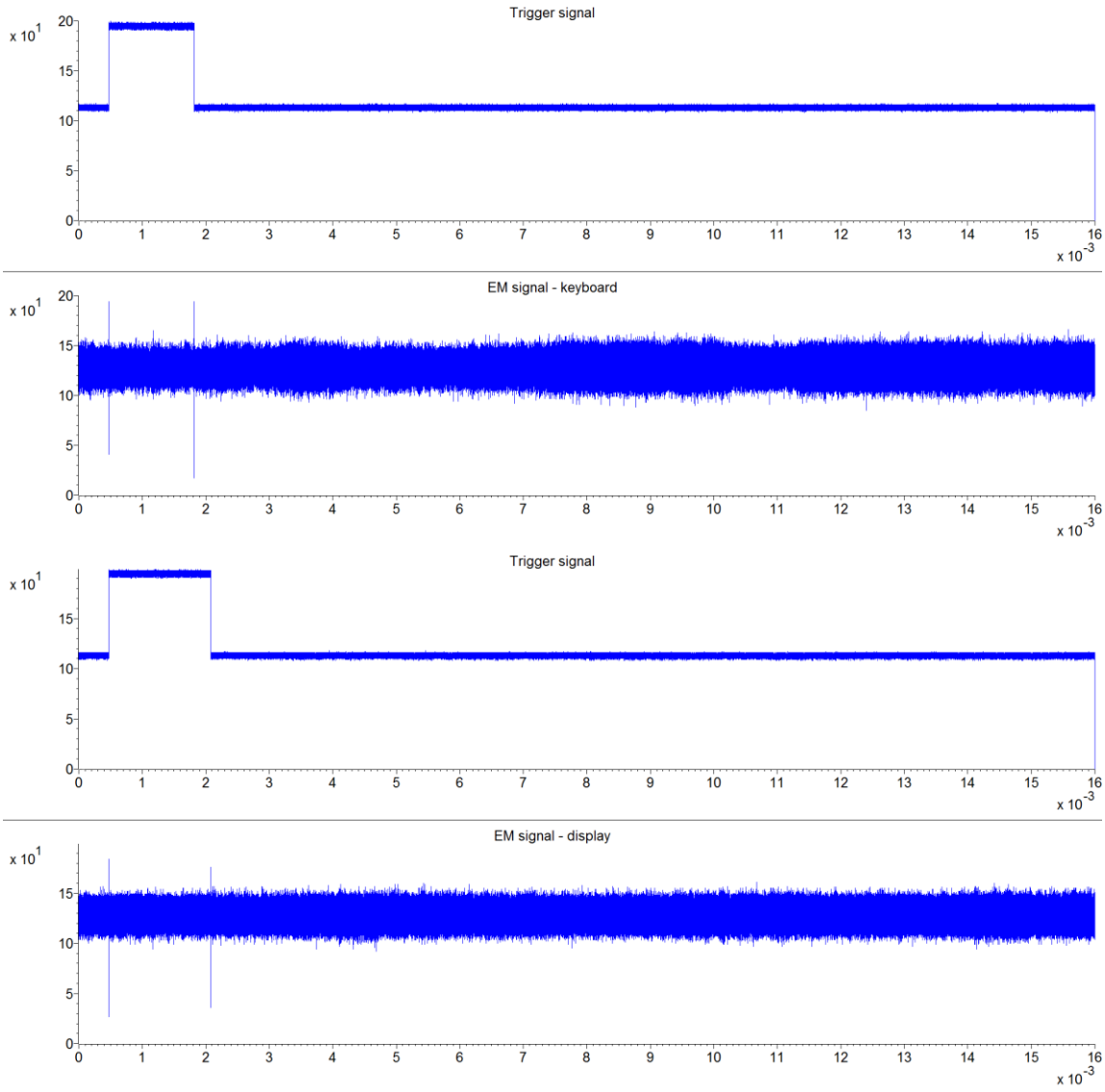

Figure 31: EM signal at the back of the TOE with shielding. The top traces are related to a key press, the bottom traces are related to the display.

This test shows that the signal levels at the back of the demonstrator are significantly reduced due to the shielding. This is a more realistic situation that an attacker should have to overcome.

D.7 Test conclusions on display emanation

This test was intended to determine if the demonstrator platform leaks information through electro-magnetic side channels while interacting with the legitimate user through the display interface.

It was possible to identify data-dependency in the EM signal related to the displayed values.

The template attacks showed that it is possible to recover information from the display. Together with the known structure of the pass-phrases, this information can reduce the search space for a pass-phrase attack significantly. In practice an attacker will also use the results of the keyboard emanation to improve the success rate (see previous chapter).

It should be mentioned again that this investigation was done on a modified test target. The modifications make identification of the interesting attack significantly easier. The shielding of the back cover makes it more difficult to collect electro-magnetic signals at the back of the demonstrator.

In practice an attack using a small spy-camera is much easier than a side channel analysis template attack. A real exploitation using a side channel analysis attack will therefore be highly unlikely. The emanation test was done for research purposes and helps the industrial partners of HECTOR in their path to commercialization.

Appendix E Perturbation attack on the passphrase re-try mechanism

E.1 Introduction

The vulnerability analysis showed that the device is well-protected against brute force attacks as result of the high entropy in the pass-phrase. The re-try mechanism is not essential for the protection of the demonstrators against brute force pass-phrase recovery. However, for research on the strength of this mechanism against perturbation attacks practical tests were devised. Information on the behaviour of the hardware platform and its configuration during laser manipulations provides useful information to the HECTOR partners for making robust commercial products.

This section describes laser perturbation testing on the retry mechanism of the HECTOR demonstrators.

E.2 Test description

This laser perturbation test mimics the behaviour of the demonstrator when incorrect pass-phrases are being input. First a normal enrolment is done. This loads the demonstrator with all keys and data required for normal operation. The pass-phrase retry counter is set to 20. Then sequences of incorrect pass-phrases are sent to the demonstrator, which reacts by decreasing the retry-counter value. During the verification and the counter-update process, a laser perturbation is done at varying moments in time at varying locations of the chip. The settings of the laser parameters (intensity, wavelength) are determined upfront by testing the sensitivity of the target for light pulses.

The reaction of the demonstrator on the perturbation pulses is determined by the replies that it provides after each operation. The replies can indicate normal behaviour of unexpected behaviour. The later ones could indicate a successful attack and need further explanation.

E.3 Test sample

Testing the security properties of the retry-mechanism is not practical on a real demonstrator because it requires repetitive and automatic entry of incorrect pass-phrases. Therefore a dedicated FPGA design was developed by MICRONIC – based on specifications of Brightsight – providing interfaces to the retry-mechanism that allows for automated testing.

