

TLS NSAM Project

Application Definition Document for NSAM Applications

|  |  |
| --- | --- |
| Reference: | AE-NSAM-0030 |
| Version: |  |
| Classification: | Secret |

Authorisation

Approved by

Rob Macmillan, Aconite

Authorised by

Alberto De Nicoló, Trans Link Systems

Accepted by

Erik van Ooijen, Trans Link Systems

Document Control

| **Date** | **Version** | **Author** | **Comments** |
| --- | --- | --- | --- |
| 26 Feb 2014 | 0.1 | Steve Nichols | Replaces AE-NSAM-0012 with AE-NSAM-0030, reflecting general update to reflect new requirements for Engagement 4. |
| 17 Mar 2014 | 0.2 | Steve Nichols | * Updates to E4 requirements * Review comments |
| 8 May 2014 | 0.3 | Steve Nichols | Address TLS review comments, including:   * Insert test values for KABi and KTR\_0 keys * S2 CHV1 selected on basis of device type & equipment vendor * MSA device certificate end date supplied for both issuance and repersonalisation * Configuring Certificate for DL/CRL/AIL signature * Change card expiry report to MSA certificate expiry report * Insert device certificate processing |
| 11 June 2014 | 0.4 | Steve Nichols | Address TLS review comments |
| 17 June 2014 | 0.5 | Steve Nichols | Insert CHV2 test values |
| 11 July 2014 | 0.6 | Steve Nichols | Identify E5 functionality |
| 16 July 2014 | 0.7 | Steve Nichols | Identify TLS test ticketing master keys |
| 4 Aug 2014 | 1.0 | Steve Nichols | Formal Issue |
| 7 Aug 2014 | 1.1 | Steve Nichols | Add test data |
| 18 Aug 2014 | 1.2 | Steve Nichols | Add test key identifiers |
| 28 Aug 2014 | 1.3 | Steve Nichols | Clarify “Global Version” |
| 9 Sept 2014 | 1.4 | Steve Nichols | Test value of global version defined |
| 13 Sept 2014 | 1.5 | Steve Nichols | Correct key padding Insert data length into DE50 for S2 |

References

| **Reference** | **Document Title** | **Contact** |
| --- | --- | --- |
| [MSA Perso] | MSA NSAM Personalisation, version 1.0.8, 14/2/14 | TLS |
| [MSA Spec] | MSA NSAM Specification, version 3.1.1, 14/2/14 | TLS |
| [GUI] | NSAM Operator Interface, AE-NSAM-0032 | Aconite |
| [S9 SAM] | S9 SAM detailed interface specification, revision 3, 8/10/09 | TLS |
| [NSAM Perso] | AE-Data Prep interface and Perso Requirements, AE-NSAM-0031 | Aconite |
| [NKAM/NCRS IFS] | DIS NKAM NCRS, Version A rev 1+ Memo SDOA changes required for S9 TKF distribution 1.0 | TLS |
| [S2 App] | S2 application specification for NSAM, version 0.9.1, 28/3/14 | TLS |
| [Test Data] | File “Delivery Aconite 140807.zip”, as supplied by Martin Potappel 7/8/14 | TLS |

Contents

[1. Introduction 1](#_Toc396126577)

[1.1 Purpose of this Document 1](#_Toc396126578)

[1.2 Scope of Document 1](#_Toc396126579)

[1.3 Glossary 3](#_Toc396126580)

[1.4 Outstanding issues 3](#_Toc396126581)

[2. Permitted Platforms 4](#_Toc396126584)

[3. Test/Production Personalisation Data 5](#_Toc396126585)

[4. Application Specific Platform Related Parameters 6](#_Toc396126586)

[4.1 MSA Application platform related parameters 6](#_Toc396126587)

[4.2 S9 Application platform related parameters 6](#_Toc396126588)

[4.3 S2 Application platform related parameters 6](#_Toc396126589)

[4.4 Platform related personalisation data 7](#_Toc396126590)

[5. Shared Personalisation Data 8](#_Toc396126591)

[5.1 Product Level Personalisation Data 8](#_Toc396126592)

[5.2 Personalisation data provided with an NSAM request 13](#_Toc396126593)

[5.3 [E5] Personalisation data provided with a Device Certificate request 16](#_Toc396126594)

[6. MSA Personalisation Data 18](#_Toc396126595)

[6.1 Business Application Level Personalisation Data 18](#_Toc396126596)

[6.2 Application Profile Level Personalisation Data 18](#_Toc396126597)

[6.3 Derived personalisation data 18](#_Toc396126598)

[6.4 Application Datamap 26](#_Toc396126599)

[6.5 Retained Data 27](#_Toc396126600)

[6.6 MSA reissuance 27](#_Toc396126601)

[7. S9 Personalisation Data 29](#_Toc396126602)

[7.1 Business Application Level Personalisation Data 29](#_Toc396126603)

[7.2 Application Profile Level Personalisation Data 29](#_Toc396126604)

[7.3 Derived personalisation data 29](#_Toc396126605)

[7.4 Application Datamap 32](#_Toc396126606)

[7.5 Retained Data 33](#_Toc396126607)

[7.6 S9 Personalisation Differences 33](#_Toc396126608)

[8. S2 Personalisation Data 35](#_Toc396126609)

[8.1 Business Application Level Personalisation Data 35](#_Toc396126610)

[8.2 Application Profile Level Personalisation Data 35](#_Toc396126611)

[8.3 Derived personalisation data 35](#_Toc396126612)

[8.4 Application Datamap 37](#_Toc396126613)

[8.5 Retained Data 39](#_Toc396126614)

[8.6 S2 Personalisation Differences 39](#_Toc396126615)

[9. MSA TKF Personalisation Data 41](#_Toc396126616)

[9.1 Request Level Personalisation Data 41](#_Toc396126617)

[9.2 Product Level Personalisation Data 41](#_Toc396126618)

[9.3 Business Application Level Personalisation Data 41](#_Toc396126619)

[9.4 Application Profile Level Personalisation Data 41](#_Toc396126620)

[9.5 Personalisation data 42](#_Toc396126621)

[9.6 Retained Data 44](#_Toc396126622)

[9.7 Test Keys 44](#_Toc396126623)

[10. [E5] Device Certificate Personalisation Data 49](#_Toc396126624)

[10.1 Business Application Level Personalisation Data 49](#_Toc396126625)

[10.2 Application Profile Level Personalisation Data 49](#_Toc396126626)

[10.3 Derived personalisation data 49](#_Toc396126627)

[10.4 Application Datamap 49](#_Toc396126628)

[10.5 Retained Data 50](#_Toc396126629)

[11. Bureau Returns 51](#_Toc396126630)

[11.1 Bureau Return Processing 51](#_Toc396126631)

[11.2 [E5] Certificate Issuance Bureau 52](#_Toc396126632)

[12. Reports 53](#_Toc396126633)

[12.1 Blacklisted Applications 53](#_Toc396126634)

[12.2 MSA Expiry Report 54](#_Toc396126635)

[12.3 [E5] Device Certificate Expiry 55](#_Toc396126636)

[12.4 Update Status 55](#_Toc396126637)

[12.5 All Devices 56](#_Toc396126638)

[12.6 Device Order Report 56](#_Toc396126639)

# Introduction

## Purpose of this Document

This document identifies the information that is required in order that Affina application builders can be configured and then used to deliver appropriate data personalised for the selected Device Type.

Where the Device Type represents an NSAM, the personalisation data will be for MSA and S2 or S9 applications, suitable for writing to a SmartMX contact smart card, including TKF data for the MSA application.

[E5] Where the Device Type represents a certificate, the personalisation data will be a device certificate, suitable for writing to a file.

Additionally, this document defines NSAM specific reports that Affina needs to provide in support of this project. Such reports would not normally be within the scope of a document such as this, but they are included here as they are reporting on the data used in application personalisation.

This document has been prepared for TLS on the basis of discussions and information made available, with the main source documents being [MSA Perso], [MSA Spec], [S2 APP] and [S9 SAM]. Once this document has been approved, it shall form a set of instructions on how Affina is to personalise the MSA, S2 and S9 applications and also how to create device certificates. At the point of approval, responsibility for the contents of this document will pass to TLS.

Following document approval, it is expected that this document will, as appropriate, be maintained to cover any changes to existing Affina products and extended to cover any additional Affina products. Note that such work is beyond the scope of the current Aconite contract.

## Scope of Document

The main scope of this document is to identify the functionality that is required for the project phase “Engagement 4”. However, discussions also lead to requirements for “Engagement 5” being captured and, whilst not currently contacted, these are also documented. For clarity, E5 requirements are identified as [E5] and are coloured blue.

This document defines the personalisation of the MSA, S2, S9 and [E5] Cert applications in terms of data that is held within Affina and in terms of data that is obtained by or generated by Affina. For the data that is held within Affina, it is further categorised by the configuration level (product, business application or application profile) at which it is stored. The following diagram depicts the configuration objects (products, business applications and application profiles) assumed within this document and the relationship between each of the configuration objects.



Data that is shared between MSA, S2, S9 and Cert applications is specified at product level within this document.

The MSA application is mandatory for all NSAMs and the MSA configuration specified here will provide the personalisation data for this application.

The S9 application is optional for all NSAMs, but may not be present if the S2 application is loaded. S9 SAMs may also be imported into the system, but containing only the S9 application.

The S2 application is optional for all NSAMs, but may not be present if the S9 application is loaded. S2 SAMs may also be imported into the system, but containing only the S2 application.

The TKF application will be configured to provide ticketing key data to the MSA application via post-issuance channels. The TKF application will not be involved with the issuance process.

[E5] The Cert application only exists for device certificate requests and therefore will only be requested on its own (i.e. it cannot exist on an NSAM and NSAM applications cannot be associated with a certificate request). Ideally, the Cert application would exist as a separate product. However, because device identifiers are used within both NSAMs and device certificates, everything has been kept within one product.

This document does not address the details of the information to be supplied in the input request (as supplied from the requesting GUI) nor the supply of other personalisation data, such as the NSAM print data.

## Glossary

| Term | Meaning |
| --- | --- |
| AE | Affina Enterprise |
| AID | Application Identifier |
| APDU | Application Protocol Data Unit |
| BE | Business Entity |
| CA | Certification Authority |
| CRL | Certificate Revocation List |
| CRT | Chinese Remainder Theorem |
| CVK | Ceiling Value Key |
| DER | Distinguished Encoding Rule |
| E4 | Engagement 4; the NKAM development phase that this document pertains to. |
| E5 | Engagement 5; A future NKAM development phase |
| GUI | Graphical User Interface |
| ISIN | Issuer Specific Identification Number |
| KLK | Key Loading Key |
| KTR | Key Transfer Reader |
| MSA | Migration Security Architecture |
| NCRS | New Ceiling Reload Server |
| NSAM | New Secure Access Module |
| TKF | Ticketing Key File |
| QCM | Query Card Manifest |
| SAM | Secure Access Module |
| SM | Secure Modules CA |
| TLA | Three Letter Acronym (or trigram) |
| TLS | Trans Link Systems |
| WS | Web Services CA |

## Outstanding issues

* S2 Application EEPROM occupancy

# Permitted Platforms

The personalisation specified in this document is applicable to the following platforms

|  |  |
| --- | --- |
| **Supplier** | NXP |
| **Brand** | SmartMX |
| **Mask descriptor** | NX011D |
| **Mask Spec.** | TBD |
| **Preloaded Applets (ROM)** | None |
| **Crypto Facilities** | TBD |
| **Available EEPROM** | 77,732 bytes |

# Test/Production Personalisation Data

With the exception of the configured cryptographic related data (such as master keys), the test and production personalisation requirements are exactly the same. The implications of this are:

* Both test and production systems will have the same fixed personalisation data, unless otherwise identified.
* Both test and production systems will have unique device IDs within their own environment, but device IDs will be duplicated between the two systems (and each system will have its own device lists and CRLs)
* Both systems will be configured with the same master keys (in functional terms), but those master keys will have different values.
* Both systems will be configured with the same cryptographic data such as certificates (in functional terms), but these data items will have different values. All such data items are indicated within this document.
* The different CAs may produce certificates with the same serial numbers.

