HECTOR

D5.5

Final Report on Data Management

Project number:	644052
Project acronym:	**HECTOR**
Project title:	Hardware Enabled Crypto and Randomness
Start date of the project:	1st March, 2015
Duration:	41 months
Programme:	H2020-ICT-2014-1

Deliverable type:	OARD (open access to research data)
Reference number:	ICT-644052/D5.5/1.0
Work package:	WP5
Due date:	July 2018 – M41
Actual submission date:	22nd August, 2018

Responsible organisation:	TUG
Editor:	Maria Eichlseder
Dissemination level:	Public
Revision:	1.0

Abstract:	The purpose of the final report on data management is to provide an update of the analysis of the main elements of the data management policy used by the applications with regards to all the datasets that were generated by the project. The datasets collected in HECTOR include samples and statistical test results of TRNGs and PUFs, VHDL code of building blocks, measurements of passive and active physical attacks, and software to reproduce the cryptanalytic and system-level analysis conducted in HECTOR. Most important aspects regarding data management, like metadata generation, data preservation, and responsibilities, were updated compared to the initial report D5.2 (Data Management Plan) according to the outcome of the project.
Keywords:	data management policy; datasets; gathering process, accessibility, sharing, archiving

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

Editor

Maria Eichlseder (TUG)

Contributors (ordered according to beneficiary numbers)

Mario Münzer (TEC)

Josep Balasch, Dave Singelée (KUL)

Oto Petura, Ugo Mureddu, Viktor Fischer (UJM)

Gerard van Battum (BRT)

Executive Summary

This document represents the Final Report on Data Management of the HECTOR project and is an updated and extended version of the Data Management Plan (DMP) submitted in D5.2. This updated document reflects the final outcomes of the HECTOR project. In addition to the expected data items described in D5.2, we identified additional data sources, particularly software for verifying our analysis and protected implementations. In summary, the most important research data produced by the HECTOR project include output streams of TRNGs during normal operation and when applying active attacks, hardware signatures of PUFs during normal operation and when applying active attacks, leaked signal traces for side-channel analysis attacks, source code of unprotected and protected hardware and software modules, output data of statistical tests, and software for verifying and extending cryptanalytic attacks. Results of statistical tests and other relevant analysis and performance data were also published in the according deliverables for dissemination and, if applicable, in the corresponding academic publications.

This report also discusses the choices made with respect to data sharing. Two primary parameters influenced the infrastructure used for data sharing: First, whether the data should be publicly available or only shared internally between project partners; and second, the size of the generated data. The internal project SVN (https://hector.technikon.com) served as sharing platform for smaller, private data items, such as source code, which particularly profit from collaboration and versioning of files. Smaller public data items, in particular public source code for data analysis and efficient implementations, were shared via public versioning repositories on GitHub (https://github.com) and similar. Medium-sized public datasets, particularly data generated by the TRNGs and PUFs under various conditions, were published on the Zenodo platform (https://zenodo.org/communities/hector). Data sets too large for publication, such as full results of side-channel measurements that amount to terabytes of data, are stored by the responsible partners for at least three years after the project's end and are available on request.

Beyond such single static TRNG samples, we provide a webservice to request dynamically generated random output (https://trng.technikon.com). All the required information for discovering and using the publicly available data and code is provided on the HECTOR project website (https://hector-project.eu) and associated dedicated websites and repositories, such as the Ascon website (https://ascon.iaik.tugraz.at). For the data provided to the public, appropriate licensing schemes were selected in order to allow third parties to use, modify, and build on the data provided by the HECTOR project without or with reasonable restrictions. Finally, interoperability and reusability are naturally tied to the HECTOR core objectives regarding standard-compliant TRNGs and standardization processes for authenticated ciphers, which is reflected in the produced datasets.

In summary, this final DMP provides an overview of the data generated and disseminated during the HECTOR project. For dissemination, the DMP documents the selected sharing options for the data items in order to make them accessible for external partners like other research organisations or universities. Dissemination also includes publishing the results achieved during the HECTOR project in public deliverables and scientific publications. The HECTOR project homepage serves as the main point of information on publicly available data.

Contents

Chapter 1 Introduction

The data management plan (DMP) is a tool to assist in managing the data created during a collaborative research project. A first version of the DMP (HECTOR D5.2) specified at an early stage of the project what data will be generated, how it will be shared between the project partners, and to what extent it will be publicly available. Furthermore, D5.2 described the data management lifecycle and served as a reference for resource and budget allocation. The present document is the final, updated DMP (HECTOR D5.5) that summarizes the final decisions and outcomes with respect to research data in the HECTOR project.

In general, the DMP specifies what data is generated, collected, and processed during the project. It also provides information whether and how data is exploited and open for public and re-use. The DMP includes information on what standards and methodologies are used and how the data is handled during and after the research project, i.e., how the data is curated and preserved.

The data created by the HECTOR project includes:

- Bit streams generated by true random number generators (TRNGs)
- Hardware signature codes generated by physically unclonable functions (PUFs) and corresponding test results
- Results of statistical testing methods for TRNGs using AIS 31, NIST SP 800-22 and NIST SP 800-90B methodologies
- Measurement data of power consumption/electromagnetic emanation of the investigated devices observed during hardware side-channel attacks (leakage traces)
- Output data acquired during active attacks (e.g. fault attacks) targeting specific modules (e.g. TRNG, PUF)
- FPGA or ASIC specific HDL code describing modules for performing efficient cryptographic calculations, particularly authenticated encryption (including lightweight, high-performance, and protected implementations)
- Data and code for verifying fault attacks on authenticated encryption
- Data and code for system-level attacks and countermeasures
- Code for verifying and further developing cryptanalytic attacks

To a large extent, these datasets were made publicly available to the research community in order to improve the verifiability of the research and to enable researchers and practitioners to re-use the information for future applications. The datasets are stored and maintained in appropriate cloud storage solutions and online repositories (Zenodo, GitHub, …) as well as IT service hosted by partners, primarily TEC. To make the data easily accessible and discoverable, the HECTOR website links to these individual services or to other portal websites, which provide download options and incorporate detailed descriptions and instructions for the data sets. Therefore, no specific data-sharing infrastructure is required at the partner sites. This approach allows providing access to interested parties outside the project by simply sharing a URL.

It must be considered that the size of some types of generated data (e.g., random data generated by a device, device-specific leakage profiles, or other attack samples) exceed several Gigabytes (GBs) or even Terabytes (TB). In such cases, it is inconvenient to use large static samples. Instead, in such cases, only a subset of the data was shared (e.g., on Zenodo), and alternative ways of obtaining more data are provided. Examples include the source code to generate/simulate/reconstruct the necessary data, an online service to request arbitrary amounts of data (https://trng.technikon.com), or on-demand solutions for future use, such as the SCA traces available at request from BRT.

The full data is typically stored by the project partner generating it, e.g., the partner who performs side-channel measurements also stores the full corresponding leakage traces locally. Sharing the data between project partners is done on demand.

For source code created during the project, the initial DMP considered unprotected cryptographic implementations (e.g., HDL code of cryptographic modules, microcontroller code) and suggested that parts which do not include protection mechanisms against, e.g., side-channel analysis attacks, would be made available for the public to allow other interested researchers to reuse the code. This reuse results in citations for the author. On the other hand, the research community can benefit from the publicly available code in the way that implementing standard algorithms (e.g., authenticated encryption algorithms submitted to the CAESAR competition: http://competitions.cr.yp.to) from scratch becomes unnecessary. During the HECTOR project, several protected implementations were also selected to be made public. This allowed a better and more reproducible evaluation of side-channel countermeasures developed during the HECTOR project, as well as providing reference points to compare protected implementations of CAESAR candidates and thus support the decision-making process in the CAESAR competition. Furthermore, accompanying software for several publications on analysis, including microarchitectural attacks and cryptanalytic attacks, was made available to assure the reproducibility of the results and to support the development of countermeasures.

Results of side-channel analysis (SCA) attacks based on leakage traces, results of statistical tests for the TRNGs/PUFs, and implementation results of the cryptographic building blocks (area numbers, runtime) were published in deliverables and academic publications. Therefore, these numbers are accessible for interested parties outside the project.

For publicly available data, appropriate open licensing schemes such as CC-BY or appropriate equivalents for software such as the MIT or Apache licenses were selected. Interested third parties should be allowed to use, modify, and build on the provided data. CC-BY allows the reuse of the data by a third party, but the original author must be cited.