In addition the laser testing requires the FPGA chip surface to be exposed. This required removal of the epoxy package above the silicon die without damaging the active circuits. As laser experiments can potentially be destructive, it also had to be possible to replace a defective sample with a new one.

Both decapsulation of the package and eventual replacement would be very inconvenient on the real demonstrators. The test function is therefore implemented on a HECTOR daughterboard using the same Microsemi SmartFusion2 FPGA as Demonstrator 2.

The Microsemi SmartFusion2 M2S025 FPGAs are Ball-Grid Arrays with 484 solder balls. The silicon die is placed with the metal side facing up (away from the ball bonds), which means the interfacing between the package substrate PCB is done using bond wires. Due to the large number of interfaces a dual row of bond wires is applied at all four edges of the die. Most FPGA manufacturers today use copper bond wires as a compromise between costs and performance over traditional gold and aluminium wires. Standard decapsulation uses fuming nitric acid to solve the epoxy resin without damaging the die or its bond wires. This can be done for packages using gold or aluminium wires, but not for devices that apply copper bonding. The acid will dissolve the copper.

Therefore a more specialized decapsulation method had to be applied using plasma (Microwave Induced Plasma, or MIP). This work was outsourced to the Dutch company MASER, which is a failure analysis lab with the required experience and equipment. A series of five FPGAs was sent for decapsulation, of which one was damaged beyond repair during handling.

The opened FPGA samples were soldered at the HECTOR daughter board and positioned in a laser manipulation setup. The daughter board is shown in Figure 32.

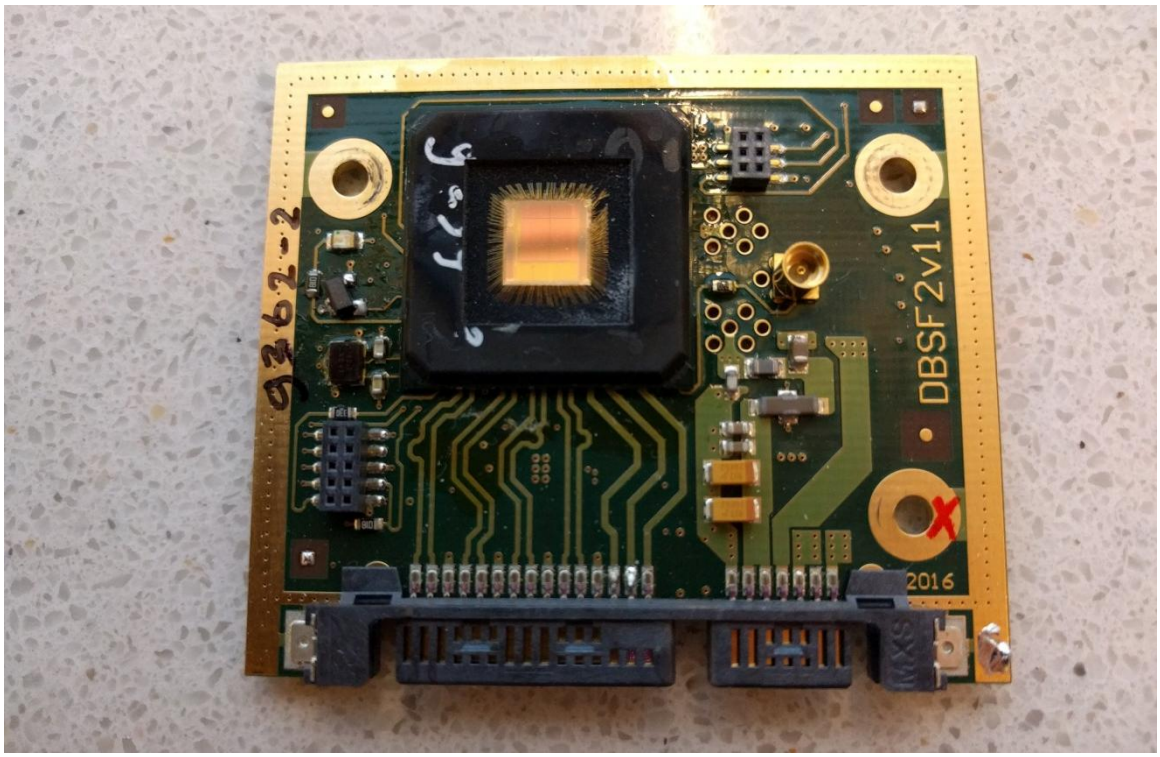

Figure 32: Decapsulated FPGA test sample mounted at a HECTOR daughter board.

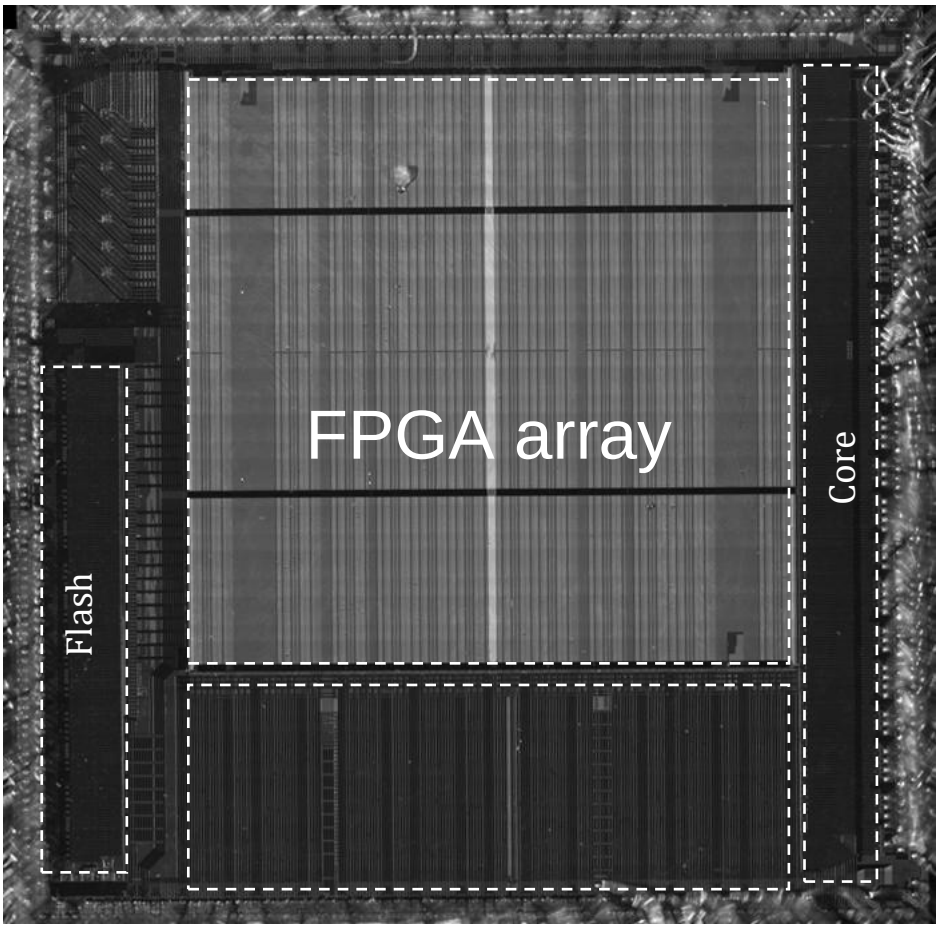

Figure 33: Surface picture of the Microsemi M2S025 SoC.