# Application Specific Platform Related Parameters

The following are the application specific parameters that are required by the platform.

|  |  |
| --- | --- |
| **Protocol** | SCP02 |

## MSA Application platform related parameters

|  |  |
| --- | --- |
| **MSA Application Size in EEPROM** | 49,192 bytes |
| **RAM Required** | TBD |
| **Load File AID** | D528005129000001020000 |
| **Module AID** | D528005129010001020000 |
| **Application AID** | D528005129010001020000 |

## S9 Application platform related parameters

|  |  |
| --- | --- |
| **S9 Application Size in EEPROM** | 21,316 bytes |
| **RAM Required** | TBD |
| **Load File AID** | D5280051290000020101 |
| **Module AID** | D5280051290100020101 |
| **Application AID** | D5280051290100020101 |

## S2 Application platform related parameters

|  |  |
| --- | --- |
| **S2 Application Size in EEPROM** | TBD bytes |
| **RAM Required** | TBD |
| **Load File AID** | D528005129000004010100 |
| **Module AID** | D528005129010004010100 |
| **Application AID** | D528005129010004010100 |

## Platform related personalisation data

### Card Manager Keys

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| NSAM ISD Encryption key (ISDS-ENC) | 16 Bytes | Derived |
| NSAM ISD Message Authentication Code key (ISDS-MAC) | 16 Bytes | Derived |
| NSAM ISD Data Encryption key (ISDDEK) | 16 Bytes | Derived |

**Description:**

The three NSAM specific card manager keys (ISDS-ENC, ISDS-MAC and ISDDEK) are derived from the same 24 byte master key (MKNISD) with different diversification data. The diversification data includes data obtained from the chip and therefore cannot be created during the data generation phase for issuance. As such the generation of these keys is detailed in [NSAM Perso].

For post-issuance, the diversification data is known (as it is provided in the bureau return data) and thus Affina shall be able to derive the NSAM specific card manager keys. See [NSAM Perso] for details of the derivation process.

Multiple ISD master keys can be configured in Affina and the correct ISD master key shall also be selected from data provided in the bureau return information supplied when the NSAM was created. The bureau return information used to select the ISD master key shall be the key ID (GP.CMKName in the bureau return) and the generation (GP.CMKKeySetVersion in the bureau return)

For **test** purposes, the configured master keys shall be:

| **Item** | **Key Id** | **generation** | **Value (in clear text)** |
| --- | --- | --- | --- |
| MKNISD (Test) | tstmksamisdtls001 | 0x01 | 'E0E0E3E3E5E5E6E6E0E3E3E5E5E6E6E9E3E3E5E5E6E6E9E9' |
| MKNISD (production) | prdmksamisdtls001 | 0x01 | 'E0E0E5E5E5E5E6E6E0E3E3E6E6E6E6E9E3E3E5E5E6E6E9E9' |

### SCP02 Sequence Counter

When Affina creates APDUs for application updates (as delivered via the NCRS interface), these APDUs are encrypted using the chip’s secure channel, the keys for which are derived using the chip’s current sequence counter (with other data).

The card’s bureau return data provides, in hexadecimal, the next Secure Channel Sequence Counter for the card (GP.SCP02SequenceCounter).

# Shared Personalisation Data

This section defines all of the personalisation parameters that are shared between MSA, S2 and S9 applications (i.e. the same parameter will be used to personalise all applications).

Note that there are several cryptograms defined within this document section. Affina does not hold the keys to decrypt these cryptograms and will just treat them as uninterpreted values.

## Product Level Personalisation Data

This section identifies data that is configured at the product level and is therefore available as personalisation data to S9, S2 and MSA applications.

### PIN

Test Configuration:

| **Parameter name** | **Format** | **Product Value** |
| --- | --- | --- |
| PIN | 8 Bytes | 0x2C219628906076FF |

### S9 Keys

The S9 keys data set will be an unclassified security data set that references the following data:

| **Parameter name** | **Format** | **Comment** |
| --- | --- | --- |
| TLSK9 | Key reference | 3DES key (16 bytes) |
| KUpload | Key reference | 3DES key (16 bytes) |
| KDownload | Key reference | 3DES key (16 bytes) |

Note that the following are the test key values in clear text

| **Parameter name** | **Key Id** | **Value** |
| --- | --- | --- |
| TLSK9 | tstiks9yke9tls001 | 0x61 85 46 1F 91 D3 CE 97 DA 32 5E C1 DF 23 98 25 |
| KUpload | tstiks9yupltls001 | 0x83 62 07 F7 49 CD 1C 4C 4C 94 E6 61 91 0D 4A DC |
| KDownload | tstiks9ydwntls001 | 0x6B 79 97 F1 68 FD E0 B0 58 0E 57 34 C8 CB 70 19 |

### S2 CHV1 data set

The S2 CHV1 data set will be classified security data set that references the following data:

| **Parameter name** | **Format** |
| --- | --- |
| S2 CHV1 | 8 Bytes |

Each data set instance shall be indexed (or classified) via a three hexadecimal characters (“000” – “FFF”), where the first character represents the Equipment Vendor Identifier (“0” – “F”) and characters two to three represent the device type (“00” – “FF”).

If an S2 CHV1 is required (it’s only used by the S2 application), the appropriate S2 CHV1 value shall be selected using the Device Type (see 5.2.2) and the Equipment Vendor (see 5.2.8) from the request. If there is no S2 CHV1 value corresponding to the requested Device Type and Equipment Vendor, the request shall be rejected.

Note that the following are the test values to be configured:

| **EV Id** | **Device Type** | **S2 CHV1 Value** |
| --- | --- | --- |
| 0 | 00 | EA8041E549A332AE |
| 0 | 01 | 8C2A75640C9B824C |
| 0 | 02 | 0664F1F25DE7B571 |
| 0 | 03 | A9BE7A2C3D72E8D8 |
| 0 | 04 | CE2D15EF462415C8 |
| 0 | 05 | 6137D526BE142286 |
| 0 | 06 | 638D8AA00556F788 |
| 0 | 07 | 8E98AA0E174226C3 |
| 0 | 08 | 1EB15C29F37F577F |
| 0 | 09 | D5029CF74A0737A9 |
| 0 | 0A | 2A61255466F5984E |
| 0 | 0B | AF45DA64863B9B09 |
| 0 | 0C | 0C2A8887DC0D3FDE |
| 0 | 0D | F436613E82E5F01C |
| 0 | 0E | 6295AEBF309E2283 |
| 0 | 0F | 33533DA3B1AF224C |
| 0 | 10 | DF43E8062827C887 |
| 0 | 11 | D087405BF3D31010 |
| 0 | 12 | 57F7B4D593E8CCA0 |
| 0 | 13 | 9E8F7F5AF8F83F0B |
| 0 | 14 | D86FE56820DC0DD4 |
| 0 | 15 | 9A2DF424372EBA0D |
| 0 | 16 | D6028D7E8E23EDB0 |
| 0 | 17 | 00201930D1B625AB |
| 0 | 18 | 74D17823B58794BC |
| 0 | 19 | 8BE22B60CCC36982 |
| 0 | 1A | CE515F6B1066D216 |
| 0 | 1B | 11B86975CDB09521 |
| 0 | 1C | 9A3E0A4CEB95FB91 |
| 0 | 1D | 06D18109EBE89DD0 |
| 0 | 1E | F5DE5CBC85F5DD02 |
| 0 | 1F | D08C33CBBE54D63E |
| 0 | 20 | 7953AFC14EE95AED |
| 0 | 21 | 86A5CF238F040249 |
| 0 | 22 | 9DA8C7DE91D60CB4 |
| 0 | 23 | DD88716FCE9D53ED |
| 0 | 24 | 62B993FEB59D2896 |
| 0 | 25 | 12E5B80B81B079A6 |
| 0 | 26 | 784094CE9B77106B |
| 0 | F6 | 9246B1E925CF6DE4 |
| 0 | F7 | 6C782F56F0EA168A |
| 0 | F8 | 298A1908723E33E7 |
| 0 | F9 | 18B76DB0A906A7F7 |
| 0 | FA | 8BB86A9A76C47831 |
| 0 | FB | 625EBEE5451D3D52 |
| 0 | FC | 1925EC3C0E6DDBD6 |
| 0 | FF | 7A433C758A786468 |
| 1 | FD | CB53CD20B2F222D9 |
| 1 | FE | 5ACDCBA8F3488C6A |

### Legacy TKC data set

The Legacy TKC data set will be an unclassified security data set that references the following data:

| **Parameter name** | **Format** | **Comment** |
| --- | --- | --- |
| S2 Ticketing Key Cryptogram | 4392 Bytes | Also referred to as [[Ktick]KTRn]KEK |
| S9 Ticketing Key Cryptogram | 4392 Bytes | Also referred to as [Ktick]TLSK9 |
| Global Version | 1 Byte | The version of the S2 and S9 Ticketing Key Cryptograms |

Note that the following are the test values:

| **Parameter** | **Version** | **Value** |
| --- | --- | --- |
| S2 Ticketing Key Cryptogram | 0x01 | [Test Data] TOK050814\_2.V01 |
| S9 Ticketing Key Cryptogram | As above | [Test Data] S9TKF050814\_2.V01 |

### S2 Key data set

The S2 Key data set will be classified security data set that references the following data:

| **Parameter name** | **Format** | **Comment** |
| --- | --- | --- |
| S2 R/W Module KTRn Cryptogram | 24 Bytes | Also referred to as [[KTRn]KTR0\_rv]KEK |
| S2 KAB0 Key | Key reference | 3DES key (16 bytes) |

Each data set instance shall be indexed (or classified) via a single hexadecimal character (“0” – “F”), representing the Reader Vendor Identifier.

If an S2 Key data set is required (it’s only used by the S2 application, with the S2 R/W Module KTRn Cryptogram required by the L2/3 S2 profile and the KAB0 Key required by the L1 S2 profile), the appropriate S2 Key data set shall be selected using the Reader Vendor from the request (see 5.2.9). If there is no data set corresponding to the requested Reader Vendor, the request shall be rejected.

Note that the following are the test values:

| **Reader Vendor Id** | **S2 R/W Module KTRn Cryptogram Value** | **KAB0 Key Id** | **KAB0 Key Value** |
| --- | --- | --- | --- |
| 0 | [Test Data] KTRn\_KTR0\_0050814\_2.V00 | tstmks2ykabtha000 | BCB932CD76B0B3151A6819617FFB231C |
| 1 | [Test Data] KTRn\_KTR0\_1050814\_2.V00 | tstmks2ykabsig001 | 7C9E9ECD349D0EAEDA2AA1989E26F438 |
| 2 | [Test Data] KTRn\_KTR0\_2050814\_2.V00 | tstmks2ykab002002 | A826FECBEA2A5DAD9DBA0BCDDA381343 |
| 3 | [Test Data] KTRn\_KTR0\_3050814\_2.V00 | tstmks2ykab003003 | DFF8BF0E7C8FF7467A15BF513E7301F8 |
| 4 | [Test Data] KTRn\_KTR0\_4050814\_2.V00 | tstmks2ykab004004 | 5D98EF344F8A3162D6324FC8A898DFE6 |
| 5 | [Test Data] KTRn\_KTR0\_5050814\_2.V00 | tstmks2ykab005005 | 4F2A86E0108F43A16BDF2537E30B0B43 |
| 6 | [Test Data] KTRn\_KTR0\_6050814\_2.V00 | tstmks2ykab006006 | 012073CD83D35E9449D08673C8CD0DBF |
| 7 | [Test Data] KTRn\_KTR0\_7050814\_2.V00 | tstmks2ykab007007 | 1FE3986B8570DCB073BC5D38FD8334EC |
| 8 | [Test Data] KTRn\_KTR0\_8050814\_2.V00 | tstmks2ykab008008 | F78C6E26B9E01C701C0B0D4345C2404F |
| 9 | [Test Data] KTRn\_KTR0\_9050814\_2.V00 | tstmks2ykab009009 | 0B0B68F823B3257604DAD0754C6757CD |
| A | [Test Data] KTRn\_KTR0\_A050814\_2.V00 | tstmks2ykab010010 | 524537703D6D4F10EFD3F89EA7D6BCAE |
| B | [Test Data] KTRn\_KTR0\_B050814\_2.V00 | tstmks2ykab011011 | 7980BC8698B0C1868334B0FB1007E03E |
| C | [Test Data] KTRn\_KTR0\_C050814\_2.V00 | tstmks2ykab012012 | F8A4ADDF852F4662088F2C6D3D372013 |
| D | [Test Data] KTRn\_KTR0\_D050814\_2.V00 | tstmks2ykab013013 | 37FD5B464A51AEF215852976E9467379 |
| E | [Test Data] KTRn\_KTR0\_E050814\_2.V00 | tstmks2ykab014014 | F29EB91908977F64BAFDEF98F8A8FEA8 |
| F | [Test Data] KTRn\_KTR0\_F050814\_2.V00 | tstmks2ykabtn1015 | 97B6B0F7F143914CF4D0DC64BF0E46AE |

