BS ISO/IEC 20248:2022+A1:2024

BSI Standards Publication

Information technology — Automatic identification and data capture techniques — Digital signature data structure schema

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20248:2022+A1:2024

National foreword

This British Standard is the UK implementation of ISO/IEC 20248:2022+A1:2024. It supersedes BS ISO/IEC 20248:2022, which is withdrawn.

The start and finish of text introduced or altered by amendment is indicated in the text by tags. Tags indicating changes to ISO/IEC text carry the number of the ISO/IEC amendment. For example, text altered by ISO/IEC amendment 1 is indicated by 〖A1〉 〈A1〗.

The UK participation in its preparation was entrusted to Technical Committee IST/6, Data communications.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 27289 5

ICS 35.040.50

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 August 2022.

Amendments/corrigenda issued since publication

Date	Text affected
28 February 2025	Implementation of ISO/IEC amendment 1:2024

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20248:2022+A1:2024

INTERNATIONAL STANDARD

ISO/IEC 20248+A1

Second edition
2022-06

Amendment 1
2024-10

Information technology — Automatic identification and data capture techniques — Digital signature data structure schema

Technologies de l'information — Techniques d'identification automatique et de capture de données — Schéma de structure de données de signature numérique

Reference number
ISO/IEC 20248:2022(E)

Documents
Document Sharing Hub
© ISO/IEC 2022

Full document details: BS ISO/IEC 20248:2022+A1:2024

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20248:2022+A1:2024

Contents

Page

Foreword ..... **v**

Introduction ..... **vii**

1 Scope **1**

2 Normative references **1**

3 Terms and definitions **2**

4 Field and data definitions, abbreviated terms, symbols, and binary data **4**
4.1 Field and data definitions 4
4.2 Abbreviated terms 4
4.3 Symbols 5
4.4 Binary data 5

5 Conformance **5**
5.1 Specification version 5
5.2 Claiming conformance 6
5.3 Test authority 6
5.4 Test specification 6

6 DigSig use architecture **6**
6.1 General 6
6.2 DigSig identification and ownership 7
6.3 DigSig certificate process 8
6.4 DigSig generation process 9
6.5 DigSig verification process 9
6.6 Error codes 10

7 DigSig certificate **10**
7.1 General 10
7.2 ISO/IEC 20248 Object Identifier 10
7.3 DigSig certificate parameter use 10
7.4 DigSig cryptography 11
7.4.1 General 11
7.4.2 Digital signatures 11
7.4.3 Private containers 11
7.5 DigSig Domain Authority identifier (DAID) 11
7.5.1 Binary encoding 11
7.5.2 Referenced DAID 13
7.5.3 [A1] DAID for GS1 legal entity(ies) [A1] 13
7.6 DigSig certificate identifier (CID) 13
7.7 DigSig validity 13
7.8 DigSig certificate management 14
7.9 DigSig revocation 14
7.10 Online verification 15

8 DigSig Data Description (DDD) **15**
8.1 General 15
8.2 DDD derived data structures 16
8.2.1 General 16
8.2.2 DDDdata 16
8.2.3 SigData 17
8.2.4 DDDdataTagged 17
8.2.5 DDDdataDisplay 18
8.3 DigSig format 18
8.3.1 General 18
8.3.2 Snips 18
8.3.3 Envelope format 19

© ISO/IEC 2022 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO/IEC 20248:2022+A1:2024

8.3.4 AIDC specific construction of a DigSig ... 19
8.4 The DigSig physical data path ... 20
8.5 DDD syntax ... 21
8.6 DigSig information fields ... 22
8.7 Data fields ... 23
8.7.1 General ... 23
8.7.2 Compulsory data fields ... 23
8.7.3 Application data fields ... 23
8.8 Data field object syntax ... 24
8.9 DDD field types and associate settings ... 25
8.9.1 General ... 25
8.9.2 Special field values ... 25
8.9.3 Field types ... 26
8.10 DigSig data presentation ... 35
8.10.1 General ... 35
8.10.2 displaystring ... 36
8.10.3 displayformat ... 36
8.10.4 DDDdataDisplay generation ... 39
8.11 Structured document processing ... 40
8.12 Application field specification by codebook ... 41

9 Pragmas (field directives) ... 42
9.1 General ... 42
9.2 entertext ... 42
9.3 structjoin ... 43
9.4 readmethod ... 43
9.5 privatecontainer ... 44
9.6 startonword ... 45

Annex A (normative) **Test methods ... 46**
Annex B (informative) **Example DigSigs ... 49**
Annex C (informative) **DigSig use in IoT ... 57**
Annex D (informative) **Typical DigSig EncoderGenerator device architecture ... 60**
Annex E (informative) **Typical DigSig DecoderVerifier device architecture ... 69**
Annex F (normative) **DigSig error codes ... 75**
Annex G (informative) **Digital Signature use considerations ... 76**
Annex H (informative) **Example of a DigSig certificate ... 77**
Annex I (informative) **Example DDD for a physical certificate ... 79**
Annex J (normative) **DigSig revocation specifications ... 84**
Annex K (informative) **ISO/IEC 15434-based message DigSig examples ... 89**
Annex L (informative) **DigSig URI envelope discussion ... 93**
Annex M (informative) **ISO/IEC 18000-63 and GS1 EPC Gen2 RFID DigSig examples ... 94**
Annex N (informative) **Typical DigSig support infrastructure ... 98**
Annex O (informative) **Example structured document ... 103**
Bibliography ... 105

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

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 documents 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 or www.iec.ch/members_experts/refdocs).

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 https://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

This document was prepared by joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 20248:2018), which has been technically revised.

The main changes are as follows:

— The relationship between the Domain Authority (data owner) and the Domain Authority ID (DAID) is clarified to be one-to-many. The DAID has been extended to cater for the GS1 Company Prefix longer than 10 digits (see 7.5.3), and a method to use the primary data carrier DAID, if present (see 7.5.2).

— The data types and specifications have been updated for easier implementation and completeness, especially to support the practice of using the data type specifications to achieve optimized schema-based data encoding. A codebook method forms part of this update.

— The `date` field type has been found to be limiting. A new human readable `isodate` has been specified to replace `date` (see 8.9.3.7).

— The format of binary data is explicitly defined to be `HexString` or `Base64String` ensuring interoperability and ease of use.

— The `bstring` DDDdata has been limited to `HexString` since Base64 decoding can be done in more than one way which may cause a valid DigSig to be rejected.

— The `digsigenv` type has been changed from `bstring` to `string` with a range of `Base64String`, which is technically the same, but explicit and clear.

— The `cidsniptext` pragma (field directive) has been removed since it is not practical, not used, and redundant. It is also difficult and convoluted to use and implement.

Documents
Document Sharing Hub

© ISO/IEC 2022 – All rights reserved

Full document details: BS ISO/IEC 20248:2022+A1:2024

— ISO/IEC 9899, *Information technology — Programming languages — C* has been removed as a normative reference. Common current coding language methods replaced the C methods.

— Example cryptography methods are provided in B.4.

— Example interfaces to potential code blocks are provided in D.3.3 and E.3.3.

— Revocation has been harmonized with conventional best practices. The CID requirement to be 0 and 1 has been removed (see Annex J).

— An example implementation architecture description has been added as Annex N.

— The structured document function (see 8.11) has been enhanced to support multiple languages. An example structured document is discussed as Annex O.

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 and www.iec.ch/national-committees.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

Introduction

This document specifies a data structure framework and data specification method for domain-authority-specified, schema-based item identification data. A domain authority is typically a brand owner, a data authority, or a data owner.

ISO/IEC 21778 (JSON) is used as the data message format for both the schema and the data, ensuring interoperability with modern Internet systems and services. The data message encapsulates both data syntax and semantics, providing meaning to the data message.

The data source, data schema and data are both offline and online verifiable using ISO/IEC 9594-8 (public key infrastructure (PKI) digital signatures and certificates), with its implementation environment. The data message format allows for the verification of the data message anywhere within the data-stack.

Data capacity and/or data transfer capacity of automated identification data carriers (barcode labels and RFID tags) are limited. This restricts the normal use of a digital signature, as specified in ISO/IEC 9594-8, within automated identification services. This limitation is overcome by the methods specified in this document, which recognizes the three classes of item; data carrier data (any combination of barcodes and RFID tags), generic data which applies to a group of items, and item specific data which may be static for that item, or volatile. Only item specific data are carried by the tag. Generic data are carried by the digital certificate associated with the tag. This method allows additional (comprehensive) data about a group of items to be readable and verifiable.

Adding additional data, especially authenticity data, to tags are often challenging for existing systems resulting in high costs and system/services unavailability. This document provides a method whereby data may be added with limited impact to incumbent systems, facilitating an interoperable add-on rather than a system redesign.

This document specifies an effective and interoperable method to specify, read, decode, and verify data stored in automated identification, independent from real-time remote control. Meta parameters included in a digital certificate are used to achieve:

— offline integrity verification of the data source and data originality,

— a verifiable data structure description to enable interoperability of deployment, domain authority and automated identification data carriers,

— a verifiable data encoding method to achieve compact data to be stored in data constrained automated identification data carriers (the JSON data format is used for both input and output of the encoder and decoder),

— a verifiable automated identification data carrier read method description, allowing for the data of a read event to be distributed over more than one carrier of the same and of different technologies, and

— a verifiable method to support key management of cryptographically-enabled automatic identification data carriers.

A successful verification of the DigSig signifies:

— the data was not tampered with;

— the source of the data is as indicated on the DigSig certificate used to verify the DigSig with;

— if a secured unique identifier of the data carrier is included in the signature of the DigSig stored on data carrier, then the DigSig stored on the data carrier can be considered unique and original.

The choice of cryptography method should be considered carefully. It is advised that only internationally recognized or standardized methods, e.g. FIPS PUB 186-4 and IEEE P1363, be used.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

This document should be used in conjunction with standard risk assessments of the use-case and environment.

