INTERNATIONAL STANDARD

ISO/IEC 17825

First edition
2016-01-15

Information technology — Security techniques — Testing methods for the mitigation of non-invasive attack classes against cryptographic modules

Techonologie de l'information — Techniques de sécurité — Methodes de test pour la protection contre les attaques non intrusives des modules cryptographiques

Reference number
ISO/IEC 17825:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 27, *IT Security techniques*.

Information technology — Security techniques — Testing methods for the mitigation of non-invasive attack classes against cryptographic modules

1 Scope

This International Standard specifies the non-invasive attack mitigation test metrics for determining conformance to the requirements specified in ISO/IEC 19790 for Security Levels 3 and 4. The test metrics are associated with the security functions specified in ISO/IEC 19790. Testing will be conducted at the defined boundary of the cryptographic module and I/O available at its defined boundary.

The test methods used by testing laboratories to test whether the cryptographic module conforms to the requirements specified in ISO/IEC 19790 and the test metrics specified in this International Standard for each of the associated security functions specified in ISO/IEC 19790 are specified in ISO/IEC 24759. The test approach employed in this International Standard is an efficient "push-button" approach: the tests are technically sound, repeatable and have moderate costs.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19790, *Information technology — Security techniques — Security requirements for cryptographic modules*

ISO/IEC 24759, *Information technology — Security techniques — Test requirements for cryptographic modules*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19790 and the following apply.

3.1
advanced SCA
ASCA
advanced exploitation of the fact that the instantaneous side-channels emitted by a cryptographic device depends on the data it processes and on the operation it performs to retrieve secret parameters

3.2
correlation power analysis
CPA
analysis where the correlation coefficient is used as statistical method

3.3
critical security parameter
CSP
security related information whose disclosure or modification can compromise the security of a cryptographic module

EXAMPLE Secret and private cryptographic keys, authentication data such as passwords, PINs, certificates or other trust anchors.

Note 1 to entry: A CSP can be plaintext or encrypted.

[SOURCE: ISO/IEC 19790:2012, definition 3.18]

3.4
CSP class
class into which a CSP is categorised

EXAMPLE Cryptographic keys, authentication data such as passwords, PINs, biometric authentication data.

3.5
differential electromagnetic analysis
DEMA
analysis of the variations of the electromagnetic field emanated from a cryptographic module, using statistical methods on a large number of measured electromagnetic emanations values for determining whether the assumption of the divided subsets of a secret parameter is correct, for the purpose of extracting information correlated to security function operation

3.6
differential power analysis
DPA
analysis of the variations of the electrical power consumption of a cryptographic module, for the purpose of extracting information correlated to cryptographic operation

3.7
electromagnetic analysis
EMA
analysis of the electromagnetic field emanated from a cryptographic module as the result of its logic circuit switching, for the purpose of extracting information correlated to security function operation and subsequently the values of secret parameters such as cryptographic keys

3.8
horizontal attack
HA
modus operandi where sensitive information is extracted from a single measurement split into several parts

3.9
implementation under test
IUT
implementation which is tested based on methods specified in this International Standard

3.10
mutual information analysis
MIA
analysis of the mutual dependence of two random variables

3.11
power analysis
PA
analysis of the electric power consumption of a cryptographic module, for the purpose of extracting information correlated to security function operation and subsequently the values of secret parameters such as cryptographic keys

3.12
rectangle attack
RA
modus operandi where the observations acquisition phase mix horizontal and vertical attacks

3.13
side-channel analysis
SCA
exploitation of the fact that the instantaneous side-channels emitted by a cryptographic device depends on the data it processes and on the operation it performs to retrieve secret parameters

3.14
simple electromagnetic analysis
SEMA
direct (primarily visual) analysis of patterns of instruction execution or logic circuit activities, obtained through monitoring the variations in the electromagnetic field emanated from a cryptographic module, for the purpose of revealing the features and implementations of cryptographic algorithms and subsequently the values of secret parameters

3.15
simple power analysis
SPA
direct (primarily visual) analysis of patterns of instruction execution (or execution of individual instructions), in relation to the electrical power consumption of a cryptographic module, for the purpose of extracting information correlated to a cryptographic operation

3.16
timing analysis
TA
analysis of the variations of the response or execution time of an operation in a security function, which may reveal knowledge of or about a security parameter such as a cryptographic key or PIN

3.17
vertical attack
VA
modus operandi where sensitive information is extracted from different algorithm executions

4 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviated terms given in ISO/IEC 19790 and the following apply.

DLC	Discrete Logarithm Cryptography
ECC	Elliptic Curve Cryptography
EM	Electro-Magnetic
HMAC	keyed-Hashing Message Authentication Code
IFC	Integer Factorization Cryptography
MAC	Message Authentication Code
PC	Personal Computer
PCB	Printed Circuit Board
RBG	Random Bit Generator
RNG	Random Number Generator
USB	Universal Serial Bus
*	multiplication symbol
^	exponentiation symbol

5 Document organization

Clause 6 of this International Standard specifies the non-invasive attack methods that a cryptographic module shall mitigate against for conformance to ISO/IEC 19790.

Clause 7 of this International Standard specifies for each non-invasive attack method the associated security functions specified in ISO/IEC 19790.

Clause 8 of this International Standard specifies the non-invasive attack test methods.

Clause 9 of this International Standard specifies the test methods for side-channel analysis of symmetric-key cryptosystems.

Clause 10 of this International Standard specifies the test methods for side-channel analysis of asymmetric-key cryptosystems.

Clause 11 of this International Standard specifies the non-invasive attack mitigation pass/fail test metrics for each non-invasive attack method to demonstrate conformance to ISO/IEC 19790.

This International Standard shall be used together with ISO/IEC 24759 to demonstrate conformance to ISO/IEC 19790.

6 Non-invasive attack methods

This clause specifies the non-invasive attack methods that need to be addressed for conformance to ISO/IEC 19790.

The non-invasive attacks use side-channels (information gained from the physical implementation of a cryptosystem) emitted by the IUT such as:

— Its power consumption,

— Its electromagnetic emissions,

— Its computation time.

The number of possible side-channels can increase in the future (e.g. photonic emissions [49], acoustic emanations, etc.)

In order to be more formal in the attacks' taxonomy, a formalism will allow the relationships to be highlighted between the different attacks and to have a systematic way to describe a new attack.

An attack is described in the following way:

<YYY>-<XXX>-<ZZZ>

YYY refers to the statistical treatment used in the attack (e.g. « S » for Simple, « C » for Correlation, « MI »for Mutual Information, « ML » for Maximum Likelihood, « D » for Difference of Means, etc.).

NOTE 1 Other statistical treatments can be inserted like « dOC » which corresponds to a correlation treatment exploiting dth order moments (obtained for instance by raising each targeted point in the traces to a power d, or by combining d points per trace before processing the correlation).

XXX refers to the kind of observed side channel: e.g. « PA » for Power Analysis, « EMA » for Electromagnetic Analysis, « TA » for Timing Analysis, etc.

ZZZ may refer to the profiled (« P ») or unprofiled (« UP ») characteristic of the attack. This is optional and the default value is « UP ».

Additionally, an adjective may prefix the attack name to refer to the attack modus operandi. It can be: "Vertical" (classical and default mode), "Horizontal" (see [43] for more details about this) or "Rectangle".

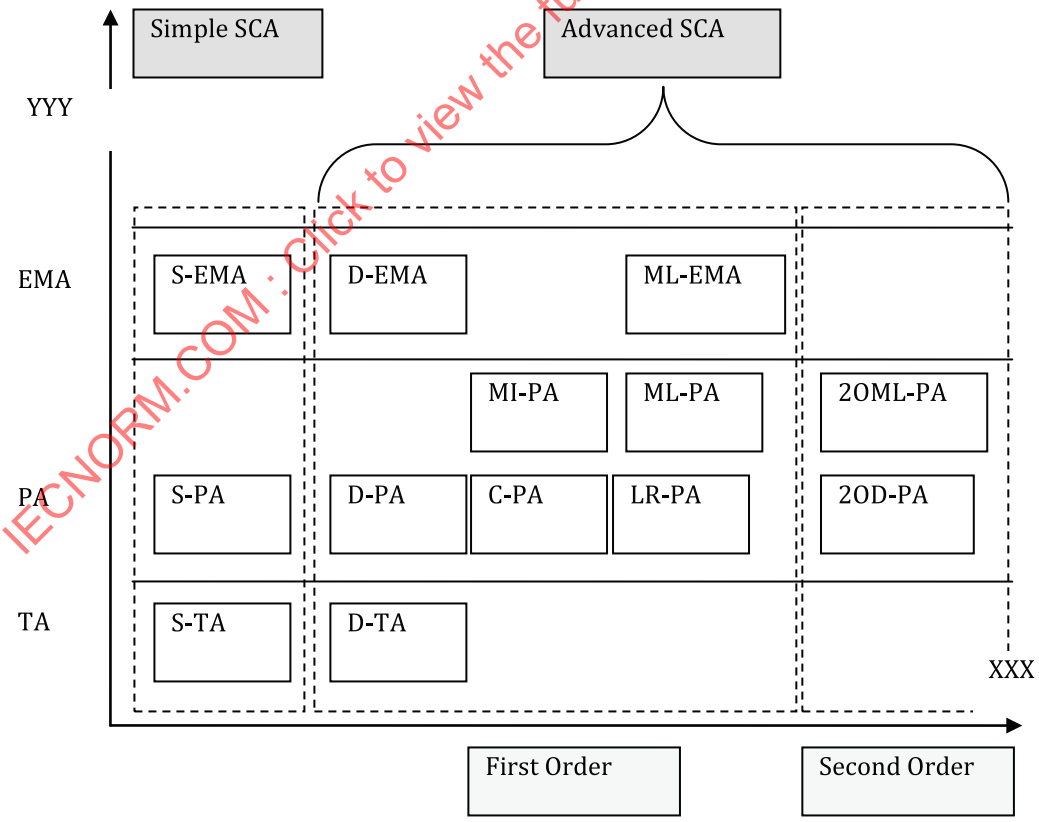

Figure 1 — Taxonomy of non-invasive attacks

NOTE 2 Collision attacks can be viewed as a classical CPA with the single difference that the hypotheses are not deduced from a hypothesis on the way that information leaks from the device but on real measurements that are simply re-arranged in order to (in) validate a hypothesis on a key difference.

NOTE 3 Instead of just splitting ASCA into univariate and multivariate cases, the classification could still be refined by separating attacks based on "variable distinguishers" (which focus on a particular moment of the distribution of the target variable) from those based on "pdf distinguishers" (which try to distinguish a pdf from another). In the first category we have ASCA based on correlation or on the linear regression techniques. In the second one, we have Maximum Likelihood and MI attacks for instance.

NOTE 4 (informative but not applicable) The SPA and SEMA attack methods include some extensions to basic SPA and SEMA attacks, such as the so called template attack. The DPA and DEMA attack methods include some extensions to basic DPA and DEMA attacks, such as so called Correlation Power Analysis (CPA) and higher-order DPA attacks. It is not mandatory to test them in this International Standard.

The variables used in the description of ASCA are:

A	cryptographic processing
C	observation processing
D	number of predictions
d_C	multivariate degree
d_D	multivariate degree
d_o	dimension of observation
F	function, i.e. manipulation
h	Observation
i	Index
K	secret key
k1	sub key 1
k2	sub key 2
M	model of leakage
N	number of observations
o_i	observation interval
(*o_i*)_*i*	observation interval number i
pred_i	Prediction
t_i	i iteration of time
x1_i	i iteration of x1
x2_i	i iteration of x2
X	input text

ASCA is described in the following steps:

a) Measure *N* observations (*o_i*) related to a cryptographic processing *A* parameterized by a known input *X* and a secret *K*.

b) [Optional] Choose a model *M* for the device leakage.

c) [Optional] Choose an observation processing C (by default C is set to the identity function).

d) Make a hypothesis h on the value of K or a subpart of it.

e) From A, h, the $(o\_i)\_i$ and possibly M deduce N predictions $pred\_i$ (one for each value of X for which an observation has been measured).

f) Select a statistical test D and compute D ($pred\_i$,$C(o\_i)$).

g) If D ($pred\_i$,$C(o\_i)$) is greater than some threshold, then validate h. Otherwise, invalidate h and go back to step 4 for a new value h.

NOTE 5 In order to stay generic, a threshold value is added in step 7. This threshold should be carefully chosen for an attack to have any chance to succeed. The classical way to choose such a threshold is to take the maximum value, over all key hypotheses, of D ($pred\_i$, $C(o\_i)$).

NOTE 6 The observations o_i may be univariate or multivariate. In the latter case, each coordinate of o_i, viewed as a vector, corresponds to a different time t_i. The dimension of o_i is denoted by d_o in the rest of this note.

NOTE 7 The observation processing $C(.)$ can always be defined as a polynomial function over the set of real-valued vectors of size d_o (denoted R^{d_o}) in the following. The multivariate degree of this polynomial is denoted by d_C. Then, the function D($pred\_i$, .): $X \rightarrow D$ ($pred\_i$, X) can also be viewed as a polynomial in X. Its multivariate degree is denoted by d_D. The value d_C*d_D is defined as the order d of the attack. For Mutual Information based attacks, only the degree d_C is used to define the attack order: we have $d=d\_C$.

