For Publication on the ICAO Website

TECHNICAL REPORT

Visible Digital Seals for Non-Electronic Documents - Visas

Version: 1.31

December 2016

File: Technical Report – Visible Digital Seals for Non-Electronic Documents V1.31

Author: ISO/IEC/JTC1/SC17/WG3/TF5 for the New Technologies Working Group (NTWG), Working group of the ICAO Technical Advisory Group on the Traveller Identification Programme (TAG/TRIP).

Release Control

Release	Date	Description
1.3	Dec. 6th, 2016	Added disclaimer in section 1.1 with respect to VDS SubCA
1.2	Dec. 5th, 2016	Updated references
1.1	July 24th, 2015	Inserted ICAO OID
1.0	May 5th, 2015	Final draft incorporating comments from WG3/TF1 Meeting May 2015 (Leiden)
0.3	April 15h, 2015	3rd draft incorporating received comments
0.2	Oct. 13th, 2014	2nd draft incorporating discussions from WG3/TF1 Meeting September 2014 (Salamanca)
0.1	Sep. 17th, 2014	First Draft Version

Table of contents

1. INTRODUCTION 4
1.1 DISCLAIMER 5
2. TERMINOLOGY AND DEFINITIONS 6
3. USE CASE (INFORMATIVE) 9
3.1 PREREQUISITE: VISA SIGNER CERTIFICATE GENERATION 9
3.2 DIGITAL SEAL GENERATION 10
3.3 DIGITAL SEAL VALIDATION 10
4. DIGITAL SEAL ENCODING 11
4.1 BARCODE FORMAT AND PRINT REQUIREMENTS 11
4.2 HEADER 11
4.3 MESSAGE ZONE 13
4.3.1 Digital Encoding of Document Features (Binary Encoding) 13
4.4 SIGNATURE ZONE 14
4.5 PADDING 14
5. DIGITAL SEALS FOR VISA DOCUMENTS 15
5.1 CONTENT AND ENCODING RULES 15
5.1.1 Header 15
5.1.2 Document Features of a VDS for Visas 15
5.1.3 Encoding Rules for Document Features 16
5.2 CRYPTOGRAPHIC SIGNATURE, PKI, AND CERTIFICATE PROFILES 17
5.2.1 Visa Signer and Seal Creation 17
5.2.2 Public Key Infrastructure (PKI) and Certificate Profiles 20
5.2.3 Visa Validation Authority 26
5.3 VALIDATION POLICY (INFORMATIVE) 27
5.3.1 Policy Rules 27
6. WORKED EXAMPLE (VISA DOCUMENT) 31
7. REFERENCES 33
ANNEX A CONVERSION OF ECDSA SIGNATURE FORMATS (INFORMATIVE) 34
ANNEX B C40 ENCODING OF STRINGS (NORMATIVE) 36

1. Introduction

Long term identity documents such as national identity cards or passports nowadays come equipped with a microchip that stores the information printed on the document in a cryptographically secure manner. This effectively prevents any faking or forging of the document, as any manipulation is easily spotted by cryptographic verification of the data stored in the microchip. Attacking the cryptographic protection is associated with very high costs or even infeasible, due to its mathematical nature.

Other types of documents, such as breeder documents[1] or visas are usually protected by physical security features alone. The overall volume of such issued documents is quite high compared to the number of issued long-term travel documents. For example for the European Schengen Area, the number of issued visas has been steadily increasing; from approximately 6.7 million visas issued in the year 2009 up to 11.7 million visas issued in the year 2012 [1]. Similar statistics exist for other nations. To combat visa fraud – and thus related effects such as illegal immigration and human trafficking – it is essential to protect the integrity and authenticity of issued visas and other breeder documents. Issuing documents with such a high-volume and short validity period makes it economically infeasible to augment their physical security features by electronic microchips.

Aside from their benefits, physical document features have three major disadvantages: First, they are symmetric. This means that the cost of faking or forging a physical document feature roughly corresponds to the cost of issuing it in the first hand. Thus in order to achieve a reasonable level of security, they have to be expensive. Second, since the equipment needed to issue the document is so expensive, it is difficult to securely personalize the document. Usually blank documents are printed with sophisticated physical security features, but personalization is done by comparatively low-cost printing equipment. A potentially dangerous attack vector is thus the loss of blank documents. Third, verification is non-trivial. Since cheap, yet high quality scanning and printing equipment is common today, it is not difficult to construct forgeries that seem authentic on a superficial level. Spotting physical document features – or the lack of them – is difficult for the untrained eye. And even for an expert it can be very challenging when facing time constraints, i.e. in a border-control situation.

This document specifies a digital seal to ensure the authenticity and integrity of non-electronic documents in a comparatively cheap, but highly secure manner using asymmetric cryptography. The information on the visa document is cryptographically signed, and the signature is encoded as a two-dimensional barcode and printed on the document itself. This approach – the *visible digital seal* – mitigates all three problems mentioned above:

1. *Asymmetry*. Due to using asymmetric cryptography, the cost of attacking a digital seal is considerably higher than the cost of issuing a visa document protected with a seal. Thus even though the cost of issuing a document is very low, it is extremely costly to fake or forge it.
2. *Personalization*. Each seal verifies the information printed on the physical document, and is thus tied to the document holder. There is no direct equivalent of a blank document, and thus no blanks can be lost or stolen.
3. *Easy verification*. Even untrained personal is able to verify a document protected with a digital seal by using low cost equipment, such as an application on a smartphone. Moreover, due to the binary nature of a digital signature, distinguishing between authentic documents and forged ones is easy.

This document is structured as follows. In Section 2, we introduce terminology needed, and give reference definitions for various concepts used in this document. In Section 3, a general overview of the concept of visible digital seals is provided. The container format of a digital seal is defined in

1 Documents that can serve as a basis to obtain other identification documents, i.e. a birth certificate that is used to obtain a passport.

Section 4, whereas Section 5 gives specific definitions for the use-case of digital seals applied to visa documents, including the generation and encoding rules, the public key infrastructure required, the validation policy for such issued visas, and a worked example.

All sections in this document are normative if not marked otherwise. For each annex it is explicitly mentioned whether the annex is normative or just for information purposes. We distinguish between two references for documents: If documents are referenced with an explicit version or date, that explicit version is normative. If documents are referenced without an explicit version or date, their latest iteration – including future versions – is normative.

1.1 Disclaimer

Future releases of this Technical Report may change the requirement for a VDS SubCA to simplify the process of distribution of Visa-signer certificates and the handling of revocation of a Visa Signer Certificate. This change may not be backwards compatible with this release of the Technical Report, and readers of this Technical Report are advised to take cognizance of this.

2. Terminology and Definitions

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [2].

MUST	This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
MUST NOT	This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
MAY	This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

We identify binary values with their hexadecimal representation preceded by 0x in `monospaced` font, e.g. `0x2A`. If clear from context, the prefix `0x` is sometimes omitted. Sometimes binary values are identified with their decimal value, written by appending `dec`, i.e. `42dec`. Throughout this document, we assume Big Endian encoding, i.e. byte sequences are read from left to right. Bit sequences are read from right to left, i.e. we assume the least significant bit (LSB) to be at (the rightmost) position 0.

Moreover we define the following terminology:

Barcode
Optical, machine-readable representation, in one or two dimensions, of data relating to the object to which it is attached

Barcode Symbology
A mapping between messages and barcodes is called a symbology. Such mapping is defined in the specification of the barcode and includes the encoding of single digits or characters, the size of a so called quiet zone around the barcode, as well as the computation of checksums for error correction.

Certificate

Electronic file attesting that a cryptographic key pair belongs to a person or a hardware or software component as identified in the certificate. A certificate is issued by a Certification Authority. By signing the certificate, the Certification Authority approves the link between the identity of a person or component and the cryptographic key pair. The certificate may be revoked if it doesn't attest the validity of this link any more. The certificate has a limited validity period.

Certificate Revocation List (CRL)
A list of certificates that have been revoked. Documents that identify a certificate from a CRL for verification shall thus no longer be trusted.

Country Signing Certification Authority (CSCA)
The Certification Authority of a country that signs document signer certificates. Document issuers, such as makers of passports, use the private keys corresponding to the document signer certificates to sign data on electronic machine readable travel documents (eMRTDs). The CSCA of each Issuing State or organization acts as the trust point for the Receiving State. The CSCA also signs SubCA-Certificates for SubCA’s below the CSCA.

Data To Be Signed (DTBS)
The message that is given as input to a signature generation algorithm of a signature scheme.

Cryptographic Signature
The output generated by a signature algorithm of a signature scheme.

Cryptographic Signature Scheme
A tuple of three algorithms. The key-generation algorithm takes as input a security parameter and outputs a key pair consisting of a private and a public key. The signature algorithm takes as input a private key, and a message, and outputs a cryptographic signature. The verification algorithm takes as input a public key, a message, and a signature, and outputs valid if the signature was generated using the signature generation algorithm with the private key of the key pair and the message as input, and invalid otherwise.

(Digital) Document Feature
A property of a document which can be used to verify the contents of the document. Examples are textual information such as the name of the holder, or the issuing date, or a printed image of the document holder. A digital document feature is the digitized version of a document feature.

Digital Seal
Short for *Visible Digital Seal*.

Document Signer (DS)
A Document Signer digitally signs data to be stored on eMRTDs; this signature is stored on the eMRTD in a Document Security Object

Document Signer Certificate (DSC)
A DSC is a certificate that contains the Document Signer's public key. Document Signer certificates are used to verify the validity of Document Security Objects that were signed with the Document Signer's private key.

