
jtsec
Applus
The new cryptographic evaluation methodology created by CCN
INTERNATIONAL COMMON
— CRITERIA CONFERENCE —
WASHINGTON DC

About me

José Ruiz Gualda

jtsec Beyond IT Security

jruiz@jtsec.es

- Computer Engineer (University of Granada)
- Expert in Common Criteria, LINCE and FIPS 140-2 & FIPS 140-3
- Member of the SCCG (Stakeholder Cybersecurity Certification Group) at the European Commission.
- Secretary of SC3 at CTN320
- Editor of LINCE as UNE standard
- Editor in JTC13 WG3 of the FITCEM Methodology
- European Commission reviewer for the ERNCIP group "IACS Cybersecurity Certification".
- Former ICCC program director

About us

- Cybersecurity evaluation & consultancy services
- Common Criteria, LINCE and ETSI EN 303 645 accredited lab.
- Developers of the most powerful tool for Common Criteria, CCToolbox.
- Involved in standardization activities (ISO, CEN/CENELEC, ISCI WGs, ENISA CSA WGs, CCUF, CMUF, ERNCIP, …)
- Members of the SCCG (Stakeholder Cybersecurity Certification Group)
- jtsec is part of the A+ group along with Lightship Security. We have labs in Canada, USA and Spain.

INDEX

INDEX

History of the Cryptographic Evaluation

USA

NIST (National Institute of Standards and Technology) → Verification of Conformity according to FIPS 140-1, FIPS 140-2 and FIPS 140-3 → CMVP - Designed for certifying cryptographic modules

→ CAVP - Designed to certify cryptographic algorithms → Publication of multiple "Special Publications" specifying cryptographic algorithms and how to test them

CMVP
Conformance through Testing
CAVP

FIPS VALIDATED 140-1 ™

FIPS VALIDATED 140-2 ™

FIPS 140-3 ™

ACVP

History of the Cryptographic Evaluation

International

Previously

Withdrawn
ISO/IEC 19790:2006

Withdrawn
ISO/IEC 19790:2006/Cor 1:2008

Now

Published
ISO/IEC 19790:2012

A standard is reviewed every 5 years
Stage: **90.92** (To be revised)

Will be replaced by

Under development
ISO/IEC WD 19790.3

Corrigenda / Amendments

Withdrawn
ISO/IEC 19790:2012/Cor 1:2015

History of the Cryptographic Evaluation

Spain

Certification Body for cryptographic modules - OC-CCN (Spanish National Cryptologic Centre)

ISO

- ISO/IEC 19790, Security Requirements for Cryptographic Modules
- ISO/IEC 24759, Test Requirements for Cryptographic Modules

INDEX

Cryptographic Evaluation Today

Europe

- SOG-IS Crypto Evaluation Scheme Harmonised Cryptographic Evaluation Procedures v0.16 (December 2020)
- First SOG-IS evaluation methodology
  - Implementation of cryptographic mechanisms
  - Pitfalls Prevention Requirements

SOG-IS HEP

- SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms v1.3 (February 2023)
  - Cryptographic mechanisms agreed and recommended by SOG-IS
    - Acceptable level of security
    - Implementation guidelines

SOG-IS ACM

Cryptographic Evaluation Today

Spain

CCN-STIC 130 Guide
Cryptologic Evaluation Requirements Guide (October 2017)

- Requirements for Approval of Encryption Products to Handle Classified National Information
- FIPS-like approach
- Security Requirements

MEC – LINCE
Cryptographic evaluation module within the LINCE methodology

Very light cryptographic conformance testing following the NIAP Protection Profiles approach

Botan-CCN Cryptographic Library
Reference implementation of CCN to perform conformity testing of the cryptographic mechanism in cryptographic evaluations

CCN-STIC 130 Guide

Cryptographic Evaluation Today

Spain

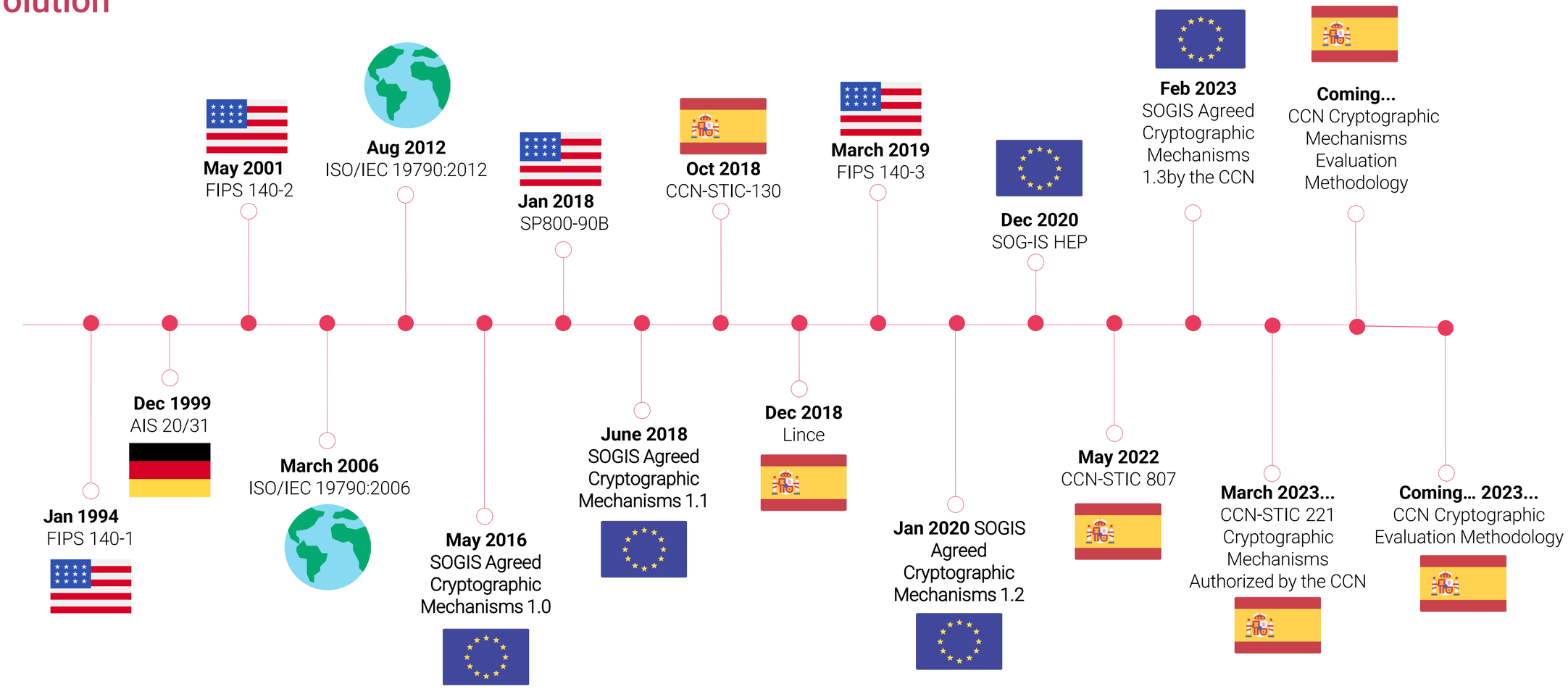
CCN-STIC 221 Guide
Cryptographic Mechanisms authorized by CCN
Includes new CCN-authorized algorithms with respect to the European ACM
Transversal use guide not limited to ENS

