Processes

This section describes processes performed by an E-SDC.

It contains the following sections:

1. E SDC Initialization
   Prior to the first use, the E-SDC has to be initialized. E-SDC must have access to the Secure Element during the initialization process in order to establish a secure connection with the TaxCore.API to obtain a set of initialization commands. The initialization commands are explained in the section Commands.

2. Standard Operation
   This section contains a description of standard E-SDC operations.

3. Malfunctions and Non serviceable Devices
   If the Secure Element is damaged and its data cannot be restored from the card, but the E-SDC is operational, the tax authority system shall be able to dump data from the E-SDC device and upload the audit packages using the same application used to upload audit packages submitted by a taxpayer.

E-SDC Initialization

Prior to the first use, the E-SDC has to be initialized. E-SDC must have access to the Secure Element during the initialization process in order to establish a secure connection with the TaxCore.API to obtain a set of initialization commands. The initialization commands are explained in the section Commands.

For instructions on how to download the initialization commands, see Get Initialization Commands.

After processing the received initialization commands, the E-SDC must upload configuration commands results to TaxCore.API.

In case of a poor or no internet connection, configuration commands results can be uploaded via file-based communication as explained in section E-SDC Stores a Command Execution Result to the SD Card or USB Drive.

NOTE:
Initialization of E-SDC has to be performed each time a new smart card is inserted into the reader.

Standard Operation

This section contains a description of standard E-SDC operations.

1. Extracting Expiration Date from Digital Certificate
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains the certificate (secure element) expiration date.

2. Extracting Taxpayer Identification Number (TIN) from Digital Certificate-from-Digital-Certificate)
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

3. Extracting UID from Digital Certificate
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains the UID of the taxpayer's secure element.

4. Extracting Taxpayer Information from Digital Certificate
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN, Business Name, Shop Name and POS location (Shop or HQ Address that will appear on the textual representation of the invoice).

5. E SDC Executes Commands
   E-SDC can receive Commands in two ways:

6. Sync Date and Time
   As an E-SDC is the source of date and time for the invoices, it is of the utmost importance to keep the device clock in sync.

7. Enter PIN to Unlock the Secure Element
   Before the Secure Element applet can be used, a valid PIN code must be supplied from the POS using the Ethernet connection. Once the E-SDC receives a PIN code, it will try to execute the *PIN Verify* APDU command.

8. Verify Secure Element Expiration Date and Time
   Each Secure Element is valid form 3 to 5 years (varies by Environment and local policies).

9. Fiscalization of an Invoice
   Invoice fiscalization is the main function of an E-SDC. Fiscalization is the process of handling an invoice request from an accredited invoicing system to produce fiscal invoices.

10.

10. Audit Process
    An audit is a process of sequential transferring of audit packages from an E-SDC to the tax authority's system and handling the response generated by the system for the specific device.

11. Notifications
    E-SDC device shall have an appropriate way to show the status of the device, information about the smart card and processes running on the E-SDC.

12. Switching Smart Cards During Operation
    During normal operation, taxpayers/cashiers might switch the smart card they are using for issuing fiscal invoices.

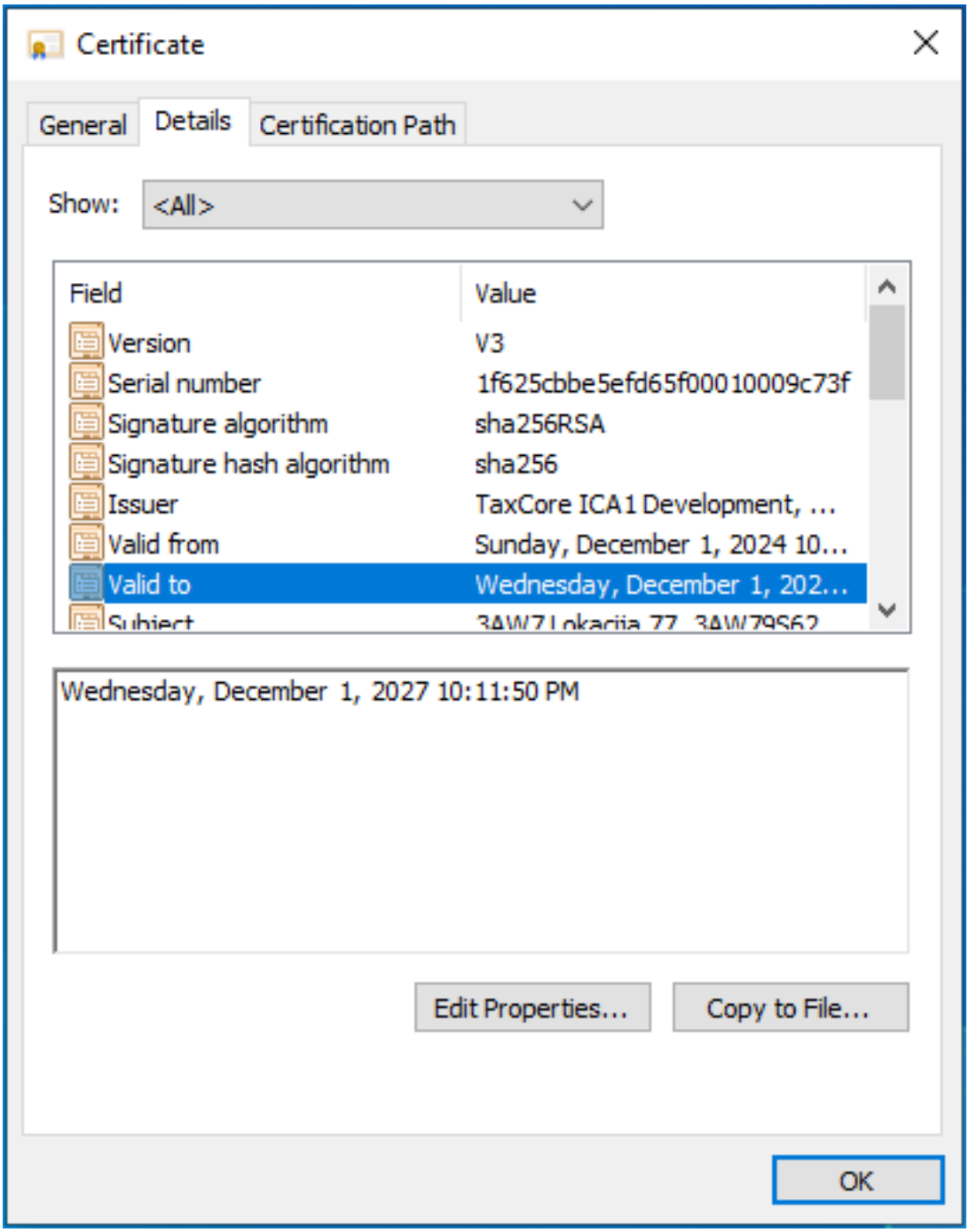
13. E SDC Logging
    E-SDC must keep a log about all required error events. It must log every error chronologically by local date and time (exact hour and minute).

Extracting Expiration Date from Digital Certificate

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains the certificate (secure element) expiration date.

Once the expiration date is extracted, it should be saved in **E-SDC's volatile memory**, until the smart card is removed/replaced or E-SDC is reset.

