# REPORT - DANIEL RYCHLÝ, IMRICH NAGY

## NXP JCOP 5.2 ON SN100.C58 SECURE ELEMENT

### TARGET OF EVALUATION (TOE)

The ToE of the certificate includes both hardware and software stack. The hardware consists of Secure Element core (NXP SN100) and integrated NFC controller (NFC controller not in the scope of evaluation). Software stack consists of Crypto LibrarySecure, Java Card OpenPlatform O/S including OS Update Component, additional software implementations (packages and applets) for eUICC and eSE domains and CSP JavaCard extension.
Any form of further personalization and user applets loaded into flash memory are outside of the scope of the evaluation.

### EVALUATION LABORATORY

Brightsight, Brassersplein 2, 2612 CT Delft, The Netherlands

### (SOME OF) ASSUMED ATTACKER MODELS

- Attacker gains possession of the ToE or the ability to communicate/operate the ToE:
    - The attacker runs own application to alter another application's data [p. 30]
    - The attacker performs unauthorized card management operations (load, install, extract, delete or alter packages or applets) [p. 30]
    - Attacker loads unauthorized Update Image [p. 31]
    - The attacker tries to modify the attack counter [p. 31]
- The attacker remotely exploits communication channel between ToE and the third party and tries to modify or disclose confidential data [p. 30]
- The attacker may predict or obtain information about random number generation process on the ToE [p. 31]

### PHYSICAL RESISTANCE

The device contains software exception handlers to react on unforeseen hardware exceptions which are caught to ensure the system goes to a secure state and increase the attack counter to resist physical manipulation and probing. [p.104] It is ensured that during command execution there are no usable variations in power consumption (measurable at e.g. electrical contacts) or timing (measurable at e.g. electrical contacts) that might disclose cryptographic keys or PINs. All functions of SF.CRYPTO except for SHA are resistant to side-channel attacks. [p. 104] Complex patterned values are used instead of boolean values which are sensible to tampering (only one bit needs to be changed to manipulate a false into a true). [p. 106] Small random delays are inserted in the program flow to make successful physical interference more difficult. [p. 106] Cryptographic coprocessors are protected against DPA and DFA. [p. 107] Enhanced security sensors for clock frequency range, low and high temperature sensor, supply voltage sensors Single Fault Injection (SFI) attack detection, light sensors. [p. 107] Secret information like Keys or PINs are stored encrypted in the TOE. The Master Key to decrypt these is not accessible during normal operation. [p. 106]

### SECURITY FUNCTIONAL AND ASSURANCE COMPONENTS

The Security Target claims to be conformant to Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, CCMB-2017-04-002, April 2017 and Part 3: Security assurance components, Version 3.1, Revision 5, CCMB-2017-04-003, April 2017. [p. 20]

Functional requirements copy most of those from Java card protection profile - open configuration, version 3.0.5 (Dec 2017), published by oracle, inc. (bsi-cc-pp-0099-2017). With some modifications. [p. 44]
SFRs for eUICC are copied from Embedded UICC for Consumer Devices, GMSA Association, 05 June 2018 (BSI-CC-PP-0100-2018) without changes. [p. 76] CSP SFRs are copied from Common Criteria Protection Profile Cryptographic Service Provider, 19 February 2019 (BSI-CC-PP-0104). [p. 79]
The detailed list of Security Functional Requirements sprawls across pages 44 through 88 and is way too long to list in this document.

### MAINTENANCE REPORT

The changes to the product are only related to the addition of a new manufacturing site used as a second source of wafer production (i.e. the physical silicon chips are now manufactured at 2 different sites). The new site has been evaluated during the re-certification of the product. The name of the ToE had to be changed because of the update of the naming conventions. There are no other changes and the software component of the ToE remains unchanged.

### OWN EVALUATION

According to the information contained in this certificate, I would certainly not hesitate to buy any product based on the ToE as described. Most of the SFRs, SARs and guarantees are based on existing, open and well-known specifications rather than new and proprietary ones, which is a good sign. The operating system and cryptographic libraries are based on Java Card OpenPlatform. This fact does not guarantee perfect security, but the product follows Kerckhoffs's principle and I would be willing to put more trust into it than closed implementations, that showed to be inappropriate many times (e.g. Infineon's flawed RSA). The document is also well-structured and detailed. The manufacturer is willing to list attack models, against which the ToE is not resistant.

## RED HAT ENTERPRISE LINUX VERSION 7.1

### TARGET OF EVALUATION (TOE)

The TOE is Red Hat Enterprise Linux, Version 7.1, which can run in two different modes: Base and MLS mode. The latter one is more strict with access policies, i.e. it enforces mandatory access control. The certification is mostly based on the Security Target. (https://www.commoncriteriaportal.org/files/epfiles/0999b_pdf.pdf)

### SCOPE

A lot is assumed about the environment of use of TOE, therefore is not in the scope of the security review.

- Appropriate physical security and competent administrative personnel are expected.
- User has complete control over their data.
- It is assumed, that all trusted counterparts of the TOE conform to the same security policy constraints as the TOE itself.
- Correct implementation of security functions by IT systems executing the TOE.
- TOE is installed and configured correctly. The required documentation is provided.