NEW

CCN-STIC 221 Guide

Cryptographic Evaluation Today

Evolution



Cryptographic Evaluation Today

Is this only a Spanish issue? | Reasons why the cryptographic mechanisms methodology is necessary

FIPS and/or ISO FIPS:

- It only works when the module has been created to meet FIPS requirements.
- It does not work well for products that integrate cryptography but do not use a third-party cryptographic module.

STIC 130

- Does not include algorithm-level conformity and includes product implementation requirements.
- Not 100% focused on cryptographic implementation.
- Provides the security point of view.

We do not have a methodology that evaluates cryptographic algorithms and protocols.

FIPS 140-3™

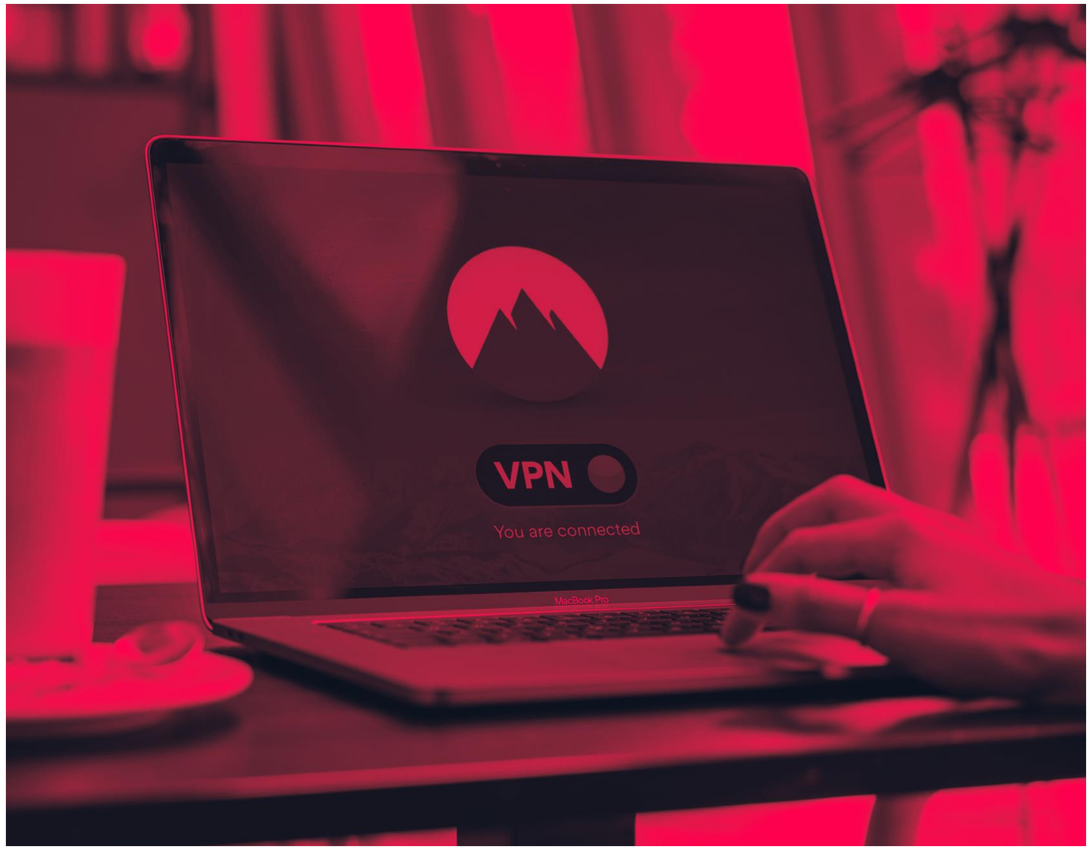
STIC 130

INDEX

Usage

CCN Cryptographic Mechanisms Evaluation Methodology

- Products whose main functionality requires cryptography (e.g., VPNs, ciphers, secure communications, etc.)
- According to three increasing Certification Levels: CL1, CL2 & CL3
- During CC, LINCE and Complementary STIC certification processes.



Index

Definition

CCN Cryptographic Mechanisms Evaluation Methodology

Document Structure

- Cryptographic Requirements
- Agreed Cryptographic Mechanisms
- Conformity Testing
- Common Implementation Pitfalls

centro criptológico nacional

Evaluation tasks and evaluation test

Structure

Each section contains:

- One or several tasks are defined. They are mandatory independently of the implementation and shall be executed for the associated certification level.

- One or several tests defined and associated to each task. They are categorized as mandatory or implementation dependant. Moreover, the required vendor inputs are detailed for each test.

CRYPTOGRAPHIC EVALUATION TASK	CCN-NAME	CERTIFICATION LEVEL
The **tester shall…** [Cryptographic Evaluation Task Definition]		**CL1 CL2 and/or CL3**

CCN-NAME/TestName	Inputs
[Cryptographic Evaluation Test Definition]	

Some specific examples:

CCN-SSP **SSP Management**	CCN-SSP/Generation	Impl-Dep
	CCN-SSP/Transport	Impl-Dep
	CCN-SSP/Storage	Mandatory
	CCN-SSP/Zeroization.1	Mandatory
	CCN-SSP/Zeroization.2	Mandatory

CRYPTOGRAPHIC EVALUATION TASK	CCN-SSP	CERTIFICATION LEVEL
The **tester shall** verify that the TOE performs secure management of Security Sensitive Parameters (SSPs) during their lifecycle, from generation to destruction, using approved cryptographic mechanisms for generation, entry/output, storage, and zeroization.		**CL2 and CL3**

CCN-SSP/Transport	Inputs
In the case of **CL2** evaluations, verify in **I1** that: - The PSP entry/output is protected in authenticity and integrity using an approved cryptographic mechanism. - The CSP entry/output is protected in authenticity, integrity, and confidentiality using an approved key protection mechanism. In addition, operate the TOE (using **I2**) to verify that: - The SSP transport methods implemented by the TOE match with those declared by the vendor in **I1**. In the case of **CL3** evaluations, use **I1**, operate the TOE (using **I2**), and perform a source code review (using **I6**) to verify the above-mentioned.	**I1** **I2** **I6**

