FINAL
DRAFT

INTERNATIONAL
STANDARD

**ISO/IEC
FDIS
20897-1**

ISO/IEC JTC **1**/SC **27**

Secretariat: DIN

Voting begins on:
2020-09-18

Voting terminates on:
2020-11-13

Information security, cybersecurity and privacy protection — Physically unclonable functions —

Part 1:
Security requirements

Reference number
ISO/IEC FDIS 20897-1:2020(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

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.

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) or the IEC list of patent declarations received (see http://patents.iec.ch).

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

A list of all parts in the ISO/IEC 20897 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document specifies the security requirements for physically unclonable functions (PUFs) for generating non-stored cryptographic parameters.

Cryptographic modules generate the certain class of critical security parameters such as a secret key using a random bit generator within the modules. Such modules can store generated security parameters in embedded non-volatile memory elements. For higher security, a combination of tamper response and zeroization techniques may be used for protecting stored security parameters from active unauthorized attempts of accessing such parameters. However, as the reverse-engineering technology advances, the risk of theft of such stored security parameters has become higher than ever.

The rapidly pervading technology called a PUF is promising to mitigate the above-mentioned risks by enabling security parameter management without storing such parameters. PUFs are hardware-based functions providing steadiness and randomness of their outputs and physical and mathematical unclonability of the functions themselves, taking advantage of intrinsic subtle variations in the device's physical properties, which are also considered object's fingerprints. PUFs can be used for security parameter generation (e.g. key, initialization vector, nonce and seed), entity authentication or device identification in cryptographic modules.

Now, security requirements of PUFs should be considered at system level, meaning that they should consider many possible attack paths, as detailed further in this document.

The purpose of this document is to define the security requirements of batches of PUFs and of single instances of PUF for assuring an adequate level of quality of the provided PUFs in cryptographic modules. This document is meant to be used for the following purposes.

a) In the procurement process of a PUF-equipped product, the procurement body specifies the security requirements of the PUF in accordance with this document. The product vendor evaluates the PUF whether the PUF satisfies all the specified security requirements, and reports the evaluation results to the procurement body.

b) The vendors evaluate the security of their PUF, publicize the evaluation results and clarify the security of their PUF.

It should be noted that all of the security requirements defined in this document are not necessarily quantitatively evaluable.

This document is related to ISO/IEC 19790 which specifies security requirements for cryptographic modules. In those modules, CSPs (e.g. key) and PSPs (e.g. ID) are the assets to protect. PUF is one solution to avoid storing security parameters, thereby increasing the overall security of a cryptographic module.

Information security, cybersecurity and privacy protection — Physically unclonable functions —

Part 1: Security requirements

1 Scope

This document specifies the security requirements for physically unclonable functions (PUFs). Specified security requirements concern the output properties, tamper-resistance and unclonability of a single and a batch of PUFs. Since it depends on the application which security requirements a PUF needs to meet, this documents also describes the typical use cases of a PUF.

Amongst PUF use cases, random number generation is out of scope in this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 18031, *Information technology — IT Security techniques — Random bit generation*

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

3 Terms and definitions

For the purposes of this document, terms and definitions given in ISO/IEC 18031, ISO/IEC 19790 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
challenge
variable parameters input to a PUF

Note 1 to entry: Some type of PUFs do not take an input challenge, and such PUFs are called a no-challenge PUF. A no-challenge PUF can be seen as a special type of PUF where a challenge length is 0 bit (see 3.10).

3.2
confined PUF
DEPRECATED: weak PUF
PUF that has a limited space of challenge-response pairs

Note 1 to entry: The term “weak PUF” does not properly express the characteristics of the PUF; nonetheless, it is the way this category of PUFs is referred to in the scientific literature.

3.3
entropy source
component, device or event which produces outputs which, when captured and processed in some way, produce a bit string containing entropy

[SOURCE: ISO/IEC 18031:2011, 3.12]

3.4
extensive PUF
DEPRECATED: strong PUF
PUF that has so large space of challenge-response pairs that not all addresses cannot be read out within the attack time scales and its entire function cannot be modelled in extenso from the knowledge of few challenge/response pairs on a different device (e.g. a general purpose processor)

3.5
false acceptance rate
FAR
probability that the inter-distance is smaller than or equal to the set threshold

Note 1 to entry: FAR is equivalent to the evaluation of the cumulative distribution function of the inter-distance at the set threshold.

3.6
false rejection rate
FRR
probability that the intra-distance is larger than the set threshold

Note 1 to entry: FRR is equivalent to the complement of the evaluation of the cumulative distribution function of the intra-distance at the set threshold.

3.7
fuzzy extractor
scheme to recover the original data from noisy data using error-correcting technique

Note 1 to entry: Fuzzy extractor uses a pair of algorithms; the first one generates a helper string from original data, and the second one recovers the original data from noisy data using error correcting techniques and the helper string.