NOTE 8 In collision attacks against block ciphers, the second step is skipped and the third step simply consists in a point selection in the traces o_i. Then, the hypothesis h typically corresponds to a hypothesis between the difference ($k1$-$k2$) of two parts of the targeted key K (e.g. two sub-keys in a block cipher implementation). Eventually, the predictions are deduced from the observations $(o\_i)\_i$ and the difference h: if for instance the attack targets the manipulation of a value $F(x1\_i+k1)$ (i.e. $C(o\_i)$ corresponds to the part of the observation related to the manipulation of $F(x1\_i+k1)$), then the attack will extract from the o_i the observations during the manipulation of another values $F(x2\_i+k2)$ and those observations will be re-arranged such that $x2\_i - x1\_i = h$. Then h_i corresponds to the part of the observation related to the manipulation of $F(x2\_i+k2) = F(x1\_i+k1)$ if h is correct. To validate the hypothesis, a correlation coefficient is usually used for D. Additionally, all the attacks described in this section can be vertical or horizontal or rectangle (i.e. horizontal and vertical). An attack is said to be vertical if each observation o_i corresponds to a different algorithm processing. If all the o_i correspond to a same algorithm processing, then the attack is said to be horizontal. If some o_i share the same algorithm processing while some other o_i do not, then the attack is said to be rectangle. The classical attacks of the Literature are vertical and this modus operandi will hence be defined as the default one. Examples of attacks performed in the horizontal mode can be found in [43] and [44].

NOTE 9 In this International Standard, it is only mandatory to mount vertical attacks.

NOTE 10 An approval authority may modify, add or delete non-invasive attack methods, the association with security functions (see Table 1) and non-invasive attack mitigation test metrics specified in this International Standard.

7 Associated Security Functions

The non-invasive attack methods specified in Clause 6 are associated with the specific security functions that use the CSPs that the attacks target. The security functions are listed in ISO/IEC 19790:2012, Annexes C, D and E.

The associations are shown in Table 1. Other non-invasive attacks and other associations between the attack methods and security functions may exist but defence against them is not currently addressed in this International Standard.

Table 1 — Associations between non-invasive attack methods and security functions covered by this International Standard

Security functions		Non-invasive attack methods		
		SPA/SEMA	DPA/DEMA	TA
Symmetric-Key	AES	A	A	A
	Triple-DES	A	A	A
	Stream Ciphers	A	A	A
Asymmetric-Key	Plain RSA (Key wrapping)	A	A	A
	RSA PKCS#1 v1.5	A	A	A
	RSA PKCS#1 v2.1	NA	NA	A
	DSA	A	A	A
	ECDSA	A	A	A
Hashing mechanisms	SHA	A	NA	NA
RNG and RBG	Deterministic	A	NA	NA
	Non-deterministic	A	NA	NA
Data Authentication Mechanisms	HMAC	A	A	NA
Key Generation		A	NA	NA
Key Derivation from Other Keys		A	A	NA
Key Derivation from Passwords		A	NA	NA
Key Establishment	DLC	A	NA	NA
	IFC	A	NA	NA
Key Entry and Output		NA	NA	NA
Operator Authentication Mechanisms	PIN/Password	A	A	A
	Key	NA	NA	A
	Biometrics	A	NA	A

Legend: A : Applicable, NA : Not-Applicable

NOTE 1 Applicable means that the security functions are susceptible to these types of attacks.

NOTE 2 Not Applicable means that the security functions are not susceptible to these types of attacks.

NOTE 3 An HMAC implementation can be compromised by applying DPA/DEMA, however Block-cipher based MAC will be covered through AES and/or Triple DES.

NOTE 4 All security functions using S Boxes such as AES, Triple-DES etc. can be compromised by applying TA, more precisely cache-timing attacks for software implementations [50].

NOTE 5 RSA PKCS#1 v1.5 can be compromised by applying DPA/DEMA since the used padding is deterministic.

NOTE 6 Timing attacks on RSA PKCS#1 v2.1 are not practicable since the used padding is probabilistic. RSA PKCS#1 v2.1 cannot be compromised by applying DPA/DEMA since the used padding is probabilistic (different random numbers are used for each new signature of the message).

NOTE 7 There are two operations in DSA (resp. ECDSA) that involve the private key or an ephemeral (secret) key:

— The modular exponentiation (scalar multiplication) of a secret value with a known parameter. This operation is vulnerable to simple side-channel analysis and to horizontal differential ones.

— The modular multiplication of a known value and the private key. If the multiplication is implemented in such a way that the multiplier is the private key and the multiplication is carried out with a variant of the binary algorithm, then this implementation is, in principle, vulnerable to side-channel analysis.

NOTE 8 SHA can be used for password hashing e.g. in Password-Based Key Derivation Function, so in this case, a non-protected SHA against SPA/SEMA or DPA/DEMA can lead to the password.

8 Non-invasive Attack Test Methods

8.1 Introduction

This clause presents an overview of the non-invasive attack test methods for the corresponding non-invasive attack methods specified in Clause 6.

8.2 Test Strategy

The goal of non-invasive attack testing is to assess whether a cryptographic module utilising non-invasive attack mitigation techniques can provide resistance to attacks at the desired security level. No standardized testing program can guarantee complete protection against attacks. Rather, effective programs validate that sufficient care was taken in the design and implementation of non-invasive attack mitigations.

Non-invasive attacks exploit a bias latent in the physical quantities non-invasively measured on or around the IUT. Such a bias is induced from and depends on the secret information the attacks target. For background see Reference [16]. The bias may be subtle but is generally persistent. In this International Standard, *the biased information that depends on the secret information* is referred to as *leakage* hereinafter. A device can fail one or more tests if experimental evidence suggests that leaking information exceeds permitted leakage thresholds. This implies that leakage demonstrates a potential vulnerability. Conversely, attacks will fail and the test passes unless leakage is observed. The *test of existence of leakage* will be called *leakage analysis* (*leak analysis*) hereinafter.

The goal is to collect and analyse measurements within certain test limitations such as maximum waveforms collected, elapsed test time, and determine the extent of CSP information leakage. Thus the test limitations and leakage thresholds constitute the test criteria.

Consider timing attack testing. If the test reveals that the computation time is biased relative to the CSP the IUT fails. For DPA if the test reveals that the power consumption during CSP related processes is biased relative to the CSP the IUT fails. The testing approach uses statistical hypothesis testing to determine the likelihood that a bias is present. Thus this International Standard provides a leakage threshold in terms of statistical significance. The test will fail if a bias exceeds the leakage threshold.

8.3 Side-Channel Analysis Workflow

8.3.1 Core Test Flow

The tester collects measurement data from the IUT and applies a suite of statistical tests on the collected data. Core test refers to testing for a single security function with a single CSP class, where CSP classes include cryptographic keys, biometric data or PINs. If some security functions deal with more than one CSP class, leakage analysis for every applicable CSP class will be performed for each security function. The test method requires repeating core tests with different CSP classes until the first fail of test occurs or all the CSP classes pass. If a core test is unable to continue if the IUT limits the number of repeated operations, the result is a pass and the core test is continued with the next CSP class. The core test is shown in Figure 2. Side Channel Resistance Test Framework is depicted in Figure 3. Leakage analysis for TA is shown in Figure 4, SPA/SEMA in Figure 5 and DPA/DEMA in Figure 6.

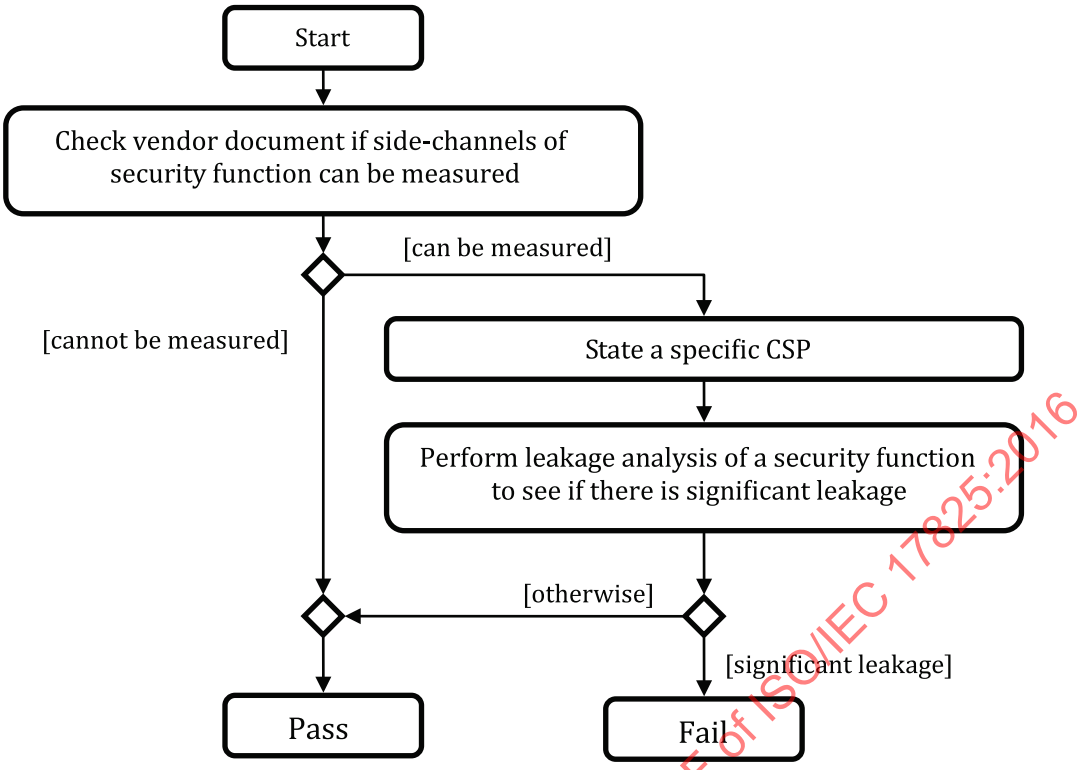

Figure 2 — Core Test Flow

Figure 2 shows the flow of a core test. First, the vendor document is verified for the specified CSP class. Second, the practicality of measuring the physical characteristics is determined. If the measurement cannot be made, the test result is Pass. Third, a set of CSPs determined by the testing laboratory will be configured into the IUT. Finally, the essential part of the core test, the analysis, which is shown in the subsequent figures, is performed and significant leakage is either observed or not.

8.3.2 Side-Channel Resistance Test Framework

As explained in Clause 7, a testing laboratory has to check the security of IUTs against TA, SPA, and DPA.

The sequential test of the three attacks leads to the attack framework depicted in Figure 3. The testing laboratory should follow the order of the operations. For example, the SPA can be tested only if TA passed.

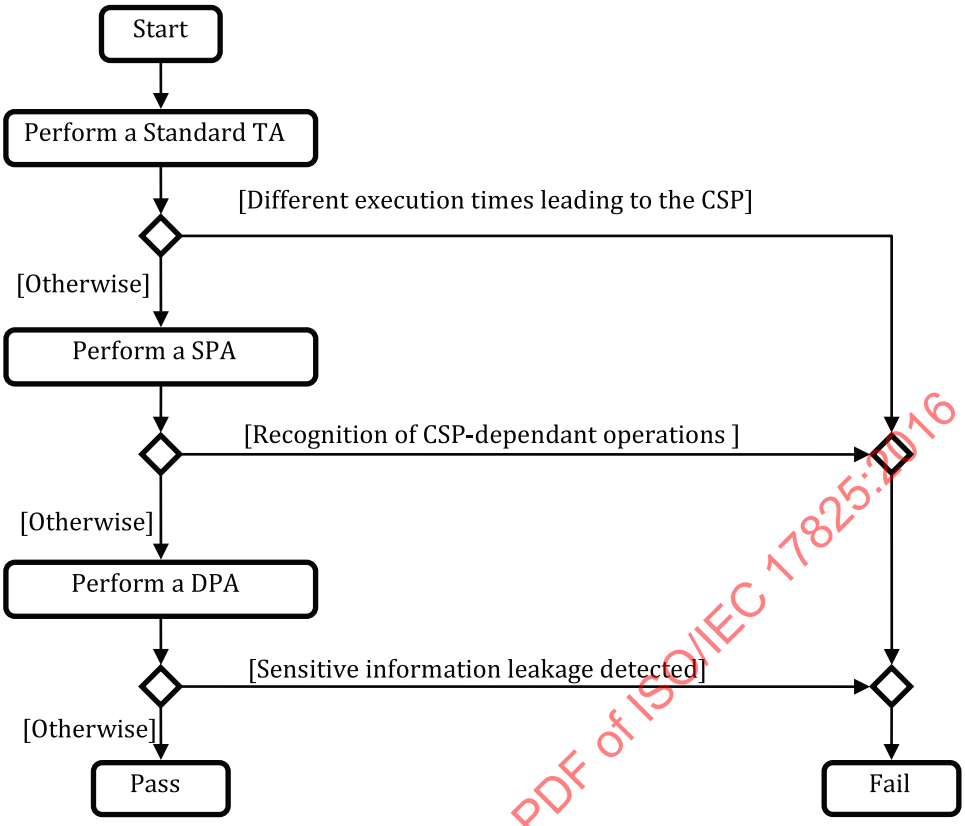

Figure 3 — Side-Channel Resistance Test Framework

The proposed methodology for side-channel resistance assessment does not need full key extraction to fail a device: an IUT can fail if significant sensitive information leakage can be demonstrated.