Cryptographic Mechanisms Evaluation Methodology

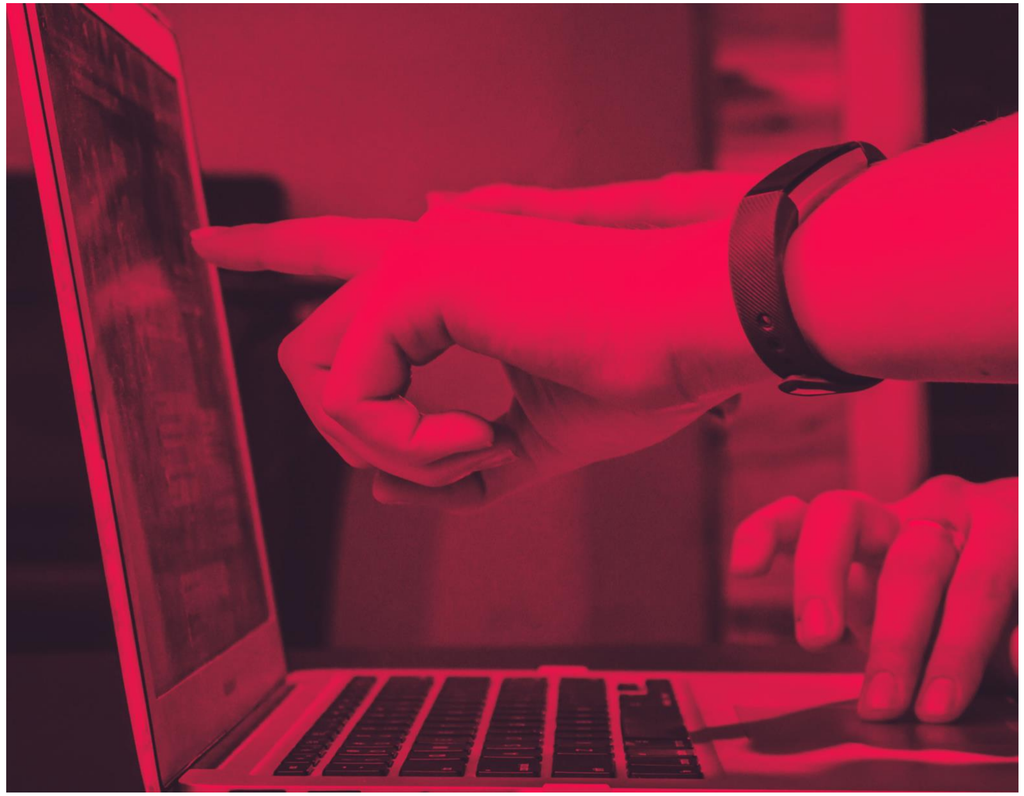
Structure

1. Cryptographic Requirements

Objective: To specify the requirements extracted by CCN from the CCN-STIC 130 guide that apply to the security of cryptographic products related to the cryptographic mechanisms and primitives implemented in relation to:

- Self-tests (not required by SOGIS nor CCN STIC-221)
- Critical Security Parameters (CSP) Management (with additional requirements than required by SOGIS)
- Mitigation of Other Attacks (not required by SOGIS nor CCN STIC-221)

Evaluation: The evaluator shall verify that the TOE complies with the cryptographic requirements listed in this section.



Cryptographic Mechanisms Evaluation Methodology

1. Cryptographic Requirements - Critical Security Parameters (CSP) Management

The methodology not only evaluates the SOGIS related Key Management requirements, but also assesses the entire life cycle of every SSP managed by the TOE.

This comprehensive approach ensures a thorough evaluation of the security posture of the TOE beyond just key management.

Example: SSP Life Cycle Management for AES_EDK

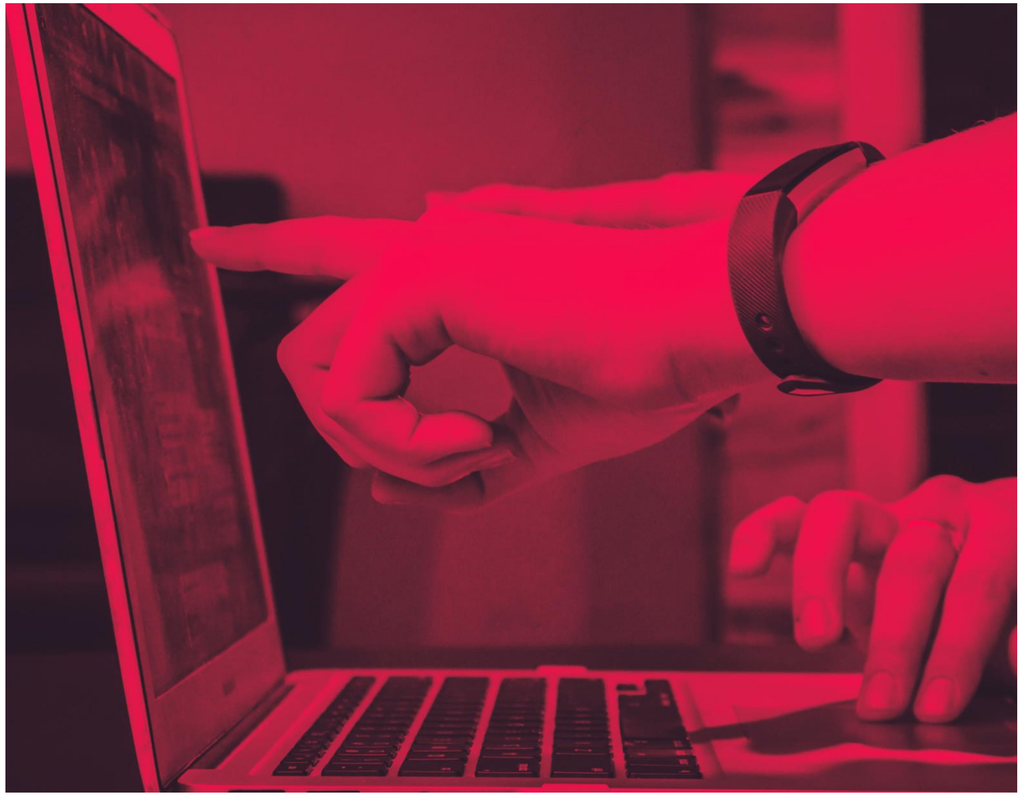
SSP	Strength (in bits)	Generation Method	Entry/Output	Storage	Applications or cryptographic operation	Zeroization	Evidence of zeroization and justification
AES_EDK	128, 256	DRBG (Hash_DRBG)	N/A	AES-256-KeyWrap	Encryption/decryption with: AES-CBC AES-CTR	Overwritten using zeros.	Include piece of source code and justification.