E.4 Test details and test results

The following tables show the details of the performed experiments and the commands that have been used. Detailed descriptions about the measurement set-up and related components can be found in Appendix F.

Test details	
Sample details	Hardware: Microsemi M2S025
Evaluator	RMAL
Reviewer	LZUS, GBAT
Hardware	LM6
Software	Matrix v3.6.1 Measurement framework v2.7.1
Equipment parameters	Objective = 50x Green (LM6) Laser wavelength = 520 nm (Green)

Table 7: Test details.

Command	File executed	Response (typical status values)
get status	get_status.tcl	Demo2 status: 0x10 - MAIN_PHRASE_BAD
get retry counter	get_retry_counter.tcl	Retry counter: X attempts remain, where X can range from 1 to 20.
send enter	send_enter.tcl	Demo2 status: 0x0E - MAIN_PHRASE_ENTRY
send passphrase	send_passphrase.tcl	Demo2 status: 0x10 - MAIN_PHRASE_BAD

Table 8: Commands used during the performed experiments.

For this experiment, initially the retry counter value is set to 20. This means that 20 incorrect passphrases can be sent before the device will block further operation. Then an incorrect passphrase is sent for each manipulation attempt, causing the retry counter to be decremented. If the retry counter is reduced to one, it will be set back to 20 by sending a correct passphrase. In case the retry counter would reach zero a new enrolment needs to be done, which unblocks and resets the retry counter and sets a new passphrase for the device. All data in the device is then lost.

In order to find the proper timing for the attack, the UART communication lines were monitored using an oscilloscope. Both transmission and reception lines were recorded and compared to the expected profiles according to the data sent through each line. Once it was verified that the correct lines were being monitored, it was decided to target the complete interval between final bit of command transmission and first bit of reply reception (see Figure 34). The verification and counter update must occur within this interval. The duration of the window is only approximately 40 µs, which allows for a fairly detailed scan resolution.

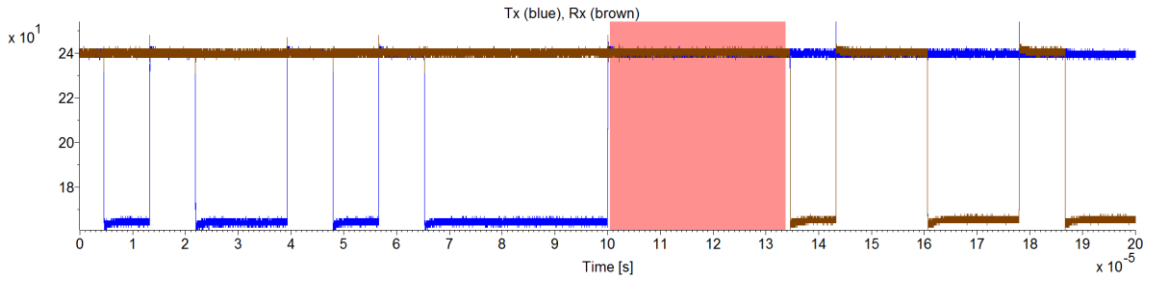

Figure 34: Transmission (Tx) and reception (Rx) lines profile.

Several experiments (surface scans) were performed combining the following laser settings:

- Laser input voltage: 4 V, 4.2 V, 4.4 V, 4.8 V
- Laser pulse width: 2 µs, 6 µs, 10 µs
- Delay: 0 µs ≤ delay ≤ 30 µs, in steps of 2 µs
- Scan locations: 420
- Attempts per location per delay: 1
- Intended attempts per complete surface scan: 10,000
- Total number of laser manipulation attempts: 126,346

The whole surface area of the chip was targeted for the surface scans. During the experiments the following events were recorded:

- *Expected responses:*
  Expected responses are those in which the retry counter is decreased by one and the return message is '`0x10 – MAIN_PHRASE_BAD`' or '`0x11 – MAIN_PHRASE_BAD_LAST`' (which means one last attempt remaining before the

device will be blocked). This is the expected behaviour and it is no security issue. This is the result for 125,747 attempts (99.23%).

- *Manipulation attempts with undefined response:*
  Only two attempts were recorded with this response. The responses returned the following message: '`0x - UNKNOWN`'. In both cases, the retry counter value returned 'invalid value 15'. In the first case the remaining number of attempts at the start of the command execution was 7 and in the other case 15. It is unclear why this happened. The following image represents the returned log (equal for both cases):

```
Connection to '\\.\COM8' opened
Passphrase to send: heck tramp 4q gw rival null ague satan
Demo2 status: 0x - UNKNOWN
Retry counter: invalid value 15
Connection to '\\.\COM8' closed
```

Figure 35: Log of the manipulation attempts for which the status '`0x - UNKNOWN`' was returned.

- *Manipulation attempts in which the retry counter did not decrease:*
  The goal of this test is to perform verification of a pass-phrase without decreasing the retry counter. Manipulation attempts in which the retry counter was not decreased were recorded in 596 occasions, but such attempts did not return the expected message '`0x10 - MAIN_PHRASE_BAD`', but instead '`0x0F - MAIN_PHRASE_TEST`' or '`0x0E - MAIN_PHRASE_ENTRY`'. In these cases it could be deduced that the sample was not processing any passphrase verification, thus no bypass could be achieved and therefore the retry mechanism was not compromised.

For each location of the surface scans, different delays for the laser pulse were used; the result is displayed using colours. For this experiment, the colour code used is the following (ordered according to the severity of the response; from no impact to worst case):

- Green dots represent an expected response.
- Yellow dots represent manipulation attempts in which an unexpected response was recorded. Unexpected responses are those different than:
  - '`0x10 - MAIN_PHRASE_BAD`'
  - '`0x11 - MAIN_PHRASE_BAD_LAST`'
  - '`0x0F - MAIN_PHRASE_TEST`'
  - '`0x0E - MAIN_PHRASE_ENTRY`' (the last two responses are known to be no security threat).
- Blue dots represent manipulation attempts in which the number of remaining pass-phrase attempts did not change after execution of the send passphrase command.