8.3.3 Required Vendor Information

The vendor shall provide the following information about the algorithms and countermeasures implemented in the IUT:

a) Implemented cryptographic algorithms.

b) Design of the implementation.

c) The conditions/mode(s) of usage where the IUT is susceptible to side-channel analysis.

Moreover, the testing laboratory shall be able to modify CSPs and cipher text when performing side-channel testing.

When performing side-channel analysis, it is common to perform signal alignment so that different traces can be compared at the same point during the cryptographic calculation. For the purposes of side-channel testing, the vendor shall provide the testing laboratory the best synchronization signal for the start of the cryptographic operation. For example, in testing mode the device may provide an external trigger point to indicate the start or stop of the cryptographic operation. If such start and stop information is not available, the testing laboratory should adopt standard signal processing- and matching-based techniques to perform alignment. In cases where traces are well aligned at the start of the cryptographic operation, the lab may be required to use standard, least-squares fit-based signal matching to perform better alignment on specific internals of the algorithm; the number and locations of these alignment points will be specified by the testing laboratory.

The vendor should then provide a so-called "calibration function" that will allow the testing laboratory to:

a) synchronize its measurements,

b) check the quality of its measurements (see 8.3.5 for more details).

8.3.4 TA Leakage Analysis

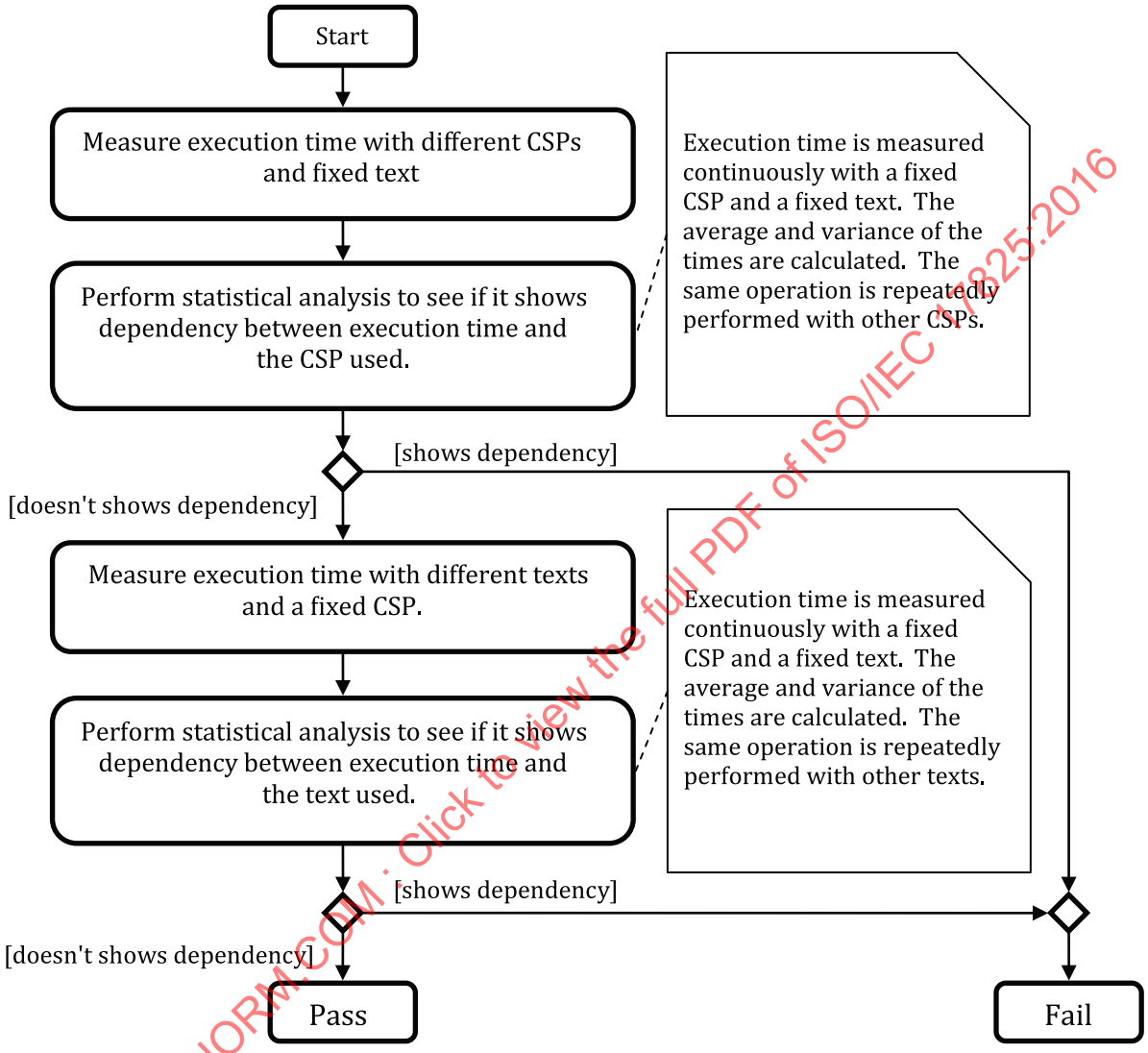

Figure 4 — Leakage Analysis for Timing Attacks

Figure 4 shows the Leakage Analysis flow for Timing Attacks. The flow can be divided into two stages. For the first stage, execution times with several different CSPs and fixed text are measured. If the measured execution time doesn't show dependency with the CSP used through statistical analysis, then the test continues to the second stage. Otherwise, the test fails. For the second stage, execution times with several different texts and a fixed CSP are measured. If the measured execution time doesn't show dependency with the text used, the test passes. Otherwise, the test fails. If the execution time is difficult to measure a tolerance value ε equals a clock cycle of the targeted chip should be used. To compare two time values (or two average time values) T_1 and T_2, the test pass if $|T_1 - T_2| < \varepsilon$, and fail otherwise.

Not only the difference of means, but also of variances, shall be computed, so as to detect second-order timing leakage. Indeed, high-order timing attacks are practical threats [58].

8.3.5 SPA/SEMA Leakage Analysis

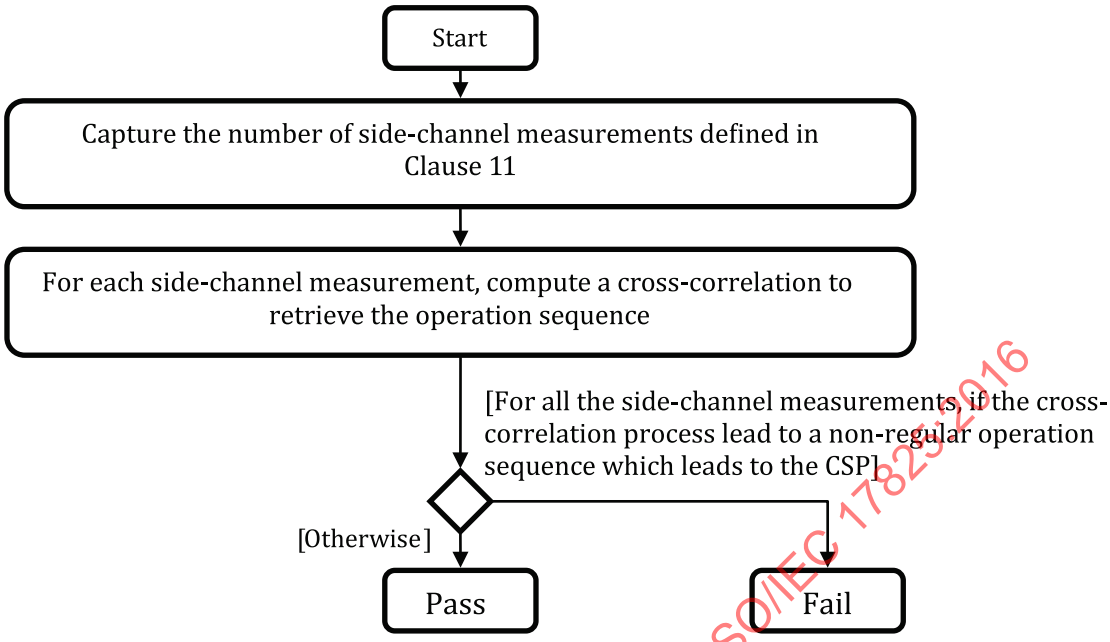

Figure 5 — SPA (SEMA) Leakage Analysis

Figure 5 shows the SPA/SEMA Leakage Analysis flow. The flow can be divided into two stages.

First, the testing laboratory shall capture the number of side-channel measurements related to the desired Security Level (see Clause 11).

Asymmetric cryptography repeatedly uses elementary operations. For RSA these are modular square (denoted "S") and multiply (denoted "M") operations. For ECC, these are point doubling and addition operations. Since the key may be derived from the order of operations it is important for the testing laboratory to distinguish these operations. As side-channel measurements can be noisy, it may be difficult to recognize these operations visually. A good method to identify a repeating operation is called "cross-correlation". This method also helps to remove subjective assessment from the testing laboratory. When the correlation is so weak that no definite statement can be taken, the testing laboratory can mount a cluster analysis.

For all the side-channel measurements, if the cross-correlation process leads to a non-regular operation sequence which leads to the CSP, the test result is Fail.

8.3.6 DPA/DEMA Leakage Analysis

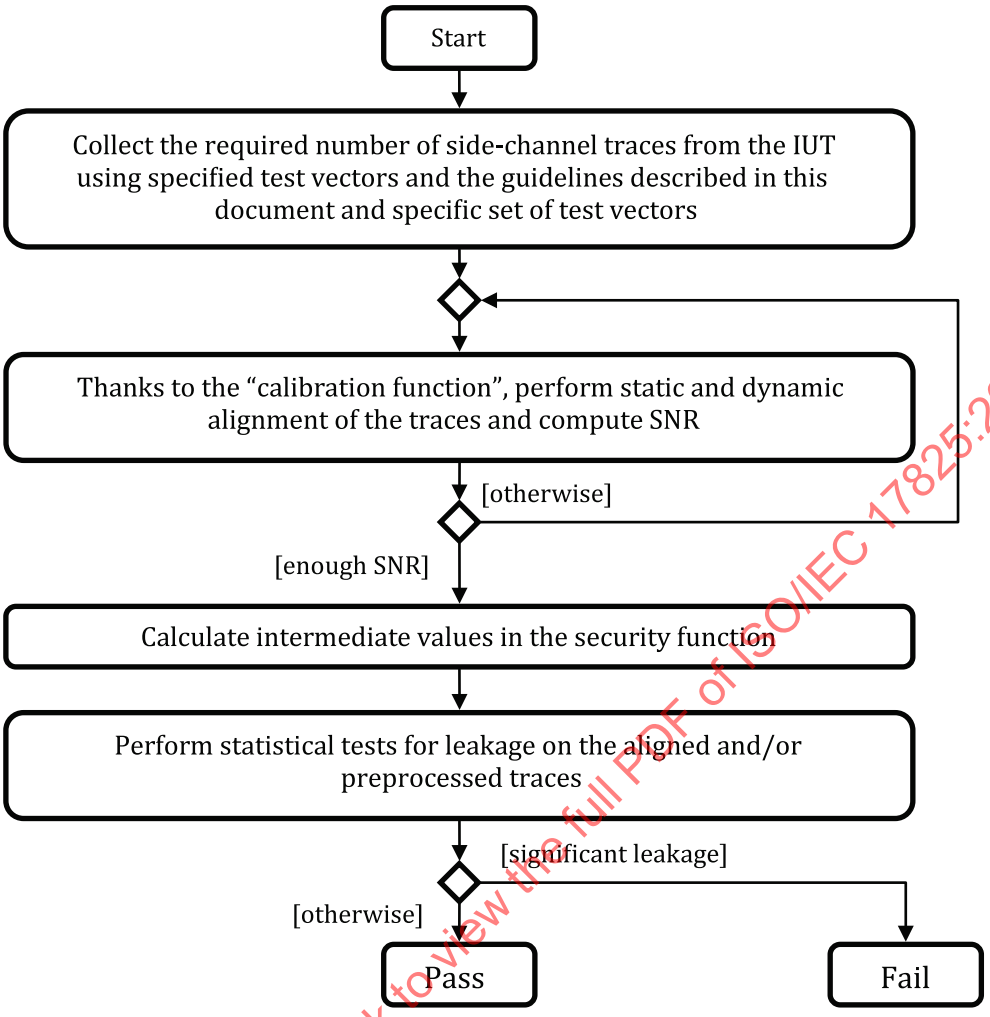

Figure 6 — DPA (DEMA) Leakage Analysis

Figure 6 shows the DPA/DEMA Leakage Analysis flow. The result represented indicates whether or not significant leakage has been observed. In case of doubt clear box or white box will be needed to clarify the ambiguity (see [45]). Indeed, if the test is a t-test, not all leakages are sensitive: typically, possible test violations are incurred by non-CSP variables, such as the plaintext or the ciphertext of a block cipher. Therefore, based on the analysis of the IUT documentation, it can be decided whether test violations depend on the CSP or not. Notice that the NICV test does not feature such ambiguity. The same expressions apply to Figure 6.