The remaining document is structured as follows. In Chapter 2, we document the data generation processes and summarize the relevant information for each type of data produced within HECTOR, including a short description, the responsible beneficiary, the intended end users, and the public availability and discoverability of the data. In Chapter 3, we provide more details and rationale for each data type, such as details on the gathering process and availability, as well as data identifiers for the individual datasets. In Chapter 4, we summarize how the data was shared, archived, and preserved. Finally, we conclude in Chapter 5.

Chapter 2 Data generation

Nr.	Responsible Beneficiary	Data set reference and name	Data set description			Research data identification				
			End user (e.g. university, research organization, SME's, scientific publication)	Existence of similar data (link, information)	Possibility for integration and reuse (Y/N) + information	D[1]	A[2]	AI[3]	U[4]	I[5]
1	UJM	Huge random bit streams and random data streams generated by proposed TRNGs in different technologies	University, research organisation, SMEs	No other similar data are available	Y; the data was used within this project for the statistical evaluation and may be reused in other projects		x	x	x	x
2	TEC	Hardware signature codes generated by proposed PUFs in individual devices	University, research organisation, consortium	Data from PUFs that were developed in the course of the FP7 project UNIQUE, http://unique.technikon.com	Y; the data was used within this project for the statistical evaluation and may be reused in other projects for advanced analysis		x	x	x	x
3	BRT	Results of TRNG statistical testing using AIS31, NIST SP 800-22 and NIST SP 800-90B methodologies	University, research organisation, SMEs	No other similar data available	Y; the data was used within this project for the statistical evaluation and may be reused in other projects		x	x		x
4	TEC	Results of PUF statistical testing	University, research organisation,	http://unique.technikon.com	Y; advanced analysis may be based on		x	x		x

[1] Discoverable
[2] Accessible
[3] Assessable and intelligible
[4] Usable beyond the original purpose of which it was collected
[5] Interoperable to specific quality standards

Nr.	Responsible Beneficiary	Data set reference and name	Data set description			Research data identification				
			End user (e.g. university, research organization, SME's, scientific publication)	Existence of similar data (link, information)	Possibility for integration and reuse (Y/N) + information	D1	A2	AI3	U4	I5
			consortium		this data					
5	BRT	Leaked signal traces observed during hardware SCA	University, research organization	Power measurements for the DPA contest, http://www.dpacontest.org/v4/rsm_traces.php	Y; Might be reused in other projects to evaluate e.g. novel attack methods		x	x		
6	UJM	Test output data acquired during active attacks on proposed modules and demonstrators	University, research organisation, SMEs		Y; may be used in other projects too			x		
7	TUG	VHDL code of building blocks for demonstration and evaluation in WP4	University, research organization	ASCON hardware implementations at GitHub, https://github.com/ascon/ascon_collection	Y; The building blocks might be reused for other projects and scientific research.	x	x	x		x
8	TUG	Protected hardware implementations of target modules	University, research organization, scientific publication; implementers and developers		Y; the building blocks might be reused for other projects and scientific research.	x	x	x	x	
9	UJM	VHDL code of TRNGs	University, research organization, SMEs		Y; may be used in other projects too	x	x	x	x	x
10	TUG	Test output data for statistical (ineffective) fault attacks	University, research organization, scientific publication		Y; the data may be used to test other statistical evaluation		x	x	x	x

Nr.	Responsible Beneficiary	Data set reference and name	Data set description			Research data identification				
			End user (e.g. university, research organization, SME's, scientific publication)	Existence of similar data (link, information)	Possibility for integration and reuse (Y/N) + information	D[1]	A[2]	I[3]	U[4]	I[5]
		(SFA, SIFA)			techniques					
11	TUG	Data and code for system-level attacks and countermeasures	University, research organization, scientific publication		Y; the tools may be useful for follow-up research and development	x	x	x	x	
12	TUG	Code for cryptanalytic attacks	University, research organization, scientific publication		Y; the tool is applicable for analysing other, similar ciphers	x	x	x	x	

Table 1: Data generation

<u>Explanation of Table 1:</u>

<u>Data set reference and name:</u>

Identifier for the data set

<u>Data set description:</u>

Description of the data that was generated or collected, its origin (in case it is collected), nature and scale to whom it could be useful, and whether it underpins a scientific publication. Information on the existence (or not) of similar data and the possibilities for integration of reuse.

<u>Research Data Identification</u>

The boxes (D, A, AI, U and I) symbolize a set of questions that should be clarified for all datasets produced in this project.

Discoverable:

Are the data and associated software produced and/or used in the project discoverable (and readily located), identifiable by means of a standard identification mechanism (e.g. Digital Object Identifier)

Accessible:

Are the data and associated software produced and/or used in the project accessible and in what modalities, scope, licenses (e.g. licencing framework for research and education, embargo periods, commercial exploitation, etc.)

Assessable and intelligible:

Are the data and associated software produced and/or used in the project assessable for and intelligible to third parties in contexts such as scientific scrutiny and peer review (e.g. are the minimal datasets handled together with scientific papers for the purpose of peer review, are data provided in a way that judgements can be made about reliability and the competence of those who created them)?

Useable beyond the original purpose for which it was collected

Are the data and associated software produced and/or used in the project usable by third parties even long time after the collection of the data (e.g. is the data safely stored in certified repositories for long term preservation and curation; is it stored together with the minimum software, metadata and documentation to make it useful; is the data useful for the wider public needs and usable for the likely purposes of non-specialists)?

Interoperable to specific quality standards

Are the data and associated software produced and/or used in the project interoperable allowing data exchange between researchers, institutions, organisations, countries, etc. (e.g. adhering to standards for data annotation, data exchange, compliant with available software applications, and allowing re-combinations with different datasets from different origins?)

It is recommended to make an “x” to each applicable box and explain it literally in more detail afterwards.

Chapter 3 Processing and explanation of generated data

The following sections provide some additional information to the listed data introduced in Chapter 2. This information includes the entity which is responsible for the data, how the data is collected, an identification of the end-users of the data, and research data identification.

3.1 Huge random bit streams generated by proposed TRNGs in different technologies

3.1.1 Responsible Beneficiary

Random data was generated and recorded by the parties performing evaluations of random number generators. This task was mainly performed by UJM, so they take the main responsibility of the data. Additional data was also produced by other parties, mainly KUL.

3.1.2 Gathering Process

Random data was essentially generated using HECTOR evaluation boards and demonstrators in various conditions, including border and corner operating conditions. Two types of data were generated: the raw random data streams and the post-processed random data streams. Random data can be bits, bytes, 16- or 32-bit words. Two data formats are available: the binary stream and the stream of random words (bytes, 16- or 32-bit words). Stream of random words can be useful for example when the raw random data is the output of a counter of random events.

The raw random bit stream files have extension *.rbs, the raw random data stream files have extension *.r08, *.r16 or *.r32 for data streams with bytes, 16- and 32-bit words, respectively. The post-processed bit stream files have extension *.pbs and the post-processed data stream files have extension *.p08, *.p16 or *.p32.

Random bit stream files with extension *.rxx or *.pxx (raw bit streams or post-processed bit streams) represent the most common file format, since this format is required by most general-purpose statistical tests (e.g. AIS 31, NIST SP 800-90B or NIST SP 800-22).

In generation and evaluation of random numbers, the order of bits, bytes and words is important, since it can change the existing pattern (if there is some). Random bytes are written into the files in the same order as they arrive. Bits are placed into the bytes in the following manner: the first arrived bit is placed to the least significant bit and the last arrived bit to the most significant bit, i.e. byte=bit8|bit7|bit6|bit5|bit4|bit3|bit2|bit1. The 16-bit words have the following format: word16=byte2|byte1 and the 32-bit words are as follows: word32= byte4|byte3|byte2|byte1.

3.1.3 End-User of the Data

The end-users of this type of data are mainly the producers of data and other partners of the HECTOR project. It can happen that the generated data would need to be shared with another institution. The data file sizes of at least 2 MB will be needed for applying the AIS20/31 tests, sizes of 1 MB for applying the NIST SP 800-90B test suite and thousands of files of 125,000 bytes for applying the NIST SP 800-22 test suite. The technique to share the data depends on the amount of data. A small amount (<100 MB) can be shared using the existing SVN. For medium amounts (<1 GB) some cloud storage infrastructure might be applied. Huge amounts (>10 GB) might require sharing USB sticks or external hard disks.