Elliptic Curve Digital Signature Algorithm (ECDSA)
A variant of the Digital Signature Algorithm (DSA) based on elliptic curve cryptography.

Machine Readable Travel Document (MRTD)
A travel document as defined in [3].

Machine Readable Zone (MRZ)
A fixed dimensional area on the MRTD as defined in [3], or on a visa as defined in [4], consisting of textual data printed in a font designed for easy Optical Character Recognition (OCR).

Master List
A Master List is a digitally signed list of the certificates that are 'trusted' by the Receiving State that issued the Master List [8].

Physical Document Features
Physical properties of a document that prevent forging or faking it. Examples are watermarks, holograms, or micro-printing.

Signature Scheme
see cryptographic signature scheme.

(Feature) Tag
A byte that uniquely identifies a document feature. The mapping between feature tags and features must be specified in a profile.

Visa Signer (VS)
The authority that receives data from a visa personalization system and that uses a VS certificate and the corresponding private key to encode and sign a visible digital seal.

Visa Signer Certificate
A certificate containing information identifying the entity that signed a visible digital seal on a visa, and containing the public key corresponding to the private key with which the signature was created.

Visa Validation Authority (VVA)
The authority that validates a visible digital seal based on a validation policy.

Visible Digital Seal (VDS)
A cryptographically signed data structure containing document features, encoded as a 2D barcode and printed on a document.

Visible Digital Seal CA (VDS CA)
Sub CA below the CSCA, and above the Visa Signer.

3. Use Case (Informative)

This section gives a general overview of using a digital seal to protect a non-electronic document. The specific use case considered here is the protection of a visa document, and depicted in Figure 1. Whereas technical details vary for other use cases, the same general principles apply.

The general workflow can be separated into three steps. As a prerequisite, Visa Signer Certificates (VSC's) have to be generated. Next, seals are generated, and then later validated.

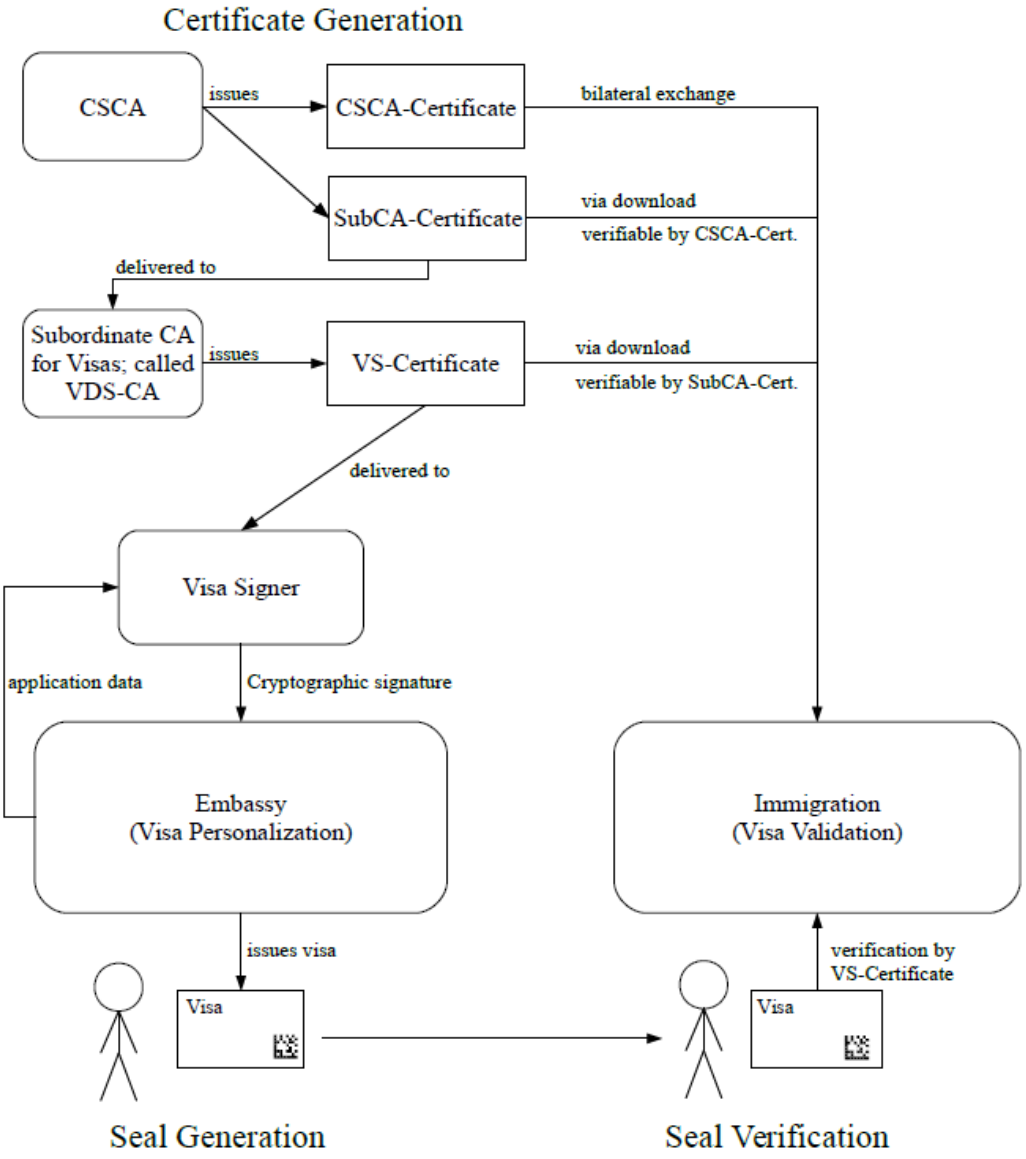

Figure 1: VDS Use Case

3.1 Prerequisite: Visa Signer Certificate Generation

The visa signing PKI is based upon the PKI set up for electronic passports defined by ICAO. At the root is the Country Signing Certificate Authority (CSCA) of each country. The CSCA publishes a CSCA-Certificate containing the public key of the CSCA. To enable trust between countries, this CSCA-Certificate is distributed in a trustworthy manner via bilateral exchange, or via master lists.

Below the CSCA are several subordinate CA's (SubCA's) for different applications. The SubCA relevant here is the one for (paper) visa documents. The certificate containing the public key of the SubCA is signed and published by the CSCA. Therefore, the SubCA-Certificate is verifiable with the CSCA-Certificate, and does not need to be distributed in a trustworthy manner. Hence simple distribution by e.g. web download or even manual exchange (without involving diplomatic channels) suffices. In the present specification, we call the Sub-CA relevant here the *visible digital seal SubCA* or VDS SubCA. The VDS SubCA itself signs and issues Visa Signer Certificates (VS-Certificates) of the Visa Signer.

The Visa Signer is the entity that actually signs digital seals. Similar to the scenario above, a VS-Certificate can therefore be verified by the VDS SubCA-Certificate, which itself is verifiable by the CSCA-Certificate, and hence distribution of these certificates can be done in a simplified and light-weight manner. Mechanisms for publication of VS certificates by the VDS SubCA are described in Section 5.2.2.4.

3.2 Digital Seal Generation

A seal is generated in two steps:

1. An applicant applies for a visa at the embassy where he resides. The embassy records the applicant's data and checks whether the applicant meets the requirements to receive a visa. If the requirements are fulfilled, the embassy sends a digital representation of the recorded data to the Visa Signer (VS). The VS can either be (1) a central entity located in the country that issues the visa, and the embassy connects to the VS via a secure channel, or (2) the VS is a decentral entity placed at each embassy, for example smartcards containing cryptographic keys that are directly attached to the personalization system. In any way, the VS cryptographically signs the recorded data.
2. For signing, the Visa Signer uses a key pair of a private key and a public key. The actual signing is done with the private key, whereas the public key is stored in a Visa Signer Certificate. The resulting signature is sent back to the Visa Personalization System if the Visa Signer is not a local part of the personalization system, printed on the visa sticker, and the visa sticker is attached to the applicant's passport.

3.3 Digital Seal Validation

When the applicant enters the issuing country, he presents his visa to a Visa Validation Authority (VVA), e.g. the immigration control of the issuing country. The VVA verifies the authenticity and integrity of the digital seal on the visa by validating the signature of the seal, and comparing the printed information on the visa sticker and on the passport with the digital information stored in the seal. The signature of the seal is verified by identifying the corresponding VS-Certificate with the help of the identifier stored in the header of the seal, and then using the public key of the VS-Certificate. As described in the previous paragraphs, the validity of the VS-Certificate itself can be verified by the VDS SubCA-Certificate, which itself can be verified by the CSCA-Certificate.

Remark
Since all certificates are publicly available, the validity of the visa can be verified by *any* third party, not just by the issuing state. The approach can thus handle use cases for unions of countries, where one country issues a visa for another country (as is done for example in the European Union). Another use case is verification of visas by airlines prior boarding a plane.

Remark
The criteria to determine if a visa document can be trusted or not based on the digital seal and the MRZs of the visa and the passport are defined in a validation policy (see Section 5.3).

4. Digital Seal Encoding

A visible digital seal is a cryptographically signed data structure containing document features, encoded as a 2D barcode and printed on a document. This section gives a definition of the encoding and structure of a visible digital seal.

4.1 Barcode Format and Print Requirements

This specification defines how visa information are encoded into a stream of bytes. Only 2D barcodes whose symbology is specified as an ISO standard SHALL be used. ISO standardized 2D barcodes symbologies include for example DataMatrix [6], Aztec Codes [19], and QR Codes [20].

The barcode SHOULD be printed in a way, that reader equipment (i.e. off-the-shelf smartphones or scanners) are capable to reliably decode the barcode; in particular [21] SHOULD be taken into account when assessing print quality.