As a general rule, it is supposed that the cryptographic operations occur always in the same moment in each measurement (consumptions or emanations). Nonetheless, the developers have the possibility to include internal clocks modifying the operation frequency or introduce randomly non-operative wait status in the algorithms execution, thus the time is no longer constant and the cryptographic operations are not performed in the same instant. This produces the well-known misalignments in the set of traces, making the analysis difficult and much more costly in terms of the number of traces needed to be processed. These modifications of the original behaviour are countermeasures implemented by the developers to counteract the possibility to acquire information through side channels, breaking the assumptions that characterize the known attacks.

In cryptographic implementations without specific countermeasures, misalignments come from errors in the measurement configuration, when starting the power consumption (or emanations) acquisition. In this case, the traces may be aligned if the vagueness can be determined when launching the measurement, displacing properly the traces. This process is called "*static alignment*". This vagueness

can also be mitigated or, at least, facilitates the alignment, if a trigger is provided (or exists) signalling when the operation starts.

When the implementation actively introduces random timing delays or clock frequency variations, the static displacement cannot attain the full alignment of the traces. In this case, the so called "*dynamic alignment*" is to be applied by matching parts of traces with different displacements and performing a non-linear sampling of the traces. After this process, the different parts along the traces are located in the same positions (e.g. the location of the different rounds matches in all the traces).

The vendor shall then collaborate with the testing laboratory by implementing in the IUT a so-called "calibration function" that helps the testing laboratory to synchronize the waveforms (by providing a trigger signalling the beginning of the cryptographic operation) and check the quality of its side-channel measurements. This can also help to test external noise reduction methods (filtering in frequency, mean calculation, etc.). This calibration function can be simply the processing/storage of a known public (non-sensitive) variable in the IUT (e.g. the public key *e* in RSA). The testing laboratory should retrieve this known value. If the SNR of the side-channel measurements portion corresponding to processing of this known value is sufficient, the testing laboratory can perform the tests. If not, the testing laboratory shall find a way to improve the quality of its measurements before performing the tests.

The testing laboratory shall then calculate intermediate values in the security function. It is feasible since the testing laboratory collects side-channel measurements using a pre-specified set of test vectors. These test-vectors are carefully chosen by the testing laboratory to expose and isolate potential leakages.

The last step consists in performing statistical tests for leakage on the aligned and/or pre-processed traces. The testing laboratory shall apply a simple statistical test (called Welch's test) to multiple, pre-specified data sets in order to detect sensitive information leakage in the side-channel.

Clauses 9 and 10 respectively describe the guidelines for assessing the DPA/DEMA resistance of symmetric and asymmetric cryptosystems.

9 Side-Channel Analysis of Symmetric-Key Cryptosystems

9.1 Introduction

This Clause focuses on Side-Channel Analysis of Symmetric-Key Cryptosystems. The framework depicted in Figure 3 is used: resistance against Timing Attacks, Simple Side-Channel Analysis, and Differential Side-Channel Analysis shall be assessed.

9.2 Timing Attacks

For (software) Symmetric-Key Cryptosystems, the only known threat related to timing attacks concerns cache-timing attacks [50]. They rely on the micro-architectural properties of the CPU (e.g. cache architecture, branch prediction unit). Cache attacks exploit the cache behaviour (i.e. cache hit/miss statistics) of cryptosystems. Cache architecture leaks information about memory access patterns. The execution time is a source of leakage (cache misses take more time to execute than a cache hit). Cryptosystems have dependant memory access patterns. Once the access patterns are extracted, the testing laboratory can recover the secret key.

If the IUT is a software/firmware implementation of a symmetric-Key Cryptosystems, and if the IUT contains a cache-memory, the testing laboratory can test the IUT against Timing Attacks following the framework described in [50]. In the contrary case, the test result is Pass.

9.3 SPA/SEMA

9.3.1 Attacks on Key Derivation Process

For Symmetric-Key Cryptosystems, the only known threat related to SPA or SEMA attacks concerns Key Derivation Process (Key Schedule). If the testing laboratory can determine the Hamming weights

of intermediate values that occur in a Symmetric-Key Cryptosystem, it allows the key to be revealed. For example, for AES [53], the testing laboratory can use the dependencies between the bytes of the round keys within the AES Key Schedule to reduce the number of possible key values. The key is then determined by using a known plaintext-ciphertext pair.

If the IUT contains a Key Derivation Process, the testing laboratory can try to apply some known methods to extract the key (e.g. [53] for AES), or any other relevant method.

9.3.2 Collision Attacks

Collision attacks also make use of the fact that side-channel measurements of the IUT depend on the processed data. Cryptographic security functions will include some steps to produce intermediate values from input value and cryptographic key. If the intermediate values become the same in value against different input values, the resultant power consumption or electromagnetic emanation would be quite similar. This kind of "collision" can be exploited in order to reduce the key space (see [38-42]).

The testing laboratory can check if the IUT is susceptible to such collision attacks, by following a known framework (e.g. [39] for AES) or any other relevant method.

9.4 DPA/DEMA

9.4.1 Introduction

The statistical test shall be made according to the following. The side-channel traces will be divided into two subsets such that the sensitive information being processed is significantly different between the two subsets [20]. This partitioning is feasible since the cryptographic algorithms are performed with known parameters and data, and all intermediate states are known.

If the side-channel traces in the two subsets are statistically different with high confidence, then information leakage is present and the device fails. Otherwise, information leakage is either not present or is suppressed.

The statistical tool which is used is the Welch t-test. A high positive or negative value of the t-test statistic value T (defined below) at a point in time indicates a high degree of confidence that the null hypothesis (i.e. the two subsets means are equal) is incorrect. A confidence level is arbitrary between 0 and 1, but normally chosen close to 1 indicating the high confidence based on the desired criteria, such as 0.99 (or 99%). The confidence level determines the threshold value C for the positive and negative threshold ($+C/-C$) for T with the t distribution. For example, the confidence level 99.99% corresponds to C=3.9 and 99.999% corresponds to C=4.5.

The t-test shall be repeated since some false positives can appear in one experiment. Two independent experiments are required, and a device can be rejected only if the t-test statistic exceeds $+/-C$ at the same time, in the same direction, in both experiments.

For each algorithm, multiple t-tests shall be performed, each targeting a different type of leakage. Each test shall be repeated twice, with two different data sets. Figure 7 describes the general statistical test procedure.

Before processing the statistical test, the testing laboratory shall specify which set of traces will be used for the test. The set of traces are divided into two divided groups, Group 1 and Group 2. These are the two disjoint data sets for performing the two independent Welch t-tests.

We denote in the following:

N_A, N_B the size of the subsets A and B

μ_A *average* of all the traces in group A

μ_B the *average* of all the traces in group B

σ_A the *sample standard deviation* of all the traces in group A

σ_B the *sample standard deviation* of all the traces in group B

T_1 $$\frac{(\mu_{A1} - \mu_{B1})}{\sqrt{\frac{\sigma_{A1}^{\ 2}}{N_{A1}} + \frac{\sigma_{B1}^{\ 2}}{N_{B1}}}}$$

T_2 $$\frac{(\mu_{A2} - \mu_{B2})}{\sqrt{\frac{\sigma_{A2}^{\ 2}}{N_{A2}} + \frac{\sigma_{B2}^{\ 2}}{N_{B2}}}}$$

In addition to the t-test, the NICV (Normalized Inter-Class Variance, aka coefficient of determination) [60], [61] can be used. The advantages are:

a) It can be multibit.

b) It is also comparable between implementations, as it is bounded between 0 and 1.

c) It relates to the Pearson correlation coefficient ρ like $0 \leq \rho^2 \leq \text{NICV} \leq 1$.

d) It generalizes naturally to high-order leakage.

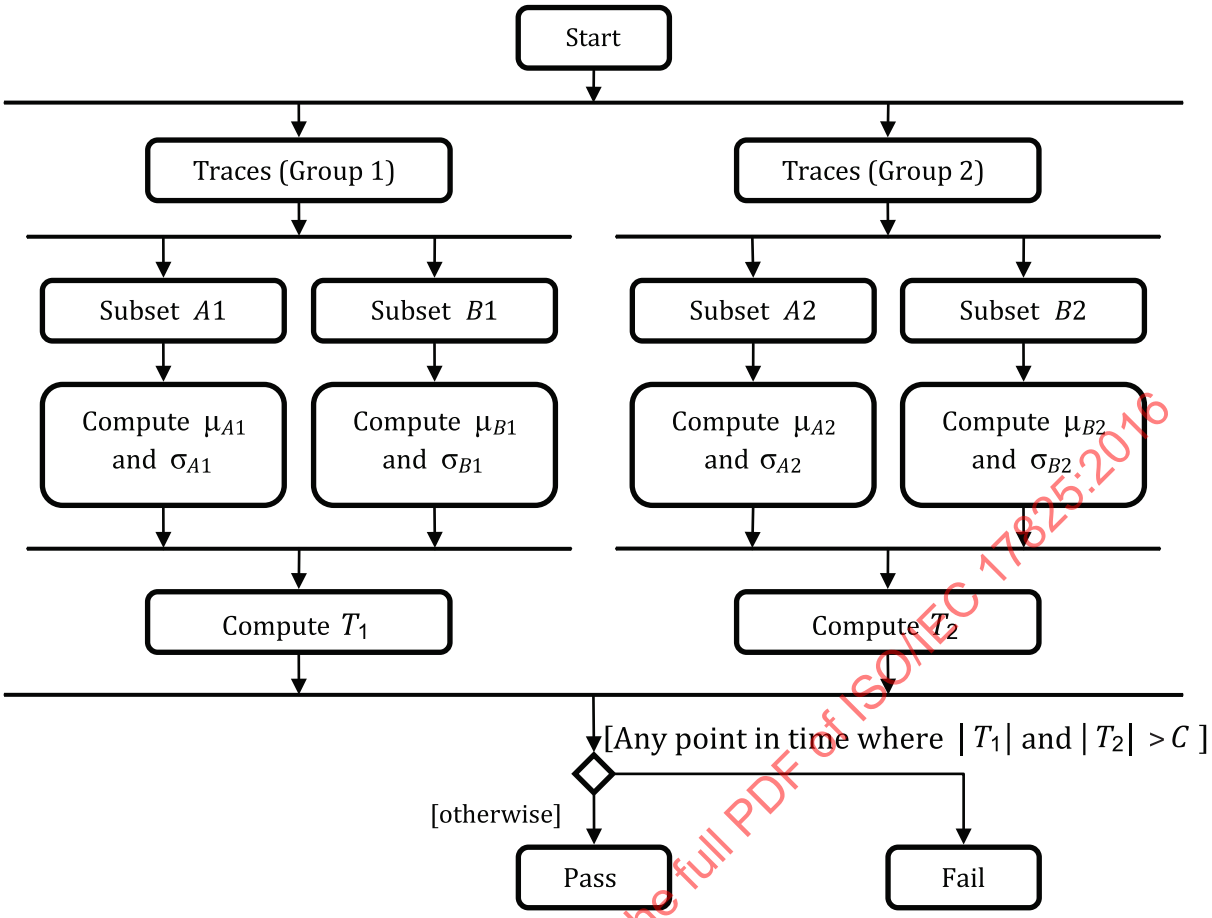

Figure 7 — General Statistical Test Procedure

9.4.2 Test Vectors

The testing laboratory shall collect two data sets DATA-SET 1 and DATA-SET 2 from the cryptographic block encryption with a specific, published key and a set of data as follows:

a) DATA-SET 1:

1) Key K is set to

i) `0x0123456789abcdef` for 64-bit cryptosystems

ii) `0x0123456789abcdef123456789abcdef0` for 128-bit cryptosystems

iii) `0x0123456789abcdef123456789abcdef023456789abcdef01` for 192-bit cryptosystems

iv) `0x0123456789abcdef123456789abcdef023456789abcdef013456789abcdef012` for 256-bit cryptosystems

2) Let n be the number of distinct samples deemed reasonable for an attacker to collect where

j is index

l_0 is input of encryption 0

l_1 is input of encryption 1...

l_{2n} is input of encryption $2n$

$2n$ encryptions shall be performed with inputs: I_0, I_1, ..., I_{2n}, where the input for the first encryption is all zeros and each subsequent encryption uses the output of the previous encryption as its input, i.e. if the cryptosystem is denoted f, $I_j + 1 = f(K, I_j)$ for $0 < j < 2n$.

b) DATA-SET 2:

1) Key K is set to

i) `0x0123456789abcdef` for 64-bit cryptosystems

ii) `0x0123456789abcdef123456789abcdef0` for 128-bit cryptosystems

iii) `0x0123456789abcdef123456789abcdef023456789abcdef01` for 192-bit cryptosystems

iv) `0x0123456789abcdef123456789abcdef023456789abcdef013456789abcdef012` for 256-bit cryptosystems

2) Data (plaintext or ciphertext) J is selected such that the following conditions are met in one middle round of the cryptosystem:

i) There is at least one byte of round_in XOR round_out equal to zero (for Test 1).

ii) There is at least one substitution part (SBox) output equal to zero (for Test 2).

iii) There is at least one byte of data XOR round key equal to zero, e.g. AddRoundKey operation in AES (for Test 3).

iv) There is at least one data byte equal to zero (for Test 4).