To allow parties outside the project to evaluate the generated data, the data files have been made publicly available. Depending on the size of the measurement data, only subsets of the data files have been publicly shared. All information concerning data acquisition are available at the same place where the generated data can be downloaded. If one interested party requires the full set of measurement data, a custom sharing method can be set up.

3.1.4 *Research Data Identification*

Some of the TRNG output data will not be discoverable in public search engines or in a global registry of research data repositories, but within the consortium internally. It is accessible by means of an existing project subversion repository or if necessary, exchanged via data storage media. The quality and reliability of the data can be evaluated by statistical evaluation. The data may be useable in upcoming projects as well, but the purpose of the data will not change from a present-day perspective. TRNG data are used within frameworks that allow the interoperability between the existing components based on the conformity to the same standards.

- To support several of the HECTOR publications, selected datasets have been made available publicly. These datasets are discoverable via the publications, search engines, and Zenodo HECTOR community: Raw output from an elementary TRNG on a Xilinx Spartan-6 FPGA: https://doi.org/10.5281/zenodo.154484
  Associated with publication https://doi.org/10.5281/zenodo.55455
- Biased bit data streams without correlations and unbiased bit data streams with bit correlations:
  https://doi.org/10.5281/zenodo.1286749
  Associated with publication 2017/15 (https://doi.org/10.5281/zenodo.1286723)
- Raw random data obtained from the DC-TRNG on Cyclone IV and Cyclone V Intel FPGAs:
  https://doi.org/10.5281/zenodo.1287612
  Associated with publication 2018/4 (https://doi.org/10.23919/DATE.2018.8342256)
- Random data obtained from the TERO TRNG, manufactured in a 28 nanometer STM technology:
  https://doi.org/10.5281/zenodo.59407
  Associated with publication 2015/1 (https://doi.org/10.5281/zenodo.60900)
- A web service to trigger the creation of a true random bitstream (up to 8,192 bytes per visit) and to download the data file was established by TEC at https://trng.technikon.com.

3.2 Hardware signature codes generated by proposed PUFs in individual devices

3.2.1 *Responsible Beneficiary*

The data of Physically Unclonable Functions (PUFs) was generated and recorded by parties performing evaluations on the data, mainly by TEC and KUL. The driving partner is TEC in this context.

3.2.2 *Gathering Process*

There were several different sources discussed which might be used for PUF data generation, such as 65nm PUF ASICs including SRAM, Latch, D Flip-flop, Buskeeper, Arbiter and Ring Oscillator PUFs (all developed during the FP7 project UNIQUE). Another possible source is an FPGA structural and behavioural emulation of an SRAM-like PUF implemented in VHDL by TEC (realized during the FP7 project HINT). However, at the later stage of HECTOR, the TERO-PUF (Transient Effect Ring Oscillator PUF) was developed and used as the main intrinsic source for the demonstrator D2 and D3.

PUF raw data are *.bin files, which are read by a MATLAB or R script, to subsequently perform statistical analysis. The sequence of bytes needs to be converted from a hexadecimal or decimal form to binary bit strings. When performing statistical analysis, the output parameters are stored within a structure array that can be saved within a text-based file (*.txt). When performing statistical tests on PUF data, a lot of data is required and needs to be shared. This might lead to an exchange of the data via USB sticks or external hard disks. If PUF data is only used for a low number of reconstructions within a framework, a small amount of responses can easily be shared via the existing SVN or a cloud storage infrastructure. A *.txt file with 5,000 responses, each with a size of 128 bits amount a file size of about 167 KB.

3.2.3 End-User of the Data

The end-user of this type of data are mainly the partners within the HECTOR project or organisations that perform analysis on PUF data.

The *.txt file with the stored responses as well as the *.mat file with the results of the statistical analysis need to be combined with a read-me file that describes the structure of the stored variables. Most of the resulting data including detailed explanations is incorporated within (public) deliverables. The primary end-users of the results are, as already mentioned, the project partners. However, universities or research organisations that may use this data to build additional statistical analysis on the given results or use them for comparisons, are considered as end-user as well. External end-user may also build up new analysis on already existing results or use the raw data for their own evaluations.

3.2.4 Research Data Identification

The PUF data is not discoverable in public search engines or in a global registry of research data repositories, but within the consortium internally. It is accessible by means of the existing project subversion repository or if necessary, exchanged via data storage media. The quality and reliability of the data can be statistically evaluated. The data may be useable in upcoming projects as well, but the purpose of the data will not change from a present-day perspective. In this context, PUF data are used within frameworks that allow the interoperability between the existing components based on the conformity to the same standards.

Published datasets by KUL:

- Monte Carlo PUF data obtained with different design parameters and temperatures on Xilinx Spartan-6 FPGA:
  https://doi.org/10.5281/zenodo.1306083
  Associated with publication 2017/11 (https://doi.org/10.5281/zenodo.897887)

Datasets used for internal statistical evaluation by TEC:

PUF data (5000 samples of 128 bits, collected under different temperature conditions): Stored in the internal project SVN repository in the context of WP4 and its deliverable D4.2 (accepted and publicly available deliverables accessible via https://hector-project.eu/publications-deliverables/deliverables). Those datasets were used for the statistical evaluation documented in deliverable D2.4.

3.3 Results of TRNG statistical testing using AIS 31, NIST SP 800-22 and NIST SP 800-90B methodologies

3.3.1 Responsible Beneficiary

The results of the statistical testing were mainly produced by those who performed evaluations on TRNG data (BRT, UJM, KUL, STR) described in Section 3.1. The responsible beneficiary is BRT.

3.3.2 Gathering Process

Test outputs (test results) are produced from TRNG output data described in Section 3.1. Results of statistical testing using AIS 31, NIST SP 800-90B and NIST SP 800-22 methodology are generated by corresponding standard tests as log (text) files. It is important to maintain the link between tested data and test output using convenient file naming. The filename before the extension must therefore be the same for input and output data of each test.

Output of tests of the raw data either use the .bin extension, or have one of the following file extensions that indicates the applied test suite:

*.r31 – for the AIS31 test suite output,

*.r22 – for the NIST SP 800-22 test suite output,

*.r9i – for the NIST SP 800-90B test suite for iid data,

*.r9n – for the NIST SP 800-90B test suite for non-iid data.

Correspondingly, output of tests of the post-processed data will have file extension:

*.p31 – for the AIS31 test suite output,

*.p22 – for the NIST SP 800-22 test suite output,

*.p9i – for the NIST SP 800-90B test suite for iid data,

*.p9n – for the NIST SP 800-90B test suite for non-iid data.

Since the NIST SP 800-90B test suite needs a different input data format: one random sample per output byte (or two-byte word) must be saved. The data formats described in Section 3.1 need to be converted to this specific format.

3.3.3 End-User of the Data

Most of the resulting data including detailed explanations are incorporated within (public) deliverables. The primary end-users of the results are the project partners and/or universities or research organisations that may use this data to build additional statistical analysis on the given results or use them for comparisons in follow-up work. External end-users may also build up new analysis on already existing results or use the raw data for their own evaluations. The evaluations conducted by STR are confidential as they were gathered through a customer-provided firmware and were used for the customer's purposes.

3.3.4 Research Data Identification

Most of the results of the statistical evaluations have been published via the Zenodo platform with the necessary metadata and are thus discoverable in public search engines. A few results, particularly STR's results with a customer product, are not accessible, but are discussed in a public deliverable. Because of the small to medium size of most output data, the data can be made accessible by means of the existing project subversion repository or the Zenodo platform. UJM's test results are significantly larger (5GB) and have thus been made available to the consortium partners via a separate data repository. The realization and the results of the statistical tests have been published together with scientific papers and deliverables within the project. Therefore, the produced data can be assessed. The interoperability is given with the exchange of the statistical evaluation between researchers.

Published data:

- The research data on the TRNG active perturbation testing on the post processing is stored in https://doi.org/10.5281/zenodo.1319588. This data is used for the work in D2.4, section 8.2.
- The research data on the TRNG Radio Frequency injection perturbation testing: https://doi.org/10.5281/zenodo.1319588. This data is used for the work in D2.4, section 8.3.
- The BRT TRNG environmental testing data on Demonstrator 1 can be found in: https://doi.org/10.5281/zenodo.1319588. This data is used for the work in D4.3, Appendix A.