Cryptographic Mechanisms Evaluation Methodology

Structure

2. Approved Cryptographic Mechanisms
Objective: To specify the cryptographic mechanisms recognized and agreed by CCN

Evaluation: The evaluator shall verify that the cryptographic mechanisms implemented by the TOE comply with the guidelines presented by the CCN in the CCN STIC-221 guide including correct parametrization.



Cryptographic Mechanisms Evaluation Methodology

Structure

3. Conformance Testing
Objective: To specify the requirements necessary to perform conformity testing of the cryptographic primitives and mechanisms implemented by the TOE. These tests shall determine whether the cryptographic primitives and mechanisms used by the TOE are correctly implemented. This is similar to what NIST does but also verifying parameterizations and limit values that often lead to errors.

Evaluation: The evaluation process is divided into four steps:

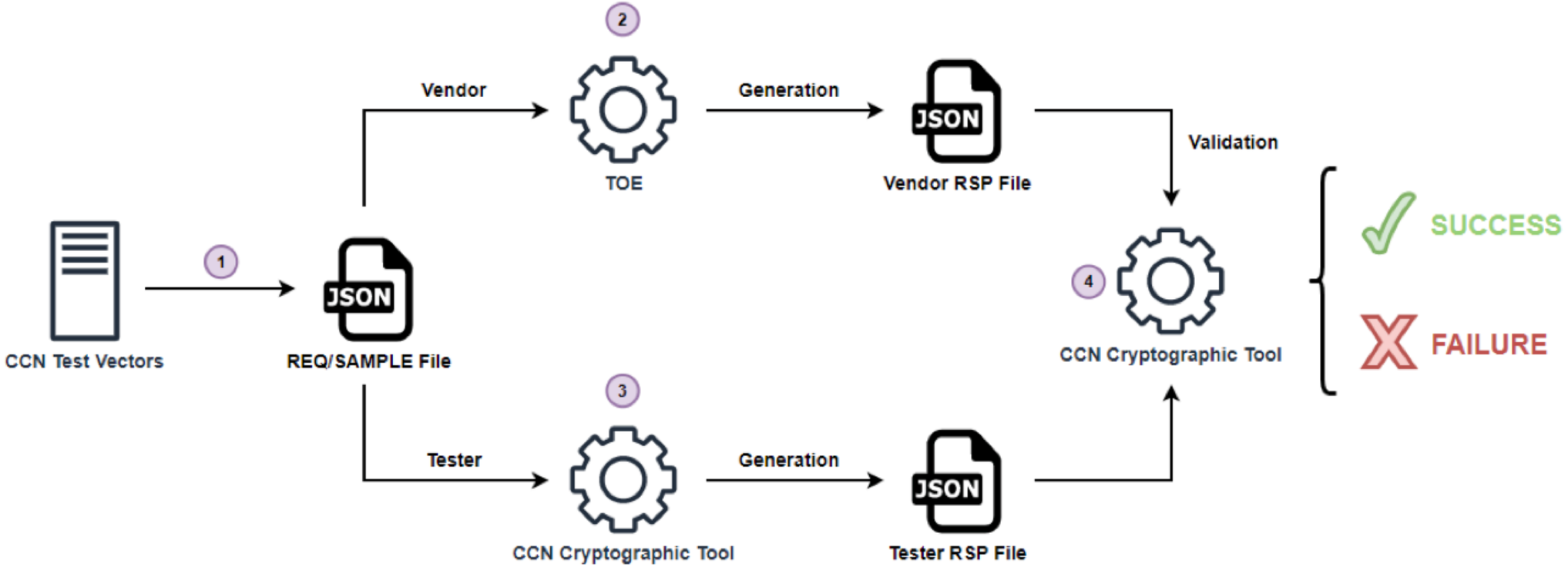
1. Generation of Test Vectors: Request and Sample files.
2. Generation of Results by the Vendor: Response File
3. Generation of Results by the Evaluator: Response File
4. Validation of Results by the Evaluator

centro criptológico nacional

CCN-STIC 221 Guide

Cryptographic Mechanisms Evaluation Methodology

Conformance Testing Evaluation Process Diagram



Cryptographic Mechanisms Evaluation Methodology

Test Vectors Generation

- The evaluator shall generate a 'REQUEST' file (in JSON format) for each cryptographic mechanism implemented by the TOE containing the test vectors associated to the supported parameterization.

- Additionally, the evaluator shall generate the 'SAMPLE' file (in JSON format) for each cryptographic mechanism implemented by the TOE containing an example solution to indicate the format of the expected result.

The evaluator shall send to the vendor a file package containing the 'REQUEST' and 'SAMPLE' files associated to all cryptographic mechanisms implemented by the TOE.

Cryptographic Mechanisms Evaluation Methodology

Generation of Results by the Vendor

- The vendor shall generate a 'RESPONSE' file associated with each cryptographic mechanism implemented, containing the output provided by the TOE for each of the test vectors provided in the 'REQUEST' file.
- The vendor shall retain the JSON format presented in the 'REQUEST' and 'SAMPLE' files for the generation of the 'RESPONSE' file.

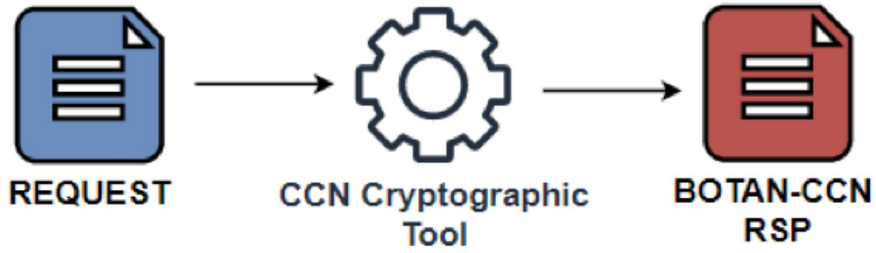
The vendor shall send to the evaluator a file package containing the 'RESPONSE' files associated with all cryptographic mechanisms implemented by the TOE.

