INTERNATIONAL STANDARD

# ISO/IEC 29192-7

First edition
2019-07

# Information security — Lightweight cryptography —

## Part 7: Broadcast authentication protocols

*Sécurité de l'information — Cryptographie pour environnements contraints —*

*Partie 7: Protocole d'authentification diffusée*

Reference number
ISO/IEC 29192-7:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
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 29192 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

Many IT environments involve broadcast communication, in which one sender communicates with multiple receivers. Securing such communication is a non-trivial task. Broadcast authentication protocols aim to enable the recipient to verify the authenticity of transmitted data and ensure entity authentication of the sender.

A straightforward way of achieving broadcast authentication is to use digital signatures, as for example described in the ISO/IEC 9796 series or ISO/IEC 14888 series. However, there are situations in which the additional communication and computational overhead of digital signatures are prohibitively expensive, as can be the case with satellites broadcasting to earth.

This document specifies lightweight broadcast authentication protocols, which offer a significantly lower implementation cost than deploying digitial signatures as a solution to the authentication of broadcast communication.

# Information security — Lightweight cryptography —

# Part 7: Broadcast authentication protocols

## 1 Scope

This document specifies broadcast authentication protocols, which are protocols that provide data integrity and entity authentication in a broadcast setting, i.e. a setting with one sender transmitting messages to many receivers. To provide entity authentication, there needs to be a pre-existing infrastructure which links the sender to a cryptographic secret. The establishment of such an infrastructure is beyond the scope of 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 9797 (all parts), *Information technology — Security techniques — Message authentication codes (MACs)*

ISO/IEC 10118 (all parts), *IT Security techniques — Hash-functions*

ISO/IEC 29192-1, *Information technology — Security techniques — Lightweight cryptography — Part 1: General*

ISO/IEC 29192-5, *Information technology — Security techniques — Lightweight cryptography — Part 5: Hash-functions*

ISO/IEC 29192-6[1], *IT Security techniques — Lightweight cryptography — Part 6: Message authentication codes (MACs)*

## 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 29192-1 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**
**broadcast**
method of communication by which information originating from a *sender* (3.11) is distributed to a group of *receivers* (3.9)

[1] Under preparation. (Stage at the time of publication: ISO/IEC DIS 29192-6:2019.)

### 3.2
**data integrity**
property that data has not been altered or destroyed in an unauthorized manner

[SOURCE: ISO/IEC 9797-1:2011, 3.4]

### 3.3
**collision-resistant hash-function**
*hash-function* (3.4) satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment.

[SOURCE: ISO/IEC 10118-1:2016, 3.1, modified — Reference to Annex C has been removed from Note 1 to entry.]

### 3.4
**hash-function**
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

— for a given output, it is computationally infeasible to find an input which maps to this output;

— for a given input, it is computationally infeasible to find a second input which maps to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment.

[SOURCE: ISO/IEC 10118-1:2016, 3.4, modified — Reference to Annex C has been removed from Note 1 to entry.]

### 3.5
**key**
sequence of symbols that controls the operation of a cryptographic transformation (e.g., encipherment, decipherment, cryptographic check function computation, signature generation, or signature verification)

[SOURCE: ISO/IEC 18033-3:2010, 2.4, modified — Between the brackets, the words after "decipherment" have been added.]

### 3.6
**MAC algorithm key**
key that controls the operation of a MAC algorithm

[SOURCE: ISO/IEC 9797-1:2011, 3.8]

### 3.7
**Message Authentication Code**
**MAC**
string of bits which is the output of a MAC algorithm

Note 1 to entry: A MAC is sometimes called a cryptographic check value (see for example ISO 7498-2).

[SOURCE: ISO/IEC 9797-1:2011, 3.9]

### 3.8
**Message Authentication Code algorithm**
**MAC algorithm**
algorithm for computing a function which maps strings of bits and a secret key to fixed-length strings of bits, satisfying the following two properties:

— for any key and any input string, the function can be computed efficiently;

- for any fixed key, and given no prior knowledge of the key, it is computationally infeasible to compute the function value on any new input string, even given knowledge of the set of input strings and corresponding function values, where the value of the $i$th input string may have been chosen after observing the value of the first $i - 1$ function values (for integers $i > 1$)

Note 1 to entry: A MAC algorithm is sometimes called a cryptographic check function (see for example ISO 7498-2).

Note 2 to entry: Computational feasibility depends on the user's specific security requirements and environment.

[SOURCE: ISO/IEC 9797-1:2011, 3.10]

**3.9**
**receiver**
entity receiving communication from the *sender* (3.11) requiring data integrity and *entity authentication* (3.10)

**3.10**
**entity authentication**
corroboration that an entity is the one claimed

[SOURCE: ISO/IEC 9798-1:2010, 3.14]

**3.11**
**sender**
entity generating messages to be *broadcast* (3.1)

**3.12**
**chain of keys**
sequence of keys generated by applying a *collision-resistant hash-function* (3.3) to a *key* (3.5) and then repeatedly applying a collision-resistant hash-function to the output of the previous execution

## 4 Symbols and abbreviated terms

$F(X)$ output of a hash-function $F$ when applied to data $X$, truncated to the length required for keys used by the MAC algorithm

MAC($K$,$M$) output of a MAC algorithm given MAC algorithm key $K$ and message $M$ as input

$X || Y$ concatenation of data items $X$ and $Y$ in the order specified

NOTE 1 For $X$, $K$, $M$, $Y$ strings of arbitrary length.