Please note that the size of the squares is not representing the actual illuminated area. The laser spot size is smaller and depends on the objective used.

A large number of surface scans were made. Some experiments only recorded expected responses. A resulting scan image looks as follows:

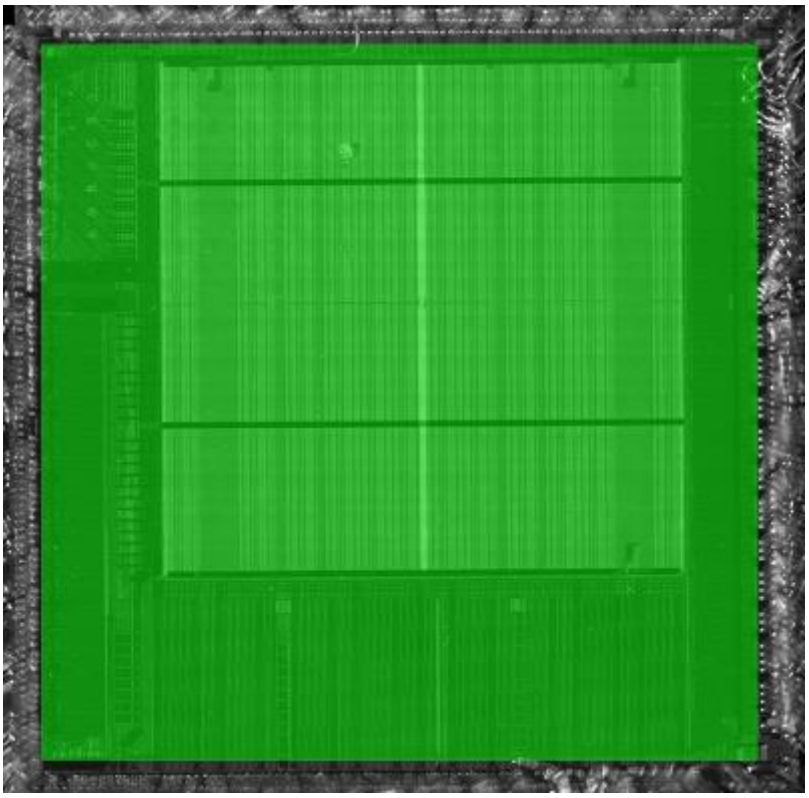

Figure 36: Results of a surface scan in which only expected responses were recorded (this one is the result of a surface scan with pulse width of 2 µs and laser input voltage of 4.2 V).

In the surface scans in which different responses than expected were recorded, an image like the following was obtained:

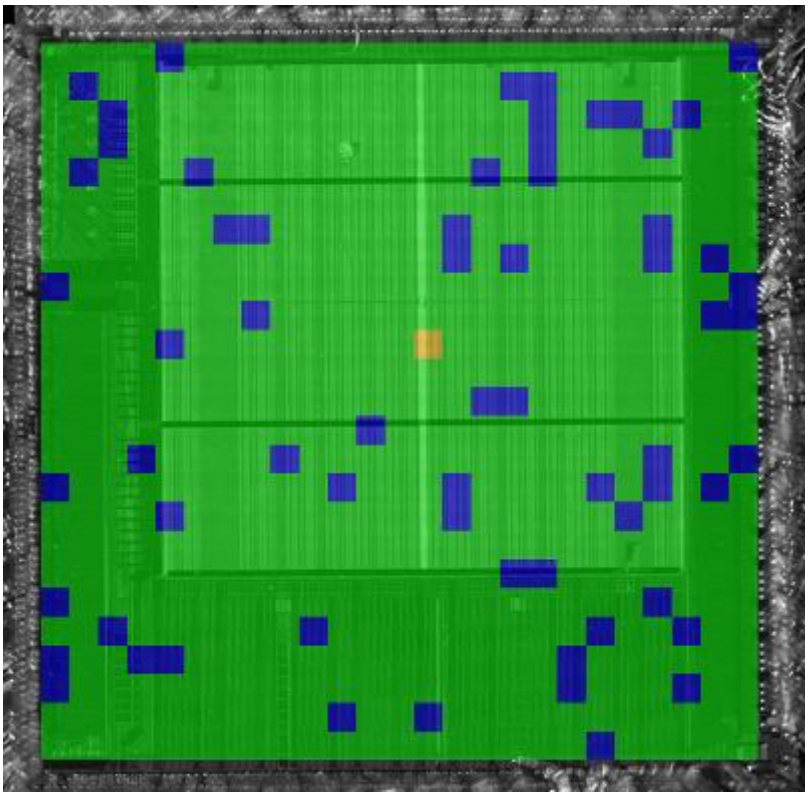

Figure 37: Results of a surface scan in which not only expected responses were recorded.

E.5 Test conclusion

Experiments using light manipulation techniques were done in order to test if the retry counter mechanism of the pass phrase entry can be bypassed. During the experiments, several unexpected responses were recorded, but no successful manipulation was achieved. None of the experiments resulted in any bypass of the retry-counter.

It is concluded that the TOE is protected against attackers with high attack potential for this attack scenario.

Appendix F Environments for Testing

F.1 Light perturbation setups

F.1.1 Description

Figure 38 and Figure 39 show schematic representations of the light manipulation measurement set-ups of LM1/LM3/LM5/LM7 and LM4/LM6/LM8, respectively. The set-ups LM1, LM3, LM5 and LM7 use a laser cutter module, which is able to produce short (4-7ns) laser bursts. The set-ups LM4, LM6 and LM8 use solid-state lasers and are therefore able to produce laser bursts of infinite duration. The minimum laser burst duration of the solid-state laser depends on the bandwidth of the used laser module.

The TOE is placed into a custom designed 'daughterboard' mounted on a 'motherboard''. Two function generators are connected to the motherboard. One function generator is used to generate the necessary 3.57MHz clock signal and the other is used to perform a cold reset. An oscilloscope is used to monitor the (filtered) power consumption and triggers a third function generator, which is used to trigger the laser module (with an adjustable delay). All the signals connected to the TOE are routed through the motherboard, which, for LM1, LM3, LM5 and LM7, is mounted on a XY-stage to target the laser. In the LM4, LM6 and LM8 set-ups, the solid-state laser itself (and not the motherboard) is mounted on an XYZ stage.

In case the TOE has active countermeasures that render it inoperable on detection of manipulation attempts, a second oscilloscope will be added to the set-up. In most cases an EEPROM or Flash (erase/)write operation is required to render a device inoperable. Using the second oscilloscope it is possible to detect this EEPROM or Flash (erase/)write operation and instantly perform a cold reset. A cold reset will interrupt the erase/write operation, which will leave the device operable.