The testing laboratory is free to use any method to generate a convenient J. The middle round R (outside first and last round) is chosen by the testing laboratory without informing the vendor.

c) Perform n encryptions with input J.

DATA-SET 2 uses the same key as DATA-SET 1, but repeatedly performs an encryption with a single fixed data value. Both DATA-SET 1 and DATA-SET 2 require the entire cryptosystem operation to be measured, recorded and checked.

9.4.3 Detailed Procedure

Based on proposed data collection for AES, the following tests described below shall be carried out: Test 0, Test 1, Test 2, Test 3, Test 4, Test 5. If the IUT fails at least one of these tests, the test result is Fail.

Test 0: Encryptions with a single fixed data value vs. random encryptions. In this test, the region of interest is the middle 1/3 of cryptosystem operation.

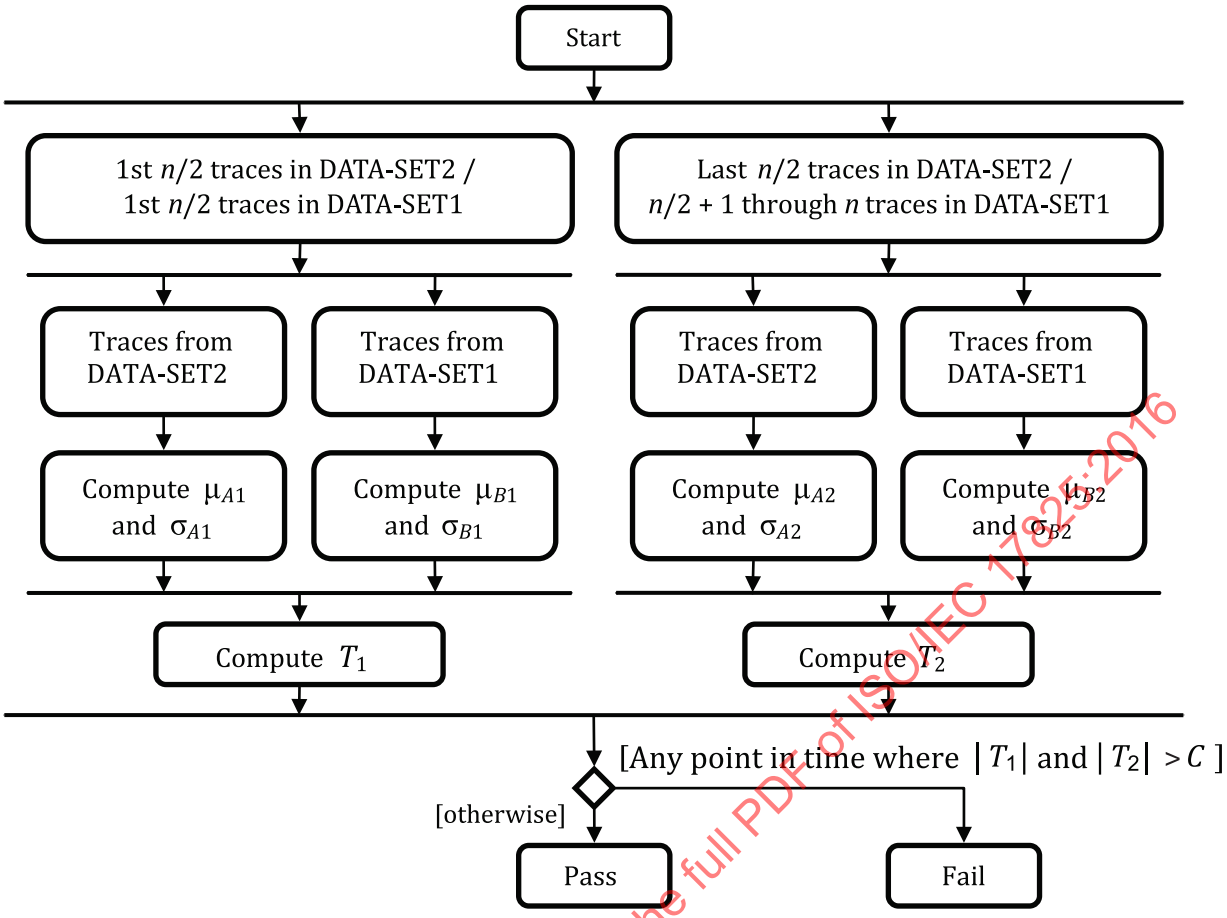

Figure 8 — Test 0

Test 1: leakage test 1. XOR of round input and output.

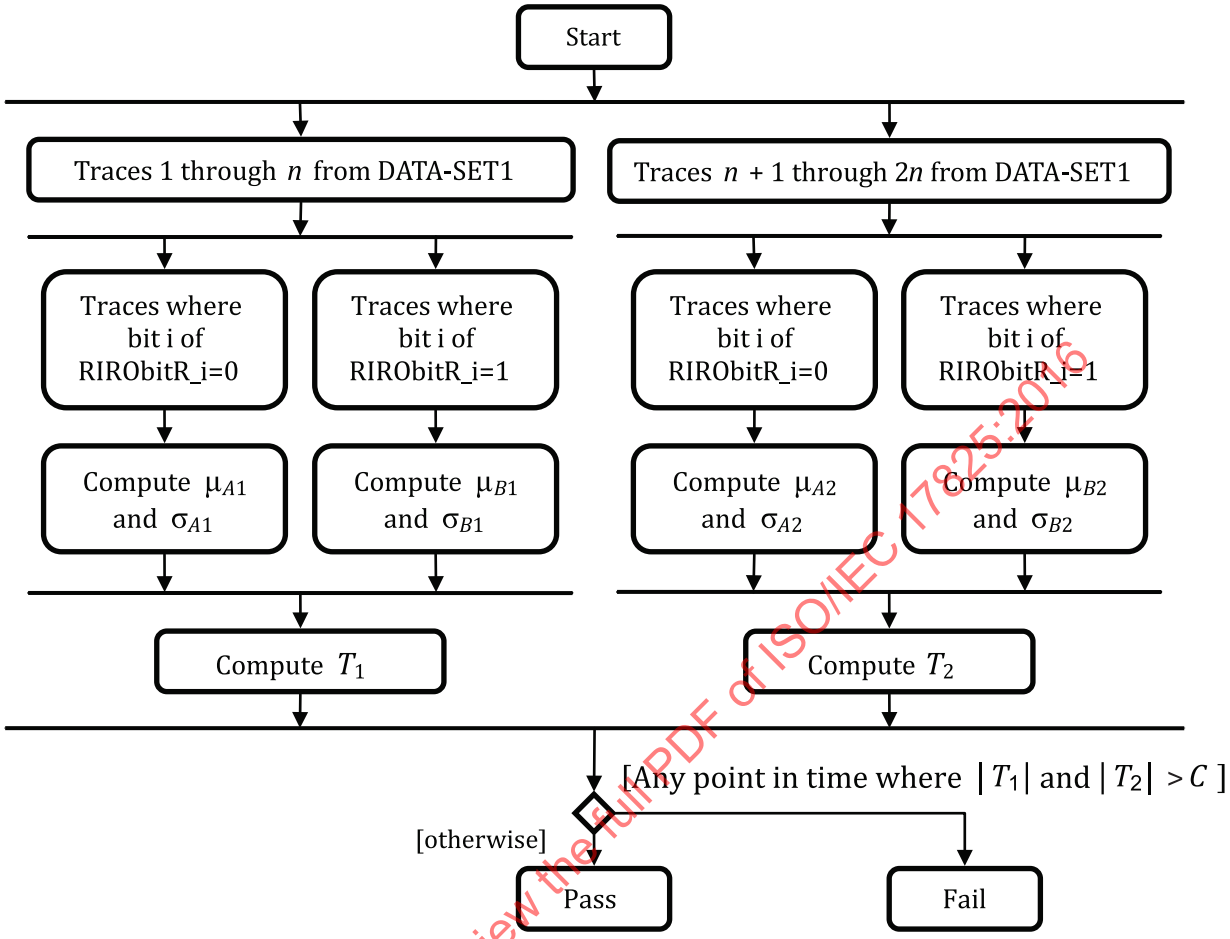

Figure 9 — Test 1 (where RIRObitR_i = Round R input XOR Round R output), for i from 0 to 127

Test 2: leakage test 2. S-box output for round.

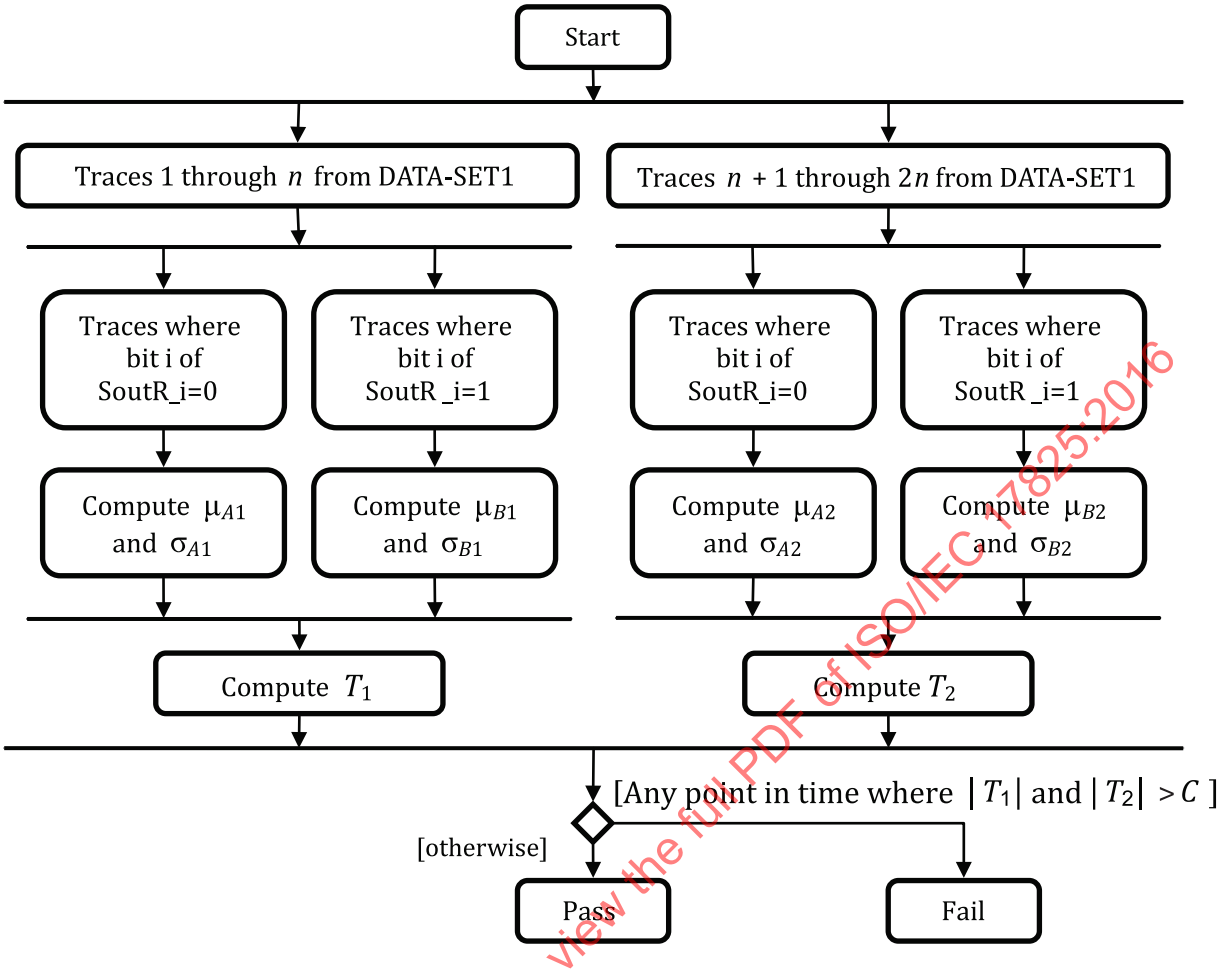

Figure 10 — Test 2 (where SoutR is the concatenated output of the 16 SBox table lookups), for i from 0 to 127

Test 3: leakage test 3. Round output.