$D$ length of each time interval

$d$ delay between the transmission of the MAC and the disclosure of the key, measured in units of $D$

$N$ length of a chain of keys

$\alpha$ a value used in the generation of the chain of keys to frustrate time-memory tradeoff attacks

$L$ length of $\alpha$

$i, j$ time intervals

$|i|$ The number of bits used to represent the $i$-th time interval. This representation shall be unique throughout the entire lifetime of the system. Whenever $i$ is used as a bit-string, it shall be converted as described by the I2BS conversion function from WG2 SD 7, 4.3.9 (2$^{nd}$ edition).

$M_i$ message sent in the $i$-th time interval

NOTE 2 $D$, $d$, $N$, $L$, and $|i|$ are called “system parameters” and are decided and made public prior to the initiation of the system.

# 5 TESLA-RD (Timed Efficient Stream Loss-tolerant Authentication — Rapid Disclosure)

## 5.1 General

The TESLA-RD protocol uses a MAC algorithm and a collision-resistant hash-function. The MAC algorithm shall be from ISO/IEC 9797 (all parts) or ISO/IEC 29192-6 and the collision-resistant hash-function shall be from ISO/IEC 10118 (all parts) or ISO/IEC 29192-5.

Annex A defines the object identifiers which shall be used to identify the algorithms specified in this document.

NOTE 1 Choosing non-lightweight MAC algorithms and collision-resistant hash-functions can reduce the benefits of using TESLA-RD in constrained environments.

The sender and all receivers shall have an internal clock. The sender’s time according to its internal clock is denoted by $T\_s$. A receiver’s time according to its internal clock is denoted by $T\_r$. The difference between the sender's time, $T\_s$, and the receiver's time, $T\_r$, shall be upper bounded throughout the lifetime of the system by some fixed and known $\varepsilon$, i.e., $|T\_s - T\_r| < \varepsilon$.

NOTE 2 The inner workings of the internal clocks, as well as the procedure on how time synchronization is obtained are outside the scope of this document.

Time shall be divided into intervals of a fixed length $D$. Each interval is associated with a key (3.5) from the chain of keys (3.12) currently in effect. The first interval is associated with the last key, the second interval is associated with the one-before-last key, etc. At the end of each time interval, a packet can be broadcast. The packet shall be processed with a MAC algorithm using the key associated with this interval, and the key shall be broadcast after a delay of $d$ time intervals.

NOTE 3 The product of the parameters $D$ and $d$ is the delay between receiving the message and authenticating it.

After $N$ keys were broadcast, the chain is exhausted and a new chain shall be used.

## 5.2 Initialization

In the initialization phase, the sender shall choose a MAC algorithm (3.8) and a collision-resistant hash function (3.3) and decide on the following system parameters: $D$, $d$, and $N$. All shall be distributed to the receivers in such a way so as to ensure data integrity and sender entity authentication.

## 5.3 Setup

The sender shall compute a chain of keys by first generating a uniformly random MAC algorithm key, $K_{N,}$ and value, $\alpha$. Then the sender shall use the collision-resistant hash-function to compute MAC algorithm keys $K_i = F(K_{i+1} \,||\, i || \,\alpha)$, for $i = N - 1 , N - 2, \ldots, 0$. The key, $K_0$, shall be distributed to the receivers in such a way so as to ensure data integrity and sender entity authentication. The value $\alpha$ shall be distributed similarly.

The distribution of the key $K_0$ may, for example, be secured by means of digital signatures or by means of another instantiation of the TESL-RDA protocol, with larger $D$ and $d$.

The length of $\alpha$ may be set to zero. However, to thwart possible time-memory tradeoff attacks $\alpha$ should have non-zero length if $K_0$ is released much before the start time of the chain.

## 5.4 Sending a message

During the $i$-th time interval, where $i = 1, \ldots, N$, the sender broadcasts the message $M_i$ processed by the MAC algorithm using the key $K_i$. The sender also broadcasts the time interval index $i$ and the key $K_{i-d}$ associated with the $(i - d)$-th time interval. A packet transmitted during the $i$-th interval is constructed as in Formula (1):

$$P_i = M_i \,\|\, i \,\|\, MAC(K_i, M_i) \,\|\, K_{i-d} \qquad (1)$$

## 5.5 Receiving a message

When the receiver receives a packet, $P_i$, with interval index, $i$, they shall first check whether the message was received in time, $T\_r$, such that $T\_r < i - d$. If so, the packet shall be stored for later processing. Otherwise, it shall be discarded.

## 5.6 Verifying the key

After a delay of $d$ packets, a packet, $P_{i+d}$, shall be released which includes $K_i$. $K_i$ shall then be verified against an already verified key $K_j$ where $j < i$ by computing Formula (2):

$$K'_j = F\left(\ldots\left(F\left(F\left(K_i \,\|\, i-1 \,\|\, \alpha\right) \,\|\, i-2 \,\|\, \alpha\right) \,\|\, \ldots\right) \,\|\, j \,\|\, \alpha\right) \qquad (2)$$

and checking whether $K'_j = K_j$. If the two values agree, $K_i$ shall be accepted as a valid key. Otherwise, it shall be discarded.

Depending on the application, once $K_i$ is accepted as valid, the receiver may opt to replace $K_j$ with $K_i$ to reduce the required computation to verify the next key.

## 5.7 Verifying the message

After the key $K_i$ has been verified, the message, $M_i$ and its MAC($M_i$,$K_i$), shall be fetched from the previously stored packet $P_i$. A putative MAC of $M_i$ shall be computed using the MAC algorithm, the retrieved message, $M_i$, and the verified key, $K_i$, and compared to the previously stored MAC. If they are equal, the message shall be accepted as valid. Otherwise, it shall be rejected.