The date can be extracted from the certificate property **Valid to**



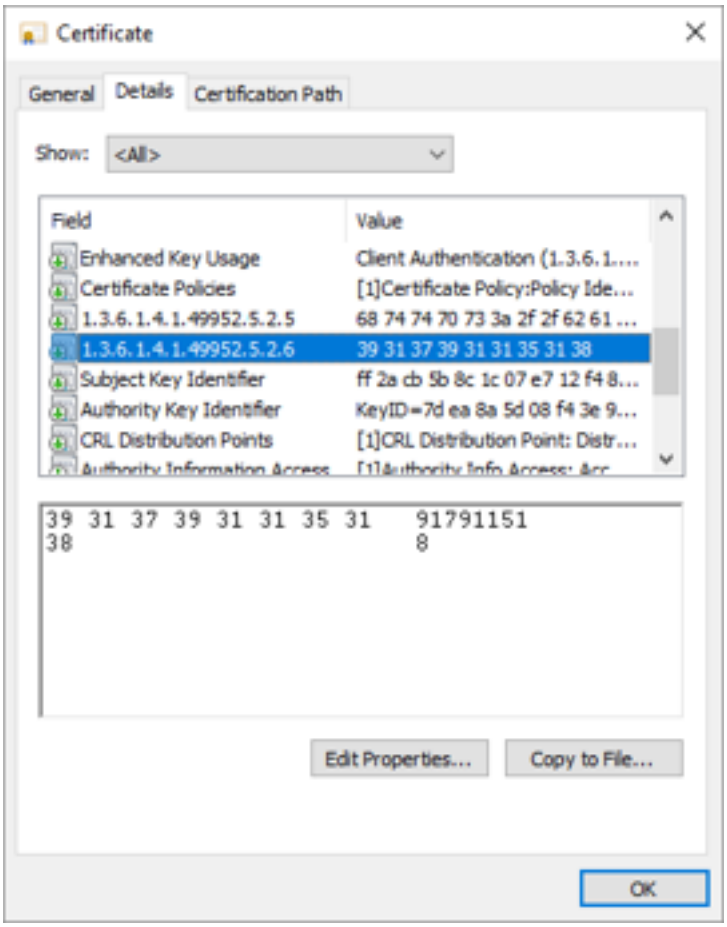
Extracting Taxpayer Identification Number (TIN) from Digital Certificate

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

TIN is stored in the digital certificate as an OID value. OID is dynamically created during a smart card personalization and depends on the target environment. The Test and Production environments will have different OIDs.

In order to use the same E-SDC with the Test and Production environments, the correct OID has to be constructed using the following procedure:

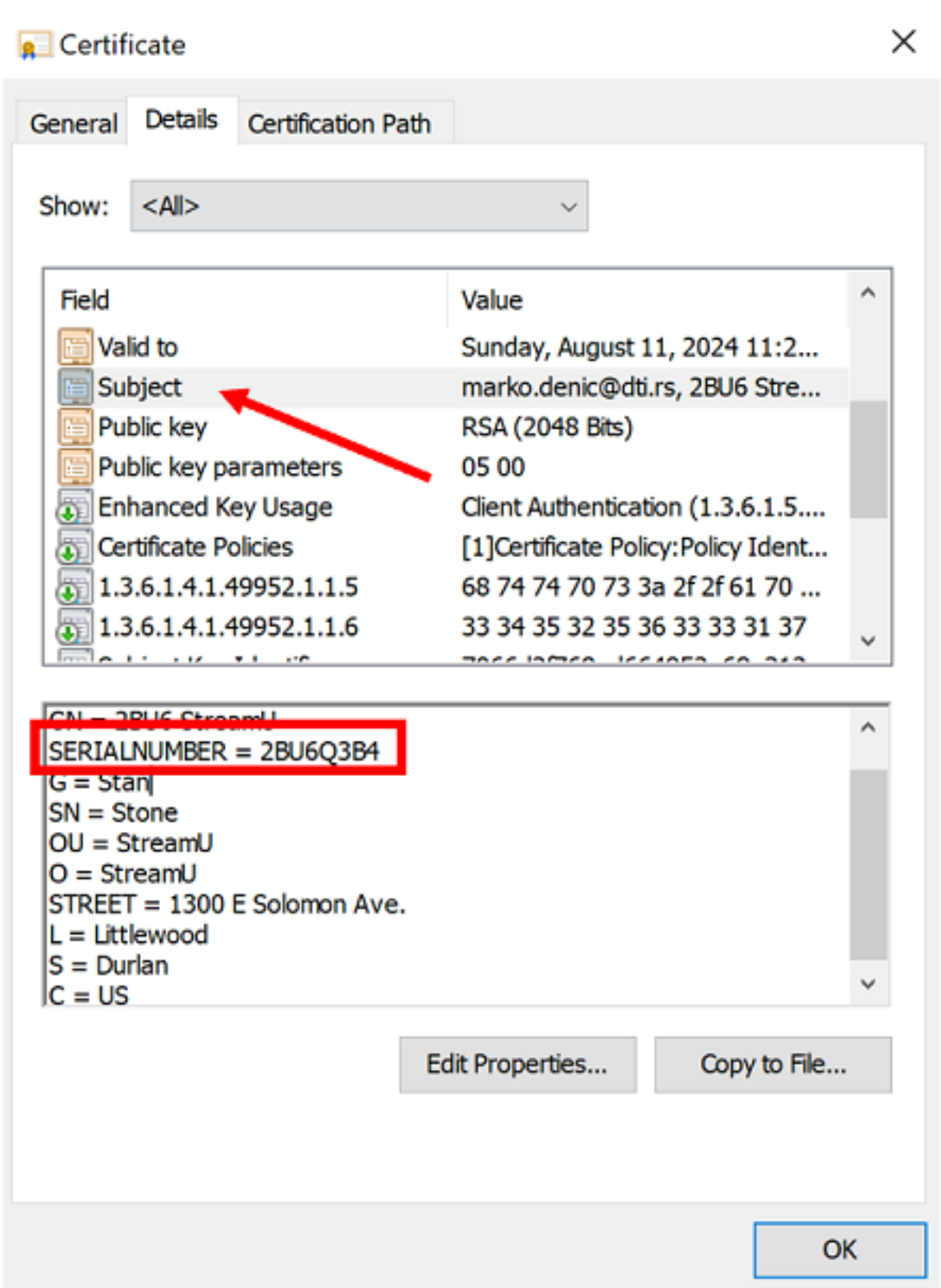
1. Get the certificate using the *Export Certificate* APDU command;
2. Read value of EnhancedKeyUsage (for example, 1.3.6.1.4.1.49952.5.2.3.3);
3. The fourth and the third integer to the right identify the environment;
4. Construct the OID that contains TIN, by replacing stars with the numbers using the following pattern - 1.3.6.1.4.1.49952...6;
5. For this example, resulting OID will be 1.3.6.1.4.1.49952.5.2.6;
6. Read the value of resulting OID containing Taxpayer TIN.



Extracting UID from Digital Certificate

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains the UID of the taxpayer's secure element.

UID is stored in the digital certificate as an **SERIALNUMBER** value in the certificate **Subject** field.



Extracting Taxpayer Information from Digital Certificate

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN, Business Name, Shop Name and POS location (Shop or HQ Address that will appear on the textual representation of the invoice).