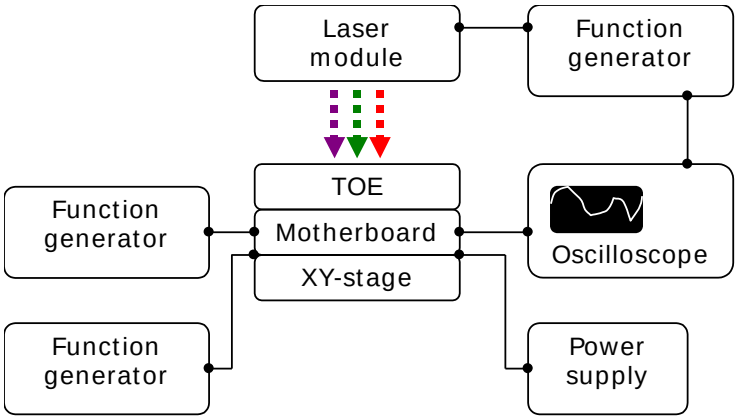

Figure 38: Schematic representation of the LM1, LM3, LM5 and LM7 set-ups (contact mode).

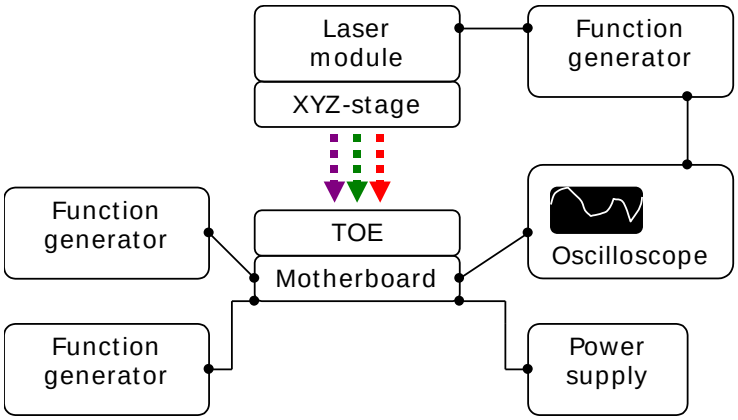

Figure 39: Schematic representation of the LM4, LM6 and LM8 set-ups (contact mode).

F.1.2 Components

Table 9 shows the components of each light manipulation set-up that is available at Brightsight (contact mode).

Set-up	Description	Manufacturer	Type
LM1	Function generator (laser trigger)	Agilent	33250A
	Function generator (clock and cold reset)	Agilent	33522A
	Power supply	Agilent	E3640A
	Primary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Secondary oscilloscope	LeCroy	9354AL
	Laser	New Wave	EzLaze II Trilite
	XY-stage	Märzhäuser	L-Step 12/2
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1
LM3	Function generator (laser trigger)	Agilent	33250A
	Function generator (clock)	Agilent	33220A
	Function generator (cold reset)	Agilent	33250A
	Power supply	Agilent	E3640A
	Primary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Secondary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Laser	New Wave	QuikLaze 1064/532
	XY-stage	Märzhäuser	Tango 2
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1
LM4	Function generator (laser trigger)	Agilent	33250A
	Function generator (clock)	Rigol	DG1022
	Function generator (cold reset)	Agilent	33220A
	Power supply	Agilent	E3640A
	Primary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Secondary oscilloscope	LeCroy	Waverunner LT372L

Set-up	Description	Manufacturer	Type
	Laser	AlphaNOV	PDM-1064 IR laser (1064nm)
		Brightsight	Blue laser 445nm
	XY-stage	Newport	M-462 series
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1
LM5	Function generator (laser trigger)	Agilent	33250A
	Function generator (clock and cold reset)	Agilent	33522A
	Power supply	Agilent	E3631A
	Primary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Secondary oscilloscope	LeCroy	Waverunner LT342
	Laser	New Wave	Dual EzLaze III 1064
	XY-stage	Märzhäuser	L-Step 12/2
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1
LM6	Function generator (laser trigger)	Agilent	33250A
	Function generator (clock)	Rigol	DG1022
	Function generator (cold reset)	Agilent	33220A
	Power supply	Agilent	E3640A
	Power supply (laser power)	Agilent	E3640A
	Primary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Secondary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Laser	AlphaNOV	PDM-1064 IR laser (1064nm)
		Brightsight	Blue laser 445nm
	XY-stage	Newport	M-462 series
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1
LM7	Function generator (laser trigger)	Agilent	33522B
	Function generator (clock and cold reset)	Agilent	33522B
	Power supply	Agilent	E3631A
	Primary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Secondary oscilloscope	LeCroy	WaveSurfer 24MXs-B
	Laser	New Wave	Dual EzLaze III 1064
	XY-stage	Märzhäuser	Tango 2
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1
LM8	Function generator (laser trigger)	Agilent	33522B
	Function generator (clock and cold reset)	Agilent	33522B
	Power supply	Agilent	E3631A
	Power supply (laser power)	Agilent	E3631A

Set-up	Description	Manufacturer	Type
	Primary oscilloscope	LeCroy	HDO4024
	Secondary oscilloscope	LeCroy	HDO4024
	Laser	AlphaNOV	PDM-1064 IR laser (1064nm)
		Brightsight	Blue laser 445nm
	XY-stage	Newport	M-462 series
	Motherboard	Brightsight	LM-Motherboard-USB-4.1, Version 3.9.1

Table 9: Measurement set-up components.

F.1.3 Laser Parameter Information

The manner in which the energy of the laser pulse is configured on the various laser set-ups differs depending on the laser module used. Table 10 gives an overview how the settings are done for each set-up.

Set-up	Laser Module	Laser Energy	Aperture
LM1, LM3, LM5, LM7	All laser cutter modules	Proprietary units: Two ranges "Lo(w)" and "Hi(gh)" Units running from 0 (lowest setting) to 255 (highest setting)	Proprietary units for X and Y: Units for X/Y running from 0 (lowest setting) to 255 (highest setting)
LM4, LM6, LM8	AlphaNOV IR laser	The laser energy is determined using a voltage that is applied to the laser module. Range: 0-5 V (5 V → max. energy)	Fixed aperture
	Brightsight Blue Laser	The laser energy is determined using a voltage that is applied to the laser module. Range: 4-6 V (~0-100 % energy)	

Table 10: Laser Parameter Information.