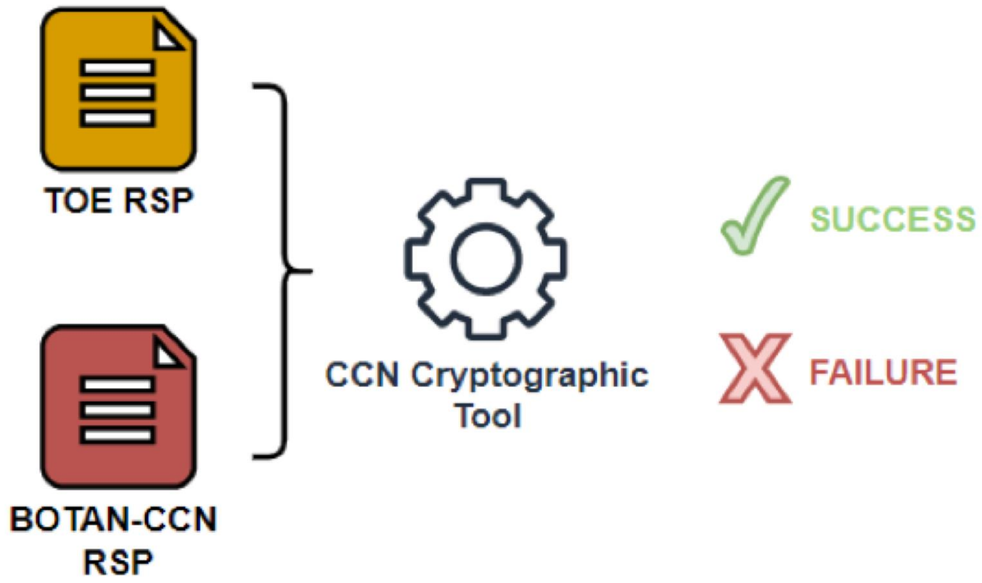
Cryptographic Mechanisms Evaluation Methodology

Generation of Results by the Evaluator

The evaluator shall generate the 'RESPONSE' file associated to each cryptographic mechanism implemented by the TOE, using the Botan-CCN library as reference cryptographic implementation.

The evaluator shall retain the JSON format presented in the 'REQUEST' and 'SAMPLE' files for the generation of the 'RESPONSE' file.





Cryptographic Mechanisms Evaluation Methodology

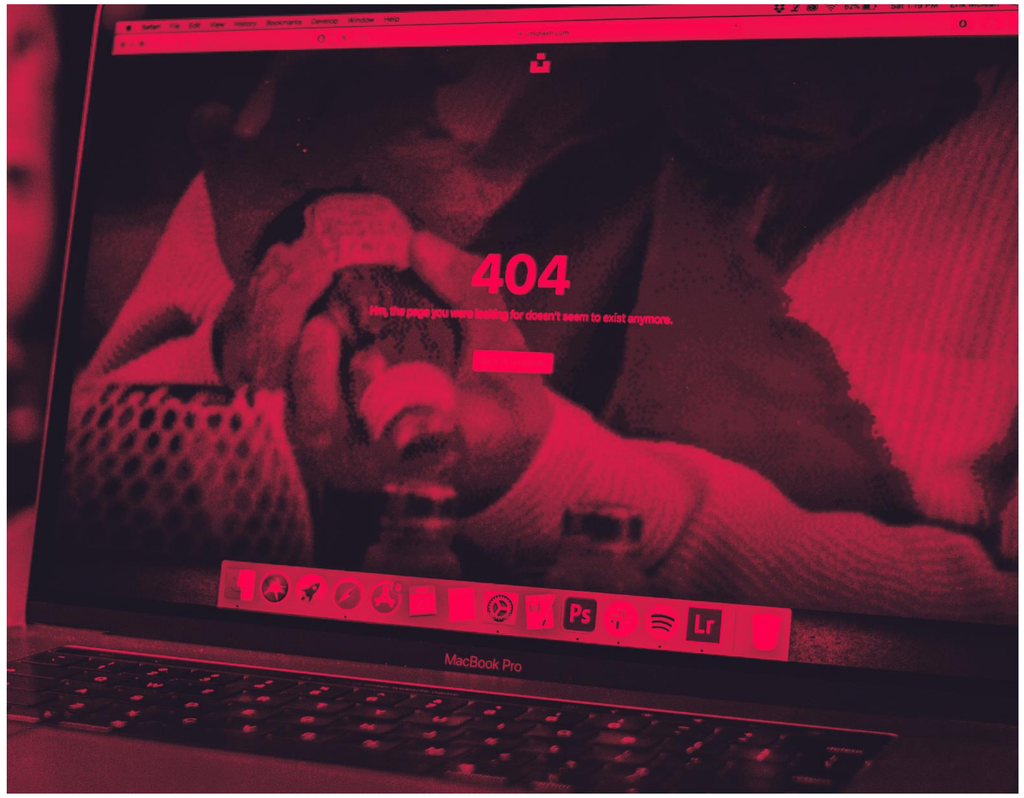
Validation of Results by the Evaluator

The evaluator shall validate the 'RESPONSE' files provided by the vendor for each cryptographic mechanism implemented by the TOE, comparing the results provided with those obtained in the previous step using the Botan-CCN cryptographic library.

The evaluator shall determine whether the TOE correctly implements the cryptographic mechanisms and primitives used and declared.

Cryptographic Mechanisms Evaluation Methodology

Structure



4. Common Implementation Pitfalls

Objective: To specify the requirements necessary to avoid implementation pitfalls in the cryptographic primitives and mechanisms implemented by the TOE.

Evaluation: The evaluator shall verify that the cryptographic mechanisms implemented by the TOE comply with the implementation pitfall avoidance guidelines presented by the SOG-IS in the SOG-IS Harmonized Cryptographic Evaluation Procedures guide.

Cryptographic Mechanisms Evaluation Methodology

Common Implementation Pitfalls -
Example: Key Derivation Implementation Pitfall

CCN-PITFALL/KeyDerivation	**Inputs**
In the case of **CL2** evaluations, verify using **I5** that invalid requests for the keying data generation are not possible. **Analysis:** The computation of the derived key starts with some size controls and that shall not be ignored. In particular, the tester shall verify that no derived key is larger than: - $255 \times h$ for HKDF constructions - $(2^{32} - 1) \times h$ for the rest of key derivation functions where h is the length (in bits) of the output block of the underlying hash function or pseudo-random function. In the case of **CL3** evaluations, use **I5** and perform a source code review (using **I6**) to verify the above-mentioned.	**I5** **I6**

Cryptographic Mechanisms Evaluation Methodology

Advantages of the Cryptographic Mechanisms Evaluation Methodology over SOG-IS

Cryptographic Mechanisms Evaluation Methodology

- Complete evaluation methodology. It establishes concrete evaluation tasks depending on the certification level (CL1,CL2 or CL3) to be followed by the evaluator for each cryptographic mechanism to assess:
    - Cryptographic Management requirements
    - Mitigation of other attacks.
    - Usage of approved mechanisms, including post quantum algorithms and specific entropy requirements
    - Conformity Testing
    - Common implementation pitfalls avoidance.
- Self-tests. It is verified that the self-tests are properly implemented for each algorithm according to CCN requirements. Several evaluation tasks are designed to evaluate their implementation and correct operation.

SOG-IS HEP and ACM

- Provides the agreed mechanisms and their associated requirements, and the evaluation tasks to:
    - Verify their correct implementation according to their associated standard
    - Perform the conformity testing.
    - Avoid implementation pitfalls.
    - Verify key management (with less requirements than the CCN evaluation methodology)
- Self-tests requirements and Mitigation of other attacks are not specified.