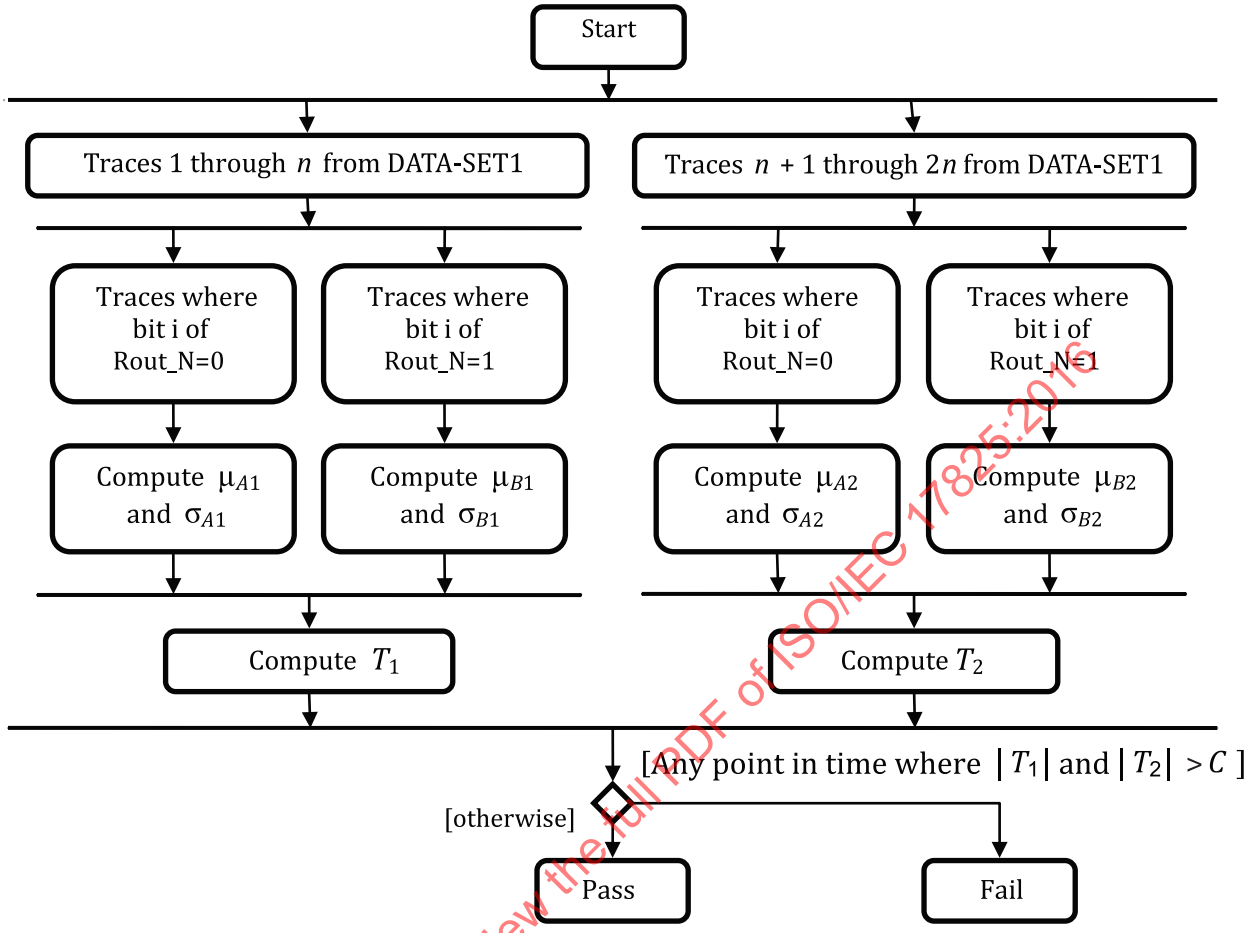

Figure 11 — Test 3 (where Rout_N = Output of Round N), for i from 0 to 127

Test 4: leakage test 4. Byte analysis of round output (each value for first byte).

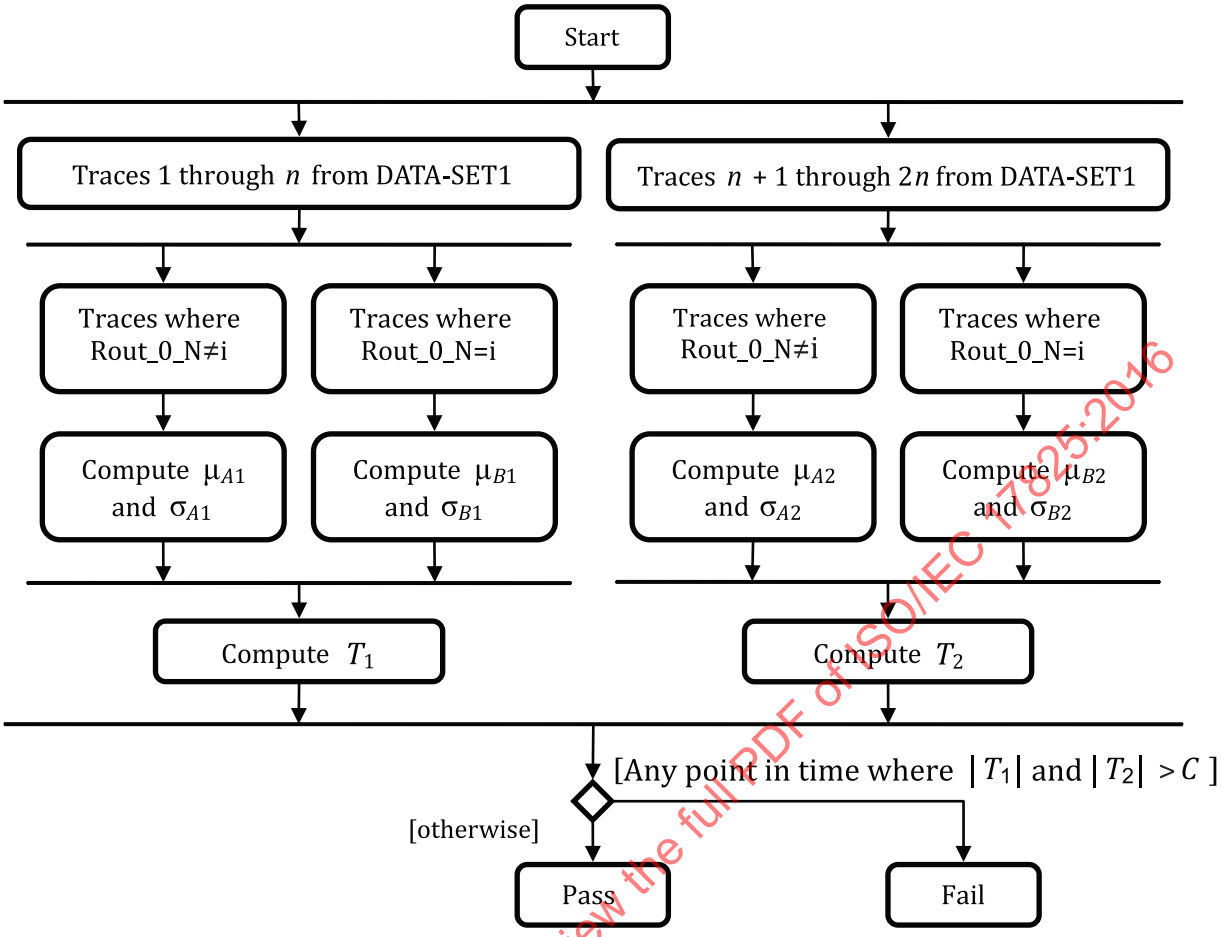

Figure 12 — Test 4 (where Rout_0_N is the first byte of Round N), for i from 0 to 127

Test 5: leakage test 5. Byte analysis of round output (each value for second byte).

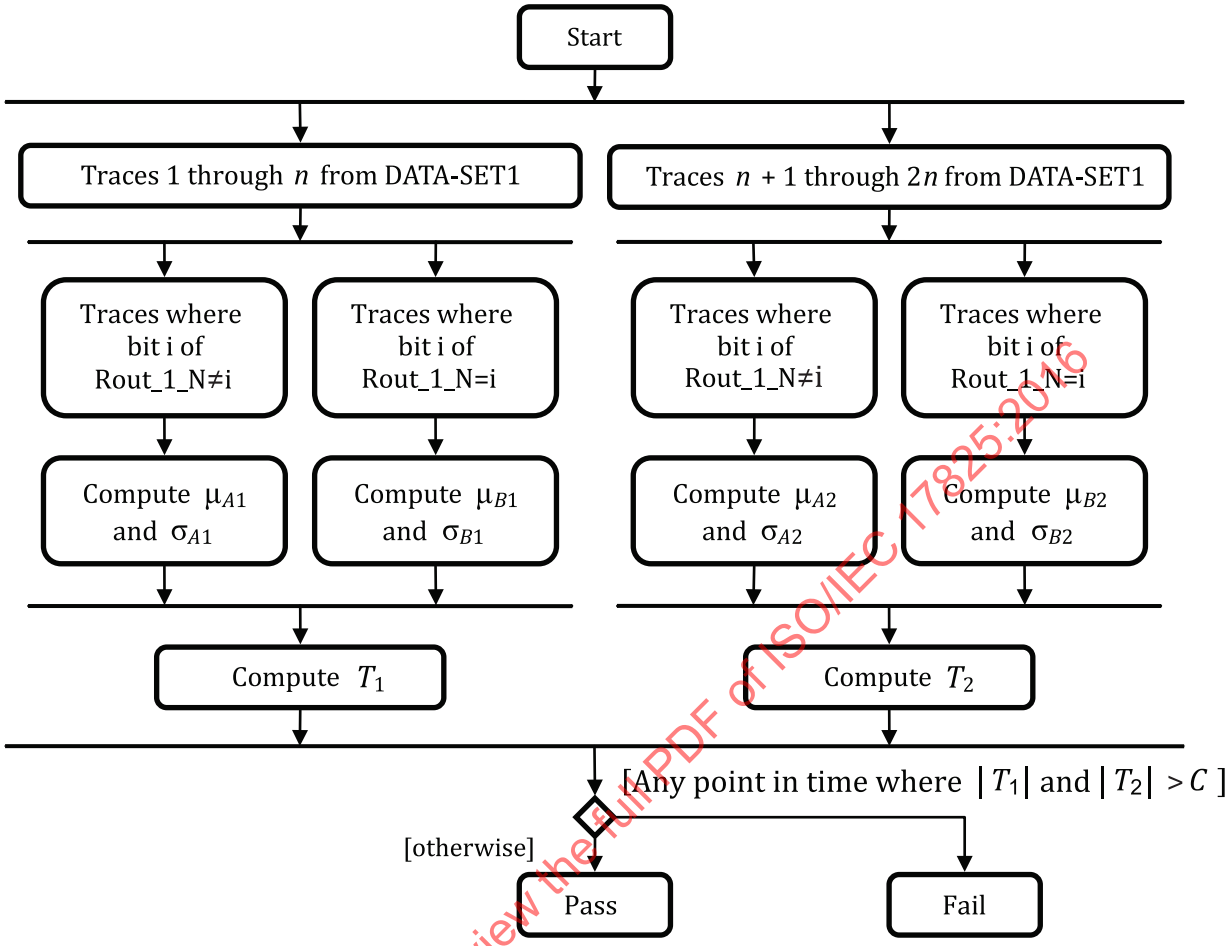

Figure 13 — Test 5 (where Rout_1_N is the second byte of Round N), for i from 0 to 127

NOTE 1 Applying DPA/DEMA on the Key Derivation Process (e.g. Key Schedule) of an IUT can also allow retrieving the key. If the IUT contains a Key Derivation Process, the testing laboratory can try to apply any relevant method to extract the key.

NOTE 2 Attacking Stream Ciphers is also possible by applying the generic methodology described in 9.3.

10 ASCA on Asymmetric Cryptography

10.1 Introduction

This Clause focuses on Side-Channel Analysis of asymmetric-key cryptosystems.

Asymmetric algorithms can fulfil three different tasks: signature, encryption and key agreement. As pointed out in Table 1, the most used asymmetric algorithms are RSA and Elliptic Curve Cryptosystems (ECC).

For signature, encryption or key agreement, the main operation is the computation of

a) a modular exponentiation in the case of RSA; that is the computation of m^d mod n for some integer m, private key d and an integer n, and

b) an elliptic curve scalar multiplication in the case of ECC; that is the computation of $d.p$ for a point on the given elliptic curve and the private key d.

There are many possibilities to compute a modular exponentiation and an elliptic curve scalar multiplication. The actual exponentiation algorithms depend on the integers' representation; on the arithmetic modular operations.

NOTE Actual exponentiation algorithm implementations also depend on the performance, complexity, compactness and targeted security level, and therefore result from a necessary tradeoff.

The framework depicted in Figure 3 is used: resistance against Timing Attacks, Simple Side-Channel Analysis, and Differential Side-Channel Analysis shall be assessed.

Regarding ECC, a complete survey of side-channel attacks is available in [59].

