Fiscalization of an Invoice

Introduction

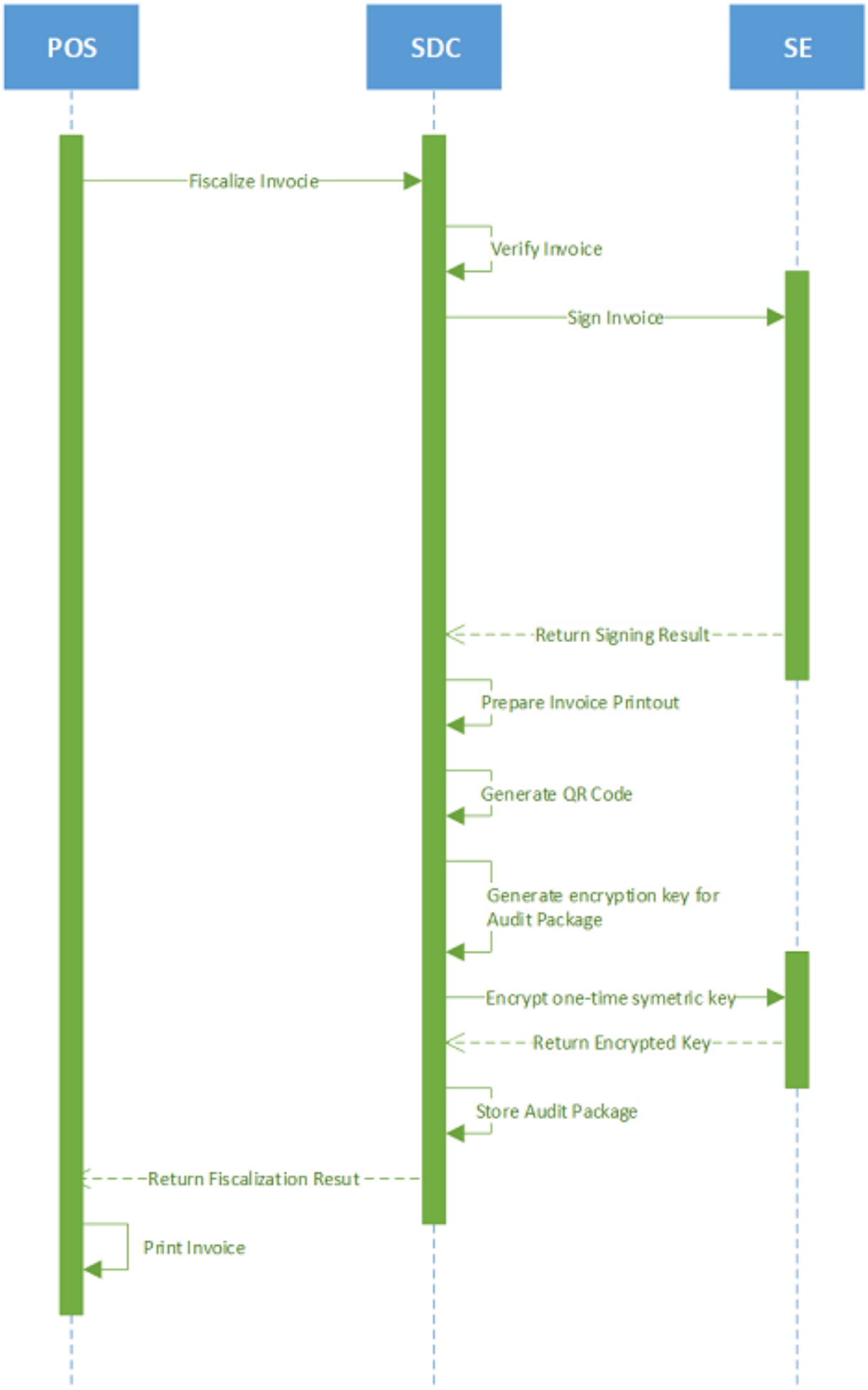
Invoice fiscalization is the main function of an E-SDC. Fiscalization is the process of handling an invoice request from an accredited invoicing system to produce fiscal invoices.

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.