### S2 Infrastructure data set

The S2 Infrastructure data set will be an unclassified security data set that references the following data:

| **Parameter name** | **Format** | **Comment** |
| --- | --- | --- |
| S2 KEK Key | Key reference | 3DES key (16 bytes) |
| S2 SAM Auth Key | Key reference | DES Key (8 bytes) |
| S2 KTRn Cryptogram for all R/W modules | 387 Bytes | Also referred to as [[KTRn]KTR0\_rv]KEK |
| S2 KABi Key | Key reference | 3DES key (16 bytes) |
| S2 R/W Module KABi Cryptogram | 24 Bytes | Also referred to as [[KABi]KTRn]KEK |
| S2 L1 SAM KABi Cryptogram | 16 bytes | Also referred to as [KABi]KEK |
| S2 Combined KABi Cryptogram | 43 bytes | Also referred to as “[[KABi]KTRn]KEK and [KABi]KEK” |

Note that the following are the test key values:

| **Parameter name** | **Key Id** | **Value** |
| --- | --- | --- |
| S2 KEK Key | tstmks2ykektls255 | [Test Data] tstmks2ykektls255\_zmk\_kcv\_5A709E59.bin |
| S2 SAM Auth Key | tstiks2yde2tls255 | [Test Data] tstiks2yde2tls255\_zmk\_kcv\_72234ACE.bin |
| S2 KABi Key | tstmks2ykbitls000 | [Test Data] tstmks2ykbitls000\_zmk\_kcv\_F5BFE6A5.bin |

Note that the following are the test data values:

| **Parameter** | **Value** |
| --- | --- |
| S2 KTRn Cryptogram for all R/W modules | [Test Data] KTR050814\_2.hex |
| S2 R/W Module KABi Cryptogram | [Test Data] KABi\_RW.bin |
| S2 L1 SAM KABi Cryptogram | [Test Data] KABi\_L1.bin |
| S2 Combined KABi Cryptogram | [Test Data] KAB050814\_2.V00 |

### [E5] NKAM Device Certificate and Key

The NKAM device certificate and its corresponding private key shall be configured via an unclassified security data set and then used for signature of legacy device lists, legacy CRLs and AILs.

The method by which this data is created will be a largely manual process, that is expected to be as follows:

1. The key management facility is used to generate an RSA key and the private component is exported from the key management facility (This is the key that is configured in the security data set)
2. The key management facility is used to create a certificate request (using PKCS#10 format), utilizing the key that has been generated
3. The NKAM is used to generate a Device ID (for the CAU device type)
4. The NKAM SM CA is used to generate the NKAM device certificate (as a certificate in the SM subordinate chain), utilizing the certificate request and the Device ID (This is the certificate that is configured in the security data set).

## Personalisation data provided with an NSAM request

This section identifies data that is supplied in a request for an NSAM, and is therefore available as personalisation data to S2, S9 and MSA applications. This does not imply that the applications will use the data but merely that it is available should it be required. The actual usage of the request supplied data is defined in sections 6, 7 and 8.

### ISIN

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| ISIN | 4 Bytes | From request |

**Source:** BatchRequestFile.CardPersonalisationData.ISIN

**Description:**

The request contains a 4 byte ISIN (which itself is made up from a 1 byte instance version, a 1 byte device type followed by a 2 byte serial number).

### DeviceID

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| DeviceID | 3 Bytes | From request |

**Source:** BatchRequestFile.CardPersonalisationData.DeviceID

**Description:**

The request contains a 3 byte DeviceID (which itself is made up from a 1 byte device type followed by a 2 byte serial number).

See [GUI] for further definition of the input DeviceID.

### Key Type

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| KeyType | 1 Byte | From request |

**Source:** BatchRequestFile.CardPersonalisationData.KeyType

**Description:**

The request contains a 1 byte key type, as follows:

|  |  |
| --- | --- |
| **Key Type** | **Value** |
| Personalisation | 01 |
| Service | 02 |
| Validation | 03 |
| Enquiry | 04 |

### Business Entity

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| BEIdentifier | 3 Bytes | From request |
| OperatorId | 4 Bytes | From request |

**Source:** BatchRequestFile.CardPersonalisationData.BE

**Description:**

The request contains a 2 byte Business Entity, with a value in the range 0x0000 to 0xFFFF.

See [GUI] for further definition of BE.

The 3 byte BEIdentifier is formed by prefixing the received BE with a single byte set to 0x00

The 4 byte OperatorId is formed by prefixing the received BE with a double byte set to 0x0000

### BE Abbreviation

**Source:** BatchRequestFile.CardPersonalisationData.BETLA

**Description:**

The request contains an acronym for the BE (e.g “CXX” for Connexxion).

This acronym is used within the S9, S2 and MSA device certificates.

### AE Instance

**Source:** BatchRequestFile.CardPersonalisationData.Instance

**Description:**

The request contains an acronym for the AE instance (and thus the CA instance) that was the source of the request (e.g “TST” for Test).

This acronym is used within the MSA certificates.

### NCRS

**Source:** BatchRequestFile.CardPersonalisationData.NCRS

**Description:**

The NCRS that the NSAM is associated with at issuance. The NCRS value is not used in application personalisation, but is stored in the NSAM database record for later use.

If an NCRS hasn’t been allocated, this field will contain the value “None”.

It should be stored as part of the MSA application data and should be configured to be modifiable via the Affina/ESP interface.

### Equipment Vendor

**Source:** BatchRequestFile.CardPersonalisationData.EquipmentVendorId

**Description:**

The equipment vendor that the NSAM is associated with at issuance. The equipment vendor value is not used in application personalisation, but is stored in the NSAM database record for later use.

### Reader Vendor

**Source:** BatchRequestFile.CardPersonalisationData.ReaderVendorId

**Description:**

The reader vendor that the NSAM is associated with at issuance. The reader vendor is used by the S2 application for data selection and is also stored in the NSAM database record for later use.

### Order Number

**Source:** BatchRequestFile.CardPersonalisationData.Order Number

**Description:**

The order number associated with the NSAM. The order number value is not used in application personalisation, but is stored in the NSAM database record for later use with reporting.

### Initialisation Year

**Source:** BatchRequestFile.BusinessApplication Id.PersonalisationData.InitYear

**Description:**

The request will contain a 4 digit initialisation year for use in both S2 and S9 device certificate personalisation.

### S2 Profile

**Source:** BatchRequestFile.BusinessApplication Id.PersonalisationData.S2 Profile

**Description:**

A request for an S2 application will contain a 2 to 4 character identification of the type of profile that is to be personalised, the valid profiles being: “L1”,”L2/3”, “L4” or “CRS”

### Device Abbreviation

**Source:** BatchRequestFile.BusinessApplication Id.PersonalisationData.DeviceTLA

**Description:**

A request for an S2 application will contain a 3 letter acronym for the Device (e.g “GAT” for Gate).

### [E5] MSA device certificate expiry date

**Source:** BatchRequestFile.BusinessApplication Id.PersonalisationData.MSAExpriy

**Description:**

The expiry date for the MSA device certificate in the form DDMMYYYY.

## [E5] Personalisation data provided with a Device Certificate request

This section identifies data that is supplied in a request for a device certificate, and is therefore available as personalisation data to the Cert application. The actual usage of the request supplied data is defined in section 10.

### ISIN

See Section

Note that the ISIN is not used in personalisation, but is stored for later use in the AIL and device list.

Further note that the ISIN is constructed from one byte 0x01 concatenated with the device ID. The first byte will not change as it can with the MSA application.

### DeviceID

See Section

### Business Entity

See Section

### BE Abbreviation

See Section

### AE Instance

See Section

### NCRS

See Section

Note that this field will always contain the value “none”.

### Certificate Expiry

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| CertExpiry | String in form DDMMYYYY | From request |

**Source:** BatchRequestFile.CardPersonalisationData.CertExpiry

**Description:**

The expiry date for the device certificate in the form DDMMYYYY.

### CA

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| CA | 2 character string | From request |

**Source:** BatchRequestFile.CardPersonalisationData.CA

**Description:**

Identifies which CA is to be used to obtain the device certificate:

* WS – use the Web Services CA
* SM – use the Secure Module CA (i.e. the CA used for MSA device certificates)

### PublicKey

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| PublicKey | Multiple bytes | From request |

**Source:** BatchRequestFile.CardPersonalisationData.PublicKey

**Description:**

* Provides an RSA public key as a PKCS#1 RSAPublicKey wrapped as an X.509 PublicKeyInfo and DER-encoded

# MSA Personalisation Data

This section defines all of the personalisation parameters that are specific to the MSA application.

This initial section discusses how the personalisation parameters are used when initially creating an MSA application, whereas later sections discuss personalisation differences during MSA application reissuance.

## Business Application Level Personalisation Data

There is no personalisation data specified at the business application level.

## Application Profile Level Personalisation Data

There is no personalisation data specified at the application profile level.

## Derived personalisation data

### Key Loading Keys

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| KeyGeneration | 1 Byte | From security data set |
| MSA KLK Authentication key | 24 Bytes | Derived  Key Loading Key Authenticity NSAM Key (NKKLK,a) |
| MSA KLK Confidentiality key | 24 Bytes | Derived  Key Loading Key Confidentiality NSAM Key (NKKLK,c) |

**Description:**

The master keys for Key loading keys are generated in pairs (i.e. each pair consists of 24 byte Authentication key loading key master key (MKKLK\_a) and a 24 byte Confidentiality key loading key master key (MKKLK\_c)) and each key pair has an associated generation. The pair of master keys and the associated generation will be stored in a security data set.

There can be multiple versions of KLK security data sets specified. The version with the highest key generation value will be used.

For test purposes, the master keys are:

| **Item** | **Key Id** | **Value (in clear text)** |
| --- | --- | --- |
| MKKLK\_a | tstmkmsaklatls001 | '010202040407070810131315151616192023232525262629' |
| MKKLK\_c | tstmkmsaklctls001 | '313232343437373840434345454646495152525454575758 |
| Generation |  | 01 |

For each NSAM, the NSAM specific key loading keys shall be created by using the following diversification mechanism:

A 24 byte diversifier shall be formed by concatenating the 4 byte NSAM ISIN (see earlier section) 6 times as follows:  
  
ISIN | ISIN | ISIN | ISIN | ISIN | ISIN   
  
In order to produce the card specific keys (set to odd parity), each diversifier shall be encrypted with the appropriate master key, using 3DES in outer CBC mode and an IV of zero

The key generation shall be read direct form the security data set that is storing the associated master keys.

### CA Data

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| P | 128 Bytes | Generated |
| Q | 128 Bytes | Generated |
| PQ | 128 Bytes | Generated |
| DP1 | 128 Bytes | Generated |
| DQ1 | 128 Bytes | Generated |
| Device Certificate | Var | Generated |
| CA Subordinate Certificate | Var | Generated |
| CA Root Certificate | Var | Generated |

**Description:**

AE will use the HSM to generate a 256 byte RSA key (NKSK1) and supply this private key in CRT form as follows:

* P is the CRT prime number p
* Q is the CRT prime number q (where p > q)
* PQ is the CRT constant q-1 mod p
* DP1 is the CRT constant d mod (p – 1)
* DQ1 is the CRT constant d mod (q – 1)

AE will request the Device Certificate from the CA using the private key and the CA shall respond with the complete key chain (device, subordinate and root certificates)

**Certificate Format**

The format of the root, subordinate and device certificates is as follows:

* All certificates shall be RFC5280 compliant
* All security objects MUST be produced in DER format to preserve the integrity of the signatures within them.