10.2 Detailed Side-Channel Resistance Test Framework

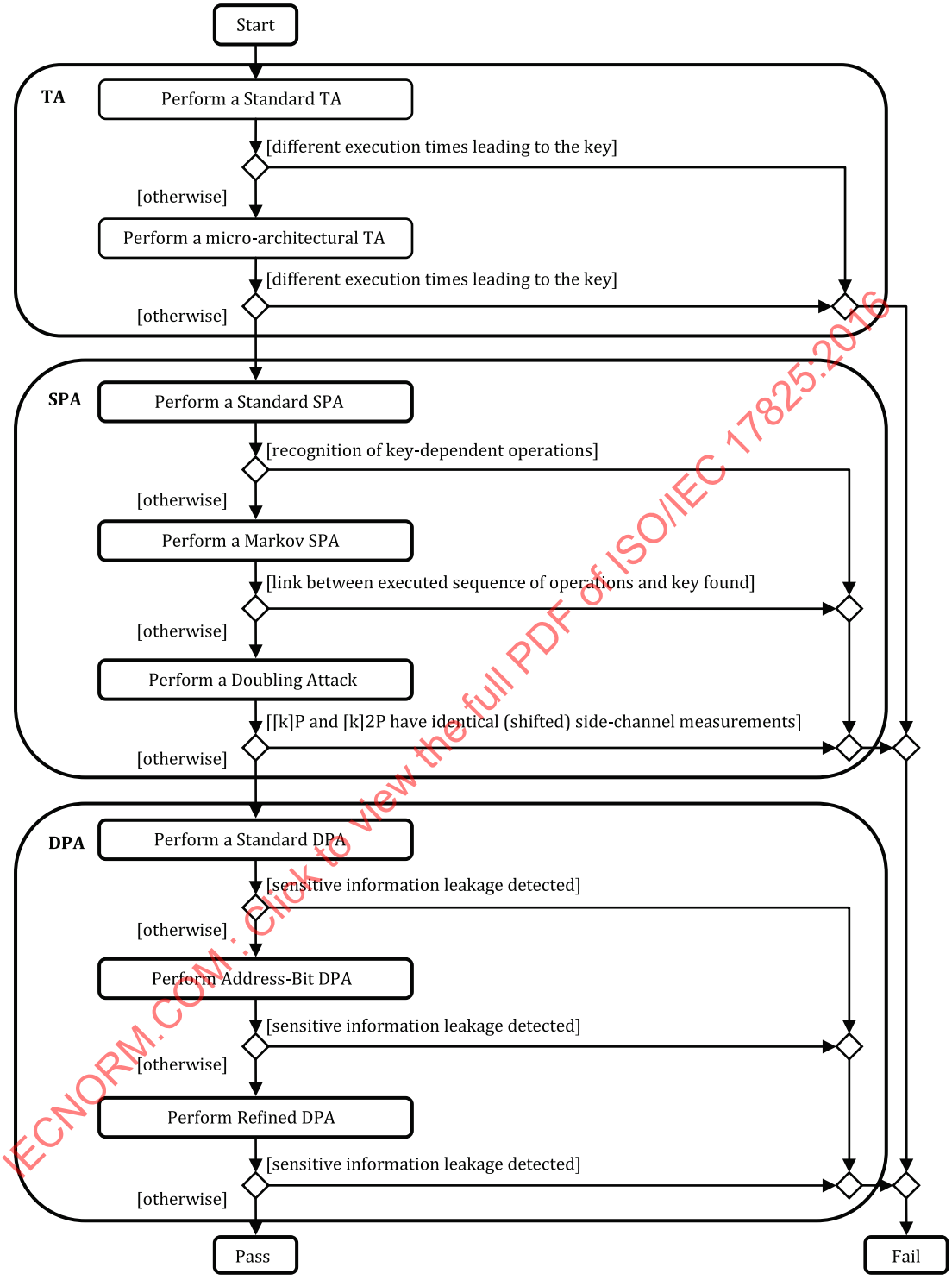

Figure 14 — Side-Channel Resistance Test Framework

Compared to symmetric-key cryptosystems, there exist more ways to attack asymmetric-key cryptosystems due to their varieties of structure and underlying arithmetic. Figure 14 describes the side-channel resistance test framework for asymmetric-key cryptosystems. The security tests are aimed to check the security against conventional attacks such as the TA, SPA, DPA but also with more sophisticated attacks (Address-bit DPA, Markov SPA, Doubling Attacks), and each of these is a particular attack on asymmetric ciphers. The details are explained in 10.3, 10.4, and 10.5.

10.3 Timing Attacks

10.3.1 Introduction

With respect to Figure 14, the testing laboratory should assess the security of asymmetric-key cryptosystems against standard and micro-architectural timing analysis.

10.3.2 Standard Timing Analysis

Since asymmetric-key cryptosystems computes key-dependant operations and can use basic mathematical operations with variable time computations (e.g. modular multiplication), they can be vulnerable to timing attacks. The following details the method described in [2] to attack RSA with a timing attack.

In [2], attacks are done on an IUT which implements:

— A modular multiplication (Montgomery method) that shows computation time variations,

— The standard square-and-multiply exponentiation routine that allows these variations to be exploited.

The result of each multiplication lies in [0,2N-1], where N is the modulus.

Concerning the exponentiation routine, the IUT computes a multiply step when the current secret key bit *d_I* is equal to 1. If the result of the multiply step is greater than *N*, a subtraction by *N* is computed.

In [2], the testing laboratory shall have knowledge of secret key bits *d_k*-1 to *d_k-i*+1 when attacking *d_k*. Knowing the message, the intermediate value after the square step (called s) at iteration *k - i* is computed. Whether the subtraction in the multiply step is required may be stated.

The attack is based on an oracle. The oracle is a clone of the IUT, in which the tester can change the CSP. The testing laboratory shall:

a) Sign with same (*d*, *N*) for many random messages,

b) Make the assumption that *d_k-i* =1,

c) Construct 2 sets of messages depending on the fact that the subtraction happens (set A) or not (set B) during the multiplication.

In case:

— *d_k-i* = 0, global times for sets A and B are not statistically distinguishable (the split is based on a multiplication which does not occur);

— *d_k-i* = 1, global times for sets A and B show a statistical difference related to the optional subtraction (the multiplication does occur).

Time measurements validate or invalidate the oracle. The testing laboratory shall compute the mean of the global duration for each subset. <A> (resp. <B>) is the mean global duration for messages A (B). The oracle criterion is the following:

— If <A> - <B> > 0, then the oracle was right (*d_k-i* =1);

— If <A> - <B> = 0, then the oracle was wrong (d[k-i] = 0).

If the testing laboratory finally retrieves the entire key with this method (coming from[2]) or any other relevant one, the test result is Fail. Otherwise, the test result is Pass.

NOTE RSA PKCS#1 v2.1, DSA and ECDSA are not vulnerable to the standard timing analysis since:

— The used padding for RSA PKCS#1 v2.1 is probabilistic and the attacker cannot predict on the intermediate values;

— DSA and ECDSA use an ephemeral exponent and scalar respectively, so the attacker cannot target a specific bit.

10.3.3 Micro-Architectural Timing Analysis

(Software) asymmetric-key cryptosystems are also vulnerable to micro-architectural attacks. This side-channel is enabled by the branch prediction capability common to all modern CPUs. The penalty paid (extra clock cycles) for a mispredicted branch can be used for cryptanalysis of cryptographic primitives that employ a data-dependent program flow. (Software) asymmetric-key cryptosystems are then susceptible to so-called "Branch Prediction Analysis" [52]. The testing laboratory should test the IUT against Timing (Micro-Architectural) Attacks following the framework described in [52], or any other relevant method.

10.4 SPA/SEMA

10.4.1 Introduction

With respect to Figure 14, the testing laboratory should assess the security of asymmetric-key cryptosystems against standard SPA, so-called "Markov-SPA" and doubling attack.

There are three main kinds of different attacks:

— Since asymmetric-key cryptosystems executes key-dependant sequence of operations, it is naturally vulnerable to "standard" SPA/SEMA (see 10.4.2);

— Some SPA/SEMA countermeasures lead to sequence of operations that can bring enough information to retrieve the key. "Markov SPA/SEMA" can then allow to clear the dependency between the sequence and the key (see 10.4.3).

— In some cases, a testing laboratory can send specific input vectors to the IUT that will allow us to retrieve the key with SPA/SEMA using a small number of waveforms. That is the principle of "doubling attacks".

10.4.2 Standard SPA/SEMA

The testing laboratory should use the framework described in 8.3.5 (see also Figure 3).

For all the side-channel measurements, if the cross-correlation leads to a regular operation sequence, for example with an RSA (here, "M" represents a multiply operation, and "S" a square one):

— "MMMMMM…"

— "MSMSMS…"

then the test result is Pass.

NOTE 1 Instead of random keys and inputs, the testing laboratory can choose particular keys and inputs:

— Keys: random, `0x80…01`, `0xff…ff`, `0xaa…aa` (in binary: 1010 1010 …).

— Inputs: random, low hamming weight, high hamming weight.

In an asymmetric cipher, the bits of the exponent/scalar highly influence the type of operation or data manipulated. The particular keys can help to distinguish the difference depending on the current bit, for example distinguish square to multiply (double from add for ECC), particular moving data for the square and multiply always (double and add always for ECC), bad use of jump instructions depending on the current bit, etc.

NOTE 2 An IUT using some SPA/SEMA countermeasures such as "atomic double-and-add algorithm" can leak the Hamming weight of the secret key. In some conditions it is sufficient to leak the entire key [57]. The testing laboratory can then optionally follow the methodology proposed in [57].

10.4.3 Markov SPA/SEMA

If the cross-correlation leads to non-regular operation sequences, and if the link between this latter and the key is not clear at first sight, the testing laboratory can apply specific attacks to finally retrieve the key. The testing laboratory can for example face a flawed randomized modular exponentiation (resp. scalar multiplication), and then it can use known weaknesses to finally retrieve the key by considering the exponentiation (or scalar multiplication) algorithm as a Markov process [51]. The attack proposed in [51] on ECC works in four steps:

— Precomputation phase: find the Markov model. The testing laboratory shall calculate the conditional probabilities for sequences of bits and sequences of executed operations.

— Data collection phase: the testing laboratory shall deduce the sequence of operations square and multiply (double and add) operations.

— Data analysis phase: the testing laboratory shall split the sequence into a number of sub-sequences.

— Key testing phase: the testing laboratory shall check all possible keys by the known ciphertext.

The testing laboratory shall test the IUT against Markov SPA/SEMA (especially when the relationships between the bits of the key and the operations is difficult to find) following the framework described in [51], or any other relevant method. If it retrieves the key, the test result is Fail.

10.5 DPA/DEMA

10.5.1 Introduction

With respect to Figure 14, the testing laboratory should assess the security of asymmetric-key cryptosystems against standard DPA, and Address-bit DPA.

10.5.2 Standard DPA/DEMA

In the general case, the DPA/DEMA resistance test for asymmetric-key cryptosystems is similar to the Symmetric-Key ones except that there are much more leakage models. For RSA, the leakage models are listed in the following table.

Table 2 — Leakage models for RSA

Representation of integers	Exponentiation algorithm	Size of words	RSA mode
Classical	Left to right square and multiply always	8	No CRT
Montgomery representation	Right to left square and multiply always	16	CRT
-	Montgomery ladder	32	-
-	Sliding windows of size window size 2	64	-
-	Sliding windows of size window size 3	-	-
-	Sliding windows of size window size 4	-	-
-	Joye's square and multiply method	-	-

For standard single-bit DPA testing and CPA testing, the tests depend on a particular leakage model. Nevertheless, each leakage model consists of the combination of the different columns. There are in total 2×7×4×2 = 112 leakage models. For ECC, various aspects should be considered: different addition/doubling formulae, different representation of points, etc. There are many more possible

leakage models. Therefore, applying standard DPA/DEMA resistance test is not compatible with "push-button" approach.

As for the symmetric cryptography case, the t-test approach is consequently less useful for building an attack against an IUT: it can resolve leaks without having to know or guess the nature of the leak in advance.

This is why the approach presented for symmetric cryptography is extended to cover public-key algorithms:

— The testing laboratory shall use a pre-specified set of test-vectors to expose and isolate potential leakages,

— The testing laboratory shall then apply Welch's t-test to multiple, pre-specified data sets in order to detect sensitive information leakage in the side-channel.

The test vectors range from those targeting very specific weaknesses that could occur for certain implementation choices, to very general tests that allow picking a very broad set of leakages. Common DPA/DEMA leaks may appear in Asymmetric-Key algorithms in:

— the main operation (modular exponentiation or scalar multiplication), i.e. during squares, multiplies or doublings and additions,

— initialization operations (initial reduction, Montgomery conversion, calculation of parameters like p^{-1} mod q for RSA-CRT or R = 2^{k} mod p for Montgomery multiplication) and intermediate operations (intermediate reductions, recombination steps of CRT, (non-modular) multiplication by p or q in RSA-CRT),

— the arithmetic unit itself.

The test vectors are divided into four categories (see Table 3).