F.2 Side channel set-ups

F.2.1 SPA/DPA set-ups

The SPA/DPA measurement set-up is used to measure the power consumption profile of a smart card or smart card chip. The TOE can be inserted (with or without external connector board) in the TOE interface. Figure 40 shows a schematic representation of the set-up.

The ground pin of the TOE is connected to either a 50 Ω resistor or the 50 Ω input impedance of the oscilloscope. The oscilloscope is used to digitise the voltage across this impedance. This signal is referred to as the power consumption profile. Not only the power consumption profile, but also the filtered power consumption profile and the IO communication signals are measured with the oscilloscope. These signals are useful to identify the parts of interest in the power consumption profile.

The oscilloscope can be triggered by the I/O signal, a specific pattern in the power consumption profile, a software trigger signal generated by the SPA/DPA interface or a combination (smart trigger). A PC is connected to the SPA/DPA interface and oscilloscope to control the commands send to the TOE and to collect the measured power consumption profiles.

The function generator is used to generate a 3.57 MHz clock signal with adjustable amplitude and offset. The power supply is used to power the chip with an adjustable voltage. A low

voltage often improves the results of SPA/DPA, but the TOE will be inoperable when the input voltage is too low.

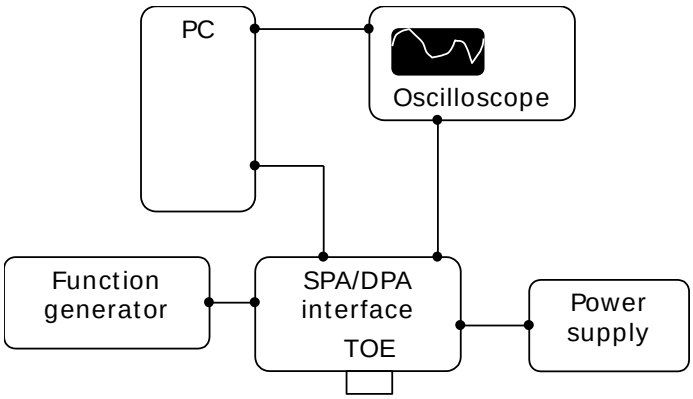

Figure 40: Schematic representation of the set-up (non-contactless).

The following table shows for a DPA set-up the components of which it consists:

Setup ID	Description	Manufacturer	Serial number	App ID
DEMAP1				
	Power supply	Tenma	72-8695	PS 50
	Oscilloscope	Lecroy	WS24MXS-B	SCOOP 21
DPA1				
	Function generator	Agilent	33120A	CLK 01
	Power supply	Agilent	E3631A	PS 02
	Oscilloscope	Lecroy	WR620ZI	SCOOP 43
	DPA Card Reader	Brightsight	-	SN001
DPA2				
	Function generator	Agilent	33120A	CLK 02
	Power supply	Agilent	E3631A	PS 60
	Oscilloscope	Lecroy	WR620ZI	SCOOP 36
	DPA Card Reader	Brightsight	-	SN010
DPA4				
	Function generator	Agilent	33250A	CLK 21
	Power supply	Agilent	E3631A	PS 38
	Oscilloscope	Lecroy	WR620ZI	SCOOP 23
	DPA Card Reader	Brightsight	-	SN011
DPA5				
	Function generator	Rigol	DG1022	CLK 20
	Power supply	Tenma	72-8695	PS 40
	Power supply	Agilent	E3631A	PS 43

Setup ID	Description	Manufacturer	Serial number	App ID
	RF Synthesizer	Hameg	HM8135	RF-SYNTH-1
	Spectrum Analyser	Hameg	HM5014-2	SPECTRUM 1

Table 11: Measurement set-up components.

F.2.2 SEMA/DEMA set-ups

The DEMA set-ups in Brightsight are capable of measuring electro-magnetic signal and power signal simultaneously or independently.

The set-up for measuring the electro-magnetic side channel on contact secure micro controllers or smart cards is placed inside a Faraday cage. The electro-magnetic signals emanated from the surface of the TOE can be measured with minimal influence from external electro-magnetic sources (e.g. GSM phone signals, contactless smart card readers, etc). A schematic view of the set-up is shown in the figure below in Figure 41.

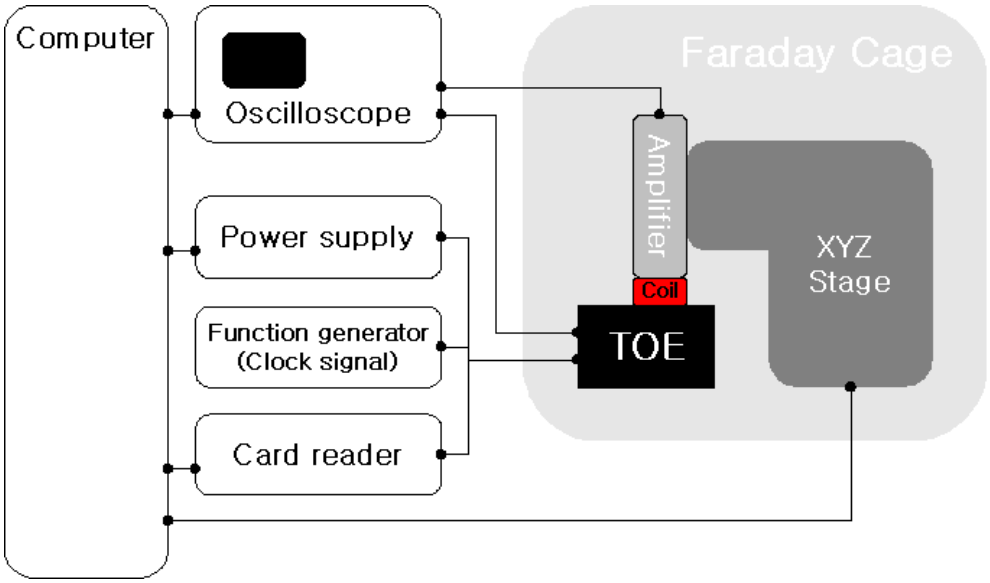

Figure 41: Schematic representation of the set-up (contact) for measuring electro magnetic side channels.

The TOE is connected to the computer through a card reader. The power supply is used to power the TOE with an adjustable voltage. A function generator supplies the TOE with a clock signal with an adjustable amplitude, offset and frequency.

The pickup coil used to measure the EM emanation is connected to an amplifier mounted on an XYZ stage. The amplifier is used to amplify the signals that are picked up by the coil. The XYZ stage can be used to automatically scan the surface of the TOE to find an interesting location to measure a larger set of EM traces for the differential analysis. The oscilloscope is used to digitise the amplified signals picked up by the coil.