SOGIS
Senior Officials Group
Information Systems Security

Cryptographic Mechanisms Evaluation Methodology

Advantages of the Cryptographic Mechanisms Evaluation Methodology over SOG-IS

Cryptographic Mechanisms Evaluation Methodology

New Algorithms: The Cryptographic Mechanisms Evaluation Methodology includes new "classical" and post-quantum algorithms recommended by the Spanish CCN in the new STIC 221 guide.

New recommended classical algorithms:

- SCRYPT, ChaCha20_Poly1305
- EdDSA

Post-Quantum Algorithms

- CRYSTALS-Kyber, CRYSTALS-Dilithium, Falcon, SPHINCS+
- FrodoKEM is also recommended. It will not be standardised as part of NIST's PQC project, mainly due to efficiency considerations, but there are currently no doubts about its security.

SOG-IS HEP and ACM

- List of classical algorithms without including references to post-quantum algorithms

Cryptographic Mechanisms Evaluation Methodology

Advantages of the Cryptographic Mechanisms Evaluation Methodology over SOG-IS

Cryptographic Mechanisms Evaluation Methodology

- **Life cycle management of each SSP managed by the TOE.** For each SSP, its strength, generation, entry/output, storage and zeroization methods are evaluated.
- **Complete list of conformity test vectors for all the agreed cryptographic mechanisms.** Example: AES Key Wrapping.

SOG-IS HEP and ACM

- Establishes general Key Management requirements, specifying only the recommended mechanism for each stage.
- The conformity test vectors of several algorithms are not defined or are not complete.

Cryptographic Mechanisms Evaluation Methodology

Link with Common Criteria

Cryptographic Mechanisms Evaluation Methodology

- **The methodology will be used in Common Criteria evaluations at the national level if crypto is a core component of the product (e.g. VPN, Ciphers, etc...)**
- **The methodology could be considered a supporting document to harmonize how to evaluate crypto mechanisms.**

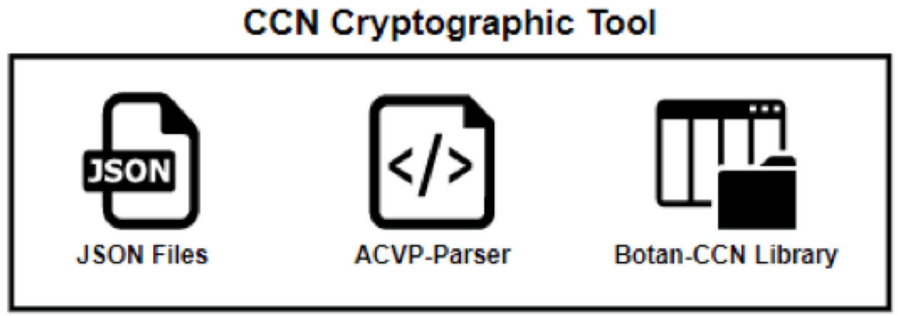
INDEX

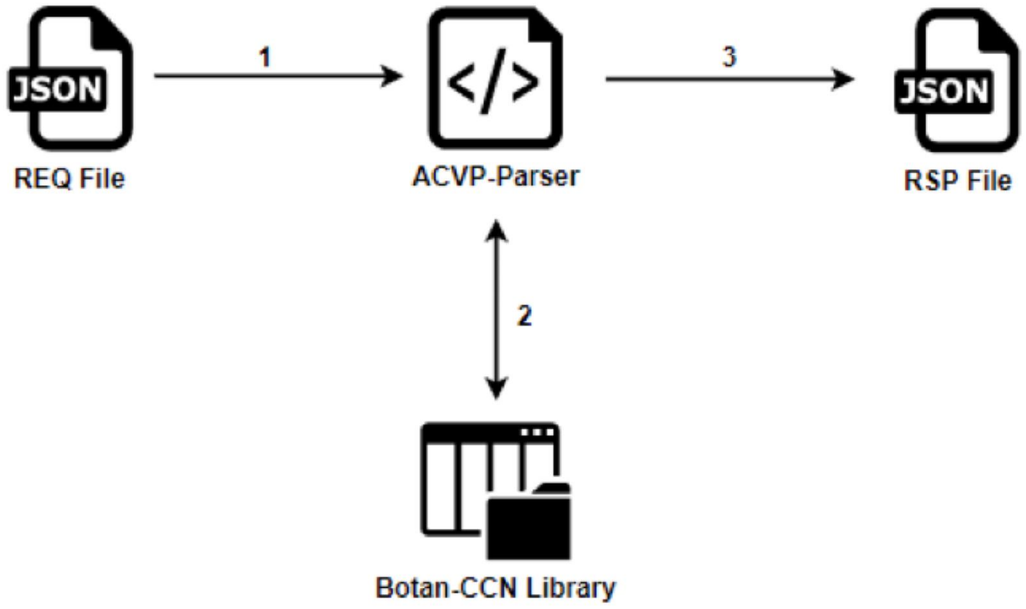
CCN Cryptographic Evaluation Tool

Definition

Performing Conformity Testing
Structure of the Tool

- JSON test files: test vectors in hexadecimal format according to SOG-IS methodology.
- ACVP-Parser: JSON file processing and extraction of parameters needed to invoke the cryptographic reference implementation.
- Botan-CCN Cryptographic Library: cryptographic reference implementation used to generate test vectors results and validate the correct cryptographic implementation of the TOE.