NOTE Many applications rely on a secure non-transferable unique data carrier identifier to tag an item uniquely. ISO/IEC 29167 gives more information on such functionality for RFID tags. This specification provides a mechanism to ensure the integrity and authenticity of the data carrier data and an irrefutable link of the data carrier data with the unique data carrier identifier. As such, alterations or insertion of false data into data carriers are detectable. It also provides a means to detect tampered data carrier data stored and communicated within systems. It does not provide any means to defend against replay attacks. As a counter the data carrier reader can use this specification to sign the read data, effectively providing integrity and authenticity to the read-transaction. A third party can then verify that the read-transaction happened at a given place and time, as well as verify the data read from the carrier. Likewise, the signed data carrier data can contain data describing unique features and security marks of the item establishing a verifiable link between the data carrier data and the physical item.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

Information technology — Automatic identification and data capture techniques — Digital signature data structure schema

1 Scope

This document is an ISO/IEC 9594-8 [public key infrastructure (PKI) digital signatures and certificates] application specification for automated identification services. It specifies a method whereby data stored within a barcode and/or RFID tag are structured, encoded and digitally signed. ISO/IEC 9594-8 is used to provide a standard method for key and data description management and distribution. The data capacity and/or data transfer capacity of automated identification data carriers are restricted. This restricts the normal use of a digital signature as specified in ISO/IEC 9594-8 within automated identification services.

The purpose of this document is to provide an open and interoperable method, between automated identification services and data carriers, to read data, verify data originality and data integrity in an offline use case.

This document specifies

— the meta data structure, the DigSig, which contains the digital signature and encoded structured data,

— the public key certificate parameter and extension use, the DigSig certificate, which contains the certified associated public key, the structured data description, the read methods, and private containers,

— the method to specify, read, describe, sign, verify, encode, and decode the structured data, the DigSig Data Description,

— the DigSig EncoderGenerator which generates the relevant asymmetric key pairs, keeps the private key secret, and generates the DigSigs, and

— the DigSig DecoderVerifier which, by using to the DigSig certificate, reads the DigSig from the set of data carriers, verifies the DigSig and extracts the structured data from the DigSig.

This document does not specify

— cryptographic methods, or

— key management methods.

2 Normative references

The following documents are referred to in the text in such a way that some or all 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 8601 (all parts), *Data elements and interchange formats — Information interchange — Representation of dates and times*

ISO/IEC 8824-1[1)], *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1*

1) ITU-T X.680 is equivalent to ISO/IEC 8824-1.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

ISO/IEC 9594-1[2], *Information technology — Open Systems Interconnection — The Directory — Part 1: Overview of concepts, models and services*

ISO/IEC 9594-8[3], *Information technology — Open Systems Interconnection —The Directory — Part 8: Public-key and attribute certificate frameworks*

ISO/IEC/IEEE 9945, *Information technology — Portable Operating System Interface (POSIX®) Base Specifications, Issue 7*

ISO/IEC 19762, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*

IETF RFC 5646[4], *Tags for Identifying Languages*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 and the following apply.

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

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

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

3.1
authenticity
quality or condition of being authentic, trustworthy, or genuine

3.2
base64url
Base64 encoding with the URL and Filename Safe Alphabet

Note 1 to entry: See IETF RFC 4648.

3.3
CIDSnip
continuous binary sequence starting with the DAID and CID encoded field values

Note 1 to entry: See 8.3.2.

3.4
DataSnip
continuous binary or text sequence of encoded field values

Note 1 to entry: See 8.3.2.

3.5
digital certificate
certificate
data construct that contains the public key, integrity parameters and use parameters of the DigSig

Note 1 to entry: The data construct shall be as specified in ISO/IEC 9594-8.

2) ITU X.500 is equivalent to ISO/IEC 9594-1, and is the commonly used reference for standard and terminology.

3) ITU X.509 is equivalent to ISO/IEC 9594-8, and is the commonly used reference for standard and terminology.

4) IEFT RFC 5646 is the reference specification of the IETF BCP 47.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

3.6
digital signature
signature
result of an asymmetric encryption method on a data construct

Note 1 to entry: The asymmetric encryption method and data construct shall be as specified in ISO/IEC 9594-8.

Note 2 to entry: In typical legal terminology, this term is the equivalent of an advanced electronic signature.

3.7
DigSig
data construct assembled according to this document which contains verifiable information obtained from one or more AIDC

3.8
DigSig envelope
envelope
data construct assembled according to this document by the EncoderGenerator

3.9
Domain Authority
entity, operating as a trusted third party, responsible for the digital signature integrity of a jurisdiction

3.10
integrity
reliability of data that are as they were created according to the required verification parameters

3.11
jurisdiction
independent domain of control in terms of the business or legal (or both) scope of the parties concerned

Note 1 to entry: Examples are independent countries, separate ministries or departments of a government, or independent companies each with their own legal or business (or both) framework.

3.12
nibble
four-bit aggregation

3.13
pragma
field directive

Note 1 to entry: See Clause 9.

3.14
Private key
key that is kept in secret and is used to generate a digital signature by encrypting data that will be verified by its associated public key

3.15
Public key
key that is publicly available and is used to verify data that were encrypted by its associated private key

3.16
Snip
continuous binary or text sequence

Note 1 to entry: See 8.3.2.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

3.17
UTF-8
8 bit variable-width encoding as specified by ISO/IEC 10646 of the Unicode Graphic code points

Note 1 to entry: See ISO/IEC 10646.

3.18
WORD
media physical memory grouping of bits

4 Field and data definitions, abbreviated terms, symbols, and binary data

4.1 Field and data definitions

Field and data objects are defined in Clause 8 and 8.10.3.

4.2 Abbreviated terms

AFI	Application Family Identifier
AIDC	Automatic Identification data capture
BRE	Basic Regular Expression
CA	Certification Authority
CID	DigSig certificate ID
DA	Domain Authority
DAID	Domain Authority identifier
DDD	DigSig Data Description
DI	Data Identifier (see ISO/IEC 15434)
DigSig IA	DigSig Issuing Authority
ERE	Extended Regular Expressions
ID	Identification number
IoT	Internet of Things
JSON	Data description construct (see ISO/IEC 21778)
MSB	Most significant bit
OID	Object identifier as specified in ISO/IEC 8824-1
PKI	Public key infrastructure
RFID	Radio-frequency identification
UID	Unique ID
URI	Uniform Resource Identifier
UTC	Coordinated Universal Time

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

X.509	ISO/IEC 9594-8

4.3 Symbols

||	concatenate or join
…	repeat the previous, as required, or fill in, as required
{…}	parameters that form one structure/grouping, a JSON object
[x]	x is optional, or a JSON array
<x>	x is compulsory
x |= y	y is a description of x
x ⇐ y	x takes the value of y
# x	x is a comment until the end of the line
F(x,y)	the function F that takes as input two parameters, x and y, to produce an output
:XXXX:…XX	depicts a hexadecimal string with "0" to "9" and "A" to "F" with 4-character groupings
0xX…	depicts a hexadecimal value with X with "0" to "9" and "A" to "F"
XXX_2	"XXX" is binary with X a "0" or a "1"

4.4 Binary data

Binary data shall be represented by a JSON string with one of the following two formats:

— `HexString`: The character 0 to 9 and A to F shall be used to represent binary values 0000_2 to 1111_2. A `HexString` is a continuum of 4-bit nibbles presented as uppercase hex grouped with 4 characters preceded by a colon (":"). Zero-bit padding shall be used to reach a 4-bit boundary.

EXAMPLE 1 `":0123:4567:89AB:CDEF:0123:4567:89AB:C"` and `":92"`

— `Base64String`: IETF RFC 4648 base64url with padding shall be used except when `Base64String` is used as an input value for bstring where multiples of 6 bits are represented each with a base64url character.

EXAMPLE 2 `"VGhlIHF1aWNrIGJzyQMjezuZcA=="`

Binary data shall be represented in network byte order (big-endian on bytes).

Applications shall auto distinguish between `HexString` and `Base64String` by using the leading colon (":") character in the JSON string.

5 Conformance

5.1 Specification version

The specification version is used by data structures defined in this document. Other systems use the specification version to identify the data structures of this document and to determine the version of the specification.

The specification version shall be set as follows:

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

specificationversionvalue ⇐ "ISO/IEC 20248:yyyy" with "yyyy" the year of the publication supported by the implementation.

5.2 Claiming conformance

To claim conformance, a service shall comply with the requirements of this document.

5.3 Test authority

The tests shall be performed by a software test authority using a norm application or by code inspection. The norm application used in this test shall be independent from the person who requests the test.

5.4 Test specification

The test specification specifies the conformance test methods for this document.

The test methods in Annex A shall be used.

The test specification is independent of

— cryptography conformance and performance; and

— AIDC data carrier conformance and performance.

The following components shall be tested:

— DigSig certificate format.

— DigSig data.

— DigSig DecoderVerifier.

— DigSig EncoderGenerator.

6 DigSig use architecture

6.1 General

This document specifies a DigSig EncoderGenerator and a DigSig DecoderVerifier system component. The DigSig EncoderGenerator is typically an application dedicated implementation and the DigSig DecoderVerifier an application independent implementation.

A DigSig is a structured set of AIDC data. A DigSig may be stored over more than one AIDC device of different types. A DigSig is cryptographically verifiable. The DigSig data structure definition, read methods and cryptographic functions (DigSigs Data Description - DDD) are specified by a Domain Authority (DA) and published in a DigSig certificate (a version 3 X.509 digital certificate). The DigSig certificate is cryptographically verifiable as certified by an X.509 Certification Authority. Each DigSig certificate has a unique identifier (see 7.6) called the certificate Identifier (CID). The {DAID, CID} is unique and contained in every DigSig as the first data of the DigSig. A reader of a DigSig uses the {DAID, CID} of the specific DigSig to reference the relevant DigSig certificate, from which the reader acquires the read methods, data structure specification and cryptographic functions to read the full DigSig, decode the DigSig and verify the DigSig. See Annex B for example DigSigs and Annex I for the DigSig data structures of an example DigSig.

The DigSig EncoderGenerator is used to generate a DigSig, on request from a data carrier programming application. The DigSig EncoderGenerator does not include the method to create or program a data carrier. See Annex D for a typical DigSig EncoderGenerator use architecture.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

The DigSig DecoderVerifier is used by a local application to instruct a data carrier reader/interrogator how to read the DigSig from a set of data carriers and other sources to decode and verify the data. See Annex E typical DigSig DecoderVerifier use architecture.