Table 3 — RSA Test Vectors and Rationale

Set Number	Test vector properties	Rationale for this test
Set 1	Constant Key, Constant Ciphertext	(Baseline. Key and Ciphertext are fairly random.)
Set 2	Varying Keys, Constant Ciphertext	The goal of this test is to highlight any systematic relationship between power consumption and secret key data. It may reveal variations due to the different key values being processed.
Set 3	Constant Key, Ciphertexts from a set of "special values" (described in 10.5.2)	The goal of this test is to expose the IUT to any special values that are known to trigger exceptional power consumption in certain implementations.
Set 4	Constant Key, Varying Ciphertexts	The goal of this test is to highlight any systematic relationship between side-channel measurements and ciphertext data. It may reveal systematic side-channel measurements due to the fact that the ciphertext values are changing.

Traces from Sets 2 and 3 shall be compared to traces in Set 1 using the Welch's test. Depending on the context of IUT usage, traces from Set 4 shall be also compared to traces in Set 1. Each t-test is repeated twice with independent test vector subsets (subset A and subset B). The IUT fails if there is a point in time where the t-value for both subset A and subset B exceeds +4.5 or -4.5 for at least one test.

A high-amplitude spike in the t-test reveals a significant statistical relationship between the power consumption (the state of the IUT) and the values being processed, i.e. the t-test has revealed a leak.

Set 4 includes vectors chosen to generate exceptional intermediates for a variety of modular exponentiation or scalar multiplication algorithms and implementation choices. Table 4 describes

some examples of various categories (or “special values”) of test vectors that can be used by the testing laboratory when the IUT uses Montgomery multiplication for modular multiplication. Some tests mainly concern RSA-CRT with 1024-bits key (when we consider ciphertexts modulo p or q) but some of them can be also used for non-CRT-RSA and ECC.

Table 4 — Description of Categories for Set 4

Category number	Ciphertext value
00	0
01	1
02	2
03	3
04	n-1 (ciphertext congruent to -1 modulo N, modulo p, and modulo q)
05	n-2 (ciphertext congruent to -2 modulo N, modulo p, and modulo q)
06	n-3 (ciphertext congruent to -3 modulo N, modulo p, and modulo q)
07	2^{-512} mod p (number whose Montgomery representation is 1 mod p)
08	2^{-256} mod p (number whose square has the Montgomery representation 1 mod p)
09	2^{-1024} mod n (number whose Montgomery representation is 1 modulo N)
10	Number whose Montgomery representation is 2 mod p, q and N
11	Number whose Montgomery representation is 3 mod p, q and N
12	Number whose square in Montgomery representation is 2^{511} -1 mod *p*, *q*, and N
13	Number whose square’s Montgomery representation is 2^{511}-1 mod p, q, and N
14	Number whose cube in Montgomery representation is much smaller than p and q
15	Number whose Montgomery representation modulo N is near 2^{1023}-1
16	Number whose square’s Montgomery representation modulo N is near 2^{1023}-1
17	Number whose cube’s Montgomery representation modulo N small

Set 4 can be expanded to include other special values more adapted to the IUT. For example, this latter can use the Barrett reduction. In this case, the testing laboratory can add to the test vectors details in Table 4 some other special values which makes for example some intermediate values equal to 0 or having high or low Hamming weight. Moreover, the ciphertexts can be chosen to provoke particular properties on the result of cryptographic operation (e.g. producing decryption outputs that are very small).

The testing laboratory shall use at least the list of ciphertexts detailed in Table 4, and some optional ones which lead to potentially high-amplitude leaks in the IUT.

RSA PKCS#1 v2.1, DSA and ECDSA are not vulnerable to the standard DPA/DEMA since:

— The used padding for RSA PKCS#1 v2.1 is probabilistic and the attacker cannot predict on the intermediate values;

— DSA and ECDSA use an ephemeral exponent and scalar respectively, so the attacker cannot target a specific bit.

10.5.3 Address-Bit DPA/DEMA

The address-Bit DPA/DEMA [14] exploits the fact that internal addresses of registers or memory locations are another type of data processed by the CPU. Hence, side-channel measurements of two intervals of an algorithm where the same instruction accesses different addresses will be less correlated than if that instruction was accessing the same address. The attack is easier if the number of registers or memory locations accessed during the algorithm depending on bits of the secret key is small.

The attacker computes the average side-channel measurements for two known keys: `0xfff…f`, and `0x80…01`. If the difference of the two averages (c_0 and c_1) shows spikes, the key bits leak and then the test result is Fail.

NOTE DSA and ECDSA are not vulnerable to the address-Bit DPA/DEMA since they use an ephemeral exponent and scalar respectively.

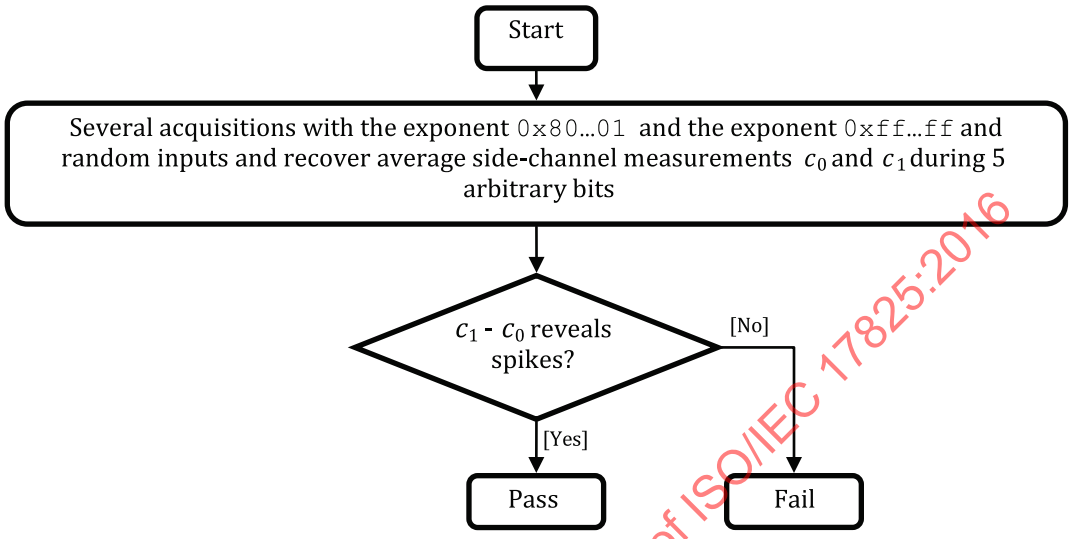

Figure 15 — Address-bit DPA / DEMA

DSA and ECDSA can be vulnerable to a specific DPA/DEMA when the private key is multiplied with a known number.

When the private key d is used for a multiplication by r, in this case, the attacker can perform a DPA/DEMA during the computation of $d.r$.

A multiplication is generally performed word by word. The leakage model can be the Hamming Weight of the first word of d (d [0]) multiplied by the first word of r (r[0]), i.e. HW(d[0].r[0]).

As for symmetric encryption, perform a t-test with the good hypothesis and a random wrong hypothesis.

The tests may be performed with different architecture sizes: 8, 16, 32 and 64.

There is no need to perform this test with all words of d: a deduction can be made that the other words are recovered the same way in case of a success and the other words are protected as well in case of a failure.

11 Non-invasive attack mitigation pass/fail test metrics

11.1 Introduction

This clause specifies the test metrics and pass/fail criteria for each associated combination of attack method and security function shown in Table 1.

In the following sub-clauses, test metrics are provided in data collection time, analysis time, and amount of data. When statistical significance test applies, the risk level of 5% is used to set the threshold.

A performance reference for computational analysis will be given as the specification of the reference computer by the validation authority to provide fair criteria in analysis time.

11.2 Security Level 3

11.2.1 Time Limit

The maximum acquisition time shall be no more than 6 h for each elementary test for Security Level 3. When this limit is reached, measurement will be terminated even if the number of measurements has not met the specified maximum. The total sequence of acquisition time shall be no more than 72 h.

11.2.2 SPA and SEMA

To complete the SPA or SEMA test at Security Level 3, the provided test suite should collect:

— 11 waveforms using the input data patterns each comprising a CSP and plaintext provided by the test suite (1 pre-determined input data pattern is used for the same-data-pair input; 1 pre-determined pair and 4 random-data pairs are used for the different-data-pair inputs);

— per the time period corresponding to one CSP bit, the resolution of each waveform is 100 points or greater.

The similarity of the resultant traces for a core test will be inspected both visually and with a statistical test. To satisfy the core test, both test results shall be passed. The test will fail, otherwise.

11.2.3 DPA and DEMA

To complete the DPA or DEMA test at Security Level 3, the provided test suite should collect:

— 10 000 waveforms of one type of side-channel leakage from different input data patterns for each CSP in a set of provided CSPs.

If the calculated leakage is determined significant against the significance level defined in the document, the test will fail. The test will pass, otherwise.

Every DPA variant attack specified in the document will be applied in sequence.

11.2.4 Timing Analysis

To complete the Timing Analysis test at Security Level 3, the provided test suite shall collect:

— 1 000 timing measurements for random CSPs and a pre-determined plaintext for the first measurement block, and

— 1 000 timing measurements for a pre-determined CSP and random plaintext for the second measurement block.

11.2.5 Pre-processing conditions in differential analysis

To complete the differential power or emanation analysis at Security Level 3, the following factors are applicable:

— A synchronisation signal is available signalling the beginning of the cryptographic operation.

— A noise reduction is to be performed by the tester calculating the mean of different traces (10 cryptographic executions should be performed for the same inputs set, to get one single trace).

11.2.6 Pass / Fail condition

a) If the traces obtained using the trigger are misaligned from the different executions the test passes and the statistical test for the cryptographic algorithm is not to be performed.

b) If the traces obtained using the trigger are aligned, then the mean is to be computed and the verdict will be the one obtained from the statistical test applied for the cryptographic algorithm and the traces.

11.3 Security Level 4

11.3.1 Time Limit

The maximum acquisition time shall be no more than 24 h for each elementary test for Security Level 4. When this limit is reached, measurement will be terminated even if the number of measurements has not met the specified maximum. The total sequence of acquisition time shall be no more than 288 h.

11.3.2 SPA and SEMA

To complete the SPA or SEMA test at Security Level 4, the provided test suite should collect:

— 21 waveforms using the input data patterns each comprising a CSP and plaintext provided by the test suite (1 pre-determined input data pattern is used for the same-data-pair input; 5 pre-determined pair and 15 random-data pairs are used for the different-data-pair inputs).

— Per the time period corresponding to one CSP bit, the resolution of each waveform is 1 000 points or greater.

The similarity of the resultant traces for a core test will be inspected both visually and with a statistical test. To satisfy the core test, both test results shall be passed. The test will fail, otherwise.

11.3.3 DPA and DEMA

To complete the DPA or DEMA test at Security Level 4, the provided test suite should collect:

— 100 000 waveforms of one type of side channel leakage from different input data patterns for each CSP in a set of provided CSPs.

If the calculated leakage is determined significant against the significance level defined in the document, the test will fail. The test will pass, otherwise.

Every DPA variant attack specified in the document will be applied in sequence.

11.3.4 Timing Analysis

To complete the Timing Analysis test at Security Level 4, the provided test suite should collect:

— 10 000 timing measurements for random CSPs and a pre-determined plaintext for the first measurement block, and

— 10 000 timing measurements for a pre-determined CSP and random plaintext for the second measurement block.

11.3.5 Pre-processing conditions in differential analysis

To complete the differential power or emanation analysis at Security Level 4 in addition to the factors specified for the Security Level 3, the following are applicable:

— A noise reduction is to be performed by the tester applying a spectrum analysis and filtering the traces in frequency (band-pass filter according to the module operation frequency).

— A static and dynamic alignment will be performed to bypass any immediate countermeasure or misalignments coming from errors in the configuration measurement when starting the power consumption (or emanations) acquisition.

11.3.6 Pass / Fail condition

Apply the filter in frequency.

a) If the traces present the inclusion of random timing delays or clock frequency variations so that they cannot be fully aligned with a static alignment, the test passes and the statistical test for the cryptographic algorithm is not to be performed.

b) If the static alignment succeeds, the verdict will be the one obtained from the statistical test applied for the cryptographic algorithm and the traces filtered and aligned.

Annex A
(normative)

Requirements for measurement apparatus

A.1 General

The statements in this Annex are required for the measurement apparatus [20].

A.2 Speed

a) Bandwidth of at least 50% of the device clock rate for software implementations and at least 80% of the clock rate for hardware implementations.

b) Capability to capture samples at 5× the bandwidth.

A.3 Resolution

a) A minimum of 8-bits of sampling resolution.

A.4 Capacity

a) Enough storage to capture the entire signal required for the test and analysis.

A.5 Probe

A probe is needed to measure the IUT currents.

If the used side-channel is the power consumption of the IUT, a resistor should be placed between the VCC line supplying the IUT and the IUT. The testing laboratory should choose the highest value resistor that allows the IUT to function.

If the used side-channel is electromagnetic emanations of the IUT, a near-field magnetic probe shall be used, provided the bandwidth of the probe is at least that of the IUT clock rate.