- The research data by KUL on the statistical testing (AIS-31 and NIST SP800-90B) of the DC-TRNG can be found in: https://doi.org/10.5281/zenodo.1323500. This data is used for the work in D2.4, section 2.2.
- UJM and MIC's statistical testing data for the PLL-TRNG implemented for the Cyclone V FPGA platform is discussed in D2.4, Section 2.3 (AIS 20/31 and NIST SP800-90B). The raw data has been made public via the Zenodo HECTOR community: https://doi.org/10.5281/zenodo.1400616.

Other data discussed in deliverables:

- STR's statistical testing data for the PLL-TRNG implemented in an ASIC for automotive applications is discussed in D2.4, Section 2.4 (AIS 20/31 and NIST SP800-90B). The raw data is confidential (see Section 3.3.3).

3.4 Results of PUF statistical testing

The statistical evaluation of PUFs is closely tied with and depends on the generation of PUF hardware signature codes, as described in Section 3.2. The responsible beneficiaries, gathering process, expected end users, and data identification are thus very similar to Section 3.2.

3.4.1 Responsible Beneficiary

The results of the statistical testing are mainly produced by those who perform evaluations on PUF data (i.e., TEC, KUL), as described in Section 3.2.

3.4.2 Gathering Process

PUF raw data are *.bin files, which may be read by a MATLAB or R script, to subsequently perform statistical analysis. The sequence of bytes needs to be converted from a hexadecimal or decimal form to binary bit strings. When performing statistical analysis, the output parameters are stored within a structure array that can be saved within a *.mat file.

For the TERO PUF evaluation, 16 daughter boards and 8 mother boards were analysed based on 5000 responses of 128 bits for each board. Selected tests were performed under different temperature conditions. Evaluated properties include intra-device distance (bit error rate, hamming weight, probability of failure, dark bit selection), inter-device distance, and correlation evaluation (bit frequency, autocorrelation).

3.4.3 End-User of the Data

The *.mat file with the resulting parameters of the statistical analysis needs to be combined with a read-me file that describes the structure of the stored variables. Most of the resulting data including detailed explanations is incorporated within (public) deliverables and internal reports. The primary end-user of the results are the project partners and universities or research organisations that may use this data to build additional statistical analysis on the given results or use them for comparisons. External end-user may also build up new analysis on already existing results or use the raw data for their own evaluations.

3.4.4 Research Data Identification

The results of the statistical evaluations are not discoverable in public search engines or in a global registry of research data repositories, but within the consortium internally. Because of the small size of the output data, the data can be easily made accessible by means of the existing project subversion repository. The realization and the results of the statistical tests have been reported in reports and deliverables within the project. Therefore, the produced data can be assessed. The interoperability is given with the exchange of the statistical evaluation between researchers.

Datasets and analysis results used for internal statistical evaluation by TEC:

- Analysis of PUF data from Section 3.2 (5000 samples of 128 bits, collected under different temperature conditions): The analysis results are stored together with the collected samples in the internal project SVN repository in the context of WP4 and the D4.2 (https://hector-project.eu/publications-deliverables/deliverables).
  The results of the statistical evaluation, including the relevant metadata, are documented in detail in an internal report for WP4 (D4.2 context), as well as in deliverable D2.4 (https://hector-project.eu/publications-deliverables/deliverables).

3.5 Leaked signal traces observed during hardware side channel attacks

3.5.1 Responsible Beneficiary

Leakage signal traces have been recorded by the parties performing evaluations of the side-channel resistance of specific cryptographic building blocks. This task has mainly been performed by BRT, so they take the main responsibility of the data. Similar data was also produced by other parties, e.g. KUL, as these parties also have expertise in side-channel measurements.

3.5.2 Gathering Process

Leaked signal traces are typically recorded using an oscilloscope, independent of whether power measurements or EM measurements are performed. Modern digital oscilloscopes allow storing the captured traces in different file formats. Such file formats include CSV (comma separated file), MAT (MATLAB data file), or a proprietary format. Since most of the formats can be easily converted into other formats it is not necessary for the different parties to agree on a common format. In case of proprietary formats (BRT and KUL, respectively), a conversion tool is provided to the partners of the consortium.

3.5.3 End-User of the Data

The end-users of this type of data are mainly the producers itself. It is common that the institution measuring the side-channel information also evaluates the amount of leakage which can be extracted out of the measurements by applying methods like differential power analysis (DPA), template attacks (TA) and others. At BRT the data sets are collected and stored in a proprietary file format (.bf), suited for the BRT proprietary analysis software. At KUL the data sets are similarly collected and stored in a custom file format (.bin), which can be loaded into Matlab (or other software) using a simple conversion script. These trace files contain all meta-data necessary for the analysis. At BRT, in addition, result files of operations are also stored (in a different file format, .bfr), along with information on the operation. Of course, cases might arise where the measurement data has to be shared with another institution wanting to test and apply novel analysis methods. Here, the technique to share the data highly depends on the amount of data. A small amount (<100 MB) can be shared using the existing SVN. For medium amounts (<1 GB) some cloud storage infrastructure can be applied. Huge amounts (>10 GB) require sharing USB sticks or external hard disks. The latter is the case for the BRT and KUL side channel analysis data.

To allow parties outside the project to reproduce the side-channel analysis results or to apply new methods, the leakage traces can be made available. Due to the large file sizes, no data is publicly shared. All information required to use the measurements (e.g. corresponding plain text and cipher text to each leakage trace) are available as meta data in the files themselves. Other relevant information (e.g. oscilloscope model which has been used for capturing the data, measurement parameters) is reported in the deliverables. If one interested party requires the full set of measurement data, a custom exchange method can be set up.

During the side channel analysis measurements on the HECTOR primitives, a set of 17 TB of measurement data is collected, of which the single largest file is more than 7 TB. Most of this data is in the proprietary file format of BRT (.bf, .brf) and can be processed and analysed with the BRT proprietary tool 'Sideways 3'. The remaining data (1.5 GB) is in the proprietary .bin format of KUL. This data is archived at BRT, respectively KUL, on a set of hard disks and can be made available at request. A tool is available and will be provided with the data to extract the raw data from the files. For information please contact BRT and KUL.

The results of the side-channel analyses have been reported in (public) deliverables. So additional end-users of the results will be project partners and/or universities or research organisations that may use this data to perform additional side-channel analysis with the given measurements, or use them for comparisons.

3.5.4 Research Data Identification

The leaked signal traces are not discoverable in public search engines or in a global registry of research data repositories, but announced within the consortium internally. Because of the large size of the data, sharing it using the existing project subversion is not applicable. The size even exceeds several terabytes, so exchange of physical data requires storage devices like backup tapes or hard disks. Results of side-channel analysis based on specific leakage traces have been published in the deliverables within the project. Therefore, the achieved results based on the measurement data can be assessed.

Stored side channel measurement data at BRT contains:

- Side channel analysis measurement traces for the analysis work in D2.4 (SCA on on-line test, Chapter 7.1; SCA on on-line entropy test, Chapter 7.2)
- Side channel analysis measurement traces for the analysis work in D4.3 (Keyboard emanation, Appendix C; Display emanation, Appendix D)

Stored side channel measurement data at KUL contains:

- Side channel analysis measurement traces for the analysis work in D3.3 (Security degradation of side-channel countermeasures)

3.6 Test output data acquired during active attacks on proposed modules and demonstrators

Investigations of the influence of active attacks on the hardware signature codes of PUFs and the random bit streams generated by the TRNGs were performed during the HECTOR project. The goal was to evaluate to what extent the investigated PUF/TRNG modules are vulnerable to active attacks in order to include appropriate countermeasures. Format and gathering process of the output data do not change when applying active attacks so for a detailed description to the corresponding data formats we refer to Section 3.1 for the TRNG case and to Section 3.2 for the PUF case, respectively.

Type and parameters of the active attacks are important information for further analyses and for development of countermeasure. This additional information is incorporated to the dataset description and poses the main difference to the data sets recorded without active attacks.

The following published datasets contain output data collected during active attacks:

- Output from an elementary TRNG on a Xilinx Spartan-6 FPGA with an oversampled clock rate:
  https://doi.org/10.5281/zenodo.154484
  Associated with publication https://doi.org/10.5281/zenodo.55455
- Output from two TERO TRNG implementations on a Xilinx Spartan-6 FPGA at different Vdd levels:
  https://doi.org/10.5281/zenodo.154524
  Associated with publication https://doi.org/10.5281/zenodo.154591

We refer to Section 3.10 for more data collected during active attacks on other primitives, particularly authenticated encryption algorithms.