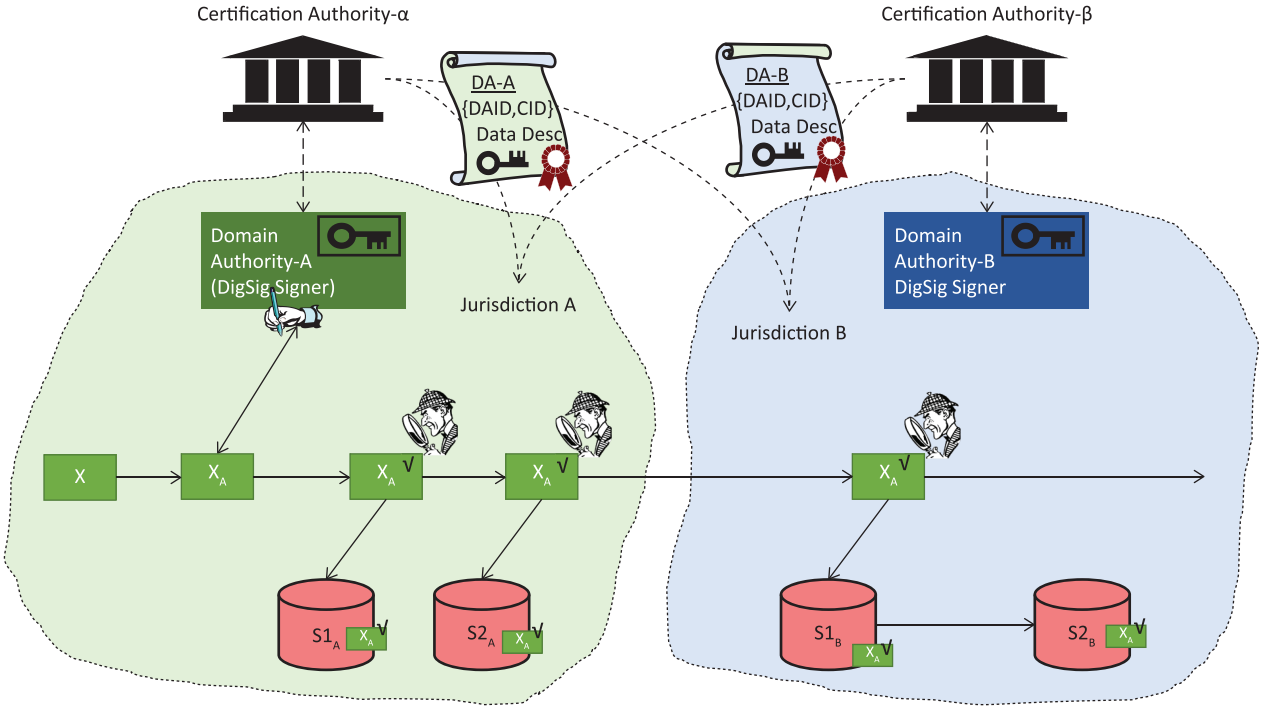

Figure 1 — DigSig use architecture

The general method and properties of this document are illustrated in Figure 1.

a) Domain Authority A (DA-A) provides DigSig issuing services for Jurisdiction A. Similarly, Domain Authority B (DA-B) provides DigSig issuing services for Jurisdiction B. “Issuing” entails the validation of the data for a DigSig, the validation of the DigSig requestor's credentials, and the generation of the DigSig. The DigSig requestor may be a human and/or an application.

b) The DigSig certificates issued by DA-A, each containing a DigSig Data Description (DDD) as applicable to Jurisdiction A applications/services, are certified by the Certification Authority α (CA-α) for a specific signing and certificate validity period in accordance with a Certification Practice Statement as specified in X.509. Similarly, Certification Authority β certifies DigSig certificates issued by DA-B. Certification Authorities α and β may be the same entity.

c) The DigSig certificates are published in a manner to allow systems $S1_A$, $S2_A$, $S1_B$, $S2_B$... to acquire them in advance or on demand. The DigSig certificates are used by the systems to read, decode, and verify DigSigs generated and stored on data carriers by the Domain Authorities, e.g. Data X is used by DA-A to generate a DigSig N X_A as specified by a DigSig certificate N. The systems of both jurisdictions use the DigSig certificates to read, decode, verify, and use the data without the need to connect to any other system.

Annex C provides more information on DigSig use in IoT. AIDC data fulfil an important role in IoT by providing physical objects with a digital identity and optional attributes.

Annex G provides more information on digital signature use in general.

6.2 DigSig identification and ownership

The Domain Authority (DA) shall be the owner of the DigSig certificate which specifies the DigSig data structure schema (the DDD). The DA shall be the issuer of the DigSigs, directly or by proxy, specified by the associated DigSig certificate.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

The Domain Authority shall be identified by its URI called the DA URI (see 7.3). The DA URI shall be verified by the X.509 Certificate Authority when issuing the DigSig certificate.

The DigSig ownership shall be identified by the tuple {Domain Authority URI (`dauri`), Domain Authority identifier (DAID, see 7.5), DigSig certificate identifier (CID, see 7.6)} as certified by the DigSig certificate to be valid at the time of the issuance of the certificate. This validity should be valid when a DigSig is issued.

A Domain Authority may be related to more than one DAID. A DAID shall only be related to one Domain Authority at a given time.

NOTE A DAID is typically assigned to a company or an operating entity of a company. These entities can be traded, terminated, suspended, and resurrected as part of normal business operations.

6.3 DigSig certificate process

Figure 2 describes the process that shall be followed to issue a DigSig certificate in accordance with X.509.

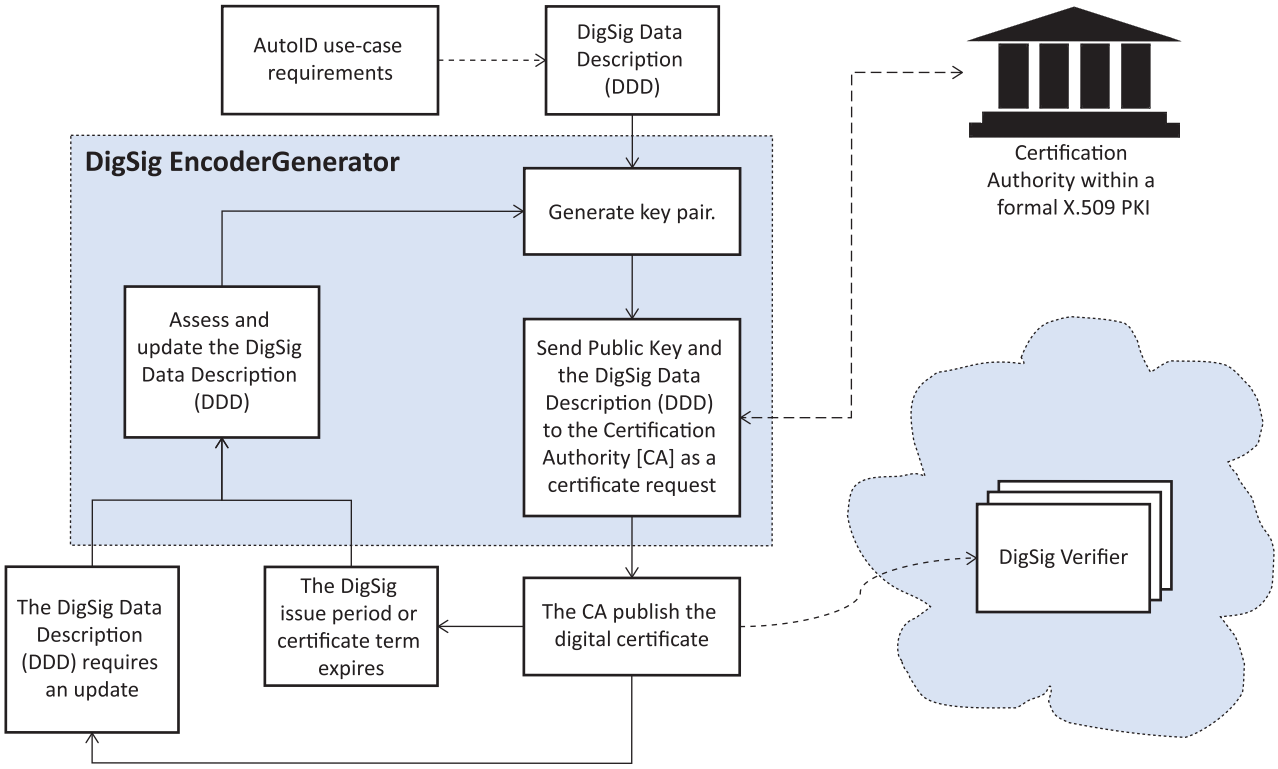

Figure 2 — DigSig certificate process

Figure 3 illustrates how data structure changes can be achieved seamlessly by allowing DigSig certificates — and therefore the data structure validity — to overlap.