3.8
intra-Hamming distance
intra-HD
Hamming distance between the two responses obtained by giving the same challenge twice to the same PUF or obtained from the same no-challenge PUF

3.9
inter-Hamming distance
inter-HD
Hamming distance between the two responses obtained by giving the identical challenges to two different PUFs or obtained from two different no-challenge PUFs

3.10
no-challenge PUF
one form of a confined PUF that does not receive a challenge as an input

3.11
physically unclonable function
physical unclonable function
PUF
function implemented in a device that is produced or configured with the security objective that random fluctuations in the production process lead to different behaviours and are hard to reproduce physically

3.12
response
output value from a PUF

3.13
static entropy source
source of entropy which is prepared in advance of its use

4 Abbreviated terms

CRP	challenge response pair
CSP	critical security parameter
DFA	differential fault analysis
DRBG	deterministic random bit generator
ECC	error correction code
FIB	focused ion beam
HCI	hot carrier injection
HMAC	hash-based message authentication code
LVP	laser voltage probing
MOS	metal oxide semiconductor
NBTI	negative bias temperature instability
PSP	public security parameter
PUF	physically unclonable function

5 Security requirements for PUFs

5.1 General

A PUF is a physical object that provides a unique digital output as a (deterministic) function of the given inputs. The mapping between the inputs and outputs of a PUF is determined by the variation of device components (e.g. transistor, wire, capacitance) arisen during manufacturing process, etc. Since the device variation is random and uncontrollable, it is virtually impossible to manufacture two PUFs that behave in exactly the same way. Actually, the idea behind a PUF is that given the identical "blueprint" or fabrication file, every instance behaves differently. Therefore, the PUF may be used for security parameter generation, device identification and device authentication.

A PUF's output is expected to be random for different challenges and for inter-modules. Another random sequence or key to be generated from a PUF should not be estimated from any other output of the PUF. In order to keep using the same generated key, the same challenge only has to be kept. Even if a challenge is leaked out, the corresponding response should not be estimated without having the same device activated. However, for those purposes, particular caution should be taken due to the PUF's properties. The raw output of a PUF contains a small amount of errors due to subtle fluctuations in the physical properties exploited. Therefore, typically an error correction scheme is combined with the output of a PUF in order to make the output the same every time the same challenge is given. In order to avoid undesired regeneration of the same key or nonce, it is required to maintain the challenge value carefully.

A PUF and DRBG are different in the following two aspects:

— an output of a DRBG is completely computable by a deterministic algorithm given a seed, whereas a response of a PUF is virtually impossible to compute from a challenge;

— different instances of a DRBG will produce completely the same outputs given the same seed, whereas different PUFs will output different responses.

Therefore, the application of a PUF for random number generation is out of scope in this document.

A PUF is seen as an instance of a design by a factory (which can be modelled as a random variable). Ideally, the properties of each PUF instance are independent and identically distributed (i.i.d.), as represented in Figure 1. Notice that, in practice, the PUFs are neither identical nor independent: it is the purpose of some security requirements to quantify how well PUF instances differ one from each other.

For the PUF to indeed provide a securely strong and reliable response, some properties shall be met. This clause concerns the definition of the security requirements which apply irrespective of PUF implementation, whereas there exist many kinds of PUF implementations (refer to Annex A).

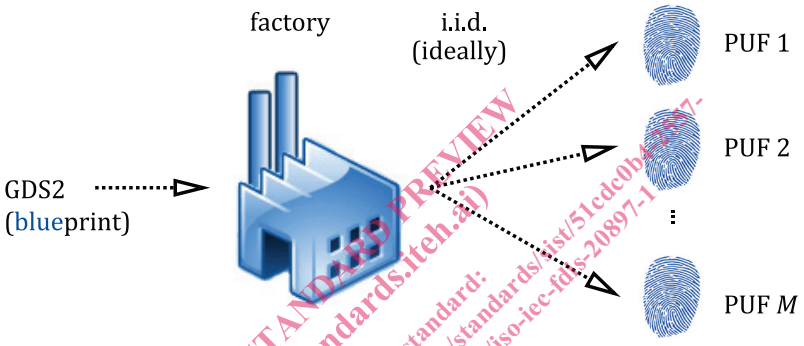

Figure 1 — (Ideal) modelization of a PUF as a random instance from a factory

5.2 PUF interface

Generally, a PUF receives an input called a *challenge*, and generates a unique output called a *response*. A challenge is a bit-string made up of *challengeLength* bits, and a response is a bit-string made up of *responseLength* bits. The response may be specific to an input challenge. In case of a no-challenge PUF, the challenge length is considered as 0 bit. Thus, a PUF can be an alternative to a memory of $2^{challengeLength}$ words where each word is *responseLength* bits.

A PUF may have optional ports, which allow for security testing. The signals from these ports are output status, and are meant to indicate whether the PUF responses can be used securely. After the security test is finished, these ports shall be disabled to avoid being exploited by an attacker.

In addition, a PUF may have other optional ports to control the PUF.

5.3 PUF building blocks

A PUF consists of a main block and an optional pre-processing block and post-processing block. The structure of a PUF is illustrated in Figure 2.