3.7 VHDL code of building blocks for demonstration and evaluation in WP4

3.7.1 Responsible Beneficiary

VHDL code for cryptographic building blocks was mainly developed by the parties KUL, STI and TUG. As the editor of D3.2 "Building Blocks for WP4" and designer of Ascon, one of the main building blocks to be implemented for WP4, TUG is the responsible beneficiary.

3.7.2 Gathering Process

Hardware building blocks are modelled using a hardware description language (HDL) such as VHDL or Verilog. For more complex building blocks, the source code can be divided into several files which then form a project. Projects are accompanied by a short readme file explaining the file structure providing a quick overview of the project.

Additional software modules (C/C++, Java) for use on microcontrollers or as reference implementations for the hardware were also developed.

In more detail, the following implementations were prepared for WP4:

- **Advanced Encryption Standard: AES-128**: According to the requirements in AIS 20/31 PTG.2 it is not allowed to use the same cryptographic primitive for RNG post-processing and data encryption/decryption. Therefore, the HECTOR consortium did not reuse Ascon/Ketje in demonstrator D2: Secure USB Stick and D3: Secure Messaging Device for the post processing of the RNG output. Instead, AES-128 was implemented.
- **Authenticated Encryption Algorithms: KETJE**: Ketje is a family of algorithms for authenticated encryption, which share the same permutation-based structure. All instantiations of Ketje are aimed at memory-constrained devices and strongly rely on nonce uniqueness for security. Ketje was designed and submitted to round 3 of the CAESAR competition by Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche, Ronny Van Keer.
- **Authenticated Encryption Algorithms: ASCON**: Ascon is a family of authenticated encryption algorithms, currently participating as a finalist in the CAESAR competition. The Ascon family was designed to be lightweight and easy to implement, even with added countermeasures against side-channel attacks. Ascon was designed by a team of cryptographers from Graz University of Technology (Christoph Dobraunig, Maria Eichlseder, Florian Mendel and Martin Schläffer).

The AEAD implementations follow the CAESAR Hardware API v1.0.3 proposed by Project ATHENA (http://cryptography.gmu.edu/athena/index.php?id=download), as required for the hardware implementations of CAESAR candidates for Round 3.

3.7.3 End-User of the Data

Several end-users can be identified for the cryptographic hardware building blocks. First, they serve as building blocks for WP4 and have been integrated by MIC and TCS. Second, the implementations provide performance figures to compare the selected algorithms with alternatives and thus judge their usefulness for different application scenarios, which is useful for any potential future user of the algorithms, as well as for providing insights for future designs. Both the performance figures and the actual implementations were made publicly available for these purposes. Third, the implementation allows to evaluate their resistance against implementation attacks such as differential power analysis (DPA) attacks or fault attacks, and compare it with protected implementations or evaluate the efficiency of different attacks and integrated countermeasures.

For CAESAR candidates of Round 3, the process required designers to submit hardware implementations of each cipher in a specific framework, the CAESAR Hardware API (http://cryptography.gmu.edu/athena/index.php?id=download), to help implementers and allow a performance comparison using different metrics on various platforms. In addition to the Ascon implementation provided here, the CAESAR Hardware API / GMU ATHENA team also provided an implementation (https://cryptography.gmu.edu/athena/index.php?id=CAESAR_source_codes). The resulting performance is extensively evaluated by the GMU ATHENA project for different FPGAs (https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/table_view).

In the initial version of the DMP, the following guidelines for deciding which implementations would be made public were proposed: "If the implementation does not include countermeasures and is likely to be reused by other parties for comparison reasons or as foundation for integrating improvements, it will be made publicly available. Implementations including specific countermeasures against implementation attacks will not be made publicly available, they can be shared using the internal project SVN service."

However, to provide verifiable data for how efficient different countermeasures (developed within HECTOR) are, and how well-suited the target algorithm Ascon is for protected implementations, several protected implementations were also developed and published, as detailed in the next section.

3.7.4 Research Data Identification

The implementations used for WP4 can be discovered via the HECTOR website at https://hector-project.eu/news/rnm:

- **Advanced Encryption Standard AES-128**:
  https://hector-project.eu/downloads/AES_HECTOR.zip
- **Authenticated Encryption Algorithm Ketje**:
  https://hector-project.eu/downloads/Ketje_Eval_pfm.zip
- **Authenticated Encryption Algorithm Ascon**:
  https://hector-project.eu/downloads/Ascon_Eval_pfm.zip
  The original implementations that this project is based on have also been updated and are available on GitHub, as well as discoverable from the Ascon website (https://ascon.iaik.tugraz.at):
  https://github.com/IAIK/ascon_hardware
  This implementation and its design choices are documented by the accompanying hardware design document:
  https://github.com/IAIK/ascon_hardware/blob/master/doc/ascon_hw_doc.pdf
  The evaluation of the CAESAR Hardware API implementation and comparison with other CAESAR candidate is available via the GMU ATHENA project:
  https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/table_view

The publicly available source code is discoverable via the HECTOR website, the Ascon website (for Ascon), and by public search engines using the name of the implemented algorithm. Derived information, particularly performance statistics, are published on the Ascon website and as part of the Ascon updated design documents and other dissemination material.

3.8 Protected hardware implementations of target modules

3.8.1 Responsible Beneficiary

As discussed in Section 3.7.3, this data was not foreseen for publication in the original DMP, but has been made available to allow a better evaluation of both the protection mechanisms (WP3) and the cipher (in the CAESAR context). TUG is the responsible beneficiary.

3.8.2 Gathering Process

The gathering process and tools are similar to Section 3.7.2. The data consists of VHDL code; implementation-specific details and toolchain are documented in detail in the corresponding research papers. The following protected implementations are published:

- **Domain-Oriented Masking (DOM)-protected AES** implementation, configurable for arbitrary protection orders
- **DOM-protected Ascon** implementation
- **Unified Masking Approach (UMA)-protected Ascon** implementation

3.8.3 End-User of the Data

The end users of the protect implementations are similar, but not identical to those of the unprotected implementations discussed in Section 3.7.3. The end-user's interests may be

- **Protection mechanisms:** The implementations are protected by two newly developed types of masking schemes, namely Domain-Oriented Masking (DOM) and the Unified Masking Approach (UMA). The protected implementations of different example ciphers (AES, Ascon) allow to evaluate the efficiency and correctness of the countermeasures in different contexts. This is useful for reviewers, other researchers working on follow-up work, but also developers and implementers from academia and industry.
- **Cipher:** The main target cipher of the protected implementations is Ascon. These implementations demonstrate how well-suited Ascon is for efficient protection mechanisms and allow the measure the cost (including area and randomness cost) of implementation for different protection orders. This is a relevant decision criterion in the context of the CAESAR competition.

3.8.4 Research Data Identification

The data is discoverable via links (from the paper, the Ascon website, etc.):

- DOM-protected AES implementation: https://github.com/hgrosz/aes-dom
  Associated with publication 2017/07 (https://doi.org/10.5281/zenodo.574261)
- UMA- and DOM-protected Ascon implementations: https://github.com/hgrosz/ascon_dom
  Associated with publications 2017/10 (https://doi.org/10.5281/zenodo.897935) and 2018/07 (https://doi.org/10.1007/s13389-018-0184-y)

The resulting performance numbers for Ascon are documented in the research papers, as well as on the Ascon website (https://ascon.iaik.tugraz.at).

3.9 VHDL code of TRNGs

3.9.1 *Responsible Beneficiary*

The scope and detail of this data is beyond what was foreseen in the original DMR. UJM is the developer and responsible beneficiary.

3.9.2 *Gathering Process*

The gathering process is similar to the cryptographic building blocks as documented in Section 3.7.2. In detail, the data covers AIS-20/31 compliant TRNG cores and includes schematics and VHDL code of the following TRNG designs:

- **Ring Oscillator (RO) based elementary TRNG**
- **Ring Oscillator (RO) coherent sampling based TRNG**
- **Multi-Ring Oscillator based TRNG**
- **Coherent Sampling based TRNG using PLLs**
- **Transition Effect Ring Oscillator based TRNG**
- **Self-Timed Ring based TRNG**

The work targets three FPGA families:

- **Xilinx Spartan 6**
- **Altera Cyclone V**
- **Microsemi SmartFusion 2**

The implementations are documented in detail on the dedicated website and in the associated research paper (https://zenodo.org/record/375453).