The following tables use the following notation:

M Mandatory field that must be present

X Field that shall not be populated

O Optional field that may be populated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Certificate Fields** | **Section in RFC 5280** | **CA Cert** | **Sub-CA Cert** | **Device Cert** | **Comments** |
| TBSCertificate | 4.1.1.1 | M | M | M | Certificate information, ASN.1 DER encoded. See below. |
| SignatureAlgorithm | 4.1.1.2 | M | M | M | Identifier for the algorithm used by the CA to sign the certificate (Must contain the same algorithm identifier as the signature field in the TBSCertificate)  SHA-256 with RSA and MGF 1. |
| SignatureValue | 4.1.1.3 | M | M | M | Signature of TBSCertificate |

**TBSCertificate** (To Be Signed)

| **TBSCertificate Fields** | **Section in RFC 5280** | **CA Cert** | **Sub-CA Cert** | **Device Cert** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| version | 4.1.2.1 | M | M | M | Integer 2 (represents certificate version 3) |
| serialNumber | 4.1.2.2 | M | M | M | Unique positive integer per certificate (within a CA), automatically generated by the CA |
| signature | 4.1.2.3 | M | M | M | Identifies the algorithm used by the CA to sign the certificate. (Must contain the same algorithm identifier as the signatureAlgorithm field in the Certificate)  SHA-256 with RSA and MGF 1 |
| issuer | 4.1.2.4 | M | M | M | UTF8String in the form:  CA Cert: “C=NL,L=TLS,O=OVC,OU=0*,*CN=ROOT.CA.0”  Sub-CA Cert: “C=NL,L=TLS,O=OVC,OU=0*,*CN=ROOT.CA.0”  Device Cert: “C=NL,L=TLS,O=OVC,OU=0*,*CN=CA\_SM.*A*”  *A:* aninteger that provide a unique reference within an “O” domain. This value will change over time as certificates are renewed (e.g. every 10 years). The initial settings are: PRD=“1”, TST=“1”, INT=”2” |
| validity | 4.1.2.5 | M | M | M | Time interval during which the CA warrants that it will maintain information about the status of the certificate, in the form of two dates:  notBefore set to certificate generation date  notAfter setting is configurable and varies with certificate type:  CA root certificate: cert generation date + 100 years  CA subordinate certificate: cert generation date + 20 years  device certificate:  [E4] Last day in the month following cert generation date + 10 years  [E5]Using the expiry date supplied in the request (see ), set to 00:00:00 of the next day. |
| subject | 4.1.2.6 | M | M | M | UTF8String in the form:  CA Cert: “C=NL,L=TLS,O=OVC,OU=0,CN=ROOT.CA.0”  Sub-CA Cert: “C=NL,L=TLS,O=OVC,OU=0,CN=SM.*A*”  Device Cert: “C=NL,L=*YYY*,O=OVC,OU=*XXX*,CN=SM.*ZZZZZZ*”  *XXX*: is the BE identifier (see 5.2.4), converted to decimal (using the shortest decimal representation)  *YYY*: is the BETLA (see 5.2.5)  *ZZZZZZ*: is the Device ID (see 5.2.2)  *A:* aninteger that provide a unique reference within an “O” domain. This value will change over time as certificates are renewed (e.g. every 10 years). The initial settings are: PRD=“1”, TST=“1”, INT=”2” |
| subjectPublicKeyInfo | 4.1.2.7 | M | M | M | The RSA public key for the holder of the certificate  The CA Cert public/private RSA key will be 512 bytes (4096 bits). All other RSA keys will be 256 bytes (2048 bits) |
| issuerUniqueID | 4.1.2.8 | X | X | X | Not present |
| subjectUniqueID | 4.1.2.8 | X | X | X | Not present |
| Extensions | 4.1.2.9 | M | M | M | See Table below |

**Extensions**

| **Extensions Fields** | **Section in RFC 5280** | **CA Cert** | **Sub-CA Cert** | **Device Cert** | **Comments** |
| --- | --- | --- | --- | --- | --- |
| AuthorityKeyIdentifier | 4.2.1.1 | O | M | M | Identification is based on subject key identifier in the issuer’s certificate  Not present in CA Cert |
| keyIdentifier | 4.2.1.1 | O | M | M | The keyIdentifier shall be composed of the 160-bit SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits).  Not present in CA Cert |
| authorityCertIssuer | 4.2.1.1 | X | X | X | Not present |
| authorityCertSerialNumber | 4.2.1.1 | X | X | X | Not present |
| SubjectKeyIdentifier | 4.2.1.2 | M | M | X | Not present in Device cert |
| keyIdentifier | 4.2.1.2 | M | M | M |  |
| KeyUsage | 4.2.1.3 | M | M | M | Marked critical  See Table below |
| CertificatePolicies | 4.2.1.4 | X | X | X | Not present |
| PolicyMappings | 4.2.1.5 | X | X | X | Not present |
| SubjectAlternativeName | 4.2.1.6 | X | X | X | Not present |
| IssuerAlternativeName | 4.2.1.7 | X | X | X | Not present |
| SubjectDirectoryAttributes | 4.2.1.8 | X | X | X | Not present |
| BasicConstraints | 4.2.1.9 | M | M | X | The BasicContraints extension shall be marked critical.  The cA boolean field in this extension shall indicate that the public key in this certificate may be used to verify certificate signatures.  The pathLenConstraint field shall be set as follows:  CA Cert: “1”  Sub-CA Cert: “0”  Device Cert: Not present |
| NameConstraints | 4.2.1.10 | X | X | X | Not present |
| PolicyConstraints | 4.2.1.11 | X | X | X | Not present |
| ExtendedKeyUsage | 4.2.1.12 | X | X | X | Not present |
| CRLDistributionPoints | 4.2.1.13 | X | X | X | Not present |
| InhibitAnyPolicy | 4.2.1.14 | X | X | X | Not present |
| FreshestCRL | 4.2.1.15 | X | X | X | Not present |
| PrivateInternetExtensions | 4.2.2 | X | X | X | Not present |
| AuthorityInformationAccess | 4.2.2.1 | X | X | X | Not present |
| SubjectInformationAccess | 4.2.2.2 | X | X | X | Not present |

**Key Usage**

The keyUsage extension shall be marked critical.

The values of the key usage bits depend on the type of certificate. The table below defines the value of the key usage bits for each type of certificate.

|  |  |  |  |
| --- | --- | --- | --- |
| **Bit description (bit nr)** | **CA Cert** | **Sub-CA Cert** | **Device Cert** |
| digitalSignature (0) | 0 | 0 | 1 |
| nonRepudiation (1) | 0 | 0 | 1 |
| keyEncipherment (2) | 0 | 0 | 0 |
| dataEncipherment (3) | 0 | 0 | 0 |
| keyAgreement (4) | 0 | 0 | 0 |
| keyCertSign (5) | 1 | 1 | 0 |
| cRLSign (6) | 1 | 1 | 0 |
| encipherOnly (7) | 0 | 0 | 0 |
| decipherOnly (8) | 0 | 0 | 0 |

## 

## Application Datamap

The personalization image for the MSA application will be provided in TLV form, where the tags are defined in the table below and the length is fixed as 2 bytes. The contents of the table are:

Tag: A 2 byte value that identifies the data to follow

Field The field(s) that are provided within the tag. The contents of each field are described elsewhere in this document. Where there is more than one field within a tag, the fields are concatenated in the order they are written

Size Number of bytes used by each field

Encrypt Identifies if the “value” part of the TLV is encrypted with the data transport key (shared between AE and desktop production). Encryption will be 3DES in ECB mode (using odd parity), right padded with ‘80’ followed by between 0 and 7 bytes of ‘00’ – making the input data a multiple of 8 byte blocks

Section Identifies the section within this document that describes the field.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tag** | **Field** | **Size (bytes)** | **Encrypt** | **Section** |
| DF00 | ISIN Key Type KLK Key Generation | 4 1 1 | N |  |
| DF10 | MSA KLK Authentication key MSA KLK Confidentiality key Padding | 24 24 8 | Y | 6.3.1 6.3.1 |
| DF20 | BE ID PIN | 3 8 | N | 5.2.4 5.1.1 |
| DF30 | P  Padding | Var 8 | Y | 6.3.2 |
| DF40 | Q  Padding | Var 8 | Y | 6.3.2 |
| DF50 | PQ  Padding | Var 8 | Y | 6.3.2 |
| DF60 | DP1  Padding | Var 8 | Y | 6.3.2 |
| DF70 | DQ1  Padding | Var 8 | Y | 6.3.2 |
| DF80 | Device Certificate | Var | N | 6.3.2 |
| DF90 | CA Root Certificate | Var | N | 6.3.2 |
| DFA0 | CA Subordinate Certificate | Var | N | 6.3.2 |

## Retained Data

The following MSA related data items shall be stored in the NSAM record within the Affina database and shall be available via a QCM request:

* ISIN
* Old ISIN (this is only populated for the period whilst waiting for a delete MSA application to be confirmed)
* Key Type
* BE ID
* PIN
* SerialNumber (Device Certificate number)
* Device Certificate expiry date
* NCRS
* Equipment Vendor
* Reader Vendor
* AE Instance

## MSA reissuance

The MSA Application can be re-issued via the post-issuance interface [NKAM/NCRS IFS] and this will cause the existing MSA application to be deleted and the new instance loaded. The trigger for re-issuance will either be an update request (which means a new version of the Affina configuration is available, which may include a new version of the application) or a reperso request.

The main differences between issuance and re-issuance are:

1. Issuance sends the personalisation data to the card personalisation system, which then loads the application and the personalisation data. Post-issuance provides both the application data and the personalisation data.
2. The personalisation data is constructed slightly differently.
3. The personalisation data is delivered as APDUs rather than DGIs

The previous sections discussed how to create personalisation data during the issuance process. This section identifies the differences for re-issuance:

### ISIN

The ISIN is obtained from the current record (4 bytes) and the first byte is incremented by one to create a new ISIN, with the last three bytes (the device ID) remaining untouched. If the increment exceeds 0xFF then the re-issuance request shall be rejected.

Note that the original ISIN value shall be used to populate the ‘Old ISIN’ parameter (which itself shall be removed, once the MSA application delete has been confirmed).

### Key Loading Keys

The master keys and generation should be read from the configuration (in case they’ve changed) and the MSA specific KLKs regenerated as the diversification data (the ISIN) has changed.

### MSA certificate

A new MSA device certificate will be generated as follows:

* A new private key will be generated
* The validity notAfter setting will be obtained from the current record (i.e. the end date does not change), unless a new MSA Expiry value is supplied as input data (which will happen in the repersonalisation case)
* The subject will use the existing Device ID
* The subject will use the BE TLA obtained from the previous application, unless a new BE TLA is supplied as input data (which will happen in the case of a BE change)

### PIN

The PIN should be read from the configuration (in case it’s changed)

### Key Type

The Key Type should be obtained from the current record

### AE Instance

The AE instance should be obtained from the current record

### APDUs

The APDUs, as delivered to the post-issuance interface, shall be formed as described in § 6.1.2[NSAM Perso]

# S9 Personalisation Data

This section defines all of the personalisation parameters that are specific to the S9 application.

Note that the SAM serial number, which is a required element of the S9 personalisation, is not mentioned in this document as this number has to be read directly from the SAM and therefore cannot be created during the data generation phase.

This initial section discusses how the personalisation parameters are used when initially creating an S9 application, whereas later sections discuss personalisation differences during S9 application reissuance.

## Business Application Level Personalisation Data

There is no personalisation data specified at the business application level.

## Application Profile Level Personalisation Data

The following personalisation data is specified at the application profile level.

### Zero

A fixed value:

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| Zero | 1 Byte | 0x00 |

## Derived personalisation data

### Initialisation Date

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| InitialisationDate | 15 Bytes | From current system time |

The current system time in GMT is used for the initialization date, with the exception of the year. The year is obtained from the request (see section 5.2.11). The resulting initialisation date is written in the format:

YYYYMMDDHHmmSSZ

Each character is written in ASCII, 15 bytes in total.

### CA Data

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| P | 64 Bytes | Generated |
| Q | 64 Bytes | Generated |
| PQ | 64 Bytes | Generated |
| DP1 | 64 Bytes | Generated |
| DQ1 | 64 Bytes | Generated |
| Device Certificate | 608 Bytes | Generated |
| CA Root Certificate | 608 Bytes | fixed value |

AE will use the HSM to generate a 128 byte RSA key (NKSK2) and supply this private key in CRT form as follows:

* PQ is the CRT constant q-1 mod p
* DP1 is the CRT constant d mod (p – 1)
* DQ1 is the CRT constant d mod (q – 1)

AE will then form a device certificate (see format in 2.6.1 of [S9 SAM]) using the following values:

|  |  |
| --- | --- |
| **SerialNumber** | Unique value for each S9 (and S2) application, 4 bytes starting with 0xFFFFFFFF and decrementing for each instance. (little endian) |
| **Issuer** | Fixed as: 0x4F553D43412C444E3D4946532C444E3D636F6D00000000000000000000000000  (OU=CA,DN=IFS,DN=com.............) |
| **notBefore** | The initialisation date (see 7.3.1) in ASCII in YYYYMMDDHHmmSSZ format |
| **notAfter** | Set to 01-01-2100 00:00:00 in ASCII in YYYYMMDDHHmmSSZ format |
| **Subject** | Fixed 30 character string in form:  “CN=XXXXXX,OU=QQQ,DN=IFS,DN=com”  Where XXXXXX is the Device ID (section 5.2.2)  And QQQ is the BE TLA (section 5.2.5) |
| **modulus** | Modulus of key just created (big endian) |
| **Exponent** | Public exponent of key just created (big endian) |
| **IssuerUniqueId** | All zeros (0x00) |
| **SubjectUniqueID** | All zeros (0x00) |

An MD5 hash shall be calculated over the first 460 bytes of the certificate, the resultant hash (16 bytes) is padded and signed using the HSM. The resultant signature is appended to the certificate.