Subject Field of the certificate contains the following information:

CN = Certificate Common Name - first 4 characters of the secure element UID and the Shop name

SERIALNUMBER = UID if the taxpayer's secure element

G = Authorized Person first name

SN = Authorized Person first name

OU = Shop name

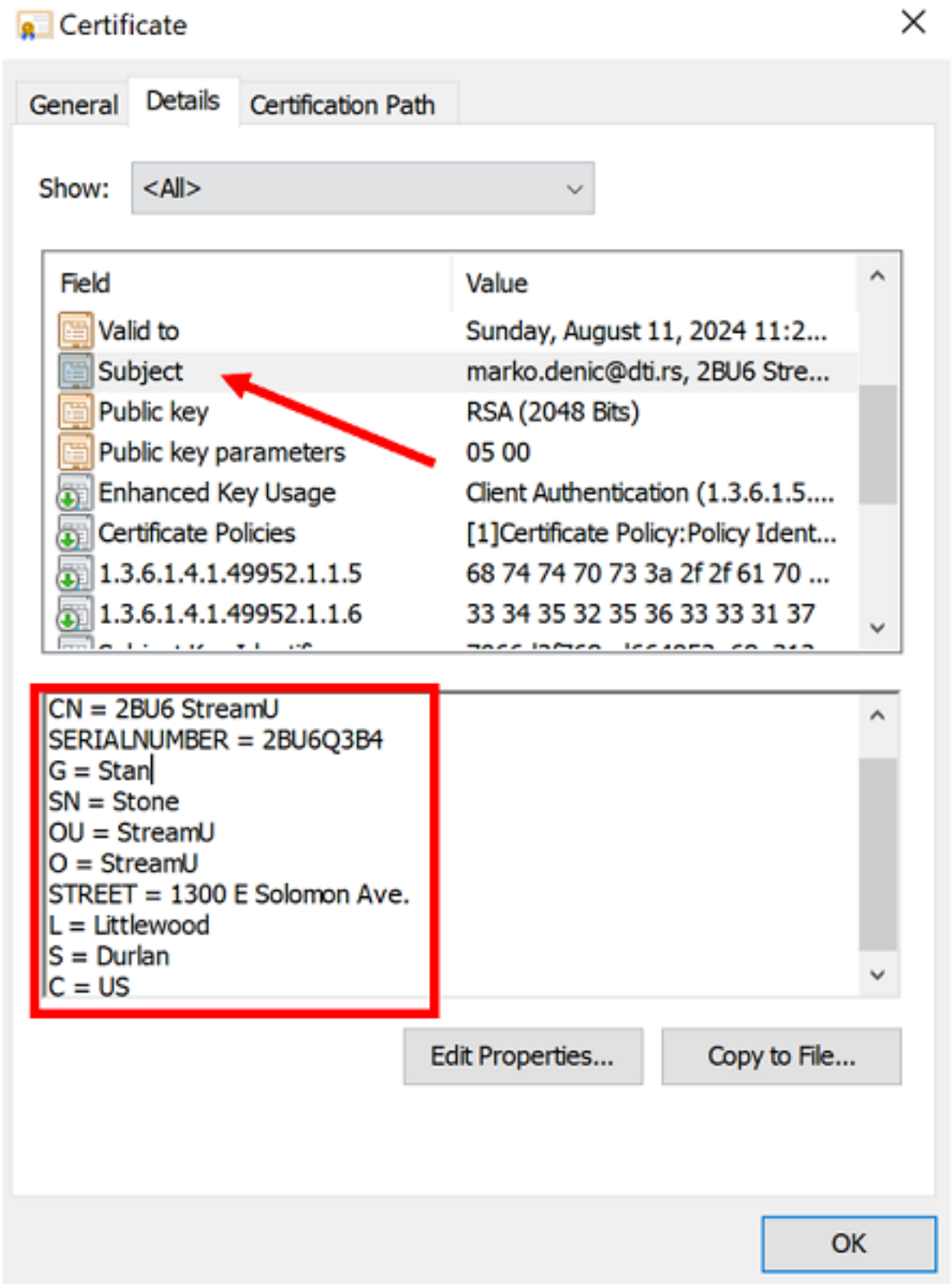
O = Business name

STREET = Physical street address

L = City/town

S = State, District or Region

C = Country



E-SDC Executes Commands

E-SDC can receive Commands in two ways:

- as a response to some TaxCore.API service calls (Notify Online Status and Submit Audit Package)
- via a external storage units

Processing commands depends on the command type. The process of execution for each command is explained

in the Commands section.

After the execution of the commands, in case the E-SDC is online, it notifies TaxCore.API about the execution success as explained in Notify Command Processed.

Sync Date and Time

As an E-SDC is the source of date and time for the invoices, it is of the utmost importance to keep the device clock in sync.

If the internet connection is available, the E-SDC shall sync time with the recommended NTP service at least once every 48h.

If the E-SDC does not support online or semi-connected operation modes, the manufacturer shall provide and document a simple way to check, set and keep date and time in sync on the E-SDC.

Enter PIN to Unlock the Secure Element

Before the Secure Element applet can be used, a valid PIN code must be supplied from the POS using the Ethernet connection. Once the E-SDC receives a PIN code, it will try to execute the *PIN Verify* APDU command.

NOTE:
Depending on the provided PIN, the Secure Element will remain either unlocked for further use or locked until a valid PIN is entered. This means that the E-SDC should not execute the PIN Verify APDU command again as long as the communication session between the E-SDC and the smart card is open.

E-SDC will send a response to the POS based on the result of the PIN Verify command execution.

It is important to note the Secure Element interprets data as byte containing digits, so the E-SDC must perform appropriate conversion before data is sent to the Secure Element. For example, if a PIN transmitted from a POS is “2017” (0x32 0x30 0x31 0x37 in ASCII hexadecimal representation), data sent to the SE shall be 0x02 0x00 0x01 0x07.

Verify Secure Element Expiration Date and Time

Introduction

Each Secure Element is valid form 3 to 5 years (varies by Environment and local policies).

Any invoice fiscalized using expired secure element will be automatically marked as invalid.

How to obtain digital certificate

E-SDC must obtain Secure Element expiration date from digital certificate returned from Secure Element using Export Certificate APDU command in DER format.

This operation must be executed whenever smartcard is inserted into reader or E-SDC is switched on.

What E-SDC must do if digital certificate has expired?

E-SDC must check current date and time against expiration date and time **BEFORE** invoice is signed by Secure Element. If current date and time is greater than expiration date and time of the digital certificate E-SDC must stop fiscalizing any invoices.

Fiscalization of an Invoice

Introduction

Invoice fiscalization is the main function of an E-SDC. Fiscalization is the process of handling an invoice request from an accredited invoicing system to produce fiscal invoices.

Process

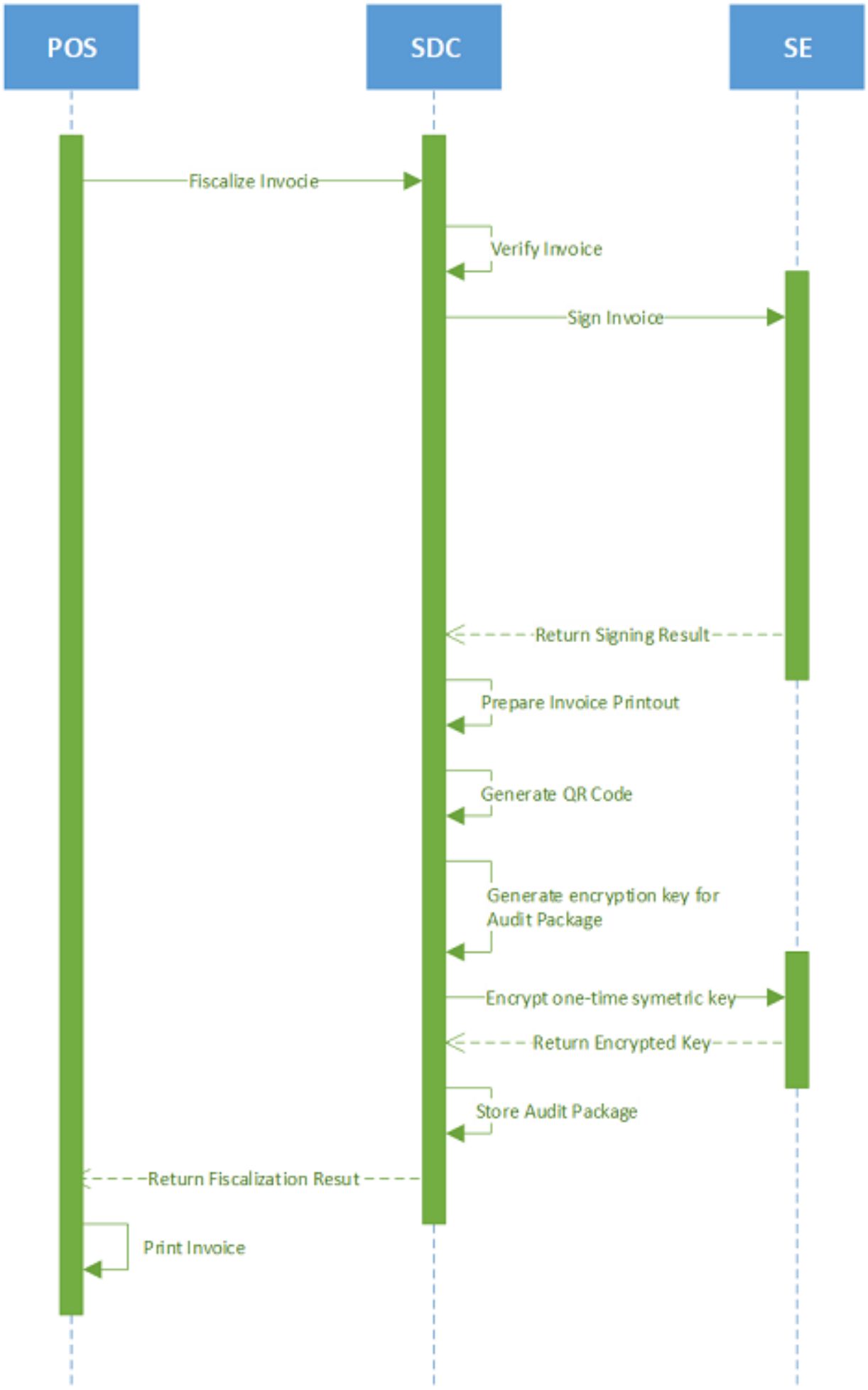
The following steps are executed by the E-SDC once an invoice request data is received from an Accredited POS:

1. POS generates a request data and sends it as an invoice request to the E-SDC;
2. E-SDC verifies the format and content of the invoice request;
3. E-SDC verifies the current E-SDC date and time value is smaller than the smart card certificate expiration date;
4. E-SDC determines which tax rate group to use based on the value of invoice type, ReferentDocumentNumber and ReferentDocumentDT fields;
5. E-SDC calculates taxes based on the selected tax rates group;
6.

E-SDC sends the invoice data to the Secure Element for fiscalization providing the current date and time and PIN code/password if required;

7. Secure element signs the invoice and returns the data to the E-SDC;
8. E-SDC produces a journal – a textual representation of an invoice;
9. E-SDC generates a verification URL;
10. [optionally] E-SDC creates a QR Code – a graphical representation of a verification URL;
11. E-SDC creates an invoice with all mandatory elements (receipt data, previously generated signature, verification URL and journal), generates a one-time key and encrypts the invoice using a symmetric algorithm. The E-SDC encrypts a one-time symmetric key using the tax authority's system public key and adds it to the package so the tax authority's system decrypts the symmetric key and access the package content once it arrives in the Service's system.
12. E-SDC returns a response to the POS and optionally generates journal data.

The process is illustrated in the figure below.



Fiscalization of An Invoice – Image of the fiscalization process

Content

Calculate Taxes

Introduction

Taxes are calculated by an E-SDC after a POS has sent a valid request. The tax amount for particular items on an invoice is defined by the tax labels associated with an item.

Process of a tax calculation depends on:

- Invoice and Transaction Type
- the tax rates for each label associated with an item on an invoice
- the Type value of tax category to which the label belongs

A POS sends an invoice fiscalization request with the line items. Items are sent with the total amounts (taxes included) and zero or more tax labels associated with them, which participated in the total price calculation.

How To Determine Which Tax Rate Group to Use

By default, E-SDC must use current `TaxRateGroup` based on current date and time of its clock to calculate taxes for each invoice.

In case of Copies and Refunds following rules are applied:

- If `referentDocumentNumber` is specified and `referentDocumentDT` is ommited or blank applicable `TaxRateGroup` must be determined based on current date and time of its clock and used to calculate taxes
- If both `referentDocumentDT` and `referentDocumentNumber` are specified, `TaxRateGroup` shall be determined based on value of `referentDocumentDT`

Algorithm

In order to calculate a tax, the following algorithm shall be implemented:

1. Determine which Tax Rate Group to use (see above).
2. Make an array of distinct tax labels associated with the items in the POS request (e.g. A, B, C, F, ...).
3. Calculate the tax amount for each individual label in the array:
    - Iterate through all items in the POS request
    - For each item, calculate tax amounts. One item has one or more tax labels, and each label represents a tax amount. Each tax amount is a part of an item's total price. These tax amounts are calculated as follows:
        - ♠ If an item has a label from the **amount-on-quantity** category applied, subtract the tax rate amount for that label, multiplied with quantity, from the item total price. The resulting amount (the remainder), is used in all further calculation steps instead of item total amount.
        - ♠ If none of the labels' tax category type is ***tax-on-total*** (category 1):
            - ♠ Tax amount for one label is:

$$\frac{item\ total\ amount\ *\ label\ rate}{(100 + \Sigma(all\ tax - on - net\ rates\ on\ item))}$$

Example 1: An item has a total price of $10 and applied labels: A(5%) and B(6%).

$$A = \frac{10\$ * 5}{(100 + \Sigma(5+6))} \quad B = \frac{10\$ * 6}{(100 + \Sigma(5+6))}$$

Tax amount for label A=$0.4505 and for label B=$0.5405.

♠

If any of the labels' tax category is ***tax-on-total*** (category 1):

♠ Tax amount for every label whose category type is ***tax-on-total*** (category 1) is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax-on-total\ rates)/100)} * \frac{label\ rate}{100}$$

♠ Tax amount for every other label from category 0 is:

$$\frac{item\ total\ amount}{(1\ +\ \Sigma(all\ tax-on-total\ rates)/100)} * \frac{label\ rate}{(100 + \Sigma(all\ tax-on-net\ rates\ on\ item))}$$

Example 2: Item has a total price $10 and applied labels: A(5% tax-on-net), B(6% tax-on-net), C(3% tax-on-total) and F(4% tax-on-total).

$$A = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{5}{(100 + \Sigma(5+6))}$$

$$B = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{6}{(100 + \Sigma(5+6))}$$

$$C = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{3}{100}$$

$$F = \frac{10\$}{(1\ +\ \Sigma(3+4)/100)} * \frac{4}{100}$$

Tax amount for label A=$0.4210 , for label B=$0.5052 , for label C=$0.2804 and for label F=$0.3738.

o

Summarize calculated tax amounts per label for items as the total amount sum for that label.

Example 3: the request contains two items from Example 1 and Example 2, the total sums for labels are: A=$0.8715 , B=$1.0457 , C=$0.2804 and F=$0.3738.

o

Summarize fixed tax amounts per label for items (each multiplied with quantity) as the respective total amount sum for that label.

Example 4: An item has quantity 2 with a total price of $10 and applied labels: A(5% tax-on-net) and E($0.10 fixed tax). The total sums for labels are: A=$0.4667 and E=$0.2000

Example 5: An item has quantity 2 with total price 10$ and applied labels: A(5% tax-on-net), C(3%

tax-on-total) and E(0.10$ fixed tax). The total sums for labels are: A=$0.4531, C=$0.2854 and E=$0.2000.

Example 6: the request contains two items: one with a total price of $5 and quantity 1, and another with a price of $10 and quantity 2. Both have applied label E(0.10$ fixed tax). The total sum for label E=$0.3000.

- Each of these summarized amounts per label represents one tax item in the array `taxItems` in Invoice Response, along with other properties related to the category for this label.