3.9.3 *End-User of the Data*

The open-source AIS-20/31 compliant implementations of the HECTOR TRNGs are useful for academic and industrial researchers alike and may be used in future products.

3.9.4 *Research Data Identification*

The data is accessible on GitHub and discoverable via the dedicated website:

- VHDL code for TRNGs: https://gitlab.univ-st-etienne.fr/oto.petura/hector_trng_designs.git
  Associated with publication 2016/11 (https://zenodo.org/record/375453)
  Discoverable via and documented in the research paper and the dedicated website:
  https://labh-curien.univ-st-etienne.fr/cryptarchi/HECTOR_TRNG_designs/

3.10 Test output data for statistical fault attacks (SFA) and statistical ineffective fault attacks (SIFA)

3.10.1 *Responsible Beneficiary*

This data goes beyond the data foreseen in the original DMP for physical attacks (Section 3.6) and is based on TUG's research on novel fault evaluation techniques. TUG is the responsible beneficiary.

3.10.2 *Gathering Process*

The data consists of faulted ciphertexts (given as human-readable .txt files as CSV). The statistical evaluation of these ciphertexts is provided in two associated research papers. The public data covers different setups and evaluation methods:

- **Statistical Fault Attacks (SFA) on Authenticated Ciphers (AEAD)**:
  - **Set 1**: Laser fault injections targeting an AES co-processor on a smartcard microcontroller.
  - **Set 2**: Clock tampering targeting an AES co-processor implemented on a general-purpose microcontroller.
  - **Set 3, 4**: Clock tampering targeting an AES software implementation (AVR crypto lib ASM) implemented on a general-purpose microcontroller (ATXmega256A3).

  The data includes the C++ code for a key-recovery attack which takes the faulty ciphertexts as inputs and recovers the used secret key using the SFA techniques described in the associated research paper (https://doi.org/10.5281/zenodo.154485).
- **Statistical Ineffective Fault Attacks (SIFA)**: Ineffectively faulted AES Ciphertexts for different platforms (Atmel AVR: ATXmega256A3 and ATXmega128D4, ARM Cortex-M4: STM32F3) and with different fault countermeasures in place (including infection-based configurations). Target platforms and setups are described in more detail in the associated research paper 2018/15 (in press, see https://eprint.iacr.org/2018/071) and include:
  - **8**-bit Software AES on ATXmega256A3
  - **32**-bit bitsliced Software AES on STM32F3
  - Hardware Co-Processor AES on ATXmega256A3

3.10.3 End-User of the Data

The data is primarily necessary to fully verify the proposed novel evaluation techniques of fault attacks.

3.10.4 Research Data Identification

- Test data for Statistical Fault Attacks (SFA) on different authenticated ciphers (AEAD) and their building blocks: https://doi.org/10.5281/zenodo.154487
  Associated with publication 2016/12 (https://doi.org/10.5281/zenodo.154485)
- Test data for Statistical Ineffective Fault Attacks (SIFA):
  https://doi.org/10.5281/zenodo.1314447
  Associated with publication 2018/15 (in press, CHES 2018, https://eprint.iacr.org/2018/071)

3.11 Data and code for system-level attacks and countermeasures

3.11.1 Responsible Beneficiary

The data was not discussed in the original DMP, as the nature and scope of the research results was not predictable. TUG is the responsible beneficiary.

3.11.2 Gathering Process

For system-level and microarchitectural attacks on CPU, Cache and RAM, proofs of concept are crucial for understanding and verifying the attack. These are usually written in C/C++ and/or Assembly, and sometimes highly platform-dependent. In some cases, the attack code is short enough to include in the paper; otherwise, the code needs to be available both during the peer-review process and to the public. In case a vulnerability is reported in a Responsible Disclosure process, embargos may prevent a timely and complete dissemination of results.

The research in HECTOR (WP3, WP4) contributed to several such attacks, as listed below:

1. **KeyDrown**: This work (paper published at https://doi.org/10.14722/ndss.2018.23027) proposes a countermeasure against certain timing-based side-channel attacks that aim to intercept keystroke timings. The countermeasure consists of three defence layers:

   1. Layer: core layer, responsible for interrupt and key injection
   2. Layer: protects the input handling library (libgtk/libgdk on x86, libinput on Android)
   3. Layer: protects the UI widgets used in applications.

   A proof-of-concept is available for both desktop and mobile setups (tested on Ubuntu 16.04 x86_64, Android 6.0.1 on an LG Nexus 5, LineageOS 7.1.1 on a OnePlus 3t).

2. **Rowhammer Bitflips**: This work (paper at https://doi.org/10.1109/SP.2018.00031) investigates countermeasures against Rowhammer attacks, and shows that they are significantly less effective than advertised by providing advanced Rowhammer attack variants:
   1. One-location hammering. A tool is provided to test whether a system is susceptible against one-sided hammering.
   2. Linux memory waylaying. A tool is provided that implements waylaying ("check tool" prints the physical address of the first page of the specified binary) and memory chasing ("relocate tool" forces relocation of a binary page).
3. **Prefetch Side-Channel Attacks**: This work (https://doi.org/10.5281/zenodo.375513) investigates prefetch side-channel attacks. Machine-/System-/library-/version-specific data is generated as a part of the attack. The results from measurements cannot directly be applied to other systems/libraries/versions or machines, so the source code to generate the dataset is published.

3.11.3 *End-User of the Data*

The primary purpose of the data is to understand and verify the proposed countermeasures and attacks, so other researchers are the main end-users. The data is highly relevant for vendors and developers of affected software and hardware. The derived introductory material (tutorial and proof-of-concept videos) are important for education and dissemination. Additionally, the test tools and countermeasure proofs-of-concept are useful for concerned software users and for developers who consider integrating the countermeasures.

3.11.4 *Research Data Identification*

The following data and code has been published:

- Code to generate data for prefetch side-channel attacks on different architectures:
  https://doi.org/10.5281/zenodo.375521
  Associated with paper 2016/17 (https://doi.org/10.5281/zenodo.375513)
  Discoverable via a URL in the paper and dissemination activities (Twitter, …)
- Proof-of-concept for KeyDrown countermeasure on different architectures:
  https://github.com/IAIK/keydrown
  Associated with paper 2018/01 (https://doi.org/10.14722/ndss.2018.23027)
  Discoverable via a URL in the paper and dissemination activities (Twitter, …)
- Tools to test susceptibility of different systems to Rowhammer attack variants:
  https://github.com/IAIK/flipfloyd
  Associated with paper 2018/08 (https://doi.org/10.1109/SP.2018.00031)
  Discoverable via a URL in the paper and dissemination activities (Twitter, …)

3.12 Code for cryptanalytic attacks

3.12.1 Responsible Beneficiary

The data was not discussed in the original DMP, as the nature and scope of the research results was not predictable. TUG is the responsible beneficiary.

3.12.2 Gathering Process

In WP3, several lightweight designs were analysed for their cryptanalytic security. Where possible, parts of the attack were implemented to validate the theoretical analysis. The resulting code includes

1. **Models** of the ciphers' cryptanalytic (differential, linear, …) behaviour in suitable search frameworks, in particular Mixed-Integer Linear Programs (MILP) and Boolean satisfiability (modulo theories) (SAT/SMT). MILP models are short enough to include directly in the paper.
2. **Dedicated search tools** for analysis, written in C/C++ or Python, which can be reused for other ciphers and should thus be published, but are too large to include in the paper.
3. **Verification** code which executes or simulates part of the attack to validate statistical assumptions. Since the executed process is described in detail in the paper, while the actual execution is randomized and depends on too much data to publish, the most important data to be published in this case is a detailed evaluation of the simulation results.

For the analysis of the lightweight tweakable block cipher MANTIS, both **(1) Models** and **(3) Verification** were published as part of academic papers and a doctoral thesis, including source code attached as supplementary material to the submission, whereas **(2) Dedicated search tool** was made freely available online.

3.12.3 End-User of the Data

The published and documented code and the data it produces are useful for other researchers, particularly cryptanalysts and designers, to

- Verify our analysis results,
- Evaluate other ciphers in similar ways, and
- Design new ciphers resistant to this type of analysis.