The hash is left and right padded as follows:

|  |  |
| --- | --- |
| **Left padding** | 0001FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF003020300C06082A864886F70D020505000410 |
| **Right padding** | CCCCCCCC |

Nothing varies within the CA root certificate, so it is just fixed data (608 bytes), including the signature:

Production Configuration:

|  |  |
| --- | --- |
| **CA Root Certificate** ||

Test Configuration:

|  |  |
| --- | --- |
| **CA Root Certificate** ||

## Application Datamap

The personalization image for the S9 application will be provided in TLV form, where the tags are defined in the table below and the length is fixed as 2 bytes. The contents of the table are:

Tag: A 2 byte value that identifies the data to follow

Field The field(s) that are provided within the tag. The contents of each field are described elsewhere in this document. Where there is more than one field within a tag, the fields are concatenated in the order they are written

Size Number of bytes used by each field

Encrypt Identifies if the “value” part of the TLV is encrypted with the data transport key (shared between AE and desktop production). Encryption will be 3DES in ECB mode, right padded with ‘80’ followed by between 0 and 7 bytes of ‘00’ – making the input data a multiple of 8 byte blocks

Section Identifies the section within this document that describes the field.

Endian Identifies, when the data is written, as to whether the most significant byte (MSB) is written to the lowest address (big) or the highest address (little). For example, the value 0xAABBCC (where AA is the MSB) is written as 0xAABBCC with big endian and as 0XCCBBAA with little endian.

| **Tag** | **Field** | **Size (bytes)** | **Encrypt** | **Section** | **Endian** |
| --- | --- | --- | --- | --- | --- |
| DF05 | DeviceID Zero Operator ID Initialisation Date | 3 1 4 15 | N | 5.2.2 7.2.1 5.2.4 7.3.1 | Little N/A Little Big |
| DF15 | TLSK9 Padding | 16 8 | Y | 5.1.2 | Big |
| DF25 | KUpload Padding | 16 8 | Y | 5.1.2 | Big |
| DF35 | KDownload Padding | 16 8 | Y | 5.1.2 | Big |
| DF45 | PIN | 8 | N | 5.1.1 | Big |
| DF55 | P Field Padding | Var 8 | Y | 7.3.2 | N/A |
| DF65 | Q Field Padding | Var 8 | Y | 7.3.2 | N/A |
| DF75 | PQ Field Padding | Var 8 | Y | 7.3.2 | N/A |
| DF85 | DP1 Field Padding | Var 8 | Y | 7.3.2 | N/A |
| DF95 | DQ1 Field Padding | Var 8 | Y | 7.3.2 | N/A |
| DFA5 | Device Certificate |  | N | 7.3.2 | N/A |
| DFB5 | CA Root Certificate |  | N | 7.3.2 | N/A |

## Retained Data

The following S9 related data items shall be stored in the NSAM record within the Affina database and shall be available via a QCM request:

* Device ID
* Operator ID
* Initialisation Date
* SerialNumber (the Device Certificate)

## S9 Personalisation Differences

The operations available for an S9 application are:

* Adding the S9 application during the issuance process
* Repersonalising an S9 application that is already present on an NSAM (e.g. changing BE)
* Updating an S9 application that is already present on an NSAM (i.e. changing the applet code)
* Deleting an S9 application from an NSAM.

Adding an S9 application to an NSAM will not be permitted (e.g. where the NSAM was issued without S9 or where S9 has been deleted)

When repersonalising or updating an application, the majority of the S9 personalisation data will be obtained from the existing S9 application. However, the following items can change:

* If the BE changed, the request to repersonalise will provide (for use in the device certificate):
  + New BE ID
  + New BE abbreviation
* The S9 device certificate shall have a new serial number and key

Note that the notBefore date in the S9 device certificate shall be retained from the existing S9 application (this is mentioned as it is a slight change to the E3 philosophy)

Post-issuance personalisation requires the NSAM serial number (which is read from the chip during issuance), so the NSAM serial number (or chip ID) shall be read from the NSAM record.

### APDUs

The APDUs, as delivered to the post-issuance interface, shall be formed as described in § 6.1.3[NSAM Perso]

# S2 Personalisation Data

This section defines all of the personalisation parameters that are specific to the S2 application.

This initial section discusses how the personalisation parameters are used when initially creating an S2 application, whereas later sections discuss personalisation differences during S2 application reissuance.

## Business Application Level Personalisation Data

There is no personalisation data specified at the business application level.

## Application Profile Level Personalisation Data

The following personalisation data is specified at the application profile level.

### Zero

A fixed value:

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| Zero | 1 Byte | 0x00 |

### Zeros

A fixed value:

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| Zeros | 16 Bytes | 0x00000000000000000000000000000000 |

## Derived personalisation data

### Initialisation Date

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| InitialisationDate | 15 Bytes | From current system time |

The current system time in GMT is used for the initialization date, with the exception of the year. The year is obtained from the request (see section 5.2.11). The resulting initialisation date is written in the format:

YYYYMMDDHHmmSSZ

Each character is written in ASCII, 15 bytes in total.

### Factory Serial Number

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| FactorySerialNumber | 8 Bytes | Generated by Affina |

Factory serial number is a unique 8 byte value for each S2 application, starting from 0x0000000000000000 and incrementing for each S2 application.

### CA Data

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| P | 64 Bytes | Generated |
| Q | 64 Bytes | Generated |
| PQ | 64 Bytes | Generated |
| DP1 | 64 Bytes | Generated |
| DQ1 | 64 Bytes | Generated |
| Device Certificate | 608 Bytes | Generated |
| CA Root Certificate | 608 Bytes | fixed value |

The S2 CA data is formed in exactly the same way as the S9 CA data. See section 7.3.2 for details.

Note that the device certificate serial number range covers both S2 and S9 device certificates (i.e. every S2 and S9 device certificate will have a unique serial number). See section 7.3.2.

### Content Device Type File

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| Content Device Type File | 21 Bytes | Derived from Device abbreviation |

Abbreviation (see 5.2.13) in ASCII, followed by a fixed 18 bytes: 0x000000000000000000000000000000000001

### Content Key Version File

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| Content Key Version File | 32 Bytes | Derived from S2 Profile |

The Content Key Version file consists of four concatenated 8 byte key version fields as follows:

| **Field** | **Size** | **Comment** |
| --- | --- | --- |
| KabiVer | 8 Bytes | Little endian |
| KEKVer | 8 Bytes | Little endian |
| KTRVer | 8 Bytes | Little endian |
| KTickVer | 8 Bytes | Little endian |

However, as all version fields that contribute to this data are defined as being 1 byte long, the version field can be represented as: 1 byte version concatenated with 7 bytes 0x00000000000000.

The content of the one byte version depends upon the requested S2 profile (see 5.2.12)[[1]](#footnote-1)

L2/3 and CRS Profiles:

| **Field** | **1 byte version** |
| --- | --- |
| KabiVer | 0x00 |
| KEKVer | 0x00 |
| KTRVer | 0x00 |
| KTickVer | Global Version (see 5.1.4) |

L1 and L4 Profiles:

| **Field** | **1 byte version** |
| --- | --- |
| KabiVer | 0x00 |
| KEKVer | 0x00 |
| KTRVer | 0x00 |
| KTickVer | 0x00 |

### Ticketing Key Cryptogram

The ticketing key cryptogram to use is selected on the basis of the selected S2 profile:

| **S2 Profile** | **TKC** |
| --- | --- |
| L2/3 | S2 Ticketing Key Cryptogram (see 5.1.4) |
| CRS | S9 Ticketing Key Cryptogram (see 5.1.4) |

The length of the selected TKC shall be calculated (in bytes) and the length shall be prefixed to the TKC in two bytes, little endian encoded.

## Application Datamap

The personalization image for the S2 application will be provided in TLV form, where the tags are defined in the table below and the length is fixed as 2 bytes. The contents of the table are:

Tag: A 2 byte value that identifies the data to follow

Profile Some DGIs are optional and are only created for certain profiles (see 5.2.12). This column identifies which DGIs are used for which profiles, with “All” meaning the DGI is used with all profiles.

Field The field(s) that are provided within the tag. The contents of each field are described elsewhere in this document. Where there is more than one field within a tag, the fields are concatenated in the order they are written

Size Number of bytes used by each field

Encrypt Identifies if the “value” part of the TLV is encrypted with the data transport key (shared between AE and desktop production). Encryption will be 3DES in ECB mode, right padded with ‘80’ followed by between 0 and 7 bytes of ‘00’ – making the input data a multiple of 8 byte blocks

Section Identifies the section within this document that describes the field.

Endian Identifies, when the data is written, as to whether the most significant byte (MSB) is written to the lowest address (big) or the highest address (little). For example, the value 0xAABBCC (where AA is the MSB) is written as 0xAA, BB,CC with big endian and as 0XCC, BB, AA with little endian.

| **Tag** | **Profile** | **Field** | **Size (bytes)** | **Encrypt** | **Section** | **Endian** |
| --- | --- | --- | --- | --- | --- | --- |
| DE00 | All | P Field Padding | Var 8 | Y | 8.3.3 | N/A |
| DE05 | All | Q Field  Padding | Var 8 | Y | 8.3.3 | N/A |
| DE10 | All | PQ Field  Padding | Var 8 | Y | 8.3.3 | N/A |
| DE15 | All | DP1 Field  Padding | Var 8 | Y | 8.3.3 | N/A |
| DE20 | All | DQ1 Field  Padding | Var 8 | Y | 8.3.3 | N/A |
| DE25 | All | Device Certificate |  | N | 8.3.3 | N/A |
| DE30 | All | CA Root Certificate |  | N | 8.3.3 | N/A |
| DE35 | All | Factory Serial Number CHV1 | 8 8 | N | 8.3.2 5.1.3 | Big Big |
| DE40 | L2/3 or L4 | S2 SAM Auth Key Zeros Zeros Zeros Zeros Zeros Zeros  Padding | 8 16 16 16 16 16 16 8 | Y | 5.1.6 8.2.2 8.2.2 8.2.2 8.2.2 8.2.2 8.2.2 | N/A |
| DE40 | L1 | S2 SAM Auth Key S2 KAB0 Key S2 KABi Key Zeros Zeros S2 KEK Key Zeros  Padding | 8 16 16 16 16 16 16 8 | Y | 5.1.6 5.1.5 5.1.6 8.2.2 8.2.2 5.1.6 8.2.2 | N/A |
| DE40 | CRS | S2 SAM Auth Key KDownload KUpload Zeros Zeros Zeros Zeros  Padding | 8 16 16 16 16 16 16 8 | Y | 5.1.6 5.1.2 5.1.2 8.2.2 8.2.2 8.2.2 8.2.2 | N/A |
| DE45 | All | Zero DeviceID Device Type[[2]](#footnote-2) OperatorId Content Device Type file Content Key Version file | 1 3 1 4 21 32 | N | 8.2.1 5.2.2 5.2.2 5.2.4 8.3.4 8.3.5 | N/A Big N/A Big Big Little |
| DE50.. DE79 | L2/3 or CRS[[3]](#footnote-3) | Data length (of TKC below)  Ticketing Key Cryptogram | 2 Var | N | 8.3.6 | Little N/A |
| DE80 | L2/3[[4]](#footnote-4) | S2 R/W Module KTRn Cryptogram S2 R/W Module KABi Cryptogram S2 L1 SAM KABi Cryptogram | 24 24 16 | N | 5.1.5 5.1.6 5.1.6 | N/A |

## Retained Data

The following S2 related data items shall be stored in the NSAM record within the Affina database and shall be available via a QCM request:

* Device ID
* Operator ID
* Initialisation Date
* SerialNumber (of the Device Certificate)
* Factory serial number
* S2 Profile

## S2 Personalisation Differences

The operations available for an S2 application are:

* Adding the S2 application during the issuance process
* Repersonalising an S2 application that is already present on an NSAM (e.g. changing BE)
* Updating an S2 application that is already present on an NSAM (i.e. changing the applet code)
* Deleting an S2 application from an NSAM.