4. After all of the items have been processed, calculate the tax amount for all tax categories found in the request. One tax category can consist of one or more tax labels (e.g. A, B...). The tax amount for a tax category is a sum of all label tax amounts related to the category. These tax amounts for the category are displayed on the invoice journal.

   Example 7: The request contains two items from Example 1 and Example 2. Labels A and B are VAT category, C is STT category and F is ET category. Total VAT=$1.9172, STT=$0.2804 and ET=$0.3738.

5. Once the Tax calculation is completed, assign `GroupId` of the tax rate group to the field `TaxGroupRevision` of `InvoiceResult`.

Rounding

E-SDC shall round all amounts to 4 decimal places using the half-round up method.

Examples:

3.44445555666 → 3.4445

3.4440012345 → 3.4440

3.44466012345 → 3.4447

3.444116012345 → 3.4441

Create Verification URL

Verification URL is created based on values submitted by a POS to an E-SDC and values returned to the E-SDC from APDU commands as follows:

1. Byte array is created:

Start	Bytes	Invoice Field	Is an invoice field	Description
0	1	version	Yes	Current version is 0x03
1	8	requestedBy	Yes	UID, ASCII encoding (e.g. JKGB3K14)
9	8	signedBy	Yes	UID, ASCII encoding (e.g. JKGB3K14)
17	4	totalCounter	Yes	Int32 Little Endian
21	4	transactionTypeCou	Yes	Int32 Little Endian
25	8	totalAmount	Yes	totalInvoiceAmount * 10000 as Uint64 bit Little Endian
33	8	dateAndTime	Yes	Unix Timestamp (number of milliseconds), 64bit unsigned integer Big Endian
41	1	invoiceType	Yes	0x00 (Normal), 0x01 (Pro Forma), 0x02 (Copy), 0x03(Training),0x04(A
42	1	transactionType	Yes	0x00 (Sale), 0x01 (Refund)
43	1	N/A	No	Buyer ID length in bytes
44	0-20	buyerId	Yes	ASCII Encoding
44-64	256 or 512	encryptedInternalD	Yes	Encrypted Internal Data received from SE after Invoice Sign APDU command, 256 or 512 bytes long
300-320 or **556-576**	256	signature	Yes	Signature received from SE after Invoice Sign APDU

				command, 256 bytes long
556-576 or **812-832**	16	N/A	No	MD5 hash of all previous bytes

2. Created byte array is encoded as base64 string, which is additionally encoded, to comply with the URL standards.
3. Encoded string is appended to the verification URL received from the Set Verification URL Command.

NOTE:
Values for `dateAndTime` and all other fields must match the values submitted to the Secure Element for digital signing (see *Sign Invoice* in Fiscalization), as well as the values in the audit package (see Create Invoice).

Create a QR Code

QR code contains a Verification URL that is described created in the section Create Verification URL.

It is the most convenient way of exposing the Verification URL because it enables customers to easily scan their fiscal invoices using a QR code reader.

How to create a QR code

Base64 encoded string is created from GIF image bytes and attached to the Invoice Response

Important parameters for creating a QR code:

- Minimal size = 40x40mm
- ErrorCorrectionLevel = L
- FixedModuleSize = 4
- QuietZoneModules = Zero
- BlackAndWhite
- ImageFormat = Gif

For more information about using QR codes in the TaxCore system, see QR Code.

Create a Textual Representation of an Invoice

A textual representation of a Receipt shall be created as described in the chapter Anatomy of a Fiscal Receipt. One row on a receipt is 40 characters long to fit 2.25 inch / 58 mm wide paper roll commonly used in thermal printers.

SDC Date and Time field printed on a journal (textual representation of an invoice) generated by E-SDC are **locally time-based**.

Any amount shall be rounded to 2 (two) decimal places using the half-round up method only on the textual representation of an invoice.

NOTE:
Although the textual representation of an invoice (journal) can optionally be omitted from the E-SDC's response to POS, it **must be submitted to the tax authority** as part of the audit package.

1. Localization of textual representation of the invoice
   E-SDCs are generally built to work in multiple environments. As part of implementation roadmap you may decide to prepare your product for multiple markets or to target one market only.

2. Mapping Digital Certificate Subject Parameters to Invoice Fields
   Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

Localization of textual representation of the invoice

Introduction

E-SDCs are generally built to work in multiple environments. As part of implementation roadmap you may decide to prepare your product for multiple markets or to target one market only.

Localization is optional

If you decide to support only one market E-SDC may support only one language. For example, Implementation for Fiji may support English language only.

What to do if you want to support multiple languages?

- Textual representation must adhere to general instructions outlined in Anatomy of a Fiscal Receipt.
- All Invoice and transaction types abbreviations must be localized. For example **NS** (Normal Sale) in English is localized as **ПП** (Промет Продаја) in Serbian Cyrilic.
- All labels on invoice must be translated
- Languages supported by your E-SDC implementation **AND** TaxCore.API must be contained in the Result of the Get Status service.
    - in case your E-SDC supports en-US, fr-FR and sr-Latn-RS and TaxCore.API supports only fr-FR E-SDC must report fr-FR as the only supoprted language
    - in case your E-SDC supports en-US only and TaxCore.API supports only fr-FR E-SDC must return configuration error and stop any initialization.
- Content generated by POS or Secure Element should not be modified or localized (i.e. item names)
- If your E-SDC and TaxCore.API supports multiple languages POS may decide language of generated textual representation of invoice using request HTTP headers when invoking Create Invoice endpoint.

Accreditation

Accreditation for specific jurisdiction may require E-SDC to support specific language and culture. For Example, E-SDCs for Serbia requires support for sr-Cyrl-RS.

Mapping Digital Certificate Subject Parameters to Invoice Fields

Digital certificate exported using the *Export Certificate* APDU command (in DER format) contains taxpayer TIN and POS location (Shop or HQ Address that shall appear on the textual representation of the invoice).

This example shows the mapping between a subject name/value pairs and invoice fields.

Subject Field parameters with examples:

CN = P22V International Trek Center

SERIALNUMBER = P22VC8VR

G = Albert

SN = Mungin

OU = International Trek Center

O = International Trek Center

STREET = 8844 Garcia

L = West Covina

S = California

C = US

Invoice Field	Subject Parameter Name	Note
TIN	N/A	obtained by OID as explained in Extracting Taxpayer Identification Number from Digital Certificate
Business Name	O	Legal name under which the business operates - see Extracting Taxpayer Information from Digital Certificate
Shop Name	OU	It may be the same as Business Name if the Company HQ and sales location are the same. - see Extracting Taxpayer Information from Digital Certificate
Address	STREET	Street name and number - see Extracting Taxpayer Information from Digital Certificate
Location	L	City or town - see Extracting Taxpayer Information from Digital Certificate
State	S	State, District or Region - see Extracting Taxpayer Information from Digital Certificate
Country	C	ISO 2-letter Country Code. Optional field on the textual representation of the invoice - see Extracting Taxpayer Information from Digital Certificate

Creating an Audit Package

Once an invoice is created (`InvoiceRequest` and `InvoiceResult`) the E-SDC is ready to create an audit package and store it in the non-volatile memory. In order to achieve that, follow these steps:

1. Convert `sdcDateTime` data to UTC;
2. Generate a random one-time symmetric key for AES256;
    - KeySize = 256
    - Padding = PaddingMode.PKCS7
    - Mode = CBC
    - BlockSize = 128
    - IV = 16bytes
    - Key =32bytes
3. Encrypt Audit Data as JSON string using the one-time key;
4. Convert the encrypted invoice to base64 string and store it in the Payload field of Audit package;
5. Get the TaxCore Public key using *Export TaxCore Public Key* APDU command).
6. Encrypt the one-time key, using RSA encryption (padding PKCS1 (fOAEP: false)), with TaxCore public key, convert it to base64 string and store it in the Key field;
7. Encrypt Initialization Vector (IV), using RSA encryption (padding PKCS1 (fOAEP: false)), with TaxCore public key, convert it to base64 string and store it in the IV field;
8. Save the Audits as an Audit Package file, named as `{UID}-{UID}-{Ordinal_Number}.json`;
9. (Optionally) Generate a QR code, and attach it to `InvoiceResult` (make sure that the QR code is not stored in the Audit Package);
10. Return `InvoiceResult` to the POS;
11. If the internet connection is available try to send the Audit Data to TaxCore.API as explained in the section Remote Audit;

NOTE:
After submitting an audit package to TaxCore.API, if status 4 is received back (see Submit Audit package), the E-SDC should immediately delete that audit package from the its local storage. If the TaxCore.API returns status 1, the E-SDC should try to resubmit an audit package. If any other status is received (other that 1 or 4), the audit package must not be deleted, and the E-SDC should not try to resubmit that audit package to TaxCore.API.