When using standard inkjet printers, it is RECOMMENDED to print with a module size (size of one block of a 2D barcode) of at least 0.3386mm width/height per module, corresponding to 4 dots per module on a 300dpi printer, or 8 dots per module on a 600 dpi printer. Smaller printing sizes MAY be acceptable, if high-resolution printers or laser-printers are used. For the placement of the barcode on the visa sticker cf. Annex E of [4].

The encoded barcode consists of a header, the message zone, and the signature zone. An overview of the structure of the barcode is given in Figure 2.

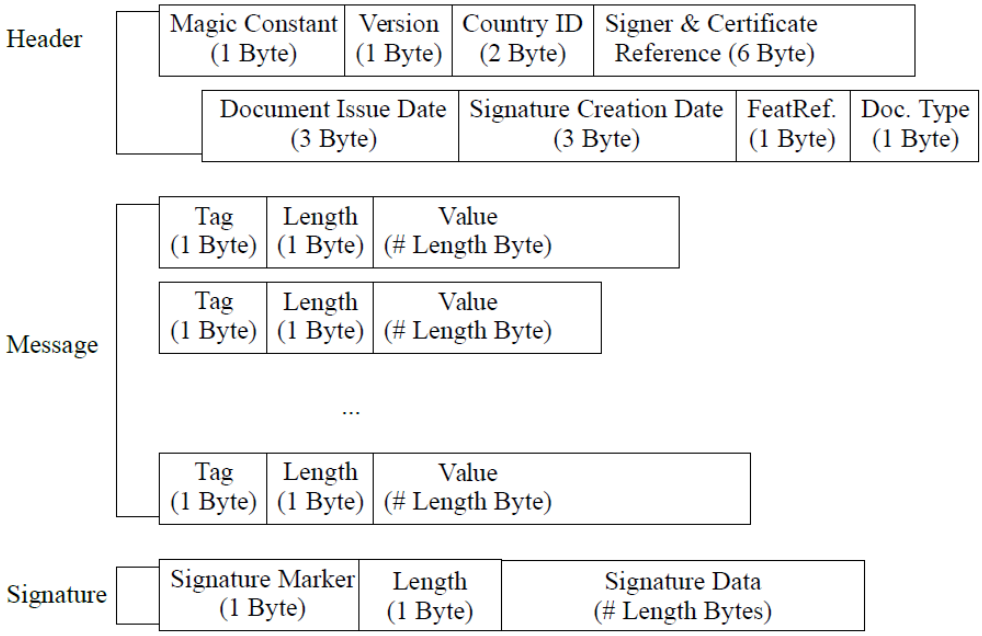

Figure 2: VDS Structure

4.2 Header

The header contains meta-data about the document and the encoding, such as a version number, and document issue and signature creation dates. The fixed overall length of the header is 18 bytes. A definition of the header is given in Table 1.

Signing Authority and Certificate Reference
Due to size restrictions, it is impossible to store the certificates that contain the public key corresponding to the signature within the barcode. Therefore, the certificate must be acquired on a different channel. In order to uniquely identify the certificate and the signer that is the subject of the certificate, and to link the certificate to the barcode, a short string containing an identifier of the Signing Authority and a reference to the certificate is stored in the header. This string consists of:

1. The *Signer Identifier*: The combination of the two letter country code according to [7] of the Signer's country and of two alphanumeric characters to identify a Signer within the above defined country. The Signer Identifier MUST be unique for a Signer in a given country.
2. The *Certificate Reference*: A hex-string of excactly five characters that MUST uniquely identify a certificate for a given issuer of the certificate.

Note that for the specific use case of visas, the Signer is the *Visa Signer*.

The Certificate Reference `00000` is reserved for testing purposes and MUST NOT be used in production.

The (Visa) Signer Identifier and Certificate Reference MUST correspond respectively to the Subject DN and the serial number of a Signer Certificate (for a Visa Signer this is described as an example in detail in Section 5.2.2.3). Thus, the Signer Certificate can be uniquely identified upon decoding the header.

The combination of the *Document Feature Definition Reference* and *Document Type Category* identify a specific set of rules, such as this specification. Future use cases can thus reuse the same barcode and header format, but reference different feature definitions or document types. This allows to reuse existing codebases, simplifies implementations and increases interoperability.

Table 1: Format of the Header

Start Position	Length (Byte)	Content
`0x00`	1	*Magic Constant*. The magic constant `0xDC` identifying a barcode conforming to the current specification
`0x01`	1	*Version*. A byte value identifying the version of this specification. The current version is identified by the byte value `0x03`.
`0x02`	2	*Issuing Country*. A three letter code identifying the issuing country. The three letter code is according to [5] and encoded by C40 (cf. Annex B) as a two-byte sequence.
`0x04`	6	*Signer Identifier and Certificate Reference*. A nine letter code identifying the (Visa) Signer and the certificate.
`0x0A`	3	*Document Issue Date*. The date the document was issued.
`0x0D`	3	*Signature Creation Date*. The date the signature was created. Encoded as defined in Section 4.3.1.
`0x10`	1	*Document Feature Definition Reference*. A reference code to a document that defines the number and encoding of document features. Encoded as defined in Section 4.3.1.
`0x11`	1	*Document Type Category*. The category of the document, e.g. (visa, birth certificate, etc.)

Sum	18

4.3 Message Zone

Following the header is the message zone. The message zone consists of the digitally encoded document features, as specified in Section 4.3. Any order of the document features is valid, as long as all mandatory document features are present.

Each document feature is preceded by

- a tag identifying the type of feature (one byte)
- the length of the feature (one byte)

The length of a feature MUST be in the range 0-254dec. Features with a length larger than 254dec bytes are currently not supported. The length value 255dec MUST NOT be used, and is reserved for future use (i.e. to encode the start of a larger length field in future versions of this standard).

4.3.1 Digital Encoding of Document Features (Binary Encoding)

Document features are encoded in the following way. As building blocks, we consider the following basic types:

1. *Alphanum*: Strings of uppercase[2] alphanumeric characters (i.e. A-Z, 0-9 and space)
2. *Binary*: Sequences of bytes
3. *Int*: Positive Integers
4. *Date*: Dates

These basic types are converted to sequences of bytes as follows:

1. Strings of alphanumeric characters are encoded as bytes by C40 encoding (cf. Annex B).
2. Sequences of bytes are taken as they are.
3. For positive integers, their unsigned integer representation is taken.
4. A date is first converted into a positive integer by concatenating the month, the days, and the (four digit) year. This positive integer is then concatenated into a sequence of three bytes as defined in the point 3) above.

Example
Consider March 25th, 1957. Concatenating the month, date and year yields the integer `03251957`, resulting in the three bytes `0x31 0x9E 0xF5`.

#

A digital document feature is a sequence of bytes. It has the following structure:

tag | length | value

Here tag is a unique integer in the range `0-254` acting as an identifier of the document feature, value is a basic type converted to a sequence of bytes, and length is an integer in the range `0-254` denoting the length (the number of bytes) of the value. Note that tag `255` is reserved to denote the start of the signature.

2 The restriction to uppercase letters is due to the limited data capacity of a barcode.

Example
Consider a document feature that encodes the string “VISA01” with assigned tag `0x0A`. The C40 encoded byte sequence (cf. Annex B) of length 4 is `0xDE515826`. The document feature is thus the byte sequence `0x0A04DE515826`.

#

A specific use case must hence augment this definition by enumerating which document features must be present and which can be optionally present, define their tag values and allowed length ranges.

Additional features, i.e. features with unknown tags MAY be present, for example for optional use of the issuing entity. Such additional features MUST NOT use the tag of the additional feature field, or the tag of any other optional or mandatory feature. The presence of features with unknown tags SHALL NOT affect the validity of the visa, if the signature is recognized as valid.Alternative recommended solution

4.4 Signature Zone

The beginning of the signature zone is marked by `0xFF`, encoded as one byte, followed by one byte denoting the length (the number of bytes) of the signature. The length of the signature MUST be in the range 0-254dec. The value 255dec MUST NOT be used, and is reserved for future use (i.e. to encode the start of a larger length field in future versions of this standard).

The input of the signature algorithm MUST be the (hash of the) concatenation of the header and the complete message zone, excluding the tag that denotes the beginning of the signature or the length of the signature. The signature zone contains the resulting signature.

Only hashing and signature algorithms defined in [8] SHALL be used. Due to the resulting signature size, ECDSA with a key length of at least 256 bit in combination with SHA-256 is (at the time this document was created) RECOMMENDED.

Remark
Applying the ECDSA signature algorithm results in a pair of positive integers `(r,s)`. This signature must be stored in raw format in the seal. The bit length of `r` and `s` respectively corresponds to the key length. Thus for example for ECDSA-256, the length of r and s is at most 256 bit = 32 byte each. The signature MUST be stored by computing the unsigned integer representation of `r` and `s`, potentially adding leading zeros to fit `r` and `s` to their expected length, and appending the resulting value of `s` to the one of `r`. See Annex A for a conversion between the ASN.1 and raw format of `(r,s)`.

4.5 Padding

If the header, message and signature together do not fill the available space of the barcode, padding characters shall be appended after the signature. All relevant 2D barcode symbologies define methods for padding in their respective standard, and padding MUST follow that definition.

5. Digital seals for Visa Documents

5.1 Content and Encoding Rules

5.1.1 Header

The *Document Feature Definition Reference* for this use-case is `93dec`.
The *Document Type Category* for Visas is `0x01`.