Adding an S2 application to an NSAM will not be permitted (e.g. where the NSAM was issued without S2 or where S2 has been deleted)

When repersonalising or updating an application, the majority of the S2 personalisation data will be obtained from the existing S2 application (e.g. it will not be possible to change the S2 profile). However, the following items can change:

* If the BE changed, the request to repersonalise will provide (for use in the device certificate):
  + New BE ID
  + New BE abbreviation
* The S2 device certificate shall have a new serial number and key
* A new factory serial number shall be used.

Note that the notBefore date in the S2 device certificate shall be retained from the existing S2 application (this is mentioned as it is a slight change to the E3 philosophy)

### APDUs

The APDUs, as delivered to the post-issuance interface, shall be formed as described in §4 [S2 App]

# MSA TKF Personalisation Data

This section identifies how to generate the ticketing key file data that is the output of the MSA TKF application.

The ticketing key file isn’t strictly an application, as it is actually data that is to be supplied to the MSA application. However, in terms of how this data is to be generated, it works well with the Affina post-issuance concepts of Add and Reperso. Thus, the generation of ticketing key file data is being treated as an application that is added and updated through post-issuance facilities. Note that ‘delete’ of a TKF is not an option (including as part of the reperso change)

The format of the data output by the MSA TKF application is defined in [NKAM/NCRS IFS]. Note that this ADD just describes how the TKF data is collated. Once the data is collated it will be stored in the database and then sent to the interface once a trigger is received (the trigger is described in the NSAM Operator Interface document). The actual output on the interface is XML, but this ADD just describes the data generated in the form of the comprising data elements (i.e. the XML can be formed by the trigger process if that’s more desirable in development terms)

## Request Level Personalisation Data

TKF Personalisation will be initiated via an update card request (i.e. a post-issuance request), which will identify the NSAM identifier (the card id), the delivery channel (the TLS channel) and whether or not the request is an Add, Reperso or an Update.

The update card request can optionally provide personalisation data:

* The CV key version to be used for the NSAMs BE.
* The new KLK version to be used.

The keys associated with these optional elements will only be provided in the TKF if the optional elements are present in the update card request.

## Product Level Personalisation Data

The majority of the data required for personalisation of the TKF application (i.e. the keys) is specified at the product level. See NSAM Operator Interface document for details.

## Business Application Level Personalisation Data

There is no personalisation data specified at the business application level.

## Application Profile Level Personalisation Data

There is no personalisation data specified at the application profile level.

## Personalisation data

### NCRS Identifier

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| NCRSIdentifier | String | From database record |

The NSAM is associated with a specific NCRS and this information is obtained from the NSAM database record using the card id.

### Issuer Specific Identification Number

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| ISIN | 4 Bytes | From database record |

The NSAM has an ISIN and this information is obtained from the NSAM database record using the card id.

### Ticketing Key File

| **Parameter name** | **Format** | **Value** |
| --- | --- | --- |
| Ticketing Key File | Base 64 | See below |

The ticketing key file is binary data encoded using Base 64 notation. The layout of the data is described in [NKAM/NCRS IFS]

The individual data items that form the ticketing key file are obtained as follows:

| **Field** | **Description** |
| --- | --- |
| MSA\_ISIN | As per 9.5.2 above. |
| Key File Sequence Number | Value that is read from the NSAM database record. It starts at 0 and is incremented by 1 (and written back to the record) each time a key file is generated. Field limit is 0xFFFF and an error should be raised and TKF not generated if limit is breached. |
| Key Type | Value that is read from the NSAM database record. This is the key type that was assigned when the NSAM was created. See 5.2.3 for creation. |
| Ticketing Key generations | For each ticketing key set that is defined in the product, specify the version of the ticketing key set. If there are less than 16 ticketing key sets defined any remaining generations are set to 0xFF. |
| CT/ULC key generations | For each CT/ULC key set that is defined in the product, specify the version of the CT/ULC key set. If there are less than 16 CT/ULC key sets defined any remaining generations are set to 0xFF. |
| Privacy key generations | For each privacy key set that is defined in the product, specify the version of the privacy key set. If there are less than 16 privacy key sets defined any remaining generations are set to 0xFF. |
| Ceiling Value Key generation | If the optional ceiling value key generation is supplied in the update request then it is specified here, otherwise it is set to 0xFF. |
| Key Loading Keys generation | If the optional KLK generation is supplied in the update request then it is specified here, otherwise set to 0xFF.  Note that once this TKF has been created this new KLK generation shall be written to the NSAM database record, overwriting the current KLK version (which is used in the next field) |
| Key Loading keys generation TKF protection | The current KLK version, as read from the NSAM database record (and as first set at issuance at 6.3.1) |
| Encrypted Key section | The keys that are being sent (see further details below), encrypted by the NSAM specific KLK confidentiality key (NKKLK,c) at the generation specified in the above Key Loading keys generation TKF protection field. Note that the NSAM specific key is derived from the master key as described at 6.3.1 |
| MAC | MAC calculated using the NSAM specific KLK authentication key (NKKLK,a) at the generation specified in the above Key Loading keys generation TKF protection field. Note that the NSAM specific key is derived from the master key as described at 6.3.1 |

With the encrypted key section containing

|  |  |
| --- | --- |
| **Field** | **Description** |
| Ticketing keys | The master ticketing keys for the key type (as selected in the key type field above); one master ticketing key per generation (as identified in the ticketing Key generations field above). For any generations set to 0xFF, the corresponding “key” will be filled with full entropy nonce random data. |
| CT/ULC keys | The master CT/ULC keys; one CT/ULC ticketing key per generation (as identified in the CT/ULC Key generations field above). For any generations set to 0xFF, the corresponding “key” will be filled with full entropy nonce random data. |
| Ceiling Value key | The NSAM specific Ceiling Value key (NKcvk) for the generation specified in the Ceiling Value Key generation field above. If the generation is set to 0xFF, the field will be filled with full entropy nonce random data. See below for key derivation.  The NSAM record will need to be consulted in order to find which BE the NSAM is associated with and thus which ceiling value key to use. |
| Key Loading Key Authentication | The NSAM specific authentication KLK (NKKLK,a) for the generation specified in the Key Loading Keys generation field above. If the generation is set to 0xFF, the field will be filled with full entropy nonce random data. See 6.3.1 for key derivation. |
| Key Loading Key Confidentiality | The NSAM specific confidentiality KLK (NKKLK,c) for the generation specified in the Key Loading Keys generation field above. If the generation is set to 0xFF, the field will be filled with full entropy nonce random data. See 6.3.1 for key derivation. |
| Privacy Keys | The privacy keys; one privacy key per generation (as identified in the privacy key generations field above). No NSAM specific derivation is required. For any generations set to 0xFF, the corresponding “key” will be filled with full entropy nonce random data. |

**Ceiling Value Key Derivation**

Each Business Entity will have a 24 byte ceiling value master key, probably at multiple versions (MKCVK,BE\_ID,v)

For each NSAM, the NSAM specific ceiling value master key (NKCVK) shall be created by using the following diversification mechanism:

A 24 byte diversifier shall be formed by concatenating the 4 byte NSAM ISIN 6 times as follows:  
  
ISIN | ISIN | ISIN | ISIN | ISIN | ISIN   
  
In order to produce the card specific keys, each diversifier shall be encrypted with the appropriate master key, using 3DES in outer CBC mode and an IV of zero

## Retained Data

The following TKF related data items shall be stored in the NSAM record within the Affina database and shall be available via a QCM request:

* MSA\_ISIN
* Key File Sequence Number
* Key Type
* All key generations used

## Test Keys

Whilst this doesn’t restrict the keys that will be used during testing/acceptance, the system shall be delivered at E3 with the following TKF related test keys available:

**Ticketing Keys**

| **Gen** | **Key Name** | **Key Id** | **Test Key** |
| --- | --- | --- | --- |
| 0x00 | Personalisation  Service  Validation  Enquiry | tstmkovapertls000  tstmkovasrvtls000  tstmkovavaltls000  tstmkovaenqtls000 | AB01AEBCAB01AEBCAB01AEBCAB01AEBCAB01AEBCAB01AEBC  AD01AEBCAD01AEBCAD01AEBCAD01AEBCAD01AEBCAD01AEBC  CD01AEBCCD01AEBCCD01AEBCCD01AEBCCD01AEBCCD01AEBC  FD01AEBCFD01AEBCFD01AEBCFD01AEBCFD01AEBCFD01AEBC |
| 0x01 | Personalisation  Service  Validation  Enquiry | tstmkovapertls001  tstmkovasrvtls001  tstmkovavaltls001  tstmkovaenqtls001 | AB02AEBCAB02AEBCAB02AEBCAB02AEBCAB02AEBCAB02AEBC  AD02AEBCAD02AEBCAD02AEBCAD02AEBCAD02AEBCAD02AEBC  CD02AEBCCD02AEBCCD02AEBCCD02AEBCCD02AEBCCD02AEBC  FD02AEBCFD02AEBCFD02AEBCFD02AEBCFD02AEBCFD02AEBC |
| 0x04 | Personalisation  Service  Validation  Enquiry | tstmkovapertls004  tstmkovasrvtls004  tstmkovavaltls004  tstmkovaenqtls004 | AB04AEBCAB04AEBCAB04AEBCAB04AEBCAB04AEBCAB04AEBC  AD04AEBCAD04AEBCAD04AEBCAD04AEBCAD04AEBCAD04AEBC  CD04AEBCCD04AEBCCD04AEBCCD04AEBCCD04AEBCCD04AEBC  FD04AEBCFD04AEBCFD04AEBCFD04AEBCFD04AEBCFD04AEBC |
| 0x05 | Personalisation  Service  Validation  Enquiry | tstmkovapertls005  tstmkovasrvtls005  tstmkovavaltls005  tstmkovaenqtls005 | AB07AEBCAB07AEBCAB07AEBCAB07AEBCAB07AEBCAB07AEBC  AD07AEBCAD07AEBCAD07AEBCAD07AEBCAD07AEBCAD07AEBC  CD07AEBCCD07AEBCCD07AEBCCD07AEBCCD07AEBCCD07AEBC  FD07AEBCFD07AEBCFD07AEBCFD07AEBCFD07AEBCFD07AEBC |
| 0x07 | Personalisation  Service  Validation  Enquiry | tstmkovapertls007  tstmkovasrvtls007  tstmkovavaltls007  tstmkovaenqtls007 | AB08AEBCAB08AEBCAB08AEBCAB08AEBCAB08AEBCAB08AEBC  AD08AEBCAD08AEBCAD08AEBCAD08AEBCAD08AEBCAD08AEBC  CD08AEBCCD08AEBCCD08AEBCCD08AEBCCD08AEBCCD08AEBC  FD08AEBCFD08AEBCFD08AEBCFD08AEBCFD08AEBCFD08AEBC |
| 0x0F | Personalisation  Service  Validation  Enquiry | tstmkovapertls015  tstmkovasrvtls015  tstmkovavaltls015  tstmkovaenqtls015 | AB0BAEBCAB0BAEBCAB0BAEBCAB0BAEBCAB0BAEBCAB0BAEBC  AD0BAEBCAD0BAEBCAD0BAEBCAD0BAEBCAD0BAEBCAD0BAEBC  CD0BAEBCCD0BAEBCCD0BAEBCCD0BAEBCCD0BAEBCCD0BAEBC  FD0BAEBCFD0BAEBCFD0BAEBCFD0BAEBCFD0BAEBCFD0BAEBC |
| 0x1E | Personalisation  Service  Validation  Enquiry | tstmkovapertls030  tstmkovasrvtls030  tstmkovavaltls030  tstmkovaenqtls030 | AB0DAEBCAB0DAEBCAB0DAEBCAB0DAEBCAB0DAEBCAB0DAEBC  AD0DAEBCAD0DAEBCAD0DAEBCAD0DAEBCAD0DAEBCAD0DAEBC  CD0DAEBCCD0DAEBCCD0DAEBCCD0DAEBCCD0DAEBCCD0DAEBC  FD0DAEBCFD0DAEBCFD0DAEBCFD0DAEBCFD0DAEBCFD0DAEBC |
| 0x1F | Personalisation  Service  Validation  Enquiry | tstmkovapertls031  tstmkovasrvtls031  tstmkovavaltls031  tstmkovaenqtls031 | AB0EAEBCAB0EAEBCAB0EAEBCAB0EAEBCAB0EAEBCAB0EAEBC  AD0EAEBCAD0EAEBCAD0EAEBCAD0EAEBCAD0EAEBCAD0EAEBC  CD0EAEBCCD0EAEBCCD0EAEBCCD0EAEBCCD0EAEBCCD0EAEBC  FD0EAEBCFD0EAEBCFD0EAEBCFD0EAEBCFD0EAEBCFD0EAEBC |
| 0x20 | Personalisation  Service  Validation  Enquiry | tstmkovapertls032  tstmkovasrvtls032  tstmkovavaltls032  tstmkovaenqtls032 | AB10AEBCAB10AEBCAB10AEBCAB10AEBCAB10AEBCAB10AEBC  AD10AEBCAD10AEBCAD10AEBCAD10AEBCAD10AEBCAD10AEBC  CD10AEBCCD10AEBCCD10AEBCCD10AEBCCD10AEBCCD10AEBC  FD10AEBCFD10AEBCFD10AEBCFD10AEBCFD10AEBCFD10AEBC |
| 0x22 | Personalisation  Service  Validation  Enquiry | tstmkovapertls034  tstmkovasrvtls034  tstmkovavaltls034  tstmkovaenqtls034 | AB13AEBCAB13AEBCAB13AEBCAB13AEBCAB13AEBCAB13AEBC  AD13AEBCAD13AEBCAD13AEBCAD13AEBCAD13AEBCAD13AEBC  CD13AEBCCD13AEBCCD13AEBCCD13AEBCCD13AEBCCD13AEBC  FD13AEBCFD13AEBCFD13AEBCFD13AEBCFD13AEBCFD13AEBC |
| 0x24 | Personalisation  Service  Validation  Enquiry | tstmkovapertls036  tstmkovasrvtls036  tstmkovavaltls036  tstmkovaenqtls036 | AB15AEBCAB15AEBCAB15AEBCAB15AEBCAB15AEBCAB15AEBC  AD15AEBCAD15AEBCAD15AEBCAD15AEBCAD15AEBCAD15AEBC  CD15AEBCCD15AEBCCD15AEBCCD15AEBCCD15AEBCCD15AEBC  FD15AEBCFD15AEBCFD15AEBCFD15AEBCFD15AEBCFD15AEBC |
| 0x25 | Personalisation  Service  Validation  Enquiry | tstmkovapertls037  tstmkovasrvtls037  tstmkovavaltls037  tstmkovaenqtls037 | AB16AEBCAB16AEBCAB16AEBCAB16AEBCAB16AEBCAB16AEBC  AD16AEBCAD16AEBCAD16AEBCAD16AEBCAD16AEBCAD16AEBC  CD16AEBCCD16AEBCCD16AEBCCD16AEBCCD16AEBCCD16AEBC  FD16AEBCFD16AEBCFD16AEBCFD16AEBCFD16AEBCFD16AEBC |
| 0x26 | Personalisation  Service  Validation  Enquiry | tstmkovapertls038  tstmkovasrvtls038  tstmkovavaltls038  tstmkovaenqtls038 | AB19AEBCAB19AEBCAB19AEBCAB19AEBCAB19AEBCAB19AEBC  AD19AEBCAD19AEBCAD19AEBCAD19AEBCAD19AEBCAD19AEBC  CD19AEBCCD19AEBCCD19AEBCCD19AEBCCD19AEBCCD19AEBC  FD19AEBCFD19AEBCFD19AEBCFD19AEBCFD19AEBCFD19AEBC |
| 0x27 | Personalisation  Service  Validation  Enquiry | tstmkovapertls039  tstmkovasrvtls039  tstmkovavaltls039  tstmkovaenqtls039 | AB20AEBCAB20AEBCAB20AEBCAB20AEBCAB20AEBCAB20AEBC  AD20AEBCAD20AEBCAD20AEBCAD20AEBCAD20AEBCAD20AEBC  CD20AEBCCD20AEBCCD20AEBCCD20AEBCCD20AEBCCD20AEBC  FD20AEBCFD20AEBCFD20AEBCFD20AEBCFD20AEBCFD20AEBC |
| 0x28 | Personalisation  Service  Validation  Enquiry | tstmkovapertls040  tstmkovasrvtls040  tstmkovavaltls040  tstmkovaenqtls040 | AB23AEBCAB23AEBCAB23AEBCAB23AEBCAB23AEBCAB23AEBC  AD23AEBCAD23AEBCAD23AEBCAD23AEBCAD23AEBCAD23AEBC  CD23AEBCCD23AEBCCD23AEBCCD23AEBCCD23AEBCCD23AEBC  FD23AEBCFD23AEBCFD23AEBCFD23AEBCFD23AEBCFD23AEBC |
| 0x2A | Personalisation  Service  Validation  Enquiry | tstmkovapertls042  tstmkovasrvtls042  tstmkovavaltls042  tstmkovaenqtls042 | AB25AEBCAB25AEBCAB25AEBCAB25AEBCAB25AEBCAB25AEBC  AD25AEBCAD25AEBCAD25AEBCAD25AEBCAD25AEBCAD25AEBC  CD25AEBCCD25AEBCCD25AEBCCD25AEBCCD25AEBCCD25AEBC  FD25AEBCFD25AEBCFD25AEBCFD25AEBCFD25AEBCFD25AEBC |