© ISO/IEC 2022 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

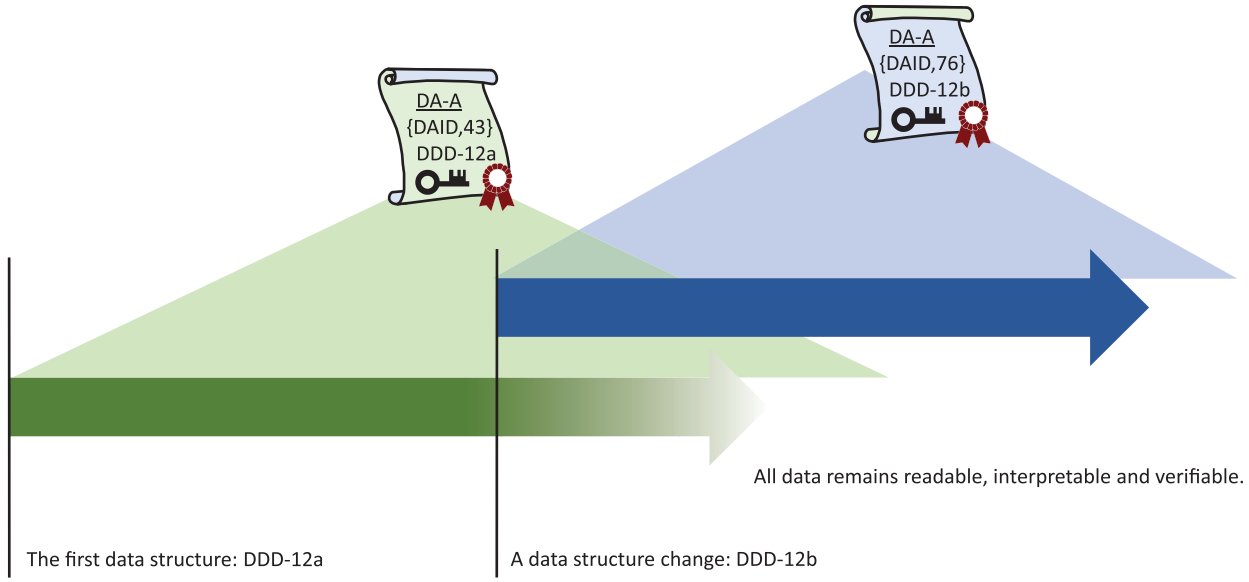

Figure 3 — DigSig data structure rollover

6.4 DigSig generation process

The DigSig is generated by the DigSig EncoderGenerator.

a) The applicable CID referenced DigSig certificate shall be used to generate SigData from DDDdata.

b) SigData shall be digitally signed using the DigSig certificate specified digital signature algorithm resulting in the values Signature and Timestamp.

c) The signature and timestamp DDDdata fields shall be assigned the values Signature and Timestamp.

d) The required DigSig shall be generated from DDDdata.

6.5 DigSig verification process

The DigSig verification shall be performed by a DigSig DecoderVerifier application. The application reads the data carriers as required by its primary function in its use case. It has also the ability to obtain the referenced DigSig certificate.

The steps to verify a DigSig shall be as follows:

a) Read the DigSig envelope.

b) Extract the {DAID, CID} from the DigSig envelope.

c) Check if the {DAID, CID} referenced DigSig certificate is contained in the local certificate store. If not, obtain it from a trusted repository.

d) Verify the {DAID, CID} DigSig certificate, and then use it in the following steps.

e) Read the remainder of the DigSig using the readmethod pragma.

f) Decode the DigSig and prepare DDDdata and SigData.

g) Perform the verification on SigData.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

6.6 Error codes

The DigSig error codes shall be in accordance with Annex F.

7 DigSig certificate

7.1 General

The DigSig certificate consists of a X.509 version 3 certificate with the DDD included in the X.509 version 3 extension.

```
DigSig certificate |= <X.509V3 certificate> || <DDD in a X.509V3 extension>
```

The DigSig certificate establishes a verifiable link between

— the Domain Authority (DA) represented by its URI (DA URI with the field `dauri`),

— the Domain Authority Identifier (DAID with the field `daid`),

— the DigSig Data Description (DDD with the fields `digsiginfo` and `datafields`), and

— the authority verification and revocation services (and array of URIs for the DigSig information fields `verificationuri` and `revocationuri`).

The relationship between a DA and a DAID shall be valid at the time of issuance of the DigSig certificate. When that link is broken the DigSig certificate should be revoked.

NOTE 1 A non-hierarchical certification path can be valid when the same DigSig is used by more than one DA.

NOTE 2 It is possible that the desired CA do not support the specific requirements of this document. In this case, an intermediate CA, a DigSig Issuing Authority (DigSig IA), can be included in the PKI, with a singular function of validating the DigSig extension. The desired CA can incorporate the DigSig IA or be the parent of the DigSig IA. X.509 recommends the inclusion of the DigSig IA function into the respective Certification Practice Statement.

7.2 ISO/IEC 20248 Object Identifier

The ISO/IEC 20248 OID shall take the value: ISO.standard.“the number of this standard”; this results in the OID: “1.0.20248”.

The ISO/IEC 20248 arch, "1.0.20248.1" shall be the container for the ISO/IEC 20248 specific digital signature methods not defined by other standard OIDs (see B.4).

7.3 DigSig certificate parameter use

The X.509 version 3 parameters (in a human readable format with notes) shall be used as follows:

```
Certificate:
  Data:
    Version: 3 (0x2)                        # Shall be version 3
    Serial Number: …                        # set in accordance with X.509
    Signature Algorithm: …                  # set in accordance with X.509
    Issuer: …                               # set in accordance with X.509
    Validity                                # set in accordance with X.509
      Not Before: …
      Not After : …
    Subject: …                              # {DA, DAID, CID} using distinguished names as
                                            #  specified in ISO/IEC 9594-1 and this clause
    Subject public key Info:                # set in accordance with X.509
      Public Key Algorithm: …               # set in accordance with X.509 (see B.4)
      Public-Key: …                         # set in accordance with X.509 (see B.4)
    X509v3 extensions:                      # set in accordance with X.509
      X509v3 Key Usage: critical
```


© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

```
        Digital Signature, Key Encipherment
      X509v3 Basic Constraints: critical
        CA:FALSE
    X509v3 Authority Key Identifier: …      # set in accordance with X.509
    X509v3 CRL Distribution Points: …       # set in accordance with X.509
    1.0.20248: <DDD>                        # set in accordance with this document
  Signature Algorithm: …                    # set in accordance with X.509
```

The common name of the subject shall have the following format:

```
CN = <DA URI>‖/"daid"/‖<DAID>‖"/cid/"‖<CID>
```

EXAMPLE 1 `CN = https://DomainAuthority.com/daid/QC%20TEST/cid/12345`

EXAMPLE 2 See Annex H.

NOTE 1 A DigSig not aware digital certificate authority (CA) can reject the issuing of the certificate if it does not know the DigSig public key algorithm.

NOTE 2 A DigSig aware digital certificate validator processes the extension and accepts or rejects the DigSig certificate depending on the content of the DigSig certificate, the extension, and the conditions under which processing occurs (e.g. the current values of the path-processing variables).

7.4 DigSig cryptography

7.4.1 General

This document specifies the use of two types of cryptography:

— For digital signatures.

— For private containers.

This document does not specify the specific cryptographic algorithms, key, or hash lengths. Only recognized cryptographic methods shall be used.

NOTE The choice of cryptography is linked to the use case risk profile, the size of memory available on the data carriers to be used, the time available to read/interrogate the data carrier reliably, the scope of the application services (i.e. open vs close loop) and the desired validity period of the DigSig.

7.4.2 Digital signatures

The digital signature cryptography methods (i.e. asymmetric encryption, key length, and hash algorithm) shall be specified within the DigSig certificate in compliance with X.509. B.4 provides example digital signature cryptography methods which may be applicable in the resource constraint AIDC environment.

7.4.3 Private containers

The private container cryptography shall be specified by the `privatecontainer` pragma (see 9.5).

7.5 DigSig Domain Authority identifier (DAID)

7.5.1 Binary encoding

The DigSig Domain Authority identifier (DAID) shall be uniquely constructed using the ISO/IEC 15459-2 Issuing Agency Code (IAC) and the Company Identifying Number (CIN) in the following manner:

```
DAID |= <IAC>‖" "‖<CIN>
```

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

An ISO/IEC 20248 Domain Authority shall be registered with a CIN.

a) The DAID is constructed as specified in Table 1.

Table 1 — Domain Authority identifier binary encoding

IAC indicator	CIN indicator	IAC code	CIN code
1 bit	2 bits	5 or 13 bits	24 or 32 bits

b) The rules in Table 2 shall be used to encode the IAC.

Table 2 — ISO/IEC 15459-2 IAC encoding

IAC indicator	IAC range	IAC code	Encoding bits
0_2	0-9	0-9	5
0_2	A-J	10-19	5
1_2	LA-UZ	20-279	13
1_2	KAA-KZZ	280-955	13
1_2	VAA-ZZZ	956-4335	13

c) The rules in Table 3 shall be used to encode the CIN. The CIN, both decimal and upper-case alphanumeric, as specified by ISO/IEC 15459-3, is treated as a number with leading zeros or spaces. Examples are provided in Table 4.

Table 3 — ISO/IEC 15459-2 CIN encoding

CIN indicator	CIN type	CIN length	Encoding bits
00_2	Numeric	1-7 digits	24 bits
01_2	Numeric	8-9 digits	32 bits
10_2	Alphanumeric	1-4 char.	24 bits
11_2	Alphanumeric	5-6 char.	32 bits

1) The numeric CIN shall be treated as a Base10 (decimal) number.

2) The upper-case alphanumeric CIN shall be treated as a Base36 number, also known as a hexatridecimal number, encoded as follows:

$$\text{Decimal CIN} \Leftarrow A_n*36^n \ldots + A_2*36^2 + A_1*36 + A_0 \Leftarrow A_n \ldots A_2 A_1 A_0$$

i) "0" to "9" takes the character values 0 to 9.

ii) "A" to "Z" takes the character values 10 to 35.

iii) Leading zeros and spaces take the character value 0.

NOTE CINs are treated as a number, as such it is padded with leading zeros which is equivalent to leading spaces.

EXAMPLE "123" is equal to "0000123" for 24-bit numerical encoding and "000000123" for 32-bit numerical encoding, and "BC1" is equal to "_BC1" and "0BC1" for 24-bit alphanumerical encoding and "___BC1" and "000BC1" for 32-bit alphanumerical encoding.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 20248:2022+A1:2024

Table 4 — DAID encoding examples

15459-2 Example IAC CIN	IAC indicator	CIN indicator	IAC	CIN	DAID calculated from IAC CIN Example
C ZZ	0_2	10_2	01100_2	:00:05:0F	:4C:00:05:0F
C ZZZZ	0_2	10_2	01100_2	:19:A0:FF	:4C:19:A0:FF
B 999	0_2	00_2	01011_2	:00:03:E7	:0B:00:03:E7
B 999999999	0_2	01_2	01011_2	:3B:9A:C9:FF	:2B:3B:9A:C9:FF
4 123456	0_2	00_2	00100_2	:01:E2:40	:04:01:E2:40
QC ZZZZ	1_2	10_2	0000010011000_2	:19:A0:FF	:C0:98:19:A0:FF
QC ZZZZZZ	1_2	11_2	0000010011000_2	:81:BF:0F:FF	:E0:98:81:BF:0F:FF
UN 999999999	1_2	01_2	0000100001011_2	:3B:9A:C9:FF	:A1:0B:3B:9A:C9:FF

7.5.2 Referenced DAID

Where the DigSig is a data element of an AIDC data carrier message, the DAID 0xFF may be used to reference the DAID of the data carrier message.