5.1.2 Document Features of a VDS for Visas

The following document features are stored in the seal:

Machine Readable Zone (MANDATORY)
The Machine Readable Zone (MRZ) of a visa contains the following information [4]:

- issuing state
- surname and first name of the document holder
- passport or visa number
- nationality of the document holder
- date of birth of the document holder
- sex of the document holder
- validity period (valid until …)

Some countries may not issue paper based visas according to [4], but instead use a domestic database to store visa applications, and merely attach a confirmation sticker to the passport. If such countries choose to adopt this standard for such stickers, the above information SHALL be encoded as either the MRZ of an MRV-A or MRV-B.

Additionally, the following document features are stored:

Number of Entries (OPTIONAL)
The number of times the visa holder may enter the territory for which the visa is valid.

Duration of Stay (MANDATORY)
This feature denotes the number of days, month or years during which the visa holder may stay in the territory for which the visa is valid. Note that this is distinct from the valid-until date of the MRZ, which is already stored in the Visa-MRZ: First, in [4] it is remarked that *in most cases this* [Valid-Until field of the Visa-MRZ] *will be the date of expiry of the MRV and indicates the last day on which the visa can be used to seek entry. For some States this will be the date by or on which the holder should have left.* Second, for some issuing countries the stay must be continuous, and for others, the stay can spread over several periods. Thus, to avoid ambiguity during validation, the feature for the duration of stay is mandatory.

Passport Number (MANDATORY)
This feature denotes the number of the passport to which the visa sticker is attached. The passport number might already be present in the MRZ: In [4] it is remarked that *at the discretion of the issuing State, either the passport number or the visa number shall be used in this field* [document number field of the Visa-MRZ]*; however, the latter option can only be exercised where the visa number has 9 characters or fewer.* To avoid ambiguity during validation, the field for the passport number (separate from the MRZ) is mandatory.

Visa Type (OPTIONAL)

This feature encodes the type of the visa. The field is especially intended to be used, if the type of the visa is not encoded as the second letter of the MRZ.

Additional Feature Field (OPTIONAL)
Reserved for future use. This field is OPTIONAL, and intended to store additional verification information in future versions of this standard.

5.1.3 Encoding Rules for Document Features

In the following, the digital encoding of document features of the visa seal is defined.

MRZ of Machine-Readable Visa of Type A (MRV-A [4])

Tag:	`0x01`
Min. Length:	48 Byte
Max. Length:	48 Byte
Value Type:	Alphanumeric
Required:	Required (if visa is of type MRV-A)
Content:	The first line of the MRZ of an MRV-A (44 chars.) and the first 28 chars. of the second line of the MRZ of an MVR-A, concatenated and encoded by C40. The filler symbol < in the MRZ is replaced by <SPACE> prior to encoding by C40.

MRZ of Machine-Readable Visa of Type B (MRV-B [4])

Tag:	`0x02`
Min. Length:	44 Byte
Max. Length:	44 Byte
Value Type:	Alphanumeric
Required:	Required (if visa is of type MRV-B)
Content:	The first line of the MRZ of an MRV-B (36 chars.) and the first 28 chars. of the second line of the MRZ of an MVR-B, concatenated and encoded by C40. The filler symbol < in the MRZ is replaced by <SPACE> prior to encoding by C40.

Number of Entries

Tag:	`0x03`
Min. Length:	1 Byte
Max. Length:	1 Byte
Value Type:	Integer
Required:	Optional
Content:	The integer in the range of 0-255 encodes the number of allowed entries. A value of 0 denotes unlimited entries.

Duration of Stay

Tag:	`0x04`
Min. Length:	3 Byte
Max. Length:	3 Byte
Value Type:	Integer
Required:	Mandatory
Content:	The duration of stay is encoded as specified in Table 2.

Passport Number

Tag:	`0x05`
Min. Length:	6 Byte
Max. Length:	6 Byte
Value Type:	Alphanumeric

Required: Mandatory
Content: The passport number of the passport of the applicant on which the visa sticker is attached.

Table 2: Encoding for the Duration of Stay

Integer Values of			Meaning
Byte 1	**Byte 2**	**Byte 3**	
0	0	0	The *valid-until* field of the MRZ denotes the last day on which the visa holder may stay in the country for which the visa was issued.
255	255	255	The *valid-until* field of the MRZ denotes the last day on which the visa holder may seek entry at the border for which the visa was issued. The duration of stay is determined by the authorities at the time of entry at the border.
number of days	number of month	number of years	The duration of stay is the sum of the number of days, the number of month, and the number of years, calculated from the time on which the visa holder enters the country for which the visa was issued. The *valid-until* field of the MRZ denotes the last day on which the visa-holder may seek entry. The triples (0,0,0) and (255,255,255), are reserved and, as seen above, MUST NOT be used in this case.

Visa Type
Tag: `0x06`
Min. Length: 1 Byte
Max. Length: 4 Byte
Value Type: Binary
Required: Optional
Content: The visa type is encoded as a binary sequence.

Additional Feature
Tag: `0x07`
Min. Length: 0 Byte
Max. Length: 254 Byte
Value Type: Binary
Required: Optional
Content: Reserved for future use by ICAO.

5.2 Cryptographic Signature, PKI, and Certificate Profiles

W.r.t. this visa profile, Visa Signer Certificates are issued in a way that allows verification by CSCA certificates. Recall that CSCAs were put in place to verify signatures of the data stored on eMRTDs. As a consequence, requirements applying to SubCA Certificates, Visa Signer Certificates and CRLs are aligned w.r.t. [8]. Note that the present specification defines additional requirements.

5.2.1 Visa Signer and Seal Creation

5.2.1.1 Architecture of the Visa Signer System

The Visa Signer receives data from a Visa Personalization System to encode a digital seal, and uses a signing key to sign it. *Figure 3: Visa Personalization* depicts one possible implementation of the Visa Signer and its client, the Visa Personalization System.

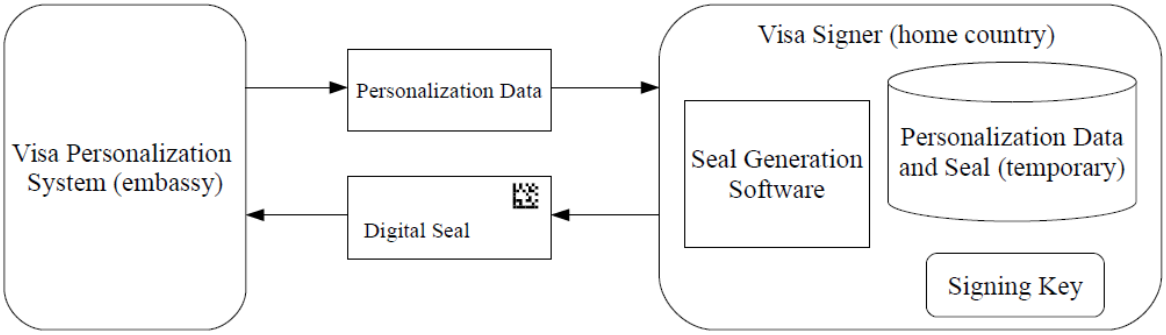

Figure 3: Visa Personalization: Scenario w/ centralized Visa Signer

The Visa Signer relies on the following software and data:

- The *seal generation software*, producing digital seals conforming to the present standard. It receives the personalization data sent by the client, signs these data with a private signing key, and encodes the personalization data and the signature to a barcode. The visa personalization data and the seal are the input and output data of the seal generation software, and must be stored temporarily in the Visa Signer during the generation of the seal.
- The *signature keys* (private and public key) to sign a digital seal. The private signing key is used by the seal generation software to sign the data of the seal. It is the most critical data of the Visa Signer.

Remark
Depending on the deployment scenario, the distinction between the visa personalization system and the visa signer is not always strict. For example, the visa-signer can be part of the personalization system at the embassy. One possible scenario is extending the personalization to include signature generation, and storing signing keys on a smartcard within the embassy. Another approach (depicted in *Figure 3: Visa Personalization*) is to set up a central visa signer in the home country, and let embassies connect to it via a secure channel. Last, some embassies might not personalize visas themselves; then the personalization system could be also set up at the home country and integrated with the visa signer.

#

The Visa Signer is a very critical component, as it produces the signature of the seal. The signature allows to verify the integrity of the data of the seal, i.e. whether the data have been manipulated, as well as their authenticity, i.e. whether they are issued by an authorized entity.
In order to achieve a sufficiently high security level, it is RECOMMENDED that the Visa Signer is a central service, and not deployed at the embassy, unless operational, technical, or logistical reasons prevent a centralized deployment. This is in order to concentrate the security measures on a limited perimeter, while taking into account best practices for ensuring recoverability and business continuity. Private signature keys shall be stored securely by the Visa Signer.

5.2.1.2 Security of the Visa Signing System

The Visa Signing System SHOULD be hosted and operated according to best security practices in the following areas: physical security, server and network infrastructure, system, development and support processes, access control, and operations security. If the visa-signer is set up as a central

service, it is RECOMMENDED to ensure compliance with ISO/IEC 27002 [10] on the perimeter of the Visa Signer in order to ensure compliance to these best security practices.

5.2.1.3 Management of Signature Keys

As mentioned, the signature keys are the most critical data of the Visa Signer and as such shall be protected according to the best security practices:

- It is RECOMMENDED to generate and confine signature keys in a cryptographic module or secure signature creation device according to one of [11,12,13].
- Access to the signature keys MUST be controlled, and signing authorized solely to the seal generation software.