3.12.4 Research Data Identification

The data is associated with papers 2018/14 (https://doi.org/10.13154/tosc.v2018.i2.111-132) and 2017/08 (https://doi.org/10.13154/tosc.v2016.i2.248-260) and has been published on GitHub:

- https://github.com/dkales/clusterfk

The data is discoverable via the URL provided in the academic paper and published under an open-source license to allow reuse, modification, etc.

Chapter 4 Research data in publications

In this chapter, we list all publicly available research data items that are directly linked to academic publications. This both supports the verifiability of the academic results and will allow other researchers to build on the HECTOR results. All research data has been made available either via the Zenodo platform for static data or on GitHub for source code that may be adapted for future work.

2015/1	
Title	A Physical Approach for Stochastic Modelling of TERO-based TRNG
Author	Patrick Haddad, Viktor Fischer, Florent Bernard and Jean Nicolai
Venue	Workshop on Cryptographic Hardware and Embedded Systems (CHES 2015)
DOI paper	https://dx.doi.org/10.5281/zenodo.60900
DOI data	http://dx.doi.org/10.5281/zenodo.59407
2016/7	
Title	TOTAL: TRNG On-the-fly Testing for Attack detection using Lightweight hardware
Author	Yang Bohan, Rozic Vladimir, Mentens Nele, Dehaene Wim, Verbaudwhede Ingrid
Venue	Design, Automation & Test in Europe Conference & Exhibition
DOI paper	https://dx.doi.org/10.5281/zenodo.55455
DOI data	https://doi.org/10.5281/zenodo.154484
Data description	Type 1: "Huge random bit streams and random data streams generated by proposed TRNGs in different technologies". Raw output from an elementary TRNG on a Xilinx Spartan-6 FPGA. Type 6: "Test output data acquired during active attacks on proposed modules and demonstrators". Output from an elementary TRNG on a Xilinx Spartan-6 FPGA with an oversampled clock rate.
2016/11	
Title	A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices
Author	Oto Petura, Ugo Mureddu, Nathalie Bochard, Viktor Fischer, Lilian Bossuet
Venue	FPL 2016
DOI paper	http://dx.doi.org/10.5281/zenodo.375453
DOI data	https://labh-curien.univ-st-etienne.fr/cryptarchi/HECTOR_TRNG_designs
Data description	Type 9: " VHDL code of TRNGs."

2016/12	
Title	Statistical Fault Attacks on Nonce-Based Authenticated Encryption Schemes
Author	Dobraunig C. E., Eichlseder M., Korak T., Lomné V. and Mendel F.
Venue	ASIACRYPT 2016
DOI paper	https://doi.org/10.5281/zenodo.154485
DOI data	https://doi.org/10.5281/zenodo.154487
Data description	Type 10: "Test output data for statistical (ineffective) fault attacks (SFA, SIFA)". Test data for statistical fault attacks (SFA) on different authenticated ciphers.
2016/14	
Title	Exploring active manipulation attacks on the TERO random number generator
Author	Yang Cao, Vladimir Rozic, Bohan Yang, Josep Balasch, Ingrid Verbauwhede
Venue	IEEE 59th International Midwest Symposium on Circuits and Systems
DOI paper	https://doi.org/10.5281/zenodo.154591
DOI data	https://doi.org/10.5281/zenodo.154524
Data description	Type 6: "Test output data acquired during active attacks on proposed modules and demonstrators". Output from two TERO TRNG implementations on a Xilinx Spartan-6 FPGA at different Vdd levels.
2016/17	
Title	Prefetch Side-Channel Attacks: Bypassing SMAP and Kernel ASLR
Author	Daniel Gruss, Clementine Maurice, Anders Fogh, Moritz Lipp, Stefan Mangard
Venue	ACM CCS 2016
DOI paper	https://doi.org/10.5281/zenodo.375513
DOI data	https://doi.org/10.5281/zenodo.375521
Data description	Type 11: "Data and code for system-level attacks and countermeasures". Code to generate data for prefetch side-channel attacks on different architectures.
2017/7	
Title	An Efficient Side-Channel Protected AES Implementation with Arbitrary Protection Order
Author	Hannes Gross; Stefan Mangard; Thomas Korak

Venue	CT-RSA 2017
DOI paper	https://doi.org/10.5281/zenodo.574261
DOI data	https://github.com/hgrosz/aes-dom
Data description	Type 8: "Protected hardware implementations of target modules". Code for a protected AES implementation.
2017/10	
Title	Reconciling d + 1 Masking in Hardware and Software
Author	Hannes Gross, Stefan Mangard
Venue	CHES 2017
DOI paper	https://doi.org/10.5281/zenodo.897935
DOI data	https://github.com/hgrosz/ascon_dom
Data description	Type 8: "Protected hardware implementations of target modules". Code for a protected Ascon (AEAD) implementation.
2017/11	
Title	The Monte Carlo PUF
Authors	Vladimir Rozic, Bohan Yang, Jo Vliegen, Nele Mentens and Ingrid Verbauwhede
Venue	FPL 2017
DOI paper	https://doi.org/10.5281/zenodo.897887
DOI data	https://doi.org/10.5281/zenodo.1306083
Data description	Type 2: "Hardware signature codes generated by proposed PUFs in individual devices". Monte Carlo PUF data obtained with different design parameters and temperatures on Xilinx Spartan-6 FPGA.
2017/15	
Title	Lightweight Prediction-Based Tests for On-Line Min-Entropy Estimation
Author	Milos Grujic, Vladimir Rozic, Bohan Yang and Ingrid Verbauwhede
Venue	IEEE Embedded Systems Letters 9(2)
DOI paper	https://doi.org/10.5281/zenodo.1286723
DOI data	https://doi.org/10.5281/zenodo.1286749
Data description	Type 1: "Huge random bit streams and random data streams generated by proposed TRNGs in different technologies". Biased bit data streams without

	correlations and unbiased bit data streams with bit correlations.
2018/1	
Title	KeyDrown: Eliminating Software-Based Keystroke Timing Side-Channel Attacks
Author	Michael Schwarz, Moritz Lipp, Daniel Gruss, Samuel Weiser, Clémentine Maurice, Raphael Spreitzer, Stefan Mangard
Venue	NDSS 2018
DOI paper	https://doi.org/10.14722/ndss.2018.23027
DOI data	https://github.com/IAIK/keydrown
Data description	Type 11: "Data and code for system-level attacks and countermeasures". Proof-of-concept for KeyDrown countermeasure on different architectures.
2018/4	
Title	Towards Inter-Vendor Compatibility of True Random Number Generators for FPGAs
Author	Milos Grujic, Bohan Yang, Vladimir Rozic and Ingrid Verbauwhede
Venue	DATE 2018
DOI paper	https://doi.org/10.23919/DATE.2018.8342256
DOI data	https://doi.org/10.5281/zenodo.1287612
Data description	Type 1: "Huge random bit streams and random data streams generated by proposed TRNGs in different technologies". Raw random data obtained from the DC-TRNG on Cyclone IV and Cyclone V Intel FPGAs.
2018/7	
Title	A Unified Masking Approach
Author	Hannes Gross, Stefan Mangard
Venue	Journal of Cryptographic Engineering / CHES 2017
DOI paper	https://doi.org/10.1007/s13389-018-0184-y
DOI data	https://github.com/hgrosz/ascon_dom
Data description	Type 8: "Protected hardware implementations of target modules". Code for a protected Ascon (AEAD) implementation.
2018/8	
Title	Another Flip in the Wall of Rowhammer Defenses

Author	Daniel Gruss, Moritz Lipp, Michael Schwarz, Daniel Genkin, Jonas Juffinger, Sioli O'Connell, Wolfgang Schoechl, Yuval Yarom
Venue	S&P 2018
DOI paper	https://doi.org/10.1109/SP.2018.00031
DOI data	https://github.com/IAIK/flipfloyd
Data description	Type 11: "Data and code for system-level attacks and countermeasures". Tools to test susceptibility of different systems to Rowhammer attack variants.
2018/11	
Title	A Closer Look at the Delay-Chain based TRNG
Author	Milos Grujic, Vladimir Rozic, Bohan Yang and Ingrid Verbauwhede
Venue	ISCAS 2018
DOI paper	https://doi.org/10.1109/ISCAS.2018.8351222
DOI data	https://doi.org/10.5281/zenodo.1289397
2018/14	
Title	Clustering Related-Tweak Characteristics: Application to MANTIS-6
Author	Maria Eichlseder, Daniel Kales
Venue	FSE 2019 / Transactions on Symmetric Cryptology 2018/02
DOI paper	http://dx.doi.org/10.13154/tosc.v2018.i2.111-132
DOI data	https://github.com/dkales/clusterfk
Data description	Type 12: "Code for cryptanalytic attacks". Search tool for differential cryptanalysis under related tweakeys.
2018/15	
Title	Exploiting Ineffective Fault Inductions on Symmetric Cryptography
Author	Christoph Dobraunig, Maria Eichlseder, Thomas Korak, Stefan Mangard, Florian Mendel, Robert Primas
Venue	CHES 2018 / TCHES Issue 3 (in press, see https://eprint.iacr.org/2018/071)
DOI data	https://doi.org/10.5281/zenodo.1314447
Data description	Type 10: "Test output data for statistical (ineffective) fault attacks (SFA, SIFA)". Test data for Statistical Ineffective Fault Attacks (SIFA) in different setups.