CCN Cryptographic Evaluation Tool

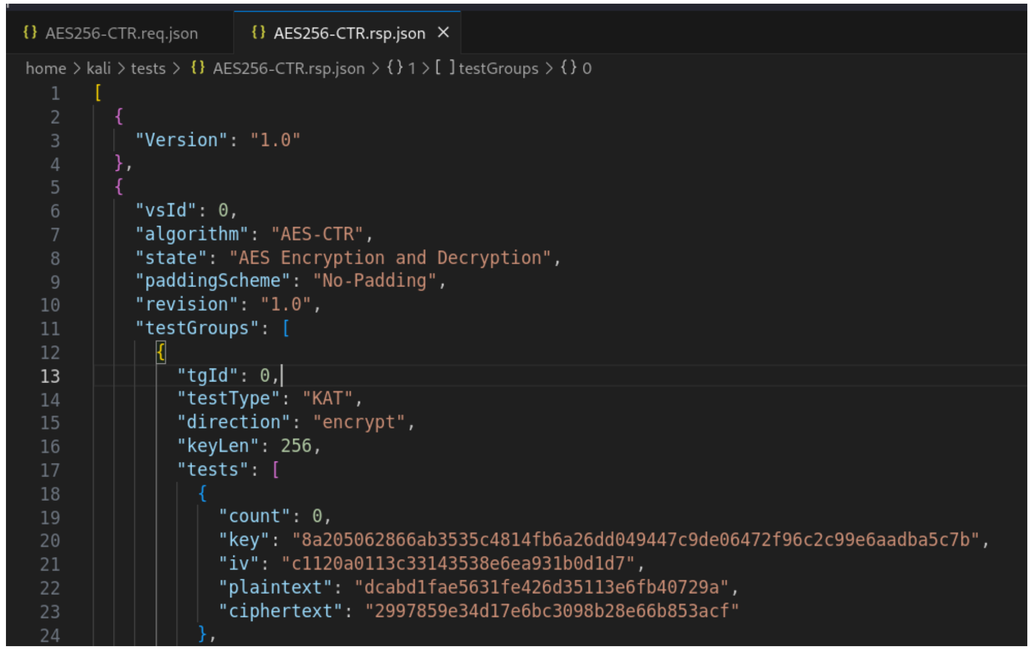
Flowchart

1. Processing of the test vectors to extract the parameters using the ACVP-Parser.
2. Invocation of the Botan-CCN cryptographic library to perform the generation of test vector results using the associated 'REQUEST' file.
3. Generation of the 'RESPONSE' file associated to a cryptographic mechanism using the associated 'REQUEST' file and the results obtained using the Botan-CCN cryptographic library.

Cryptographic Evaluation Tool

Cryptographic Evaluation Tool - Usage Example: SHA-256

‘REQUEST’ file



‘RESPONSE' file generated by the Tool

Cryptographic Evaluation Tool

Cryptographic Evaluation Tool - Usage Example: SHA-256

AES256-CTR.req.json | AES256-CTR.rsp.json

home > kali > tests > AES256-CTR.rsp.json > {} 1 > [ ] testGroups > {} 0

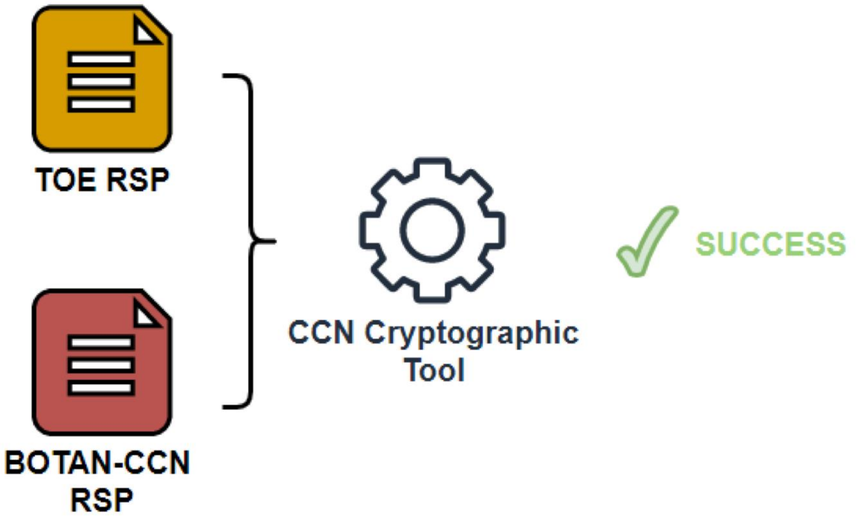
```
[
  {
    "Version": "1.0"
  },
  {
    "vsId": 0,
    "algorithm": "AES-CTR",
    "state": "AES Encryption and Decryption",
    "paddingScheme": "No-Padding",
    "revision": "1.0",
    "testGroups": [
      {
        "tgId": 0,
        "testType": "KAT",
        "direction": "encrypt",
        "keyLen": 256,
        "tests": [
          {
            "count": 0,
            "key": "8a205062866ab3535c4814fb6a26dd049447c9de06472f96c2c99e6aadba5c7b",
            "iv": "c1120a0113c33143538e6ea931b0d1d7",
            "plaintext": "dcabd1fae5631fe426d35113e6fb40729a",
            "ciphertext": "2997859e34d17e6bc3098b28e66b853acf"
          },
```

PASS

‘RESPONSE' file generated by TOE

```
$ ./acvp-parser -e AES256-CTR.rsp.json KNOWN-AES256-CTR.rsp.json -v
[PASSED] compare AES256-CTR.rsp.json with KNOWN-AES256-CTR.rsp.json
$
```

Validation of results



Cryptographic Evaluation Tool

Cryptographic Evaluation Tool - Usage Example: SHA-256

AES256-CTR.rsp.json

home > kali > tests > {} AES256-CTR.rsp.json > {} 1 > [ ] testGroups > {} 0 > [ ] tests > {} 1

```
[
  {
    "Version": "1.0"
  },
  {
    "vsId": 0,
    "algorithm": "AES-CTR",
    "state": "AES Encryption and Decryption",
    "paddingScheme": "No-Padding",
    "revision": "1.0",
    "testGroups": [
      {
        "tgId": 0,
        "testType": "KAT",
        "direction": "encrypt",
        "keyLen": 256,
        "tests": [
          {
            "count": 0,
            "key": "8a205062866ab3535c4814fb6a26dd049447c9de06472f96c2c99e6aadba5c7b",
            "iv": "c1120a0113c33143538e6ea931b0d1d7",
            "plaintext": "dcabd1fae5631fe426d35113e6fb40729a",
            "ciphertext": "ec977fc1a287cdaaf86726819781470d4c46"
          },
```

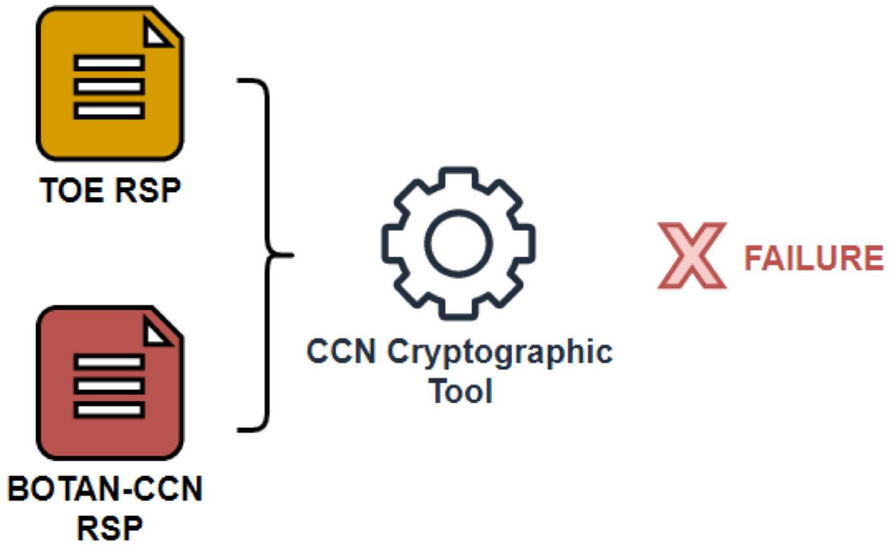
FAIL

ERROR

'RESPONSE' file generated by TOE

```
$ ./acvp-parser -e AES256-CTR.rsp.json KNOWN-AES256-CTR.rsp.json -v
[FAILED] compare AES256-CTR.rsp.json with KNOWN-AES256-CTR.rsp.json
$
```