The CIN of the primary data carrier data shall comply with 7.5.1.

NOTE Annex K a) and Figure M.2 illustrates the inclusion of a DigSig in a data carrier message (primary data carrier data). Often the primary data carrier data, as illustrated, includes a DAID (IAC and CIN). It is common for the issuer of the primary data carrier data and the DigSig signer to be the same entity (Domain Authority) as indicated by the DAID (IAC and CIN). As a result, the overall encoded data message is shortened, which is beneficial for memory and/or communication constraint data carriers.

7.5.3 〈A1〉 DAID for GS1 legal entity(ies)

GS1 General Specifications Standard shall be used to specify the GS1 Party Global Location Number (PGLN), corresponding to GS1 Application Identifier (417), as the GS1 identifier for legal entity(ies).

The DAID shall be "GS1 <PGLN>". The PGLN is a 13-character base 10 (digits 0 to 9) number.

The DAID for GS1 shall be encoded as follows:

— DAID encoding type identifier: 0xFE.

— PGLN encoding: 44-bit binary number.

EXAMPLE The PGLN 9506000151540 is the DAID "GS1 9506000151540" encoded as ":FE:8A549C323F4". 〈A1〉

7.6 DigSig certificate identifier (CID)

The CID is a 16-bit unsigned integer reference number to a DigSig certificate. The Domain Authority manages the CID and its associated DigSig certificate.

The tuple {DAID, CID} shall be unique within the DigSig certificate validity period.

A CID may be reused. Substantial time between the uses is recommended.

7.7 DigSig validity

The UTC time zone shall be used for all validity periods.

DigSigs are issued within an issuing period as specified by the range of the DigSig timestamp (see 8.5). DigSigs are typically required to be valid for a much longer period than the issuing period. A DigSig

© ISO/IEC 2022 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

expires when its associated certificate expires (see 7.3). The DigSig is therefore valid from the date of signing to the expiry of its associated DigSig certificate as illustrated in Figure 4. The DigSig remains verifiable, though the associated DigSig certificate has expired.

NOTE 1 A generally accepted best-practice is to limit the issuing period to one year to ensure that a new key pair is generated annually. This reduces the exposure risk of a key compromise. However, a DigSig is typically valid for a much longer period, e.g. a DigSig for a work of art may need to be valid for 40 years. In contrast, a DigSig for medicine may need to be valid for only 2 years. A much stronger crypto algorithm, resulting in a much larger signature, is required for the art than the medicine. "Re-keying" and DigSig certificate "re-issuing" ensure efficient and effective authentication of items, which is illustrated in Figure 4.

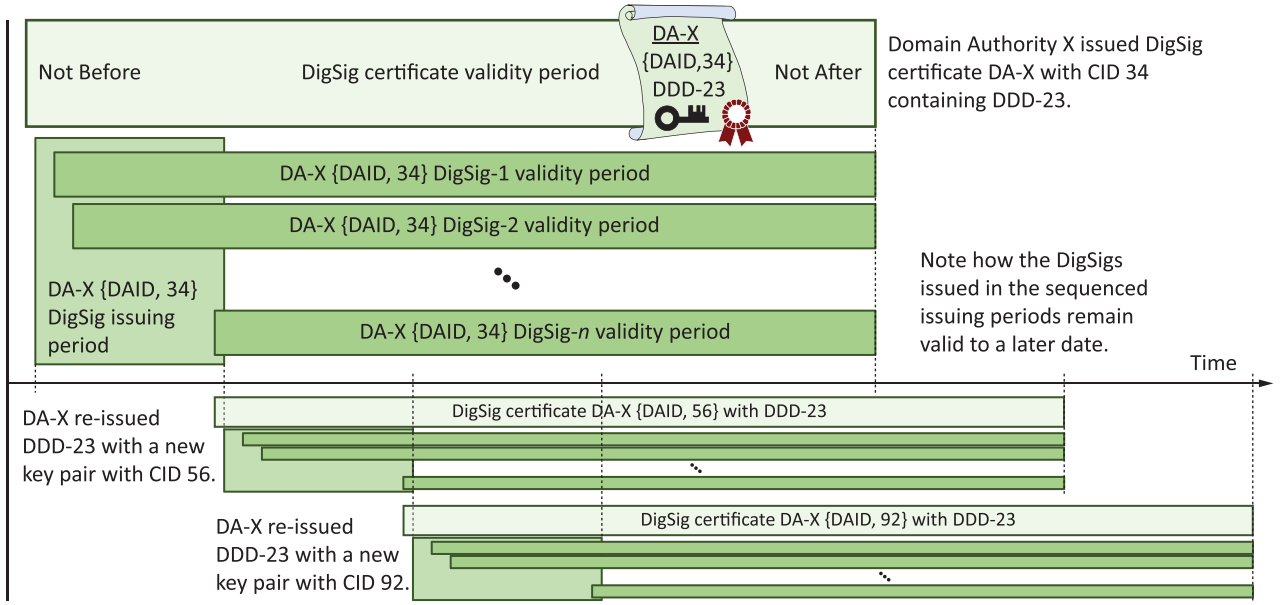

Figure 4 — DigSig validity period

NOTE 2 X.509 allows for the periodic re-issuing of a certificate to extend a certificate validity period.

EXAMPLE A re-issue period is discussed in the following use-case:

A university selects a one-year issuing period; they generate a new key pair annually and publish a new DigSig certificate with a new CID annually.

The university determines that the risk profile indicates that the DigSig certificate should be valid for 10 years. They therefore issue the DigSig certificate with a validity period of 10+1 years (to include the issuing period).

University certificates need to be verifiable for many years. The university therefore re-issues the DigSig certificate, with the same CID, DDD and key pairs, every decade. Both the DigSig and the current DigSig certificate remain therefore verifiable.

NOTE 3 Encryption strength is measured in the brute-force attack duration. The DigSig lifespan is limited by this duration. Re-issuing of the DigSig with new key pairs extends the lifespan of a DigSig.

7.8 DigSig certificate management

DigSig certificate management and DigSig certificate revocation shall comply with ISO/IEC 9594-1.

NOTE The revocation of a DigSig certificate revokes all DigSigs issued with it. The DigSigs need not be individually revoked.

7.9 DigSig revocation

The revocation of DigSigs is a recommended service typically performed by a trusted service (see Annex E).

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

The Domain Authority of a DigSig shall be the authority of the revocation of the DigSig.

The DDD may list additional authoritative revocation URIs (see 8.5).

The DigSig revocation shall comply with Annex J.

NOTE A chain of trusted digital signatures can be compromised in several possible ways (e.g. exposure of a private key, innovations in the mathematics of hashing algorithms, or availability of greater computing power). Any implementation of this document which does not implement revocation services for digital signatures assumes some risk of wrongly trusting a digital signature.

7.10 Online verification

Online verification, using commonly available browsers, may be performed by submitting the URI envelope (see 8.3.3.3) and a RAW envelope (8.3.3.2) prefixed with a verification URI to a trusted online service (see Annex E).

The DDD may list authoritative verification URIs (see 8.5).

The trusted online service shall perform the `entertext` pragma (see 9.2) through its user interface.

The trusted online service should use standard mobile app services to perform the other readmethod pragmas.

NOTE A primary goal of this document is to provide an offline method, whereby a reader/interrogator reads, decodes, and verifies data carrier data in an open and interoperable manner. The data read, and passed on to another application, as in IoT, remain verifiable.

8 DigSig Data Description (DDD)

8.1 General

The DDD is a hierarchical schema-based specification of the data fields contained or directly associated with an item, the DDD data. The DDD data are structured, signed, verified, encoded, decoded, stored, and read from data carriers as specified by the associated DDD. The encoded form of the DDD data is called a DigSig.

AIDC data carriers' data storage and data transfer are limited. This limitation should be considered when developing the DDD.

The DDD is a JSON construct. It has the following format:

`{<digsiginfo>,<datafields>}`

`digsiginfo` is a JSON object containing management definition fields.

`datafields` is a JSON array of objects each specifying a compulsory and application field.

The DDD is a tree structure. The leaves are the DDD data containers and the branches are the structuring of the containers.

Figure 5 illustrates all the components of DigSig envelope as specified by a DigSig Data Description (DDD).