Chapter 5 Accessibility - Data sharing, archiving and preservation

Access to and sharing of data helps to advance science and to maximize the research investment. A whitepaper[6] by the University of Michigan reported that when data is shared through an archive, research productivity and often the number of publications increases. Protecting research participants and guarding against disclosure of identities are essential norms in scientific research. Data producers should take efforts to provide effective informed consent statements to respondents, to identify data before deposit when necessary, and to communicate to the archive any additional concerns about confidentiality. With respect to timeliness of data deposit, archival experience has demonstrated that the durability of the data increases and the cost of processing and preservation decreases when data deposits are timely. It is important that data is deposited while the producers are still familiar with the dataset and able to fully transfer their knowledge to the archive.

In particular, potential users can find out about generated and existing data most likely through the project's dissemination activities (scientific publications and papers), deliverables, presentations and technical events (conferences, trade shows) etc. During the project lifetime these documents and data have been published on our official project website (www.hector-project.eu) where a broad community has access to the project information. Besides the HECTOR public website, flyers and newsletters, which target a broad interest group, also dedicated websites for significant results and the SVN repository are used for sharing data. The public datasets and corresponding websites and archive entries are in most cases discoverable via the corresponding research paper, which provides the URL or DOI where possible.

We provided detailed information and links in the respective “Research Data Identification” sections in Chapter 3 and refer there for more specifics.

In principle, the data is shared within the HECTOR consortium according to our Consortium Agreement (with respect to any IPR issues) via a secured data repository as soon as the data is available. To the public community, data has been shared according to the dissemination level of the data via the public project website or dedicated websites. Besides the data repository and the websites, the consortium is also willing to handle requests directly. Public deliverables have been made available as soon as they have been approved by the European Commission.

Generally, the consortium's opinion is that it will not be necessary to destroy any data for contractual, legal, or regulatory purposes. However, as described before, there is the case that the confidential deliverables are restricted. The data generated will serve as basis for future scientific research work and reports on device performance as well as for benchmarking.

With regards to the retention and preservation of the data, HECTOR will retain and/or preserve the produced data at least for three years after the project end. Due to the broad range of data generated during the HECTOR project, there is not a single solution for data sharing. An SVN repository has been used for sharing data between partners throughout the project runtime. It has to be noted that only project partners have access to the project SVN. Therefore, publicly available data needs to be shared in another way. Small amounts of human-readable data (e.g., source code of hardware modules or software implementations) have been shared using the already existing internal project SVN repository https://hector.technikon.com and/or public code repositories on GitHub (https://github.com/), depending on the dissemination level. This allows easy synchronization

[6] http://deepblue.lib.umich.edu/handle/2027.42/78307

as well as data versioning. GitHub has established itself as the primary platform for sharing open-source implementations and makes it particularly easy for other researchers and interested parties to build on existing code. Medium-sized datasets (for example measurement data) up to 100MB have been shared via the SVN repository and, in case of public data, via Zenodo. For very large amounts of data in the range of gigabytes which needs to be shared, it was originally foreseen to utilize commodity clouds with usage of internal infrastructure and databases from partners or external platforms, such as Dropbox. Large confidential/proprietary datasets will be stored locally at partners' premises. For large public datasets, particularly TRNG output, in addition to publishing relevant datasets on Zenodo (see Section 3.5.1), a dedicated website was set up (https://trng.technikon.com) to serve fresh random output from UJM's TRNGs, which is hosted by TEC.

To allow third parties to access, mine, exploit, reproduce, and disseminate the publicly available data, adequate license schemes were put in place. For publicly available data provided at the GitHub repository or via another sharing infrastructure from the HECTOR homepage, open-source licenses such as *Creative Commons* (http://creativecommons.org/licenses/?lang=en) and related licenses for software (e.g., MIT, Apache) were selected.

Chapter 6 Summary and conclusion

This data management plan (DMP) outlines the handling of data generated within the HECTOR project, during and after the project lifetime. The document is based on the original DMP in D5.2 and has been updated to reflect the different datasets produced and disseminated during the runtime of the HECTOR project. Compared to the original plan, significantly more and more varied data than anticipated has been produced.

The original DMP focused particularly on output bit-streams and signatures produced by TRNGs and PUFs, statistical tests of the latter, the side-channel traces produced by the implementations under different conditions, and the AEAD building blocks necessary for WP4. In addition, this final report covers significantly more VHDL implementations that have been made publicly available, including different protected implementations of authenticated ciphers using newly developed masking-based side-channel countermeasures, TRNG implementations, test output data of fault attacks for evaluation using newly developed statistical evaluation methods, software for cryptanalytic attacks, and proofs-of-concept for system-level and microarchitectural attacks and countermeasures. This broad spectrum of published data tangibly reflects the achievements made with respect to the HECTOR project objectives. In particular, the public datasets contribute to increase the security and confidence levels for products designed in Europe and allow to verify the quality of the security building blocks researched in HECTOR (objective O12).

The generated data such as the hardware implementations, software proofs-of-concept and leaked output data and traces under physical attacks are clearly not only of interest for the project partners, but also for the scientific community outside of the HECTOR project. The open-source hardware implementations of standard-compliant TRNGs and authenticated ciphers, including new countermeasures, provide important reference points for comparing performance and quality metrics with other solutions, including statistical properties, area, randomness consumption, runtime, and more. For example, the unprotected and protected implementations of Ascon have proved the efficiency of Ascon on resource-constrained platforms and contributed to its selection as a finalist in the CAESAR competition: In the unprotected case, it compares favourably to other ciphers (https://cryptography.gmu.edu/athenadb/fpga_auth_cipher/table_view); for the protected case, open-source implementations of new designs are very rare, but Ascon and Ketje can clearly be protected for a much lower cost than AES-based schemes. The research on active physical attacks (SIFA) and microarchitectural attacks have already sparked follow-up research works in other projects which profited from the available code and data. The leaked signal traces may serve as foundation for practically verifying new methods for, e.g., security evaluations. The same is true for the random bit streams generated by the TRNG designs applied in the HECTOR project. Not all institutions have the facility to generate this data on their own and thus benefit from the data provided by the HECTOR project, and in particular from the “TRNG-as-a-service” website. As another advantage, the public data sharing enables comparing TRNG designs across the HECTOR project borders. This will further result in citations of HECTOR project results in external scientific publications.

The HECTOR consortium is aware of proper data documentation requirements and relies on each partners’ competence in appropriate citation etc. The Consortium Agreement (CA) forms the legal basis in dealing with IPR issues and covers clear rules for dissemination or exploitation of project data. Besides the HECTOR public website, flyers and newsletters, which target a broad interest group, also dedicated websites for significant results and the SVN repository are used for sharing data. With regards to the retention and preservation of the data, HECTOR partners will retain and/or preserve the produced data for several years, three years after the project end at least. The datasets made available via persistent platforms (e.g., Zenodo, GitHub) are expected to be available for a significantly longer period.

Chapter 7 List of Abbreviations

Abbreviation	Explanation
ASIC	Application-Specific Integrated Circuit
DMP	Data Management Plan
DPA	Differential Power Analysis
FPGA	Field-Programmable Gate Array
HDL	Hardware Description Language
IID	Independent and Identically Distributed
IPR	Intellectual Property Rights
NIST	National Institute of Standards and Technology
PUF	Physically Unclonable Function
SCA	Side-Channel Analysis
SRAM	Static Random-Access Memory
SVN	Subversion
TA	Template Attack
TRNG	True Random Number Generator
URL	Uniform Resource Locator
VHDL	Very High-Speed Integrated Circuit Hardware Description Language