Audit Process

Introduction

An audit is a process of sequential transferring of audit packages from an E-SDC to the tax authority's system and handling the response generated by the system for the specific device.

There are two specific scenarios: **Remote Audit** and **Local Audit**.

NOTE:
Basic rules and processes described in this section apply to both scenarios. Details are explained in separate sections - see Remote Audit and Local Audit.

Depending on the scenario, an audit may be triggered periodically, if one or more invoices are created, or after the insertion of an external memory device into an E-SDC.

An audit is always an asynchronous process. Depending on the amount of data and means of communication, it can take from less than a second to a couple of hours.

Once the E-SDC receives a response from the Secure Element (signed invoice), it must be encrypted and stored in E-SDC's non-volatile memory.

An E-SDC device must be fully functional during an audit. The POS must be able to sign new invoices as long as the Secure Element permits. There must be a mechanism in place that is responsible for the continuous operation of the Secure Element and E-SDC while audit packages are being transmitted to the tax authority's system or an external memory unit.

Remote Audit

Remote audit is the process of transferring data to the tax authority's system using Internet connection. It is the most common way to perform audits for any device with stable Internet connection.

An E-SDC checks if TaxCore.API is reachable. If it is, the E-SDC authenticates the tax authority's system by using a server-side certificate installed on the TaxCore.API endpoint, enabling HTTPS protocol. The tax authority's system authenticates the E-SDC using a digital certificate issued on the Secure Element and issues a token for that session.

The E-SDC starts sending audit packages, performing a series of audits until all the data stored on its non-volatile memory is audited.

Remote audit step-by-step:

1. E-SDC submits an audit package by invoking TaxCore.API service Submit Audit Package
2. The tax authority's system verifies the data and returns a response containing:
    - the audit package status
    - a set of Commands (including the Proof of Audit command).

There are two scenarios for obtaining a POA:

- **As part of the Local Audit process** (see Proof of Audit for more information);
- **Via direct communication with TaxCore.API as described below**

Click here to see more details

1. E-SDC signals the beginning of the audit to the Secure Element (invokes *Start Audit* APDU command);
2. The Secure Element returns ARP (260 bytes) to the E-SDC;
3. E-SDC starts the audit by sending audit data (over HTTPS). ARP is submitted to the tax authority's system using the same communication channel;
4. If the request is correct, the system returns the HTTP status code 200 (OK);
5. Tax authority's system generates a set of Commands (including the POA command);
6. E-SDC submits a commands request to the tax authority's system via one of the following channels:
    - by invoking TaxCore.API service Notify Online Status
    - by invoking TaxCore.API service Submit Audit Package
7. Tax authority's system returns the commands to the E-SDC as a response;
8. E-SDC processes the commands as described in E-SDC Executes Commands, i.e. passes the payload to the *End Audit* APDU command;
9. The Secure Element resets its current unaudited amount to 0.00 and returns an OK message to the E-SDC;
10. E-SDC reports the results of the commands' execution to TaxCore.API as described in Notify Commands Processed.

Local Audit

Local audit initiated by a taxpayer is a common scenario for devices that lack the ability to connect to the internet due to the technical limitations of the devices or limited infrastructure.

Unlike in the Remote Audit process, during the Local Audit, the E-SDC doesn't submit the ARP file and audit packages to TaxCore.API. Instead, those files are copied to an SD Card or a USB Flash Drive.

Local audit step-by-step:

1. An audit is initiated by inserting an SD card or a USB Flash drive into an E-SDC device
2. E-SDC signals the beginning of the audit to the Secure Element (invokes *Start Audit* APDU command)
3. E-SDC copies audit packages to external memory unit (e.g. SD card, USB flash drive) in a piecemeal fashion, starting with the oldest unaudited package, as described in E-SDC Stores Audit Files on SD Card

or USB Drive

NOTE:
Audit packages remain stored in E-SDC's local memory until it receives a Proof-of-Audit command from TaxCore.API.

4. The Secure Element returns ARP (260 bytes) to the E-SDC

NOTE:
When performing the Local Audit, the ARP file should be generated and saved each time when at least one audit package is submitted to the tax authority's system, by using an external memory unit (see File-based communication).

5. An operator uploads the audit packages and the ARP file to the tax authority's system (by using the Taxpayer Administration Portal or the tax authority's Back Office Portal)
6. After the successful upload, a set of Commands (including the Proof of Audit command) are generated by the tax authority's system
7. An operator downloads the commands to an external memory unit (by using the Taxpayer Administration Portal or the tax authority's Back Office Portal)
8. An operator inserts the external memory unit with the commands file into an E-SDC, which processes the commands as described in E-SDC Executes Commands, i.e. passes the payload to the *End Audit* APDU command

NOTE:
The Proof of Audit command might be missing due to many reasons, such as: not all packages have been submitted, the tax authority system is experiencing a delay in data processing, etc. In some cases, the Proof of Audit command for a specific local audit might never be returned.

9. The Secure Element resets its current unaudited amount to 0.00 and returns the OK message to the E-SDC
10. The E-SDC generates the `Commands Execution Results` structure (as described in Commands) and saves it to a file on the external media, as described in E-SDC stores a command execution result to the SD card or USB drive

11. The operator uploads the `Commands Execution Results` file to the tax authority's system (by using the Taxpayer Administration Portal or the tax authority's Back Office Portal).

Proof of Audit

Introduction

Proof-of-Audit (POA) is generated by the tax authority's system once all expected audit packages have been received and securely stored on the tax authority's system.

Even if the taxpayer is unable to send one or more audit packages due to the failure of the EFD component or some other reason, the tax authority can still issue a POA if it can determine the tax liability for that secure element.

A POA cycle begins with E-SDC initiating the *Start Audit APDU command* (**Audit Start**). A POA cycle finishes with the tax authority's system receiving a confirmation that the secure element has successfully executed the issued POA command (**Audit End**).

NOTE:
This article describes the default operation mode of the *Proof of Audit Service*. However, upon the tax authority's decision, the service can use different strategies to issue a POA.

For the currently applicable non-standard issuance strategies, see Currently Applicable Non-Standard Strategy for Issuing Proof of Audit.

There are two scenarios for obtaining a POA:

- **As part of the Local Audit process**;
- **Via direct communication with TaxCore.API**

NOTE:
Regardless of which scenario was used to initiate the POA cycle, once the commands (containing the POA command) have been generated by the tax authority's system, the cycle can be completed using the other scenario.

POA cycle frequency

Audit Start should be initiated periodically, and the length of the period between two Audit Starts should be set dynamically. In regular conditions, there is no reason to initiate an Audit more often than every 30 minutes; although this can be extended to a period of a couple of hours as well (depending on the turnover). There is no need to initiate unnecessary Audits if the secure element(s) does not register sale amounts that will cause it to cross its assigned limit.