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

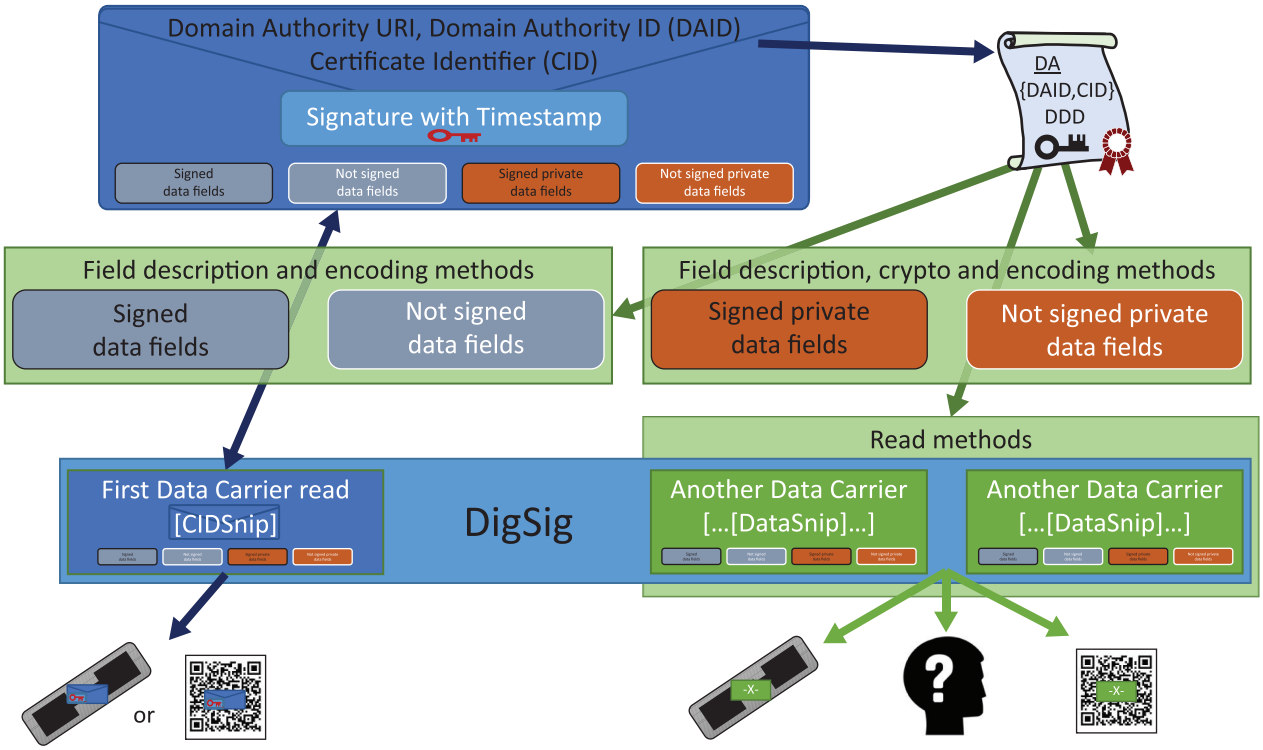

Figure 5 — Components of DigSig envelope as specified by a DigSig Data Description (DDD)

8.2 DDD derived data structures

8.2.1 General

DDD derived data structures are JSON constructs.

DDD derived data structures shall follow the DDD syntax as specified in 8.5.

The specified field order and hierarchy shall be maintained between a DDD and all its derived data structures.

EXAMPLE I.3 shows a DDD example.

8.2.2 DDDdata

DDDdata shall be a JSON array of arrays of values following the structure of the associated DDD `datafields` object (see 8.1), as such, DDDdata shall NOT contain JSON objects.

DDDdata shall only contain the values of the DDD leaf-fields, field values.

The JSON types for field values shall be:

— Single value fields (see 8.9.3.1 to 0): A JSON string, number, true, false, or null, e.g.

`[…"abc","XX",123,true…]` the values of a string, an enumeration, a number and a boolean field.

— A structure defined by struct (see 8.9.3.9): A JSON sub-array, e.g.

`[…["abc",123,true]…]` the values of a structure with a string, a number and a boolean field.

— An array of a field values, cardinality (see 8.9.2.2): a JSON array of the field values.

`[…["abc","def"…]…]` the values of a list of a string field.

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

— An enumeration where the enumeration selects a set of fields, `enumvaluefields` (see 8.9.3.8): a JSON string followed by a JSON sub-array, e.g.

`[…"XX",[123,true]…]` the values of the enumeration "XX" set of fields which is in this case a number and boolean field.

The values of the fields signature and timestamp of DDDdata^{input} shall be ignored.

NOTE DDDdata are used as an input and output structure: see in Figure 5 and 8.3.3.4 The values of the fields signature and timestamp are generated by the DigSig EncoderGenerator.

EXAMPLE For DDDdata^{input} see I.5. For DDDdata see I.6.

8.2.3 SigData

SigData shall be a JSON array, a subset of DDDdata.

SigData shall only contain the data fields specified to be signed and verified. SigData shall be generated by deleting all field values of which the field descriptor bsign equals false.

It is important that signed data are computed to be consistent, else a signature verification may be rejected even when the data is correct, but the formatting different. As such the following rules apply:

— The data to be signed is the full JSON array, starting with the first "[" and ending with the last "]".

— 〚A1〛 *Text deleted.* 〚A1〛

— All whitespaces and formatting, outside of string values shall be removed.

— Numbers shall only contain the characters 0 to 9, the decimal point when the fraction is non-zero and the leading negative sign. Numbers shall not contain leading zeros and fraction trailing zeros.

— 〚A1〛 The binary value presentations shall be as follows: bstring uses HexString, digsigenv uses Base64String and privatecontainer uses HexString. 〚A1〛

NOTE The above rules are based on the output of typical JSON message generation methods supported by most common coding languages.

EXAMPLE See I.7.

8.2.4 DDDdataTagged

DDDdataTagged shall be a JSON object whereby each array and field value of DDDdata is tagged with its fieldid value as set in the DDD to become a JSON pair.

EXAMPLE 1 This example is abbreviated and edited for readability purposes. It has two single value fields (A and B), a field with cardinality (a list C with 2 values), an simple enumeration (E), an enumeration with value fields (for F1: F11 and F12, and F2: F21 and F22) and a structure (S) with two single value fields (S1 and S2).

DDD datafields:

```
[{"fieldid":"idA"},                                                    # A single value
 {"fieldid":"idB"},                                                    # A single value
 {"fieldid":"idC","cardinality":"{2}"},                                # A list
 {"fieldid":"idE","enumvalues":["E1","E2"]},                           # An enumeration
 {"fieldid":"idF","enumvalues":["F1","F2"],                            # An enumeration
  "enumvaluefields":[[{"fieldid":"idF11"},{"fieldid":"idF12"}],        #  with fields
                     [{"fieldid":"idF21"},{"fieldid":"idF22"}]]},      #
 {"fieldid":"idS","fields":[{"fieldid":"idS1"},{"fieldid":"idS2"}]}] # A structure
```

DDDdata:

```
["valA","valB",["valC1","valC2"],"E1","F1",["valF11","valF12"],["valS1","valS2"]]
DDDdataTagged:
{"idA":"valA","idB":"valB",                          # Single values
 "idC":["valC1","valC2"],                            # A list
```

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

```
 "idE":"E1",                                          # An enumeration
 "idF":"F1","F1":{"F11":"valF11","F12":"valF12"},     # An enumeration with fields
 "idD":{"idS1":"valS1","idS2":"valS2"}}               # A structure
```

EXAMPLE 2 See I.8.

8.2.5 DDDdataDisplay

DDDdataDisplay is used to display DDDdata in the desired language and time zone. DDDdataDisplay shall only contain information for display purposes. All other information shall be purged.

See 8.10 for details.

EXAMPLE The following is the DDDdataDisplay `datafields` value in an abbreviated form for the example in 8.2.4.

```
[{"fieldid":"idA","displayvalue":"valA"},
 {"fieldid":"idB","displayvalue":"valB"},
 {"fieldid":"idC","displayvalue":["valC1","valC2"]},
 {"fieldid":"idE","displayvalue":"E1"},
 {"fieldid":"idF","displayvalue":"F1"},
 {"fieldid":"F1","displayvalue":[{"fieldid":"F11","displayvalue":"valF11"},
                                 {"fieldid":"F12","displayvalue":"valF12"}]},
 {"fieldid":"idD","displayvalue":[{"fieldid":"idS1","displayvalue":"valS1"},
                                  {"fieldid":"idS2","displayvalue":"valS2"}]}]
```

8.3 DigSig format

8.3.1 General

See Figure 5. The encoded form of the data of a DDD data is called a DigSig. A DigSig may be stored over more than one data carrier, including human memory and observation. It contains fields which are signed or not signed and open or private.

The DigSig envelope is the encapsulation of the DigSig. The first part of DigSig envelope shall be the CIDSnip with the first two fields the {DAID, CID}. The {DAID, CID} references the DigSig Certificate, which contains the DDD, which in turn defines methods to read, decode and verify data of the DigSig.

8.3.2 Snips

8.3.2.1 Snip types

A Snip is a continuous binary or text sequence of encoded field values.

The following Snips shall be used:

— CIDSnip: The CIDSnip is the head, the first Snip, of the DigSig. It shall contain the DAID, CID as the first two fields, and all data fields without a `readmethod` or `entertext` pragma.

— DataSnips: DataSnips are stored in another part of the CIDSnip data carrier, or on another data carrier as specified by the `readmethod pragma` (see 9.4). The `entertext pragma` (see 9.2) provides a method for a human to be the data carrier (memory and observation) of a DataSnip.

NOTE A Data Carrier will typically provide a CIDSnip in an automated manner, e.g. when reading a barcode, the content of the barcode will be the CIDSnip as a singular read. In contrast, inventory of an ISO/IEC 18000-63 will only provide the UII. It is possible for the UII to contain the complete DigSig, though it is expected that the TID and some user memory can also be read. In the latter case, the TID and user memory are provided as separate DataSnips.

8.3.2.2 Snip encoding

A Snip shall be encoded in binary using network byte order (see 4.4) using the DigSig Data Description (DDD).

© ISO/IEC 2022 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

A DataSnip may be provided as text (see 9.2 and 9.4). It shall be validated and converted to the DDD specified format.

DataSnips shall be identified by the `fieldid` value, the field short name, of the field contained by the DataSnip (see 8.8, 9.2, 9.4 and D.3.9).

The following shall apply to padding a Snip:

— The head of a Snip shall NOT be padded.

— The tail of a Snip may be padded with any bit value.

NOTE The DigSig decoder knows exactly the encoding format of the Snip, as such, padding is inherently ignored.

EXAMPLE See I.4.

8.3.3 Envelope format

8.3.3.1 General

The envelope contains fields as specified in 8.7.

8.3.3.2 RAW envelope

A RAW envelope is the CIDSnip.

The DataSnips are included in the envelope by reference (see 9.2 and 9.4) as specified by the DDD.

8.3.3.3 URI envelope

The URI envelope shall be the CIDSnip encoded in base64url preceded with a URI in accordance with IETF RFC 3986. The DataSnips are included in the envelope by reference (see 9.2 and 9.4) as specified by the DDD.

The URI envelope shall have the following format:

```
URI envelope |= <verificationuri> || "?" || Base64url_encoding (CIDSnip)
```

The verification URI used should be one of the listed `verificationuri`s (see 8.5).

NOTE The URI envelope is useful for online verification where a data carrier is read using a handheld device with generic data carrier readers, e.g. a smart phone with a barcode and/or RFID reader.

8.3.3.4 GS1 envelope

The GS1 envelope shall comply with the GS1 digital signature directives.