### ASSUMED ATTACKER MODEL

There are several assets the TOE needs to protect:

- any unauthorized access to stored data (read, deletion etc.)
- transient, such as network, data

- TOE security functions and resources, such as keys

Threat agents are either external or internal entities attempting unauthorized access to assets by masquerading as an authorized entity or using security functions without authorization. Various scenarios are given, such as:

- An attacker might delegate rights granted to a role that he does not possessor that he is not allowed to delegate.
- A threat agent might masquerade as an authorized entity including the TOE itself or a part of the TOE to gain unauthorized access to user data...

and many more.

## SECURITY ASSURANCE REQUIREMENTS (SAR)

TOE Security Assurance Requirements allegedly meet EAL 4 according to the certificate. The TOE is already conforming to the Operating System Protection Profile.
The TOE is delivered in the form of several download components, such as the iso file and additional packages. Those are delivered via the Red Hat Network. The packages are built and signed by the developer (their public key is widely available). SHA-256 is used for checking integrity.

## SECURITY FUNCTIONAL REQUIREMENTS (SFR)

SFRs are implemented by various functionality, such as:

- auditing – it can intercept all system calls.
- cryptographic support – TOE provides both, cryptographic primitives and secured communication channels. The primitives are based on various RFCs, SSHv2 protocol, as well as OpenSSH and TLS, are supported. Public key authentication defined in RFC4252 is supported. IPSEC and IKE protocol families are implemented. Confidentiality of data storage is achieved using dm_crypt with LUKS headers.
- Packet filter – TOE provides a packet filter for common IP-based communication. There is also a packet filter for virtual machines bridged to the TOE.
- Identification and Authentication – it all relies on authentication information provided interactively by the user. The mechanisms are PAM-based.
- Runtime protection – TOE tries to minimize the risk of buffer overflow.

## TESTING

At least one CPU from each family and virtualization type was tested. All the tests passed. The developer test coverage was as follows: system calls, security-critical configuration files and trusted programs and the corresponding network protocol. A reasonable set of tests used by the developer were rerun by the evaluator. The evaluator ran additional tests, including SSH cipher test, IPsec ciphers and certificates tests, permission settings of relevant config files or code vulnerability protection functions. Some penetration testing was done as well, for example in the form of fuzzing, CVE exploits or syscall thrashing.
There were objections raised regarding the use of low key size (<= 100) for various cryptographic functions.

## CONCLUSION

Most of the Report is just referencing the Security Target, which is reasonable, given its rigor, but we'd expect more work done from the side of the report itself. We'd like to see way more pen testing, which is critical for operating systems, as they are mostly facing the open world via the Internet. Also, the development process of TOE should be reviewed more thoroughly, in our opinion. However, we are confident with the product itself.

## ST33TPHF2E MODE TPM 2.0, TPM FIRMWARE 0X49.0X40 & 0X49.0X41

### TARGET OF EVALUATION (TOE)

The products ST33TPHF2ESPI and ST33TPHF2EI2C are hybrid TPM products targeting PC, server platforms and embedded systems. TOEs interface is irreversibly locked after first loading. Those two TPM products operating in TPM 2.0 mode are TOE, including both hardware and firmware. For a TPM to be trusted, it needs to correctly perform some operation and protect relevant data while doing it. There are three Roots of Trust in the TPM: for measurement, for reporting and storage. TOE is built on ARM processors.
The TOE supports various cryptographic primitives, such as RSA with 2048 bits, AES-128 thru AES-256 in CFB mode, SHA-256, ECC algorithms etc. RNG, self-test and physical protection are also included.
TOE supports two Endorsement keys, 2048-bit RSA and 256-bit ECC with associated certificates. CA keys are stored encrypted with the 3-DES key.
There is only one assumption, which requires the TOE to be properly installed and configured. There is no guarantee of physical security.

### ASSUMED ATTACKER MODELS

The threat models count with a variety of scenario: an active attacker, incompetent user, insecure communication with TOE, fault of the TOE. Some of the scenarios are:

- Undetected compromise of the data as a result of an attacker performing unauthorized action
- Unauthorized tampering with the attributes of security functions and resulting in the compromise of TOE assets
- Exporting data with insecure security attributes
- incorrect implementation of cryptographic key generation or operation.
- The TOE starts in an insecure state

### SECURITY FUNCTIONAL REQUIREMENTS (SFR)

The main defined roles are USER, ADMIN and DUP. The objects to be protected are various hierarchies, root keys, context, user keys, NV storage provided to the user, RNG, Clock and encrypted credentials.
Moreover, requirements for RNG, key generation, key destruction and various cryptographic operations are presented (mostly based on FIPS). We see the requirements as sufficient, as they follow contemporary limits deemed secure.
There is no specific information given regarding the Security Assurance Requirements. It only references those of EAL4 as defined in CC.

### PHYSICAL SECURITY

The TOE preserves secure state by detecting a physical attack or suspicious environment condition, such as high voltage or MPU error, and responding by the shutdown, and it does some preventive measures, such as bus encryption and memory scrambling.

### EVALUATION

We are very disappointed with the report itself. We found the certification lacking in many ways. First, the description of testing the TOE is only referenced but nowhere to be found, so we can't evaluate its rigor and reliability. The report is very vague in general (and in French). However, that does not mean the TOE itself is bad. Mostly the SFR as described in Security Target are quite comprehensive. We don't understand why 3-DES is used for storing CA keys.