Note that between E3 and E4, TLS defined the following as the official test ticketing keys. Whilst not a deliverable, these can be used for testing:

| **Gen** | **Key Name** | **Key Id** | **Test Key** |
| --- | --- | --- | --- |
| 0x00 | Personalisation  Service  Validation  Enquiry | tstmkovapertls000  tstmkovasrvtls000  tstmkovavaltls000  tstmkovaenqtls000 | 0102020404070708080B0B0D0D0E0E101013131515161619  16190102020404070708080B0B0D0D0E0E10101313151516  151616190102020404070708080B0B0D0D0E0E1010131315  1315151616190102020404070708080B0B0D0D0E0E101013 |
| 0x01 | Personalisation  Service  Validation  Enquiry | tstmkovapertls001  tstmkovasrvtls001  tstmkovavaltls001  tstmkovaenqtls001 | 1010101004070708080B0B0D0D0E0E101013131515161619  10101010020404070708080B0B0D0D0E0E10101313151516  101010100102020404070708080B0B0D0D0E0E1010131315  1010101016190102020404070708080B0B0D0D0E0E101013 |
| 0x02 | Personalisation  Service  Validation  Enquiry | tstmkovapertls002  tstmkovasrvtls002  tstmkovavaltls002  tstmkovaenqtls02 | 2020202004070708080B0B0D0D0E0E101013131515161619  20202020020404070708080B0B0D0D0E0E10101313151516  202020200102020404070708080B0B0D0D0E0E1010131315  2020202016190102020404070708080B0B0D0D0E0E101013 |
| 0x03 | Personalisation  Service  Validation  Enquiry | tstmkovapertls003  tstmkovasrvtls003  tstmkovavaltls003  tstmkovaenqtls003 | 3131313104070708080B0B0D0D0E0E101013131515161619  31313131020404070708080B0B0D0D0E0E10101313151516  313131310102020404070708080B0B0D0D0E0E1010131315  3131313116190102020404070708080B0B0D0D0E0E101013 |

**CT/ULC Keys**

| **GeN** | **Key Name** | **Key Id** | **Test Key** |
| --- | --- | --- | --- |
| 0x00 | CT\_ULC | tstmkcltulctls000 | A801AEBCA801AEBCA801AEBCA801AEBCA801AEBCA801AEBC |
| 0x01 | CT\_ULC | tstmkcltulctls001 | A802AEBCA802AEBCA802AEBCA802AEBCA802AEBCA802AEBC |
| 0x04 | CT\_ULC | tstmkcltulctls004 | A804AEBCA804AEBCA804AEBCA804AEBCA804AEBCA804AEBC |
| 0x05 | CT\_ULC | tstmkcltulctls005 | A807AEBCA807AEBCA807AEBCA807AEBCA807AEBCA807AEBC |
| 0x07 | CT\_ULC | tstmkcltulctls007 | A808AEBCA808AEBCA808AEBCA808AEBCA808AEBCA808AEBC |
| 0x10 | CT\_ULC | tstmkcltulctls016 | A80BAEBCA80BAEBCA80BAEBCA80BAEBCA80BAEBCA80BAEBC |
| 0x1E | CT\_ULC | tstmkcltulctls030 | A80DAEBCA80DAEBCA80DAEBCA80DAEBCA80DAEBCA80DAEBC |
| 0x1F | CT\_ULC | tstmkcltulctls031 | A80EAEBCA80EAEBCA80EAEBCA80EAEBCA80EAEBCA80EAEBC |
| 0x20 | CT\_ULC | tstmkcltulctls032 | A810AEBCA810AEBCA810AEBCA810AEBCA810AEBCA810AEBC |
| 0x22 | CT\_ULC | tstmkcltulctls034 | A813AEBCA813AEBCA813AEBCA813AEBCA813AEBCA813AEBC |
| 0x24 | CT\_ULC | tstmkcltulctls036 | A815AEBCA815AEBCA815AEBCA815AEBCA815AEBCA815AEBC |
| 0x25 | CT\_ULC | tstmkcltulctls037 | A816AEBCA816AEBCA816AEBCA816AEBCA816AEBCA816AEBC |
| 0x26 | CT\_ULC | tstmkcltulctls038 | A819AEBCA819AEBCA819AEBCA819AEBCA819AEBCA819AEBC |
| 0x27 | CT\_ULC | tstmkcltulctls039 | A820AEBCA820AEBCA820AEBCA820AEBCA820AEBCA820AEBC |
| 0x28 | CT\_ULC | tstmkcltulctls040 | A823AEBCA823AEBCA823AEBCA823AEBCA823AEBCA823AEBC |
| 0x2A | CT\_ULC | tstmkcltulctls042 | A825AEBCA825AEBCA825AEBCA825AEBCA825AEBCA825AEBC |

**Privacy Keys**

|  |  |  |  |
| --- | --- | --- | --- |
| **Gen** | **Key Name** | **Key Id** | **Test Key** |
| 0x00 | Privacy | tstikovaprvtls000 | A701AEBCA701AEBCA701AEBCA701AEBCA701AEBCA701AEBC |
| 0x01 | Privacy | tstikovaprvtls001 | A702AEBCA702AEBCA702AEBCA702AEBCA702AEBCA702AEBC |
| 0x04 | Privacy | tstikovaprvtls004 | A704AEBCA704AEBCA704AEBCA704AEBCA704AEBCA704AEBC |
| 0x05 | Privacy | tstikovaprvtls005 | A707AEBCA707AEBCA707AEBCA707AEBCA707AEBCA707AEBC |
| 0x07 | Privacy | tstikovaprvtls007 | A708AEBCA708AEBCA708AEBCA708AEBCA708AEBCA708AEBC |
| 0x10 | Privacy | tstikovaprvtls016 | A70BAEBCA70BAEBCA70BAEBCA70BAEBCA70BAEBCA70BAEBC |
| 0x1E | Privacy | tstikovaprvtls030 | A70DAEBCA70DAEBCA70DAEBCA70DAEBCA70DAEBCA70DAEBC |
| 0x1F | Privacy | tstikovaprvtls031 | A70EAEBCA70EAEBCA70EAEBCA70EAEBCA70EAEBCA70EAEBC |
| 0x20 | Privacy | tstikovaprvtls032 | A710AEBCA710AEBCA710AEBCA710AEBCA710AEBCA710AEBC |
| 0x22 | Privacy | tstikovaprvtls034 | A713AEBCA713AEBCA713AEBCA713AEBCA713AEBCA713AEBC |
| 0x24 | Privacy | tstikovaprvtls036 | A715AEBCA715AEBCA715AEBCA715AEBCA715AEBCA715AEBC |
| 0x25 | Privacy | tstikovaprvtls037 | A716AEBCA716AEBCA716AEBCA716AEBCA716AEBCA716AEBC |
| 0x26 | Privacy | tstikovaprvtls038 | A719AEBCA719AEBCA719AEBCA719AEBCA719AEBCA719AEBC |
| 0x27 | Privacy | tstikovaprvtls039 | A720AEBCA720AEBCA720AEBCA720AEBCA720AEBCA720AEBC |
| 0x28 | Privacy | tstikovaprvtls040 | A723AEBCA723AEBCA723AEBCA723AEBCA723AEBCA723AEBC |
| 0x2A | Privacy | tstikovaprvtls042 | A725AEBCA725AEBCA725AEBCA725AEBCA725AEBCA725AEBC |

**Ceiling Value Keys**

**BE: CXX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Gen** | **Key Name** | **Key Id** | **Test Key** |
| 0x00 | CVK for CXX | tstmkmsacvkcxx000 | A401AEBCA401AEBCA401AEBCA401AEBCA401AEBCA401AEBC |
| 0x01 | CVK for CXX | tstmkmsacvkcxx001 | A402AEBCA402AEBCA402AEBCA402AEBCA402AEBCA402AEBC |
| 0x02 | CVK for CXX | tstmkmsacvkcxx002 | A404AEBCA404AEBCA404AEBCA404AEBCA404AEBCA404AEBC |
| 0x03 | CVK for CXX | tstmkmsacvkcxx003 | A407AEBCA407AEBCA407AEBCA407AEBCA407AEBCA407AEBC |