However, if the secure element is reaching its limit, the Audit should be initiated immediately and the period between two Audit Starts should be shortened to 10 minutes (unless there were no new invoices created in that period).

NOTE:
Although the Audit Start should be initiated periodically, audit packages should be submitted continuously, as soon as they are created (regardless of the Audit Start and Audit End).

The [[*TaxCore.TaxCoreConfiguration.ElectronicMonitoringShortName*]] system needs about 10 minutes to process submitted data and generate a POA command, provided that there are no missing audit packages for that Audit.

Once the Secure Element receives a valid POA, the E-SDC can delete the audit packages included in that Audit.

NOTE:
E-SDC must store the SDC number of the last invoice created before an Audit Start (the last invoice included in that Audit) to know which invoices to delete upon receiving a valid POA.

Be mindful of these cases

Sometimes, audit packages can arrive in [[*TaxCore.TaxCoreConfiguration.ElectronicMonitoringShortName*]] database after the Start Audit command - in that case, the system will again need at least 10 minutes after the arrival of the last audit package to generate the End Audit command (**see case 2a below**).

If two Audit Starts are initiated before an End Audit command is generated, the POA for the first Audit Start will not be valid. Only the next POA (covering both Audit Starts) will be valid (**see Case 3 below**). Moreover, the POA for the first Audit Start will be replaced by the next POA, so E-SDC will receive only the latest one. But if E-SDC happens to receive the first POA, it will be invalid and rejected by the Secure Element.

If one or more audit packages issued by the same secure element (same UID) do not arrive in the database, the End Audit command (POA) will never be generated.

Audit cycle cases

This means that there are three possible scenarios for completing the Audit cycle:

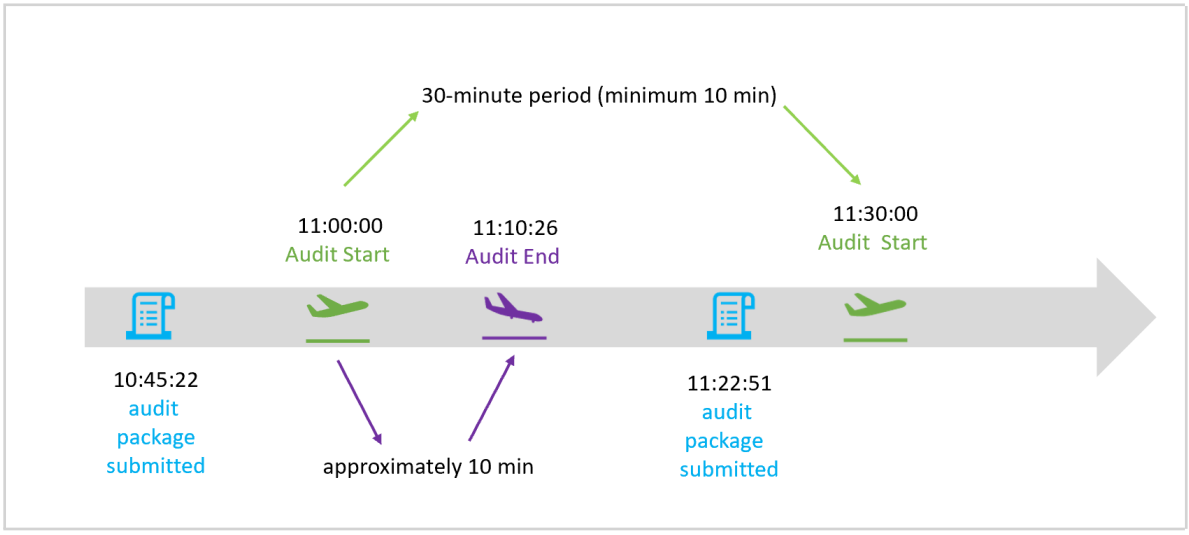
1. One audit package is created and one Audit Start is initiated between completing two Audit Ends (two POAs) - **see Case 1 below**
2. Multiple audit packages are created and one Audit Start is initiated between two Audit Ends (two POAs) - **see Case 2 below**
    - Sometimes, an E-SDC can initiate an Audit Start before submitting all the audit packages from that Audit. In that case, the system will wait for the last audit package from that Audit to arrive before it starts generating the End Audit command (POA) - **see Case 2a below**
3. Multiple audit packages are created and multiple Audit Starts are initiated between two Audit Ends (two POAs) - **see Case 3 below**

Case 1 – Audit is performed after the creation of each audit package

This is the simplest case, where no additional audit packages are generated during the whole audit process, as follows:

1. Create an audit package
2. Initiate the Audit process by invoking the Start Audit APDU command
3. Receive POA and pass it to the End Audit APDU command
4. If EndAudit returns the value "true," you can safely delete the audit package(s)
5. If EndAudit returns the value "false," continue until a valid POA is received
6. The period until the next Audit Start must be **at least 10 minutes (recommended from 30 minutes to a couple of hours)** after the previous Audit Start

The figure below illustrates the process:

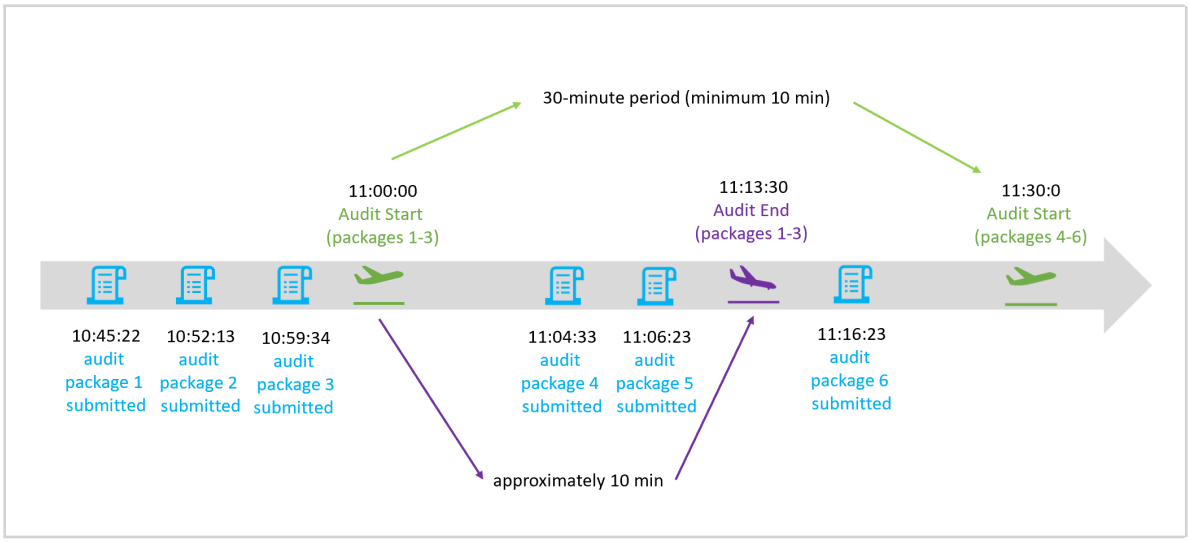


Case 2 – Audit is performed after multiple audit packages have been created

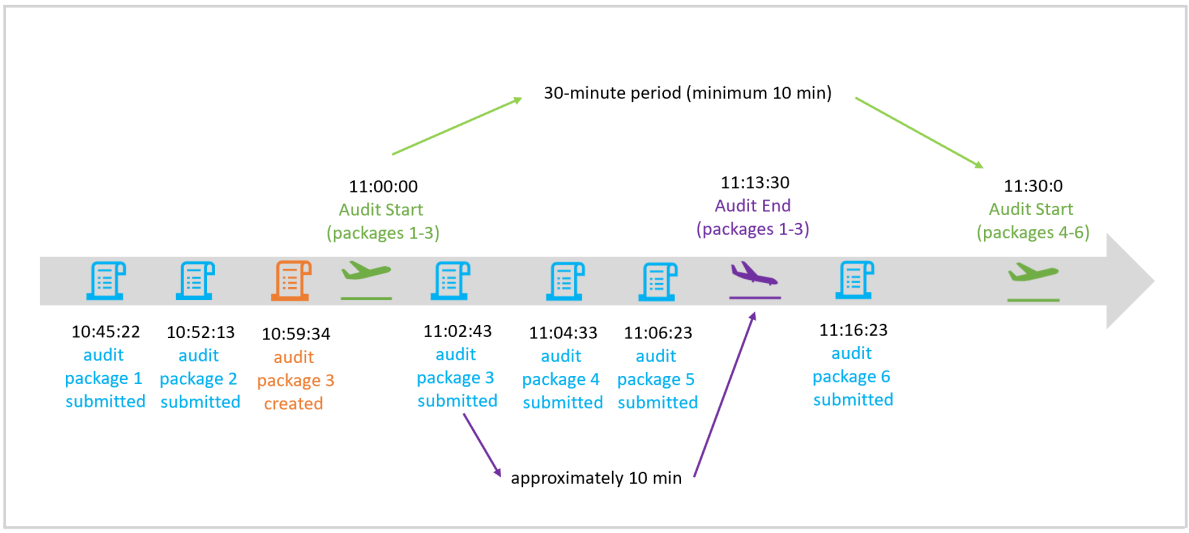
In this case, new packages can be created after an audit start:

1. Create audit packages 1-3 (as shown on the diagram)
2. Initiate the audit process by invoking the Start Audit APDU command
3. Continue to fiscalize invoices and create audit packages 4-6
4. Receive POA and pass it to the End Audit APDU command
5. If EndAudit APDU command returns value true, you can delete remaining audit packages 1-3 because it is the last initial audit being invoked by E-SDC. Audit packages 4-6 are created after the call to BeginAudit APDU command so they are not audited in this cycle
6. If EndAudit APDU command returns value false, continue (return to point 1) until valid POA is received
7. The period until the next Audit Start must be **at least 10 minutes (recommended from 30 minutes to a couple of hours)** after the previous Audit Start

The figure below illustrates the process:



Case 2a - Audit Start is initiated before all audit packages are submitted

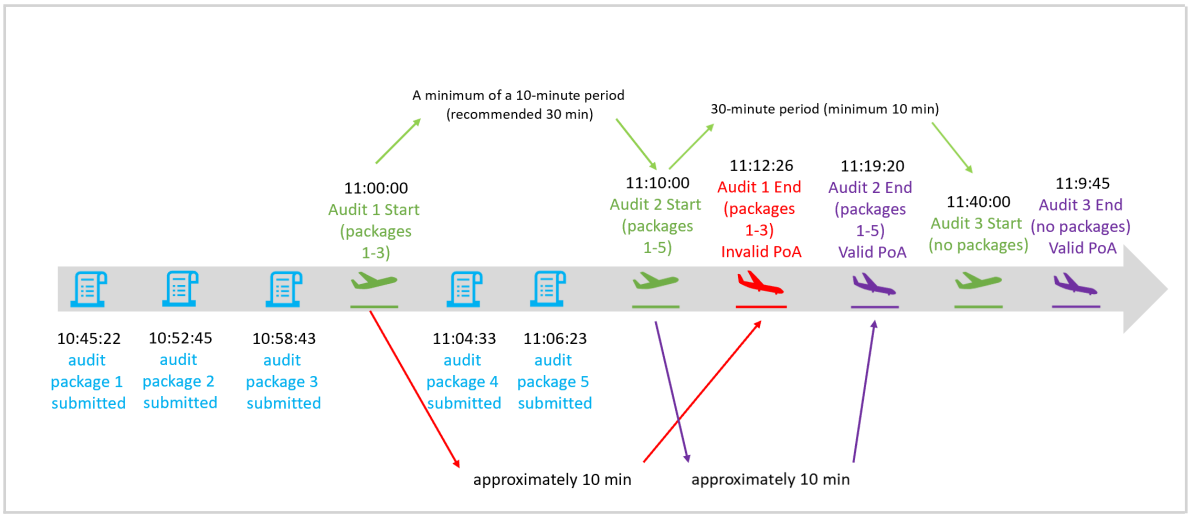


Case 3 – Audit is started multiple times before POA is generated

This case involves multiple audit starts:

1. Create Audit Packages 1-3
2. Initiate the Audit process by invoking the Start Audit APDU command
3. Continue to fiscalize invoices and create Audit Packages 4 and 5
4. Initiate another audit process by invoking the Start Audit APDU command – the previous audit is canceled
5. Receive POA and pass it to End Audit APDU command
6. If EndAudit returns value true, you can delete remaining audit packages 1-5 because it is the last BeginAudit being invoked by E-SDC
7. If EndAudit APDU command returns value false, continue until valid POA is received
8. POA generated for the first Audit Start (Audit 1 below) is not considered valid. Only the POA generated for the last Audit Start (Audit 2 below) is considered valid and will be forwarded to the secure element
9. The period until the next Audit Start must be **at least 10 minutes (recommended from 30 minutes to a couple of hours)** after the previous Audit Start.

The figure below illustrates the process:



Notifications

Introduction

E-SDC device shall have an appropriate way to show the status of the device, information about the smart card and processes running on the E-SDC.

A cashier could get the device notifications by receiving an onscreen message, by observing the colors from the light-emitting diodes (LED) or any other similar component set for displaying visual notifications.

Required Notification

The following visual notifications shall be available to a cashier:

1. Smartcard is inserted but the E-SDC is not yet configured with the tax rates, verification URL or NTP service address. This is a common situation before initialization commands are executed by E-SDC;
2. Enter PIN Code for the Secure element – Smart card is inserted but E-SDC has not received the PIN Code from POS;
3. E-SDC is ready to sign an invoice;
4. Smart card is missing or unavailable;
5. Audit package transfer is in progress (Local audit on an SD card or USB flash drive, or an online audit);
6. Firmware update is in progress (if applicable);
7. Audit data storage is almost full;
8. Audit data storage is full;
9. Time for audit;
10. Commands in progress (currently running)

Switching Smart Cards During Operation

During normal operation, taxpayers/cashiers might switch the smart card they are using for issuing fiscal invoices.

In that case, the E-SDC must perform the following activities:

- **if the new smart card is for the same environment** - the E-SDC will first submit the unsubmitted invoices that were created with the previously used smart card
- **if the new smart card is for a different environment** - the E-SDC will keep the unsubmitted invoices (created with the previously used smart card) in its internal memory until they can be submitted (old smart card is returned)

For more information about different environments, see Identification of Environments and Important Endpoints.

E-SDC Logging

Introduction

E-SDC must keep a log about all required error events. It must log every error chronologically by local date and time (exact hour and minute).

E-SDC log must be available for easy export (download, USB flash drive...) and presented in a human-readable format.

Required Logging

The following error events must be logged:

- Any Invoice Request sent by POS that E-SDC failed to process
- Any APDU error returned by SE
- Any error returned by TaxCore
- Any error caused by internal E-SDC operations
- Any error during the E-SDC Initialization process (Enter PIN and Command processing).
- Any error during the Invoice Fiscalization process
- Any error during the Audit process (Local and Remote).
- Any error during the process of Date and Time Synchronization

The above errors are the minimum requirements, but E-SDC can also keep a log of other events.

Malfunctions and Non-serviceable Devices

Dump Audit Packages Kept on E-SDC when Secure element is damaged

If the Secure Element is damaged and its data cannot be restored from the card, but the E-SDC is operational, the tax authority system shall be able to dump data from the E-SDC device and upload the audit packages using the same application used to upload audit packages submitted by a taxpayer.