The oscilloscope can be triggered by a specific pattern in the EM emanation profile or power consumption profile, IO signal, a software trigger signal generated by the card reader or a combination (smart trigger). A PC is connected to the card reader and oscilloscope to control the commands sent to the TOE and to collect the measured power and EM emanation profiles.

The power supply is used to power the chip with an adjustable voltage. A low voltage often improves the results of SPA/DPA, but the TOE will be inoperable when the input voltage is too low.

For signal processing (e.g. performing alignment, DPA/DEMA analysis as well as key search operations) dedicated Brightsight software is used.

For power measurement using the DEMA set-up, the same set of equipment and software are used. Table 12 shows the components of DEMA setups.

Setup ID	Description	Manufacturer	Serial number	App ID
DEMAP1				
	Power supply	Tenma	72-8695	PS 50
	Oscilloscope	Lecroy	WS24MXS-B	SCOOP 21
EM1				
	Function generator	HP	33120A	CLK 04
	Power supply	Agilent	E3631A	PS 05
	Oscilloscope	Lecroy	WR620ZI	SCOOP 35
EM2				
	Function generator	Agilent	33220A	CLK 17
	Power supply	Agilent	E3631A	PS 42
	Oscilloscope	Lecroy	WR620-ZI	SCOOP 24
EM3				
	Function generator	Agilent	33220A	CLK 08
	Power supply	Agilent	E3631A	PS 36
	Oscilloscope	Lecroy	WR620ZI	SCOOP 28
EM4				
	Function generator	Agilent	33522B	CLK 31
	Power supply	Agilent	E3631A	PS 49
	Oscilloscope	Lecroy	WR620ZI	SCOOP 31
EM5				
	Function generator	Agilent	33522A	CLK 26
	Power supply	Agilent	E3631A	PS 57
	Oscilloscope	Lecroy	WR620ZI	SCOOP 27

Table 12: Measurement set-up components.

F.3 Template attack method

F.3.1 Introduction to the template attack environment

The Brightsight template attack environment allows a measurement set to be used for both the construction of templates and for subsequent testing of those templates, by simply using the first *N* traces out of each class for the creation of the templates, and then, after all templates have been created, attempting to classify the *M* subsequent traces ('challenges'). Both *N* and *M* are user-defined. The Brightsight template attack environment also allows two separate measurement sets to be used for the construction of templates and for testing of those templates respectively. Depending on an attack scenario, it can be decided which approach will be used.

In order to determine the success rate of the template attack, several metrics can be calculated:

1. The *overall success rate*, that is, the overall percentage of individual challenge traces which were correctly classified. For the classification, the regular 'maximum likelihood' method is used, calculating a score representing the likelihood that the challenge trace belongs to the probability distribution defined by each of the templates, and then determining the highest score, and deciding the candidate is matched to the corresponding template. This metric is essentially the mean conditional probability of good classification, over all possible values of the secret parameter.
2. The *per-candidate worst case success rate*, that is, out of all possible candidate values c_i, the highest value of the success rate for classification of challenges corresponding to only c_i. (a per-candidate best case success rate can be defined accordingly, but is less relevant as a metric). This metric is essentially the maximum of the conditional probability of good classification, over all possible values of the secret parameter.
3. The *combined classification success rate*, that is, the success rate of a classification process which combines multiple challenge traces into a single classification as follows: all traces related to a single candidate value c_i are compared to all templates T_i. For each of those comparisons, the calculated likelihood is stored. After this, the likelihood of all comparisons to template T_i are combined to obtain an aggregated score S_i. The classification of the set of traces is established to be the candidate value for which S_i is the highest. This is repeated for the challenge sets corresponding to all candidate values c_i. Thus, in the example with 256 classes (or candidate values), 256 combined classifications would be calculated. The percentage of those that are correct is the *combined classification success* rate.
4. (Optional) The O*verall Combined Classification Success rate (OCCS)*, that is, the overall percentage of individual sets which were correctly classified. The only difference with the combined classification rate is that multiple sets of challenge traces per class are used. As it takes much more traces and processing times, it is not always applicable. For example, if a combined classification is computed with 500 traces per each class, the number of classes is 256, and the number of experiments for computing the overall combined classification success rate is 100, the total number of challenge traces would be 500x256x100 = 12.8 M traces.
5. (Optional) The *worst-case Combined Classification Success rate (CCS)*, is the highest combined classification success rate among several individual combined classification success rates. Similarly the *best-case Combined Classification Success rate* is the lowest one.

Template size

The toolset calculates each of the metrics, for a number of different template sizes (*a.k.a.* number of interesting points). Usually, a set of templates that work is characterised by a growth in the success rate as the template size grows, until over-training occurs and the success rate starts to decrease. A non-functional set of templates will show success rates more or less equal to the likelihood of correct random guessing.

Prior and posterior probabilities

The toolset allows calculating either prior or posterior probabilities. Prior probability is the probability that a challenge from a given class is classified as belonging to that class. Posterior probability is the probability that a challenge which has been classified as belonging to a class actually belongs to that class.

As an illustration, imagine the following scenario: two classes *A* and *B* exist, each equally likely to occur in challenges, but the classification result classifies any challenge as belonging to class *A*, except one in every 100 challenges belonging to *B* is correctly classified as *B*. In this situation, the prior probability for class *B* is only 1%, but its posterior probability is 100%.

The posterior probability is considered more representative for a real attack scenario as the attacker has no knowledge of the correct result before the attack.

Classes: templates for value vs. Hamming Weights

When performing a template attack, typically the target value is some value internal to a product. Popular examples are key segments as they are transported over internal data buses or intermediate values from cryptographic algorithms.

Given the way most embedded hardware works, it is reasonable to attack not the value itself, but rather it's Hamming Weight. In many contexts, knowledge of the Hamming Weight only already poses sufficient threat that the product should be considered compromised.

A frequent application is the attack of the Hamming Weight of a byte. In this scenario, nine classes are present, and although it is reasonable to use an equal amount of training and challenge traces for each of the nine classes, in an actual attack these classes do not have equal probability of occurring. The toolset contains logic to properly calculate success probabilities in this application.

F.3.2 Interpretation of template attack results

After executing the toolset, a figure showing several success rates is generated, for example, as shown in Figure 42. The x-axis shows the template size (*a.k.a.* number of interesting points) and the y-axis represents the success rate, which takes value from 0 to 1. Success rates can be computed either by prior probability or posterior probability. On top of the figure, the numbers of the training and the challenge traces per class are shown with the number of classes.