The Visa Signer is a specific type of signature server used to sign a unique type of document, i.e. a visa. As such, following the best practices in the field, a limited number of signing keys (a lower one-digit number of keys, unless operational requirements make this difficult or impossible) can and SHALL be used in parallel to sign digital seals. To ensure the continuity of the activity of the Visa Signer in case of a security incident related to the key, measures to ensure business continuity (e.g. preparation of back-up keys, backup site, etc.) should be in place.

In order to facilitate the publication of the corresponding certificates (see Section 5.2.2.4), the number of signature keys MUST be limited to five signature keys per year per SubCA.

5.2.1.4 Key Requirements (Validity Period)

Validity periods are as follows:

CSCA Certificates (as specified in [8])

Private Key Usage Time:	3 to 5 years
Certificate Validity:	Private Key Usage Time + Max. of Key Lifetime (= Certificate Validity) of Document Signer Certificates, VDS Certificates and other LDS2 CA Certificates, or other SubCA certificates below the CSCA – whichever is longer

VDS Certificates (aka SubCA Certificates)

Private Key Usage Time:	2 to 5 years
Certificate Validity:	Private Key Usage Time + Key Lifetime (= Certificate Validity) of Visa-Signer Certificates

Visa-Signer Certificates

Private Key Usage Time:	1 to 2 years
Certificate Validity:	Private Key Usage Time + Visa Validity Timeframe

Example

Suppose visas with a validity period of 10 years are issued, and the private key usage time of the Visa-Signer Certificate is 2 years. Then validity of the Visa Signer Certificate is 2 + 10 = 12 years. Suppose further that the private key usage time of the VDS Certificate is 3 years. Then the validity of the VDS Certificate must be 3 + 12 = 15 years. If the usage time of the private key of the CSCA Certificate is 5 years, then the validity of the CSCA Certificate is 5 + 15 = 20 years.

#

Remark

It can be seen from the above example that due to the intermediate SubCA and the long validity time of the visa, the overall validity of the CSCA Certificate is quite long. While subject to the issuing state, it might be appropriate to consider to shorten the visa validity timeframe, i.e. to consider only visas with shorter validity periods – for example visas with validity up to about five years – with this

approach. Indeed, some countries already issue electronically enabled residence permit cards to long-term residents.

The used hash function, signature algorithm and (domain) parameters must be present in the `SubjectPublicKeyInfo` Extension of the Visa Signer Certificate (cf. Section 5.2.2.3).

5.2.2 Public Key Infrastructure (PKI) and Certificate Profiles

The Visa Signer Certificates shall be distributed using the CSCA-PKI [8]. A general outline of this procedure is depicted in Figure 1.

In the sectiosn below we use the following terminology for presence requirements of each of the components/extensions in certificates:

m	mandatory – the field MUST be present
x	do not use – the field MUST NOT be populated
o	optional – the field MAY be present

For the criticality of certificate extensions we use the following terminology:

c	critical – the extension is marked critical, receiving applications must be able to process this extension
nc	the extension is marked non-critical, receiving applications that do not understand this extension must ignore it

For detailed certificate profiles, see the sections below.

5.2.2.1 CSCA Certificate Profile

The CSCA Certificate profile is defined in [8]. Note that the profile is however under discussion, as defined in [17]: The main change here is that in order to allow SubCA's below the CSCA, the `PathLenConstraint` must be set to '1' for a CSCA that supports SubCA's - as in the case of digital seals. It must be set to '0', if there is no SubCA below the CSCA[3].

5.2.2.2 VDS Certificate Authority Profile

The SubCA certificate profile must comply with the CSCA certificate profile [8] in general. SubCAs serve a distinct role compared to CSCAs, and thus their profile deviates in some respects. In Table 3 we define the complete certificate body of the VDS SubCA certificate.

Table 3: VDS SubCA Certificate Profile: Certificate Body

Certificate Body	Presence	Remark
Certificate	m	
TBSCertificate	m	see below
signatureAlgorithm	m	dependent on selected algorithm
signatureValue	m	dependent on selected algorithm
TBSCertificate		

3 Note that the `PathLenConstraint` is also set to '1' for a CSCA-Link Certificate.

Certificate Body	Presence	Remark
version	m	2 (Version 3)
serialNumber	m	must be a positive integer of minimal length with a maximum of 20 Octets. Leading bit must be zero in DER encoding (cf. Appendix B in [8])
signature	m	value inserted here must be the same as that in signatureAlgorithm component of certificate sequence
issuer	m	must be the value of the CSCA Distinguished Name
validity	m	Must terminate with Zulu (Z). The seconds element must be present. Dates through 2049 must be in UTCTime, represented as YYMMDDHHMMSSZZ. Dates in 2050 and beyond must be in GeneralizedTime. GeneralizedTime must not have fractional seconds, and must be represented as YYYYMMDDHHMMSSZ. The validity (i.e. difference between notBefore and notAfter) must be according to Section 5.2.1.4
subject	m	the following two MUST be present; other attributes than that MUST NOT be present. commonName: MUST NOT exceed nine characters in length, printableString format countryName: must consist of the two letter country code [7] of the VDS SubCA, uppercase characters, printableString format
subjectPublicKeyInfo	m	must adhere to [16] and [8]
issuerUniqueID	x	
subjectUniqueID	x	
extensions	m	cf. next table and [8] on which extensions should be present. Default values for extensions must not be encoded.

VDS SubCA Certificates MUST contain the extension defined in Table 4. Any other certificate extension MUST NOT be present.

Table 4: SubCA Certificate Profile: Extensions

Extension Name	Presence	Criticality	Remark
AuthorityKeyIdentifier	m	nc	
keyIdentifier	m		
authorityCertIssuer	x		
authorityCertSerialNumber	x		
SubjectKeyIdentifier	m	nc	
keyIdentifier	m		
KeyUsage	m	c	
keyCertSign	m		
cRLSign	m		
All other values	x		
PrivateKeyUsagePeriod	m	nc	
notBefore	m		both notBefore and notAfter must be present. The validity period (notAfter – notBefore) must not exceed the validity period of the certificate
notAfter	m		
Basic Constraints	m	c	
cAm			
pathLenConstraint	m		Must always be 0
ExtKeyUsage	m	c	Analogous to LDS2.0 [17], the EKU extension for each Visa Signer must be populated as: (Paper) Visa Signer Certificate, OID 2.23.136.1.1.11.1 The VDS CA itself is not authorized to sign visas

5.2.2.3 Visa Signer Certificate Profile

The Visa Signer certificates must comply with the LDS2.0 Signer certificate profile (currently under development) in general. Since Visa Signer certificates serve a different role than LDS2.0 certificates, their profile deviates in some respects. In particular, the `subject DN` of the visa signer certificate contains an identifier, and the serial number is of special form; cf. Section 5.3. In Table 5, we list the complete certificate body of a Visa Signer certificate.

Table 5: Visa Signer Certificate Profile: Certificate Body

Certificate Body	Presence	Remark
Certificate	m	
TBSCertificate	m	see below
signatureAlgorithm	m	dependent on selected algorithm
signatureValue	m	dependent on selected algorithm
TBSCertificate		
version	m	2 (Version 3)
serialNumber	m	must be the positive integer that results from interpreting the *five character hex-string* that uniquely identify a Visa Signer Certificate for one CA as a positive integer. Leading bit must be zero in DER encoding (cf. Appendix B in [8]).
signature	m	value inserted here must be the same as that in the signatureAlgorithm component of certificate sequence
issuer	m	must be the value of the subject DN of the VDS CA certificate with which the Visa Signer certificate was signed.
validity	m	Must terminate with Zulu (Z). The seconds element must be present. Dates through 2049 must be in UTCTime, represented as YYMMDDHHMMSSZZ. Dates in 2050 and beyond must be in GeneralizedTime. GeneralizedTime must not have fractional seconds, and must be represented as YYYYMMDDHHMMSSZ. The validity (i.e. difference between notBefore and notAfter) must be according to Section 5.2.1.4.
subject	m	the following two MUST be present; other attributes MUST NOT be present. commonName: must consist of *two uppercase characters*, printableString format, that uniquely define the Visa Signer within one country, and must be present in the header of a VDS as the *Visa Signer reference*

Certificate Body	Presence	Remark	
		countryName:	must consist of the two letter country code [7] of the Visa Signer, uppercase characters, printableString format
subjectPublicKeyInfo	m	must adhere to [16] and [8]	
issuerUniqueID	x		
subjectUniqueID	x		
extensions	m	cf. next table and [8] on which extensions should be present. Default values for extensions must not be encoded.	

Extensions are depicted in Table 6. No other certificate extensions must be present.

Table 6: Visa Signer Certificate Profile: Extensions

Extension Name	Presence	Criticality	Remark
AuthorityKeyIdentifier	m	nc	
keyIdentifier	m		
authorityCertIssuer	o		
authorityCertSerialNumber	o		
ExtKeyUsage	m		Analogous to LDS2.0 [17], the EKU extension for each Visa Signer must be populated as: Visa Signer Certificate, OID 2.23.136.1.1.11.1

5.2.2.4 Certificate Publication

As the Visa Signer Certificates and VDS SubCA Certificates are not contained in the digital seal itself, the CA must publish its certificates. The distribution mechanism for Visa Signer Certificates is the issuance of MasterLists via web-download, and optionally additional other mechanisms. Hence, a country that issues visas protected with digital seals MUST publish a MasterList via web-download containing all its VDS SubCA Certificates and Visa Signer Certificates. One MasterList SHALL be issued for Visa-Signer Certificates, and one for VDS SubCA Certificates. For convenience, the CSCA Certificate SHOULD be published as well via web-download. Publication must adhere to the following principles:

1. As soon as a new certificate is created, it must be published with a delay of no more than 48 hours.
2. The certificates must remain published until their expiration.