Validation of results



INDEX

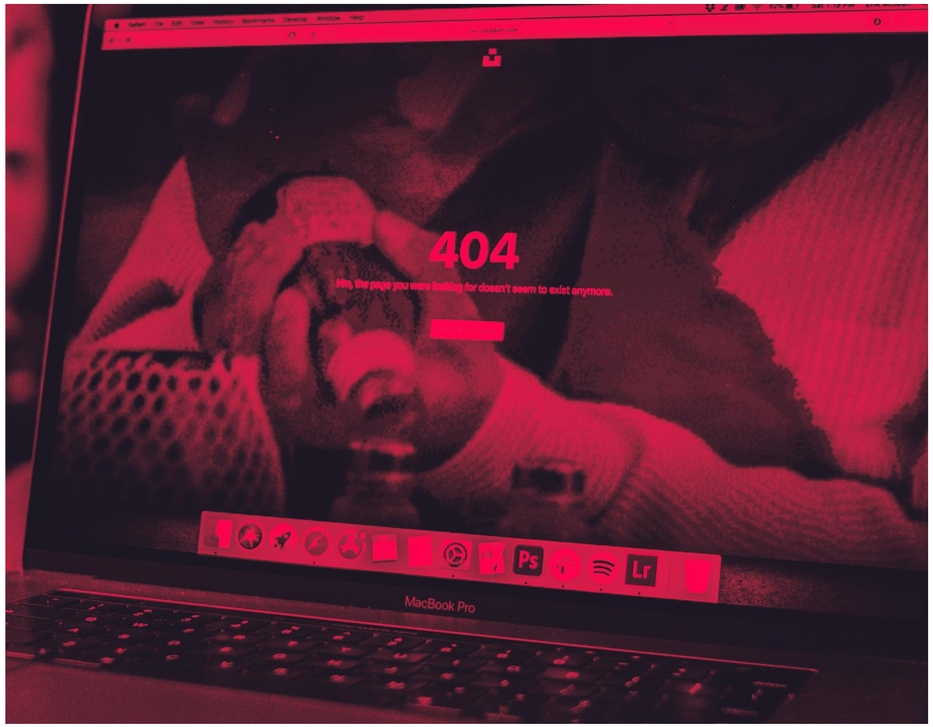
Cryptographic Evaluation Methodology

1. Usage

Cryptographic evaluation methodology to address the security requirements of the CCN-STIC 130 guidance.

This methodology aims to **evaluate the implementation of a TOE** beyond the requirements associated with cryptographic mechanisms such as:

- Cryptographic Module Design
- Authentication
- Physical Security
- Logical Security
- RNG design
- Configuration management system
- Etc…



Cryptographic Evaluation Methodology

2. Security Levels:

CCN STIC-130 defines 3 increasing qualitative levels of security that are directly mapped to the 3 evaluation levels of the Cryptographic Mechanisms Evaluation Methodology:
- CL1: Low Level of CCN-STIC 130 (Restricted)
- CL2: High Level of CCN-STIC 130
- CL3: Advanced Level of CCN-STIC 130

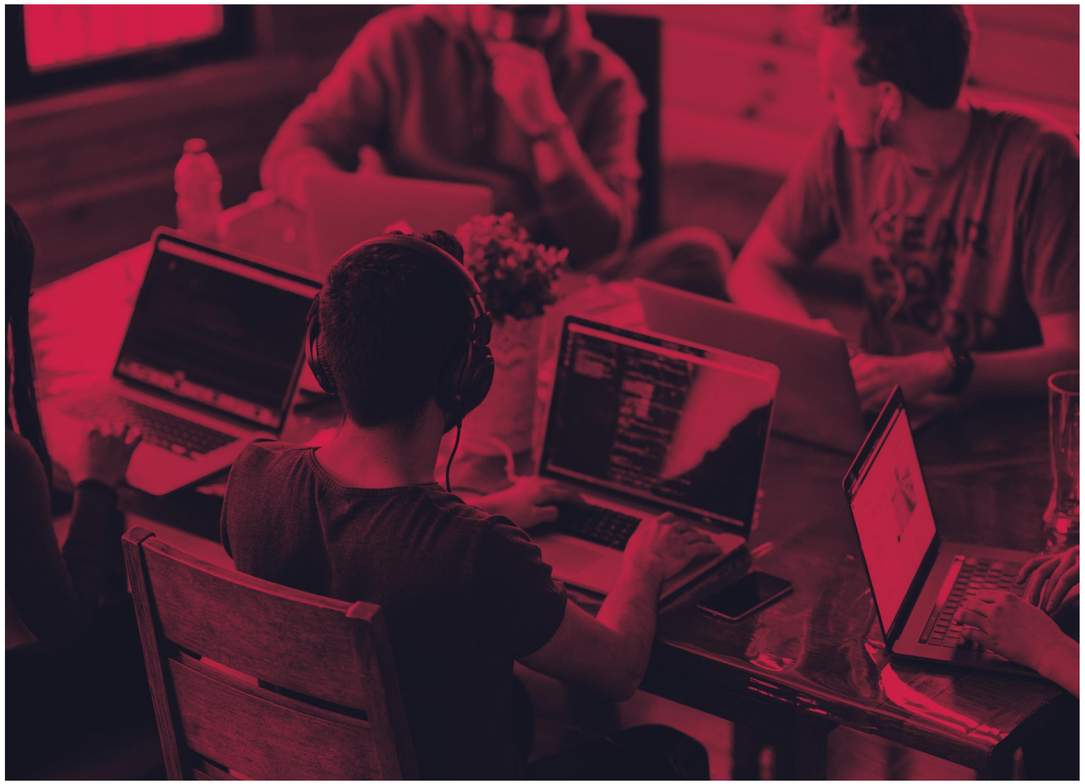
Each TOE will be evaluated according to the level of sensitivity of the information it handles and the global evaluation methodology to which the Cryptographic Methodology is being applied to.

Some evaluation tasks will be common for all levels and others will only apply depending on the security level.

INDEX

Conclusions

- Spain is pioneer in creating a Cryptographic Evaluation Methodology **for mechanisms**.
- The **usage in Common Criteria evaluations is straight forward**.
- The methodology is in **trial usage** and will be published soon
- All this work is a contribution to complement **European efforts**
- This effort is necessary to **unify criteria** in the sector in order to make life easier for laboratories and vendors.




jtsec
Applus
Thank you