**BE: RET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Gen** | **Key Name** | **Key Id** | **Test Key** |
| 0x00 | CVK for RET | tstmkmsacvkret000 | A201AEBCA201AEBCA201AEBCA201AEBCA201AEBCA201AEBC |
| 0x01 | CVK for RET | tstmkmsacvkret001 | A202AEBCA202AEBCA202AEBCA202AEBCA202AEBCA202AEBC |
| 0x02 | CVK for RET | tstmkmsacvkret002 | A204AEBCA204AEBCA204AEBCA204AEBCA204AEBCA204AEBC |
| 0x03 | CVK for RET | tstmkmsacvkret003 | A207AEBCA207AEBCA207AEBCA207AEBCA207AEBCA207AEBC |

**BE: CCV**

|  |  |  |  |
| --- | --- | --- | --- |
| **Gen** | **Key Name** | **Key Id** | **Test Key** |
| 0x00 | CVK for CCV | tstmkmsacvkccv000 | A101AEBCA101AEBCA101AEBCA101AEBCA101AEBCA101AEBC |
| 0x01 | CVK for CCV | tstmkmsacvkccv001 | A102AEBCA102AEBCA102AEBCA102AEBCA102AEBCA102AEBC |
| 0x02 | CVK for CCV | tstmkmsacvkccv002 | A104AEBCA104AEBCA104AEBCA104AEBCA104AEBCA104AEBC |
| 0x03 | CVK for CCV | tstmkmsacvkccv003 | A107AEBCA107AEBCA107AEBCA107AEBCA107AEBCA107AEBC |

# [E5] Device Certificate Personalisation Data

This section defines all of the personalisation parameters that are specific to the Device Certificate application.

## Business Application Level Personalisation Data

There is no personalisation data specified at the business application level.

## Application Profile Level Personalisation Data

There is no personalisation data specified at the application profile level.

## Derived personalisation data

### Certificate Format

A certificate shall be requested from either the WS or SM CA, as directed by the CA field in the request.

The format of the certificate shall be exactly the same as described in section for a device certificate, with the following differences:

* In the issuer field for the WS device certificate, ‘CN’ will be set to “CA\_WS.*A*”
* In the subject field for the WS device certificate, ‘CN’ will be set to “WS.*ZZZZZZ*”

The inputs to the CA for requesting the certificate shall be (as with the MSA device certificate):

* CertExpiry for the validity field
* BE Identifier, BE Abbreviation and Device ID for the subject field
* Public Key for the subjectPublicKeyInfo field

The resulting certificate shall be DER encoded.

## Application Datamap

The personalization image for the MSA application will be provided in TLV form, where the tags are defined in the table below and the length is fixed as 2 bytes. The contents of the table are:

Tag: A 2 byte value that identifies the data to follow

Field The field(s) that are provided within the tag. The contents of each field are described elsewhere in this document. Where there is more than one field within a tag, the fields are concatenated in the order they are written

Size Number of bytes used by each field

Section Identifies the section within this document that describes the field.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tag** | **Field** | **Size (bytes)** | **Section** |
| 0101 | Certificate | Var |  |

## Retained Data

The following device certificate related data items shall be stored in the record within the Affina database and shall be available via a QCM request:

* ISIN
* Device ID
* Certificate expiry date
* Certificate serial number

# Bureau Returns

NSAM requests are sent to the “DP” bureau and once the NSAM has been produced, a bureau return message is provided, as described in [NSAM Perso].

[E5] Certificate requests are sent to the “Certificate Issuance” bureau, which processes the request and then provides a bureau return message, as described below.

## Bureau Return Processing

Upon receipt of the bureau return message, the chip ID will be used to substitute the existing textual plastic number.

The production data item GP.CPLCEEPROM, that is supplied in the bureau return from the DP bureau, will be stored in the PMA database so that it can be searched using the QCMbyCardPersoData query and will be displayed in the results of any QCM query. Additionally, this CPLC data (42 bytes in total) shall be deconstructed, so that the first few bytes are also stored separately (as strings) and may be queried. The bytes to be stored are:

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Length (bytes)** | **Location in CPLC** |
| IC Fabricator | 2 | Characters 0-3 |
| IC Type | 2 | Characters 4-7 |
| OS Identifier | 2 | Characters 8-11 |
| OS Release Date | 2 | Characters 12-15 |
| OS Release Level | 2 | Characters 16-19 |

A standard card record will be produced with no additional return blocks (note that the PAN will contain the chip ID).

## [E5] Certificate Issuance Bureau

The purpose of the Certificate Issuance bureau is to read the delivery packages produced by NKAM that contain certificates, write these certificates to disk and then provide feedback to NKAM, identifying that the certificate has been “issued”.

The Certificate Issuance Bureau shall consist:

* A Java program that can run as a service and shall read all of its configuration properties from a properties file.
* Maintain a log file, where any errors are written and the fact that messages are received and sent shall be written. Full contents of messages received and sent shall only be written if logging is in debug mode.
* The properties file shall include:
  + Queue manager configuration
  + Configuration of MQ queue read from and MQ queue written to.
  + Directory location where certificates are written to.
  + The first 4 bytes of the card ID (default will be 0xFAFAFAFA)

The Certificate Issuance Bureau shall:

* Read delivery package messages from the configured read queue and:
  + Extract the certificate from DGI 0101
  + Write the certificate to the configured directory location as a binary file, using the file name format:  
    CERT.*BE TLA.Order Number.Instance*.CER  
    Where BE TLA, Order Number and Instance are available from the delivery package fulfilment data.
* Write a bureau return message to the configured write queue, where:
  + The tracking reference is extracted from the delivery package message
  + The card personalisation date is current system time
  + The card identifier is 8 bytes, consisting the configured first 4 bytes concatenated with the ISIN (from the delivery package fulfilment data)[[5]](#footnote-5).

# Reports

Affina shall provide the following TLS specific reports:

## Blacklisted Applications

This report shall list all of the non-expired applications that have been blacklisted for the following reasons:

* They’ve been added to the CRL due to being classified as lost/stolen (NSAM or device certificate has “suspended” status)
* The NSAM has been classified as end of life (NSAM or device certificate has “cancelled” status)

Note that this is an application report, so the same NSAM may appear on the report multiple times for different applications.

The report shall have the following columns:

|  |  |
| --- | --- |
| **Column Heading** | **Usage** |
| Device ID | This uses the “Name” field of the card record.  In most cases, the name field will contain the Device ID, although for S2’s disabled via a CRL, it will contain the certificate serial number as the Device ID is unknown. |
| App Type | S2, S9, MSA or Cert |
| ISIN | For MSA apps only, the app’s ISIN |
| Certificate Number | The SerialNumber field, as stored with each application. |
| Status | Lost/Stolen (if status is “suspended”)  End of Life (if status is “cancelled”) |

The report shall be ordered using the Status column, followed by Device ID.

## MSA Expiry Report

### [E4] MSA Card Expiry

This report shall ask the operator for a date range and shall then list all of the MSA applications whose certificates expire within the date range.

The report shall have the following columns:

|  |  |
| --- | --- |
| **Column Heading** | **Usage** |
| ISIN | The MSA app’s ISIN |
| Certificate Number | The SerialNumber field, as stored with each application. |
| Expiry Date | The certificate expiry date |
| NCRS | The identifier of the NCRS that the NSAM is associated with. |
| BE | The identifier of the BE that the NSAM is associated with. |

The report shall be ordered using the NCRS column, followed by BE, followed by expiry date, followed by ISIN

### [E5] MSA Certificate Expiry

This report shall ask the operator for a date range and shall then list all of the MSA device certificates that expire within the date range.

The report shall have the following columns:

|  |  |
| --- | --- |
| **Column Heading** | **Usage** |
| NCRS | The identifier of the NCRS that the NSAM is associated with (or “none”, if there is no NCRS). |
| BE | The identifier of the BE that the NSAM is associated with. |
| Expiry Date | The certificate expiry date |
| ISIN | The MSA app’s ISIN |
| Certificate Number | The SerialNumber field of the device certificate. |
| NSAM Chip ID | The NSAM’s chip identifier |
| MSA State | “Delivery” – waiting for confirmation that the APDUs have been applied  “Deployed” – The MSA application is on an NSAM and no replacement has been issued  “Deleting” – The MSA application is on an NSAM, but a replacement has been issued |

The report shall be ordered using the NCRS column, followed by BE, followed by expiry date, followed by ISIN

Note that if an MSA application is being replaced (and the response has not yet been received) and both certificates expire within the reporting period, both MSA applications will be included in the report.

## [E5] Device Certificate Expiry

This report shall ask the operator for a date range and shall then list all of the device certificates that expire within the date range.

The report shall have the following columns:

|  |  |
| --- | --- |
| **Column Heading** | **Usage** |
| BE | The identifier of the BE that the device certificate is associated with. |
| Expiry Date | The certificate expiry date |
| Device ID | The device certificate’s device ID |
| Certificate Number | The SerialNumber field of the device certificate. |

The report shall be ordered using the BE column, followed by expiry date, followed by Device ID

## Update Status

This report shall list all of the outstanding post issuance deliveries covering the period time from a specified date to current system time.

The report shall list, for each NCRS in turn:

* Those updates that have not yet been delivered
* Those updates that have been delivered , but not yet had a response
* Those updates that have been rejected by NCRS/Terminal and are in the retry loop (number of retries used will be identified)
* Those updates that have been rejected by NCRS/Terminal and have terminated (either because retry is not appropriate or maximum number of retries have been reached)

Each update shall be identified by:

* Destination NSAM device ID
* Destination NSAM card ID
* Date of update (not delivered: date update was available;   
   delivered: date of delivery;   
   retry: date of last delivery;  
   terminated: date of termination)

## All Devices

This report shall list all of the non-expired valid SAMs/NSAMs/Device certificates in the system.

The report shall have the following columns:

|  |  |
| --- | --- |
| **Column Heading** | **Usage** |
| Business Entity | The contents of the BE field, as stored with each application. |
| NCRS | The identifier of the NCRS that the NSAM is associated with. Left blank for SAMs and device certificates. |
| Device ID | This uses the “Name” field of the card record. |
| MSA ISIN | If MSA is present the app’s ISIN, if not ‘No’ |
| MSA Ver | If MSA is present the app’s version, if not then leave blank |
| S9 | If S9 is present ‘Yes’, else ‘No’ |
| S2 | If S2 is present ‘Yes’, else ‘No’ |
| [E5] Device Certificate | If this is a device certificate device ID ‘Yes’, else ‘No’ |

The report shall be ordered using the Business Entity column, followed by NCRS, followed by Device ID.

## Device Order Report

In this context, a “Device” refers to a device type as requested by ordering an NSAM or a certificate.

The order report shall be requested either by:

* 1. Entering an order number and, if that order number exists, the report shall contain the data for all Devices requested with that order number (the same order number may have been used with multiple requests).
  2. Entering a date range and the report shall contain the data for all Devices requested during the date range
  3. Requesting a report for all Devices in the database

The report’s header shall provide:

* + Date of the report.
  + Date range or order number (depending upon how the report was requested)
  + Total number of operator requests
  + Total number of Devices requested.
  + Total number of Devices produced.
  + Total number of Devices rejected.
  + Total number of Devices awaiting production.

The report’s body shall provide one entry per Device comprising:

* + Order Number.
  + Date the order was created.
  + Device ID.
  + Chip ID (if available).
  + Date produced or rejected (if available).
  + Current Status: issued, rejected, to be issued, expired, end-of-life, lost/stolen.
  + The current BE (it can change).
  + The device type.
  + The equipment vendor (blank for Cert devices)
  + The reader vendor (blank for Cert devices).
  + For each application (MSA/S2/S9/Cert):
    1. The certificate number
    2. The certificate expiry date
    3. The application version (blank for Cert devices)

1. For many of the keys, there now exists only one version, version zero. [↑](#footnote-ref-1)
2. The repetition of device type after device ID is intentional [↑](#footnote-ref-2)
3. DGIs not sent for L1 or L4 S2 profiles [↑](#footnote-ref-3)
4. DGI is not sent for L1, CRS or L4 S2 profiles [↑](#footnote-ref-4)
5. The card ID is required by Affina in order to complete the issuance cycle and thus enable the ability to query the card record. The Card ID isn’t something that NKAM requires. [↑](#footnote-ref-5)