5.2.2.5 Certificate Revocation List (CRL) and CRL Profile

Concerning the CRL of the Visa Signer Certificates and SubCA certificates issued by the CSCA and VDS SubCA, the following principles apply:

1. If a certificate has to be revoked, the corresponding CRL is renewed and published by the issuing CA within 48 hours.
2. In absence of any security incident, the issuing CA must renew the CRL at least every 90 days.

CRLs must comply with the CRL profile defined in [9,17] in general, which is listed in Table 7. Analogous to [17], the `issuerAltName` extension deviates here.

Table 7: CRL Profile: Basic fields

Certificate List Component	Presence	Remark
CertificateList	m	
tBSCertList	m	see below
signatureAlgorithm	m	dependent on selected algorithm
signatureValue	m	dependent on selected algorithm
tBSCertList		
version	m	1 (Version 2)
signature	m	value inserted here must be the same as that in signatureAlgorithm component of certificate sequence
issuer	m	must be the value of the subject DN of the VDS CA certificate
thisUpdate	m	Must terminate with Zulu (Z). The seconds element must be present. Dates through 2049 must be in UTCTime, represented as YYMMDDHHMMSSZZ. Dates in 2050 and beyond must be in GeneralizedTime. GeneralizedTime must not have fractional seconds, and must be represented as YYYYMMDDHHMMSSZ.
nextUpdate	m	Must terminate with Zulu (Z). The seconds element must be present. Dates through 2049 must be in UTCTime, represented as YYMMDDHHMMSSZZ. Dates in 2050 and beyond must be in GeneralizedTime. GeneralizedTime must not have fractional seconds, and must be represented as YYYYMMDDHHMMSSZ.

Certificate List Component	Presence	Remark
revokedCertificates	m	if present, must not be empty
crlExtensions	m	cf. next table and [8] on which extensions should be present. Default values for extensions must not be encoded.

Only extensions *deviating* from the CRL profile of [8] are depicted in Table 8. For the remaining extensions, the CRL profile of [8] MUST be adhered.

Table 8: CRL Profile: Extensions

Extension Name	Presence	Criticality	Remark
issuerAltName	m	nc	must be the same value as the `subjectKeyIdentifier` of the SubCA certificate

5.2.2.6 Certificate Generation

The certificate generation process consists of the following steps:

- The signature key pair of the Visa Signer is generated in a cryptographic module or a secure signature creation device.
- A certificate request is created by the Visa Signer. This request contains the public key of the signature key pair of the Visa Signer that should be certified, and is signed with the private key of the Visa Signer.
- This certificate request is sent to the VDS CA on a secure channel.
- The VDS CA verifies the signature of the certificate request, and creates a certificate corresponding to the public key of the Visa Signer.
- The VDS CA returns the certificate to the Visa Signer.
- The VDS CA publishes the certificate via MasterLists as described in previous sections.

5.2.2.7 Certificate Renewal

A new certificate must always contain a newly generated key pair. New certificates are regularly created when the corresponding signature keys reach the end of their signing validity period.

5.2.2.8 Certificate Revocation

A Visa Signer Certificate must be revoked in case of a security incident concerning the signature key. The certificate revocation of a Visa Signer Certificate is decided by the country that issues the visa. The revocation of a certificate is published in a CRL as described in section 5.2.2.5.

5.2.3 Visa Validation Authority

The Visa Validation Authority validates a digital seal by applying a Validation Policy. Section 5.3specifies validation criteria and algorithms to generate a validation status in detail.

Figure 4 illustrates the functional architecture of the Visa Validation Authority. The Visa Validation Authority relies on validation software which can be deployed on any computer used by the border control authorities.

The validation software is connected with a reader that takes an image of the visa to retrieve the barcode and the MRZ of the visa, and also, an image of the passport to retrieve the passport's MRZ. To verify the validity of the signature of the digital seal, the validation software SHOULD BE synchronized with the PKI publication point at least every 24 hours to retrieve the latest Visa Signer Certificates and CRLs.

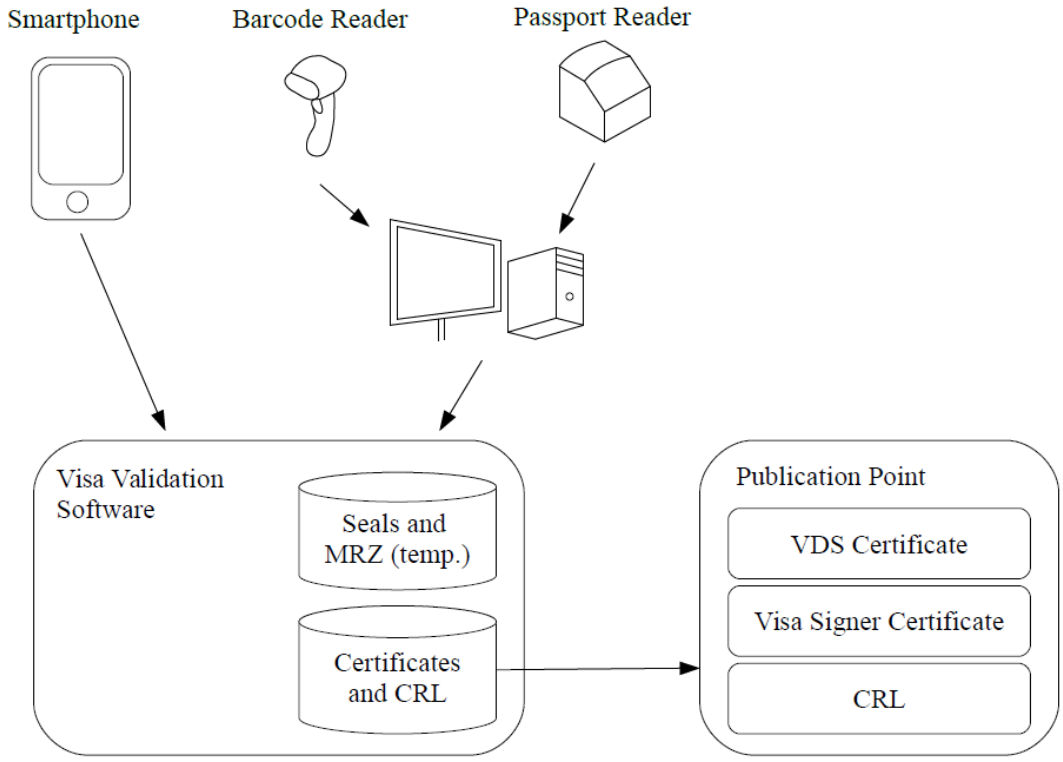

Figure 4: Visa Validation

The visa validation software decodes the digital seal and the MRZs of the visa and passport, validates the signature of the digital seal, and applies a validation policy to generate a validation status of the visa.

In mobile scenarios, the validation software can also be directly run on a smartphone. Whereas the validity of the seal can be verfied by the software on the smartphone, the comparison between the (signed) data insided the seal and the printed MRZs (of the visa and passport) must be done either manually, or by OCR of the MRZs out of the captured image, the latter being often a challenging problem in practice.

The following data are processed by the visa validation software:

- Input data provided by readers, i.e. the images of visas and passports
- Certificates and CRLs

5.3 Validation Policy (Informative)

5.3.1 Policy Rules

The Validation Policy is a set of validation rules that allow to determine the validity of the seal on the document. The application of this Validation Policy outputs a status indication with one of the following values:

1. *VALID*. The seal's authenticity and integrity has been confirmed. Here authenticity means that the data in the seal were indeed signed by a Visa Signer of the issuing country of the visa, and the corresponding Visa Signer Certificate is valid. Integrity means that the data of the MRZ of the visa were not modified, and the visa sticker was not swapped from the passport on which it was originally attached to.
2. *INVALID*. The seal is not valid, and further investigation is needed. Invalidity may occur due to the following three reasons:
    a) *Fraud/Forgery*. This includes unauthorized personalization of a visa based on a stolen blank sticker, changes of the personalization data of a visa based on an original sticker, or swapping a visa sticker from a stolen passport to another one, or other falsifications.
    b) *Damage/Tear*. The barcode cannot be decoded due to wear, tear or stains.
    c) *Unknown and/or Unexpected Errors*. This includes unpredictable errors, for example due to bugs in the software implementation used for decoding, or erroneous encoding during personalization.

Attached to the status indication INVALID are status sub-indications. These indicate the reasons for the invalidity of the seal. Since the chance of a fraud is dependent on these reasons, it is RECOMMENDED to map the status indications and sub-indications to the three trust levels “trustable”, “medium fraud potential”, and “high fraud potential”. The recommended mapping is illustrated in Table 9.

This Validation Policy considers the following questions:

1. Is the visible seal valid?
2. Is the MRZ of the visa valid?
3. Does the MRZ of the visa match with the visible seal?
4. Is the MRZ of the passport valid?
5. Does the MRZ of the passport match with the MRZ of the visa?

Below we give the validation rules for each type of control, list the validation criteria, expected results for each criteria, and resulting status sub-indications.

Visible Digital Seal Validation

1. Format Validation
    - if the physical encoding format is not compliant with the specification, or if errors due to physical noise cannot be corrected, the status is INVALID with sub-indication READ_ERROR
    - if the encoding format (i.e. the seal structures consisting of header, message zone and signature zone, or the binary/C40 encoding) is not compliant with the specification, or
    - if values expected in the header are unknown, or
    - if a mandatory field in the message zone is missing, or
    - if the format of a field in the message zone is not compliant with the specification of the version defined in the header, then the status is INVALID with sub-indication WRONG_FORMAT, otherwise continue.
    - if an unknown field is present in the message zone, then the sub-indication UKNOWN_FEATURE should be set. The status indication will be VALID or INVALID depending on the validity of the signature verified in the steps below. Note

that if the signature is valid, the presence of an unknown feature alone SHOULD NOT violate the validity of the seal however.