8.3.4 AIDC specific construction of a DigSig

A DigSig envelope (a formatted CIDSnip, see 8.3.3.3) may be stored on all AIDC data carriers capable of carrying binary data or a Base64 URL. The following methods apply:

— The application knows by design it is reading a DigSig from the AIDC data carrier. No special data carrier encoding is required to indicate the carried DigSig.

— An AIDC data carrier that is specified to carry a URL shall use the URI envelope (see Annex L).

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

— An AIDC data carrier that is specified to carry a DigSig:

  — For example, the ISO/IEC 15961-2 data construct register has assigned AFI 0x92 and DSFID 0x11 to indicate a DigSig (see Annex M).

  — For example, an ISO/IEC 15434-based message, having the ANSI MH10.8.2 Data Identifier "6R" (see Annex K) as part of the message, indicates a DigSig.

    The DAID encoding may be replaced with the 8-bit encoding 0xFF (reference DAID) where the AIDC data message contains an unambiguous DAID as the first part of the message (see M.4 for an ISO/IEC 17367 example).

The optional DataSnips may be stored on all AIDC data carriers capable of carrying binary or text, since the DDD specifies how a DataSnip is read.

An AIDC text-based data message stored in a DataSnips shall be pruned from all control and whitespace characters for signature generation and signature verification.

The construction and mapping of a DigSig in an AIDC data carrier is illustrated in Figure 6.

The data carrier should have sufficient carrying and transfer capacity for the specified Snip.

NOTE 1 A DigSig using a null signature (see B.4.4) can be as small as 32 bits, however, the smallest "applicable" signature is 256 bits resulting in a minimum DigSig size of 320 bits (see B.3.1). Using a DataSnip for the 256-bit signature can assist in reducing the impact of the carrier capacity and read limitations. It also provides a method to seamlessly introduce authenticity in legacy systems.

NOTE 2 Removing all control and whitespace characters from the text-based message eliminates any ambiguities that they can bring during the coding process.

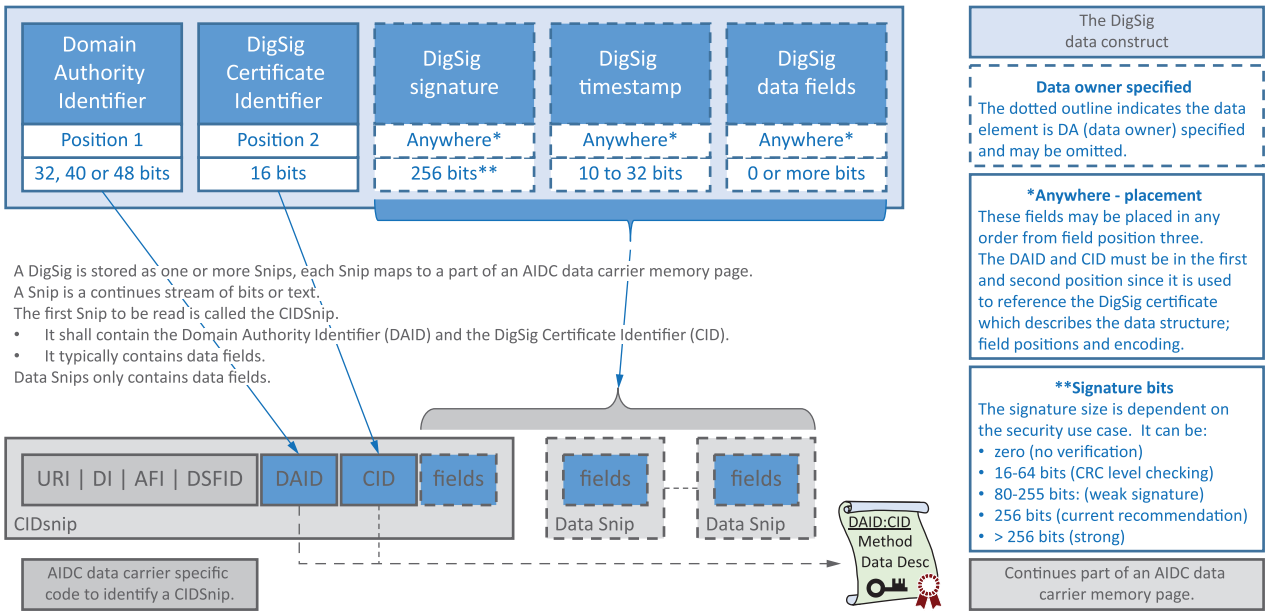

Figure 6 — The construction and mapping of an DigSig in an AIDC data carrier

8.4 The DigSig physical data path

The physical data path of a DigSig is illustrated in the steps below as depicted in Figure 7, and illustrated in Annex N.

$DDDdata^{input} \Leftarrow LocalApplicationGenerateDDDdata(Certificate^{\{DAID,CID\}}, ApplicationData)$

$SigData \Leftarrow ExtractSigData(Certificate^{\{DAID,CID\}}, DDDdata^{input})$

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

{Signature, Timestamp } ⇐ Sign(Certificate$^{\{DAID,CID\}}$, SigData)

DigSig ⇐ EncodeDigSig(Certificate$^{\{DAID,CID\}}$, DDDdata^{input}, {Signature, Timestamp})

Programmed data carrier… ⇐ WriteDigSig(DigSig)

{CID', DigSig'} ⇐ ReadDigSig(Data Carrier…)

{DDDdata', SigData', Signature'} ⇐ DecodeDigSigCertificate$^{\{DAID,CID\}'}$, DigSig')

VerificationOutcome ⇐ Verify(Certificate$^{\{DAID,CID\}'}$, SigData', Signature'}

ApplicationData ⇐ TranslateFromLocalApplication(Certificate$^{\{DAID,CID\}'}$, DDDdata',

VerificationOutcome)

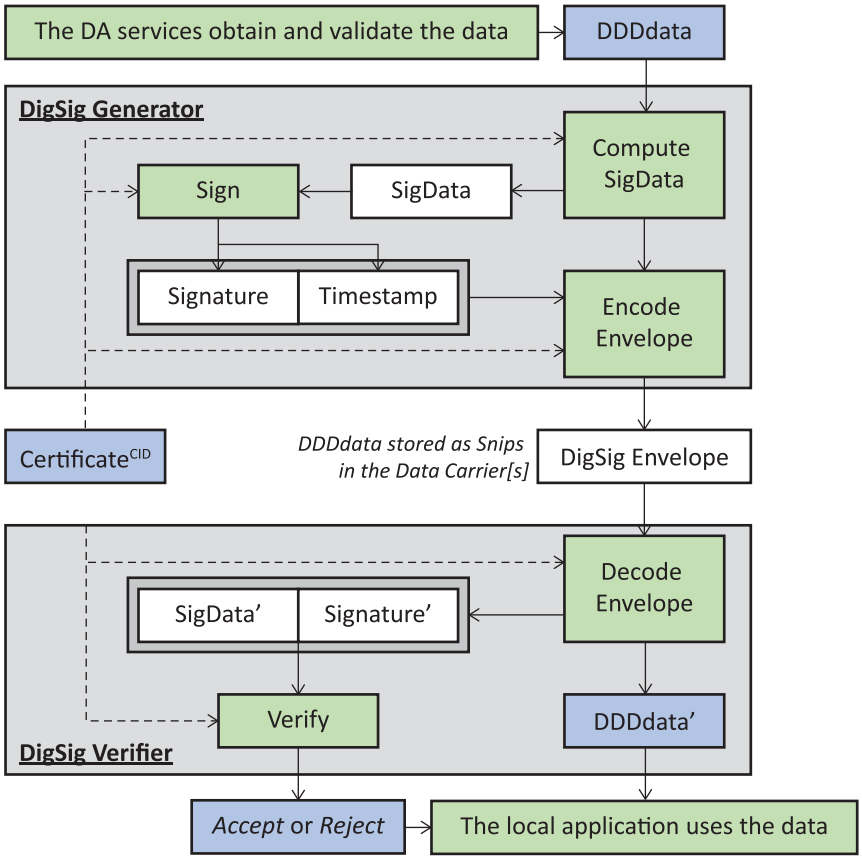

Figure 7 — Physical data path

8.5 DDD syntax

The DDD shall have the following syntax:

```
{"digsiginfo":
  {"specificationversion":<specificationversionvalue>,        # See 5.1.
   "dauri":<dauri>,
   "daid":<DAID>,                                             # See 7.5.
   "cid":<CID>,                                               # See 7.6.
   "verificationuri":[<verification URI>,…],                  # See 7.10.
   "revocationuri":[<revocation URI>,…],                      # See 7.9.
   "preverify":<display string before verification>,
   "acceptverify":<display string on verification accept>,
   "rejectverify":<display string on verification reject>,
   "postverify":<display string after verification>,
   "structureddocuri":{<language code>:<document URI>,…},     # See 8.11.
   "custom":<JSON value with custom information>},
```

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

```
"datafields":
 [{"fieldid":"specificationversion",      # Shall be the first field.
   "type":"string",
   "bsign":true},
  {"fieldid":"dauri",                     # Shall be the second field.
   "type":"string",
   "bsign":true},
  {"fieldid":"daid",                      # Shall be the third field (see 7.5).
   "type":"string",
   "bsign":true},
  {"fieldid":"cid",                       # Shall be the fourth field (see 7.6).
   "type":"unsignedint",
   "bsign":true},
  {"fieldid":"signature",                 # This field may be placed elsewhere.
   "type":"bstring",                      # The elected signature method length applies
   "bsign":false,                         # as a fixed length (see B.4 for examples).
   "binaryformat":"{<signature length>}"},
  {"fieldid":"timestamp",                 # This field may be placed elsewhere.
   "type":"isodate",                      # The time zone shall be UTC. range is used
   "bsign":true,                          # to configure the timestamp span to be at least
   "range":<DigSig issuing period>},      # as long as the issuing period (see 7.7).
  …]}                                     # Optional data fields.
```

`specificationversion`, `dauri`, `cid`, `signature` and `timestamp` are compulsory data fields. The compulsory fields shall only use the field descriptors as specified in this clause. The `timestamp range` field descriptor shall be included in the DDD. The other field descriptors may be omitted from the DDD.