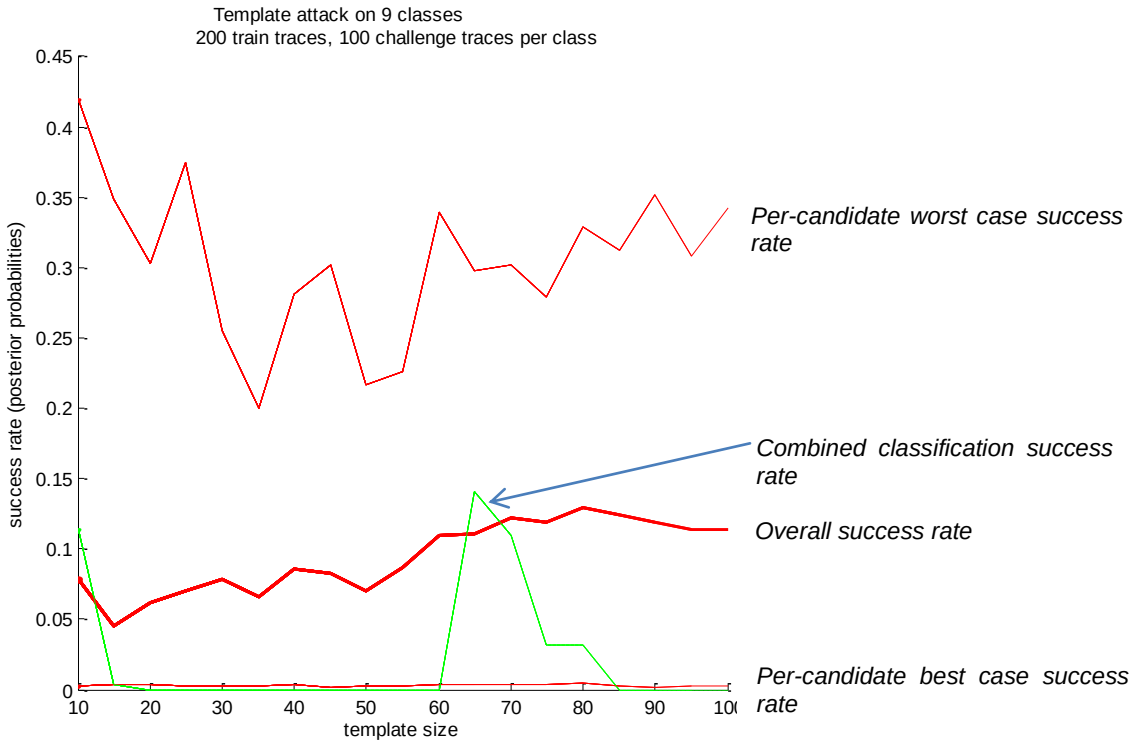

Figure 42: An example of template attack results.

Interpretation of success rates:

1. The *overall success rate*, depicted with a thick red line, shows the average success probability when an attacker uses a single challenge trace to find the secret. This is useful in the following attack scenario: an attacker first generates templates using *N* traces per each class. Then the attacker decides which candidate value is in the target device by measuring only single challenge trace from the target device and then performing the matching process. Note that this is the canonical model in which template attacks were first introduced. In many cases, this is not expected to give a better result than the combined classification success rate. However, in certain cases, for example, when a target device randomizes its processing per iteration, the combined classification success rate does not represent a realistic attack scenario.
2. The *per-candidate worst case success rate,* depicted with an upper thin red line, shows the success rate for the class that can be identified best[3]. For example, if the success rate for the class of Hamming weight 0 (it is assumed that 9 Hamming weights are used to generate templates) reaches almost to 1, it implies that the toolset can almost always identify the secret when the secret is 0.
3. The *combined classification success rate*, depicted with a dashed green line, shows the average success probability when an attacker uses multiple challenge traces as a set to find the secret. An attacker first generates templates using *N* traces per each class then decides which candidate value is in the target device by measuring *M* traces from the target device and performing the matching process using all *M* traces as a set.

[3] Note that this may not be the same class for each of the template sizes considered

4. (Optional) The O*verall Combined Classification Success rate (OCCS)*, depicted with a dashed green line, shows a mean of the Combined Classification Success rate (CCS) based on the results of the multiple experiments.
5. (Optional) The *worst-case combined classification success rate*, depicted with a dashed blue line, shows the highest CCS among multiple experiments per each template size.

Figure 43 shows an example of template attack results when overall, worst case, and best case combined classification success rates are used.

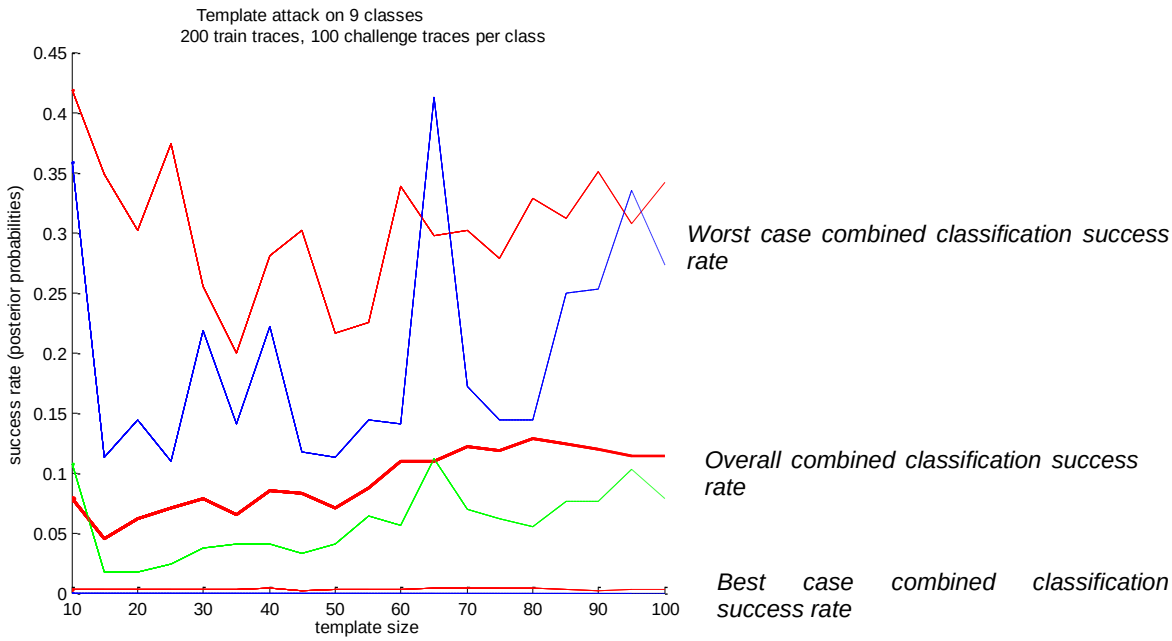

Figure 43: An example of template attack results with optional overall, worst case, and best case combined classification success rates.