2. Signature Validation
    - if the Visa Signer Certificate referenced in the header of the seal or the SubCA-Certificate is not present, the status is INVALID with sub-indication UNKNOWN_CERTIFICATE.
    - if the Visa Signer Certificate referenced in the header of the seal was not signed by the SubCA, or the signature verification fails, the status is INVALID with sub-indication UNTRUSTED_CERTIFICATE
    - if the Visa Signer Certificate referenced in the header of the seal or the SubCA-Certificate are expired, the status is INVALID with sub-indication EXPIRED_CERTIFICATE
    - if the Visa Signer Certificate referenced in the header of the seal is revoked, the status is INVALID with sub-indication REVOKED_CERTIFICATE
    - if the signature verification of the header and message zone using the Visa Signer Certificate referenced in the header of the seal fails, the status is INVALID with sub-indication INVALID_SIGNATURE
    - otherwise continue

3. Issuer Validation
    - if the CSCA or the SubCA for visas below the CSCA is not trusted by the Visa Validation System on its trust domain, the status is INVALID with sub-indication UNTRUSTED_CERTIFICATE, otherwise continue.

4. Visa-MRZ Validation
    - if the checksums of the visa MRZ are not compliant with the applicable norm – dependent on the visa type – then the status is INVALID with sub-indication INVALID_VISA_MRZ
    - if there is a mismatch between a field of the Visa MRZ and the corresponding document feature stored within the seal, then the status is INVALID with SEAL_VISA_MISMATCH. Additional information on the mismatch SHOULD BE provided. Otherwise, continue.

5. Passport MRZ Validation
    - if the checksums of the passport MRZ are not compliant with the applicable norm – dependent on the passport type – then the status is INVALID with sub-indication INVALID_PASSPORT_MRZ. Otherwise continue.

6. Passport-link Validation
    - If any of the fields of the passport MRZ listed as follows do not correspond to their equivalent feature stored in the digital seal, then the status is INVALID with sub-indication SEAL_PASSPORT_MISMATCH. The MRZ fields of the passport are: 1.) passport number and 2.) passport issuing country. Otherwise if all fields match, the status of the Visible Seal is VALID.

The above validation rules cover a comparison of the data stored in the seal against data stored on the MRZ of the visa and the passport. On top of that, a manual inspection of those data that are stored in the seal and printed on the visa, but are not present in the MRZ of the visas, could be conducted.

Table 9: Recommended Trust Levels of the Visa Policy

Status Indication	Sub Status Indication	Trust Level
VALID	-	*trustable*
	UNKNOWN_FEATURE	
INVALID	READ_ERROR	*medium fraud potential*
	EXPIRED_CERTIFICATE	
	WRONG_FORMAT	*high fraud potential*
	UNKNOWN_CERTIFICATE	
	UNTRUSTED_CERTIFICATE	
	REVOKED_CERTIFICATE	
	INVALID_VISA_MRZ	
	SEAL_VISA_MISMATCH	
	INVALID_PASSPORT_MRZ	
	SEAL_PASSPORT_MISMATCH	

6. Worked Example (Visa Document)

The following example shows a visible digital seal that results from encoding the data shown in Table 11. To generate the signature, ECDSA-256 with the curve brainpoolP256r1 was used. The domain parameters of brainpoolP256r1 and the private key encoded as Base64 are:

```
-----BEGIN EC PARAMETERS-----
MIHgAgEBMCwGByqGSM49AQECIQCp+1fboe6pvD5mCpCdg41ybjv2I9UmICggE0gd
H25TdzBEBCB9Wgl1/CwwV+72dTBBev/n+4BVwSbcXGzpSktE8zC12QQgJtxcbOlK
S0TzMLXZu9d8v5WEFilc9+HOa8zcGP+MB7YEQQSL0q65y35XyyxLSC/8gbevud4n
4eO9I8I6RFO9ms4yYlR++DXD2sT9l/hGGhRhHcnCd0UTLe2OVFwdVMcvBGmXAiEA
qftX26Huqbw+ZgqQnYONcYw5eqO1Yab3kB4OgpdIVqcCAQE=
-----END EC PARAMETERS-----
-----BEGIN EC PRIVATE KEY-----
MIIBUQIBAQQgNN2C+Njrq+F9bmAQ5FEgW/GCdul78V+XgV9h+dMyw7eggeMwgeAC
AQEwLAYHKoZIzj0BAQIhAKn7V9uh7qm8PmYKkJ2DjXJuO/Yj1SYgKCATSB0fblN3
MEQEIH1aCXX8LDBX7vZ1MEF6/+f7gFXBJtxcbOlKS0TzMLXZBCAm3Fxs6UpLRPMw
tdm713y/lYQWKVz34c5rzNwY/4wHtgRBBIvSrrnLflfLLEtIL/yBt6+53ifh470j
wjpEU72azjJiVH74NcPaxP2X+EYaFGEdycJ3RRMt7Y5UXB1Uxy8EaZcCIQCp+1fb
oe6pvD5mCpCdg41xjDl6o7VhpveQHg6Cl0hWpwIBAaFEA0IABB1CQwfc2PkvPYKu
gQ3qA0tqEhzH0ox4M9cOq8ajzKotHG2jrwlIuHaemRad0qG1pltDHgZOC59HwI0P
yLNvXHc=
-----END EC PRIVATE KEY-----
```

Encoding input data yields a byte stream, which are both depicted in Table 11. Hashing the header (cf. Table 10) and message with SHA-256 and signing them with the above private key gave the following signature `(r,s)`:

```
r: 56BCBFEDFD2DC884247426A240A7068D32B37C6CE370AEEAB62B548B5FCC16FA
s: 6A098CA74CB22559435FD4DBDE709B45F6FC4C850DA421A6E75CD05A88707CBB
```

For the sake of completeness, the signature as DER encoded ASN.1:

```
3044022056bcbfedfd2dc884247426a240a7068d32b37c6ce370aeeab62b
548b5fcc16fa02206a098ca74cb22559435fd4dbde709b45f6fc4c850da4
21a6e75cd05a88707cbb
```

Table 10: Header of Example

Header Field	Content	Hex Dump
Magic Constant	0xDC	dc
Version	3dec	03
Issuing Country	UTO	d9c5
Certificate Authority and Certificate Reference	DE01FFAFF	6d15224c5a8c
Document Issue Date	25th of March, 2007	319f27
Signature Creation Date	26th of March, 2007	31c637
Document Feature Definition Reference	93dec	5d
Document Type Category	1dec	01

Table 11: Message Zone of Example

Message Field	Content	Hex Dump		
		Tag	Length	Value
MRZ-B	*Full First Line (36 chars):* VCD<<DENT<<ARTHUR< PHILIP<<<<<<<<<<< *Second Line (first 28 chars):* 1234567XY7GBR52031 16M2005250	02	2c	dd52134a74 da1347c6fe d95cb89f9f ce133c133c 133c133c20 3833734aaf 47f0c32f1a 1e20eb2625 393afe31
No. of entries	2	03	01	02
Duration of Stay	90 days	04	03	5a0000
Passport Number	ABC424242	05	06	59e932f926c7

The complete barcode content consists of the header, the message zone, bytes `0xFF` and `0x40` that denote the start and length of the signature, and the signature block `(r,s)`:

```
dc03d9c56d15224c5a8c319f2731c6375d01022cdd52134a74da1347c6fe
d95cb89f9fce133c133c133c133c203833734aaf47f0c32f1a1e20eb2625
393afe3103010204035a0000050659e932f926c7ff4056bcbfedfd2dc884
247426a240a7068d32b37c6ce370aeeab62b548b5fcc16fa6a098ca74cb2
2559435fd4dbde709b45f6fc4c850da421a6e75cd05a88707cbb
```

Suppose that `seal.bin` contains the header and message zone, i.e. contents of Table 10 and Table 11 (note that start and length of the signature zone 0xFF and 0x40 are excluded), that the signature is DER encoded in `sig.bin`, and the above PEM encoded private key in `priv_key.pem.` The signature can then be verified with `openssl` by:

```
openssl dgst -sha256 -prverify priv_key.pem -signature sig.bin -
sha256 seal.bin
```

7. References

[1] Directorate-General Home Affairs, Overview of Schengen Visa Statistics, 2012

[2] S. Bradner, RFC2119: Key words for use in RFCs to indicate Requirement Levels, 1997

[3] ICAO, Doc 9303: Part 4 - Specifications for Machine Readable Passports (MRPs) and other TD3 size MRTDs, 2015

[4] ICAO, Doc 9303: Part 7 - Machine Readable Visas, 2015

[5] ISO/IEC, ISO 3166-2 alpha-3 – three-letter country codes

[6] ISO/IEC, ISO/IEC 16022 Information technology -- Automatic identification and data capture techniques -- Data Matrix bar code symbology specification, 2006

[7] ISO/IEC, ISO 3166-1 alpha-2 – two-letter country codes

[8] ICAO, Doc 9303: Part 12 - Public Key Infrastructure for MRTDs, 2015

[9] - placeholder -

[10] ISO/IEC, ISO/IEC 27002: Information technology – Security techniques – Code of practice for information security management, 2013

[11] NIST, FIPS PUB 140-2: Security Requirements for Cryptographic Modules, 2002

[12] CEN, CEN/TC 224: prEN 14169-1 - Protection profiles for secure signature creation device — Part 2: Device with key generation, 2009

[13] Bundesamt für Sicherheit in der Informationstechnik (BSI), BSI-CC-PP-0045-2009: Cryptographic Modules, Security Level "Enhanced", 2009

[14] ANSI, ANSI X9.62-2005: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005

[15] ICAO, Doc 9303: Part 1 - Machine Readable Passports. Volume 2 - Specifications for Electronically Enabled Passports with Biometric Identification Capability, 2006

[16] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, RFC5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2008

[17] ICAO, LDS2.0 PKI Draft, 2015

[18] - placeholder -

[19] ISO/IEC, ISO/IEC 24778:2008: Information technology – Automatic identification and data capture techniques – Aztec Code bar code symbology specification, 2008

[20] ISO/IEC, ISO/IEC 18004:2006: Information technology – Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification, 2006

[21] ISO/IEC, ISO/IEC 15415:2011 : Information technology – Automatic identification and data capture techniques -- Bar code symbol print quality test specification – Two-dimensional symbols, 2011

Annex A Conversion of ECDSA Signature Formats (Informative)

Integer Encoding in DER/BER.