EXAMPLE See B.3.1.

8.6 DigSig information fields

The DigSig information fields describe the use environment of the DigSig and DDD. `digsiginfo` fields are not included in the DigSig. `digsiginfo` fields, as part of the DDD, are included in the DigSig certificate, as such, it is a verifiable set of information applicable to all the DigSigs associated with a specific DigSig certificate.

NOTE DigSig information fields and the descriptions of the DigSig data fields are applicable to all the DigSigs generated using the DDD authenticated by the DigSig certificate. These "static" fields and information are not encoded on the AIDC data carrier with an explicit aim to keep the encoded bit count as low as possible. The AIDC data carrier is therefore only encoded with dynamic data (see 8.7).

`digsiginfo` is a JSON object. It shall contain only the fields specified in Table 5. These fields shall not contain DDDdata values.

Table 5 — digsiginfo fields

Name	Purpose and value	Type	Inclusion
specificationversion	Set according to 5.1.	JSON string	compulsory
dauri	The Domain Authority URI (see 6.2).	JSON string	compulsory
daid	The Domain Authority identifier (see 7.5).	JSON string	compulsory
cid	The DigSig certificate identifier (see 7.6).	unsigned integer	compulsory
verificationuri	These URIs point to the online verification service. It is also used for the URI envelope.	JSON array of strings	compulsory to list at least one
revocationuri	These URIs point to a revocation service.	JSON array of strings	compulsory to list at least one
preverify	The text shall be displayed before verification.	displaystring	optional
acceptverify	The text shall be displayed on a successful verification.	displaystring	optional
rejectverify	The text shall be displayed on a rejected verification.	displaystring	optional
postverify	The text shall be displayed after verification.	displaystring	optional

© ISO/IEC 2022 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

Table 5 *(continued)*

Name	Purpose and value	Type	Inclusion
structureddocuri	This URI shall point to a Media Type (IEFT RFC 6838) resource which is used by a direct substitution method to create a populated document reflecting the verified DigSig display data (see 8.11).	A JOSN object of JSON objects	optional
custom	JSON object with DDD custom information.	JSON object	optional

8.7 Data fields

8.7.1 General

The DDD datafields field specifies the information, structure, read method and encoding of the DDDdata fields. Only the values of the datafields are included in a DigSig. The static data is contained in the DigSig Certificate (see 8.6).

— `datafields` is a JSON array of objects.

— `datafields` shall contain the compulsory data fields as depicted in Table 6.

— `datafields` may contain application data fields.

8.7.2 Compulsory data fields

The compulsory DigSig data fields are specified in Table 6.

Table 6 — Compulsory DDD data fields

Names	DDD Type (type)	Signed (bsign)	DDDdata Order	DigSig Order
specificationversion	string	Yes	First	Not included
dauri	string	Yes	Second	Not included
daid	DAID rules (see 7.5)	Yes	Third	First
cid	16-bit unsignedint	Yes	Fourth	Second
signature	fixed length bstring	No	Default fifth	Default third
timestamp	isodate	Yes	Default sixth	Default fourth

The following special rules shall apply:

— Compulsory data fields should not contain field descriptors. The settings of Table 6 shall be used.

  — timestamp shall contain `range`.

  — signature may contain `binaryformat` to specify they signature length for DDD encoding analysis. The `binaryformat` shall be ignored. The signature length shall be set according to the signature method used.

— signature shall be generated at the point of signing using the DigSig certificate specified signature algorithm and its specified bit length.

— timestamp shall be generated at the point of signing using the specified range and accuracy (see 8.9.3.7.3).

8.7.3 Application data fields

Application data fields are specified by the domain authority for a specific use case. Application data fields are optional.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

Application data fields may be included by codebook reference (see 8.11).

8.8 Data field object syntax

Data fields are JSON objects with the following syntax:

```
{"fieldid":<field short name>,
 "type":<DDD type>,
 <optional field descriptors>}
```

A data field object may contain field descriptors as specified in Table 7. These field descriptors provide a method to specify the field syntax, semantics and encoding. The semantics use the multi-language displaystring (see 8.10.2).

Table 7 — DDD field descriptors

Name	Purpose and value	Default	Type	Inclusion
fieldid	The short name of the field used as its identifier. The fieldid shall be unique within the DDD and shall only contain [A-Za-z0-9].	-	JSON string	compulsory
type	Specify the field type of application fields as boolean, unsignedint, number, string, bstring, digsigenv, isodate, enum, or struct. The type of the compulsory fields is specified in 8.7.2 and should be omitted.	-	JSON string	compulsory
fieldname	A display string that is intended to name the field differently from fieldid. This is for display purposes on capturing, printing, or extracting the data.	the value of fieldid	displaystring	optional
description	A display string to provide meaning to a field when capturing, printing, or extracting the data.	null	displaystring	optional
pragma	See Clause 9.	null	JSON object	optional
bsign	<true | false>; if true then the field value is included in the signature.	true	boolean	optional
cardinality	Specify a list of values for a field (see 8.9.2.2).	"{1}"	JSON string	optional
nullable	<true | false>; if true then the field value may be set to null (see 8.9.2.1).	false	boolean	optional
nulldescription	A display string to describe a nullable field when it contains a null.	null	displaystring	optional
nonnulldescription	A display string to describe a nullable field when it contains a value.	null	displaystring	optional
binaryformat	This field descriptor specifies the value format to achieve optimal binary encoding (see 8.9.3.x).	as specified by type	JSON string or number as specified by type	optional
range	The allowed value range of the field (see 8.9.3.x).	as specified by type	as specified by type	optional
decimalunit	Specifies the accuracy of a number (see 8.9.3.3).	0	integer	optional with the type number
truncate	Specifies the truncation of the input value of a number (see 8.9.3.3).	"round"	string	
timezone	Specify the time zone to be used with DDDdataDisplay (see 8.9.3.7 and 8.10.3.6).	"UTC"	JSON string	

© ISO/IEC 2022 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

Table 7 *(continued)*

Name	Purpose and value	Default	Type	Inclusion
enumvalues	Enumeration values (see 8.9.3.8).	null	JSON array of JSON strings	used with the type enum
enumvaluedesc	The description of the enumeration values (see 8.9.3.8).	null	JSON array of displaystring	optional with the type enum
enumvaluefields	A set of fields associated with the enumeration value (see 8.9.3.8).	null	JSON array of objects	
fields	A JSON object that contains the fields of the type struct (see 8.9.3.9).	at least one field	As specified by struct	used with the type struct
defaultvalue	The default value for the field when capturing the data.	null	As specified by type	optional
displayformat	The string format for DDDdataDisplay of the field value (see 8.10).	according to the type	As specified by type	optional
custom	JSON object with DDD custom information. Components of this standard shall ignore this field.	null	JSON object	optional

8.9 DDD field types and associate settings

8.9.1 General

JSON supports the following types: number, string and boolean, and the special value null. This document specifies rules for their use.

This document specifies the following additional DDD types: `unsignedint`, `bstring` and `isodate`. The JSON type for unsignedint and date shall be number and bstring shall be string.

This document further specifies complex field types `enum` and `struct`.

— `enum` maps to a JSON string.

— `enumvaluefields` maps to a JSON array of object containing the fields of the specific `enum` value.

— `struct` maps to a JSON array of object containing the fields of the `struct`.

8.9.2 Special field values

8.9.2.1 The DDD field value "null"

The value null is a special value, meaning "no value".

The DDDdata value shall be the JSON literal null if the field is set to null.

A field shall only take the value null if `nullable` is set to true.

The default is false.

A field which is nullable shall have the null bit in the MSB position, the first bit of the binary encoding of the field.

— If the null bit is set to false (1_2) then the field contains a value.

— If the null bit is set to true (0_2) then the field contains no value. The binary encoding of the field shall be collapsed.

© ISO/IEC 2022 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024

If `nullable` is set to true; then, if set, the value of `nulldescription` or `nonnulldescription` shall be displayed within DDDdataDisplay.

See 9.4 for the null binary encoding where the readmethod pragma is used.

8.9.2.2 Array of field values - cardinality

`"cardinality":<field cardinality>`

The type cardinality specifies multiple field values for a field, a list of values.

The DDDdata shall be a JSON array of the field values.

The value of cardinality shall be a JSON string:

— `"{m}"` with m > 0 specifies a fixed number of values. The binary encoding shall reserve m fields.

EXAMPLE 1 "binaryformat":"{5}"

— `"{m,n}"` with 0 ≤ m < n specifies variable length list with m to n values. The encoded value count shall be relative to m, 0 to n-m (n-m+1 values).

EXAMPLE 2 "binaryformat":"{5,10}"

The binary encoding of a variable cardinality field shall be prefixed with `countbits`. `countbits` shall be encoded as follows:

`countbits |= {<countsize>‖<fieldcount>}`

countsize is a variable length bit field terminating with a 0_2, with each preceding 1_2 representing a nibble in `fieldcount`.

fieldcount is the number of field values. `fieldcount` is encoded as a multiple of nibbles ($CCCC_2$).

EXAMPLE 3 countsize 0_2 specifies m field values.

EXAMPLE 4 countsize 10_2 $CCCC_2$ specifies m plus 0 to 15 field values.

EXAMPLE 5 **countsize 110_2 $CCCC_2$** $CCCC_2$ specifies m plus 0 to 255 field values.

8.9.3 Field types

8.9.3.1 boolean

`"type":"boolean"`

A boolean field is a binary switch.

The DDDdata value shall be the JSON literals true, false.

EXAMPLE A DDDdataTagged example "IamSMART":true

The binary encoding of a boolean field shall be a single bit set to 0_2 when false, and 1_2 when true.

NOTE If the boolean field is marked as nullable, then the encoding is two bits long when not null (false: 10_2; true: 11_2) and one bit when null (0_2).

The following Table 7 field descriptors are valid for this field: `fieldname`, `description`, `pragma`, `bsign`, `cardinality`, `nullable`, `nulldescription`, `nonnulldescription`, `defaultvalue`, `displayformat` and `custom`.

8.9.3.2 unsignedint

© ISO/IEC 2022 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO/IEC 20248:2022+A1:2024