Integers are encoded according to both the Basic Encoding Rules (BER) and Distinguished Encoding Rules (DER) as the signed big endian encoding of minimal length, after which Tag-Length-Value (TLV) scheme is applied. We distinguish the following cases:

1. Suppose the integer value is positive, and the most significant bit (MSB) is zero in the minimal unsigned integer representation. Then the unsigned integer representation has the form below, which is the BER/DER value.

   `|0bbbbbbb| ...`

2. Suppose the integer value is positive, and the MSB is one in the minimal unsigned integer representation, i.e. has the form `|1bbbbbbb| ...` Then a byte containing zeros is put in front and the BER/DER value is

   `|00000000|1bbbbbbb| ...`

3. Suppose the integer value is negative. Then that value is encoded as the two's complement, for example by taking the unsigned minimal integer representation, inverting, and adding one. Afterwards the MSB is set to one. For example for -25357 we have the unsigned minimal integer representation

   `|0110 0011|0000 1101|`

   This is inverted to

   `|1001 1100|1111 0010|`

   One is added

   `|1001 1100|1111 0011|`

   and results in the BER/DER value. Note that the fact that the number is negative can be directly inferred by the fact that the MSB (here leftmost) is one.

Finally, one yields a TLV value by putting two bytes in front of the above encoded BER/DER values. The first byte is the tag with the constant `0x02`. The second byte contains the length (i.e. number of bytes) of the following encoded BER/DER value. Decoding can be simply done by e.g. distinguishing according to the MSB whether a negative or positive integer is encoded, and applying the above steps in reverse.

Example
Table 12 gives some examples of DER/BER encoded integers.

#

Table 12: DER/BER encoding examples for some integer values.

Value (dec)	Tag (hex)	Length (hex)	Value (hex)	Value (binary)
0	0x02	0x01	0x00	00000000
127	0x02	0x01	0x7F	01111111
128	0x02	0x02	0x00 0x80	00000000 10000000
-129	0x02	0x02	0xFF 0x7F	11111111 01111111

ECDSA signatures in ASN.1/DER
The ASN.1 description of an ECDSA signature is

```
Signature ::= SEQUENCE {
      r INTEGER, s INTEGER
}
```

This sequence is encoded according to DER as a TLV triple with tag `0x30`, the length as the number of bytes of the following value, and the value as the concatenation of the TLV triples of the encoding of `r` appended with the encoding of `s`.

Two example sequences – integers `r` and `s` of an ECDSA signature are of course much larger in practice – are given in Table 13.

Table 13: DER encoded sequences of two integers

Integers		TLV of Sequence		
R	**S**	**Tag**	**Length**	**Value**
127	1	0x30	0x06	0x02 0x01 0x7F 0x02 0x01 0x01
128	127	0x30	0x07	0x02 0x02 0x00 0x80 0x02 0x01 0x7F

Note that `r` and `s` are always positive integers for an ECDSA signature. Therefore to convert from a raw signature to DER, one has to first split the raw signature in half to get `r` and `s` individually, and then encode them as a DER encoded ASN.1 sequence according to the definition above. Conversely, to decode from an ECDSA signature in DER, one has to first decode the sequence, extract the unsigned integer representation of `r` and `s` and set both `r` and `s` to a fixed length (= length of key size) representation by stripping or adding leading zero bytes if required (e.g. both `r` and `s` must have a length of 256 bit = 32 byte in the case of ECDSA-256), and appending the value resulting from `s` to the value resulting from `r`.

Annex B C40 Encoding of Strings (Normative)

In order to save space in encoding alphanumeric characters and the filler symbol <, the encoding scheme C40 is used, as defined in [6]. In the following we define how these definitions are used in the current setting. The following two definitions apply for document features and their digital encoding:

1. Strings consist only of upper case letters, numbers, <SPACE>, and the symbol '<'. The latter is used as a filler symbol for the MRZ of travel documents. If '<' occurs in the string, all occurrences of '<' are replaced by <SPACE> before encoding. A string MUST NOT contain any other symbols.
2. Given a string of length L, the length (i.e. the number of bytes) of the corresponding digital encoding is the least even number, that is larger or equal to L.

In the following calculations, we implicitly convert between a byte value and the corresponding unsigned integer equivalent. For example we define the value of a byte by a formula consisting of integer arithmetic on integer values.

Encoding
Encoding a string of characters into a sequence of bytes works as follows: First, the string is grouped into tuples of three characters, and each character is replaced with the corresponding C40 value according to Table 16, resulting in a triple `(U1,U2,U3)`. Then for each triple, the value

```
U = (1600 * U1) + (40 * U2) + U3 + 1
```

is computed. The result is in the range from 1 to 64000, giving an unsigned 16 bit integer value. This 16 bit value `I16` is packed into two bytes by

```
Byte 1 = (I16) div 256

Byte 2 = (I16) mod 256
```

Here `div` denotes integer division (no remainder), and `mod` denotes the modulo operation. Note that these operations can be implemented by bit-shifting.

Decoding
The encoding can be easily inverted. Given a pair of bytes, let `(I1,I2)` denote their unsigned integer values. The 16 bit value `I16` is recalculated as

```
V16 = (I1 * 256) + I2
```

The triple `(U1,U2,U3)` can be recomputed by

```
U1 = (V16 – 1) div 1600

U2 = (V16 - (U1*1600) - 1) div 40

U3 = V16 - (U1*1600) – (U2*40) - 1
```

Here again, `div` denotes integer division. Characters can be decoded from the triple `(U1, U2, U3)` by simply looking up the corresponding values in Table 16.

Padding

The above definition is only well defined, if the length of the string to be encoded is a multiple of three. Akin to the padding-definitions given in [6], the following padding rules apply:

1. If two C40 (=two characters) values remain at the end of a string, these two C40 values are completed into a triple with the C40 value 0 (Shift 1). The triple is encoded as defined above.
2. If one C40 value (=one character) remains, then the first byte has the value 254dec (0xFE). The second byte is the value of the ASCII encoding scheme of DataMatrix of the character corresponding to the C40 value. Note that the ASCII encoding scheme in DataMatrix for an ASCII character in the range 0-127 is the ASCII character plus 1.

Example 1
Suppose the string “XK<CD” is to be encoded. By definition, all occurrences of '<' are replaced by <SPACE> before encoding. The resulting string is thus “XK CD”, i.e. “XK<SPACE>CD” (one space inserted). The C40 encoding/decoding of the string “XK<SPACE>CD” is depicted in Table 14.

#

Example 2
Suppose the “XKCD” is to be encoded. The string solely consists of uppercase letters. Its C40 encoding/decoding is depicted in Table 15.

#

Table 14: Encoding/Decoding example for the string “XK<SPACE>CD”.

Operation	**Result**			
original string	“XK<SPACE>CD”			
grouping into triples	(X, K, <SPACE>)		(C, D,)	
replacing with C40 values and padding	(37, 24, 3)		(16, 17, padding)	
calculating the 16 bit integer value	60164		26281	
	Byte 1 (div)	**Byte 2 (mod)**	**Byte 1 (div)**	**Byte 2 (mod)**
resulting byte sequence (decimal)	235	4	102	169
resulting byte sequence (hex)	0xEB	0x04	0x66	0xA9

Table 15: Encoding/Decoding example for the string "XKCD".

Operation	**Result**			
original string	“XKCD”			
grouping into triples	(X, K, C)		(D, ,)	
replacing with C40 values and padding	(37, 24, 16)		(unlatch C40 and encode in ASCII)	
calculating the 16 bit integer value	60177			
	Byte 1 (div)	**Byte 2 (mod)**	**Byte 1**	**Byte 2**

Operation	Result			
resulting byte sequence (decimal)	235	11	254	69
resulting byte sequence (hex)	0xEB	0x11	0xFE	0x45

Table 16: C40 Encoding chart and correspondence to ASCII.

C40 Value	Character	ASCII Value	C40 Value	Character	ASCII Value
0	Shift 1	n/a	20	G	71
1	Shift 2	n/a	21	H	72
2	Shift 3	n/a	22	I	73
3	<SPACE>	32	23	J	74
4	0	48	24	K	75
5	1	49	25	L	76
6	2	50	26	M	77
7	3	51	27	N	78
8	4	52	28	O	79
9	5	53	29	P	80
10	6	54	30	Q	81
11	7	55	31	R	82
12	8	56	32	S	83
13	9	57	33	T	84
14	A	65	34	U	85
15	B	66	35	V	86
16	C	67	36	W	87
17	D	68	37	X	88
18	E	69	38	Y	89
19	F	70	39	Z	90