Point Contactless

Document Name	: point_contactless_10.pdf
Document Type	: Technical Interface Specification
Version	: 1.0
Date	: 09-03-2012
Confidentiality	: YES
By	: Carsten Sharpe Mikkelsen

POINT TRANSACTION SYSTEMS A/S
Knapholm 7, 2730 Herlev, Tlf.: 44 53 16 10 Fax: 44 53 46 20 www.point.dk
Reg.nr. A/S 202086 CVR nr. 15 40 12 81

Content

Introduction

We plan to use four message types in the communication between the terminal and the host, two in each direction. All messages are sent in XML format complying with the EPAS specifications.

The transaction flow is described in detail in section “Flow of Transactions (Sequence Diagrams)” on page 18.

Other Documentation

The official documentation for the ISO 20022 standard can be found here: ISO 20022 Documentation.
A more detailed usage guide of all available commands can be found here: EPAS Usage Guide.

Format for Sending

All messages are prefixed by four bytes specifying the length of the message. The length is sent in network order, i.e. most significant byte first.

Message Structure

Every message contains three building blocks, a header, a message block and a security trailer. The security trailer contains a message authentication code, computed on the message building block with a cryptographic key. It allows the authentication of the initiator and protects the content of the message building block against any unauthorised alteration. The security trailer is only described in the AcceptorAuthorisationRequest section as it is identical, except for the MAC value, for all section types.

The documentation is built up in tables with this form:

Message Item	XML Tag	Description
Element		Description or possible value.
AnotherElement		
SubElement		**Possible value**: Explanation.

Notes on the Security Trailer

As the platform does not support DUKPT (Derived Unique Key Per Transaction) at the moment, we are using a trimmed version of the security trailer that only includes the actual MAC and no info about the KEK (key encryption key). This means that the security trailer is not EPAS compliant until the platform supports DUKPT.

AcceptorAuthorisationRequest

The AcceptorAuthorisationRequest message is sent by the card acceptor to the acquirer or its agent when an online authorisation is required for the card payment transaction.

Message Item	XML Tag	Description
AcceptorAuthorisationRequest	AccptrAuthstnReq	

Message Item	XML Tag	Description
Header	Hdr	
MessageFunction	MsgFctn	**AUTQ**: Request for authorisation without financial capture. (*TransactionCapture*=FALSE) **FAUQ**: Request for authorisation with financial capture. (*TransactionCapture*=TRUE)
ProtocolVersion	PrtcolVrsn	MM.mm (assigned by EPASOrg). Current version is 1.0.
ExchangeIdentification	XchgId	Used in combination with *CreationDateTime* to allow the Recipient to identify retransmissions. It is a cyclic counter that increments by one with each new message, starting at 0.
CreationDateTime	CreDtTm	Time accuracy has to be at least tenth of a second. (ISO 8601 format)
InitiatingParty	InitgPty	
Identification	Id	Terminal id.

Message Item	XML Tag	Description
AuthorisationRequest	AuthstnReq	
Environment	Envt	
POI	POI	
Identification	Id	
Identification	Id	Terminal id.
Card	Card	
PlainCardData	PlainCardData	
PAN	PAN	n-digit PAN without spaces.
ExpiryDate	XpryDt	Format: YYYY-MM
Context	Cntxt	
PaymentContext	PmtCntxt	
CardDataEntryMode	CardDataNtryMd	**CTLS**: Contactless proximity reader. **MGST**: Magnetic stripe.
Transaction	Tx	
TransactionCapture	TxCaptr	TRUE/FALSE based on *MessageFunction*.
TransactionType	TxTp	**BALC**: Balance enquiry. **CACT**: Card activation. **CAFT**: Transfer of funds to and/or from a card account. **CAVR**: Card verification. **CRDP**: Card payment. **RFND**: Refund transaction. **VALC**: Card validity check.
MerchantCategoryCode	MrchntCtgyCd	Category code conform to ISO 18245, related to the type of services or goods the merchant provides for the transaction. List of MCC codes: http://www.irs.gov/irb/2004-31_IRB/ar17.html
TransactionIdentification	TxId	
TransactionDateTime	TxDtTm	UTC date time with offset or local date time.

		(ISO 8601 format)
TransactionReference	TxRef	Identification of the transaction that has to be unique for a time period. Max 35 characters.
TransactionDetails	TxDtls	
Currency	Ccy	
TotalAmount	TtlAmt	Use a “.” (dot) as decimal point.

Message Item	XML Tag	Description
SecurityTrailer	Scty	
ContentType	CnttTp	**AUTH**: MAC (Message Authentication Code), with encryption key - (ASN.1 Object Identifier: id-ct-authData).
AuthenticatedData	AuthntcdData	Data protection by a message authentication code (MAC).
Recipient	Rcpt	Information related to the transport key.
KEK	KEK	Encryption key using previously distributed symmetric key.
KEKIdentification	KEKId	
KeyIdentification	KeyId	Maximum 140 characters.
KeyVersion	KeyVrsn	Activation date or version of the key to differentiate several keys with the same name (*KeyIdentification*) using the format YYYYMMDDhh where: YYYY is a 4-digits numeral representing the year, 0000 is prohibited MM is a 2-digits numeral representing the month (from 01 to 12) DD is a 2-digits numeral representing the day of the month (from 01 to 31) hh is a 2-digits numeral representing the hours (from 00 to 23)
DerivationIdentification	DerivtnId	Identification used for derivation of a unique key from a master key provided for the data protection. Between 5 and 16 bits.
KeyEncryptionAlgorithm	KeyNcrptnAlgo	Algorithm to encrypt the KEK.
Algorithm	Algo	**DKPT**: DUKPT (Derived Unique Key Per Transaction) algorithm, as specified in ANSI X9.24-2004, Annex A, and ISO/DIS 13492-2006. - (ASN.1 Object Identifier: id-dukpt-wrap).
EncryptedKey	NcrptdKey	Maximum 140 bits.
MACAlgorithm	MACAlgo	Algorithm to compute MAC.
Algorithm	Algo	**MCCS**: Retail-CBC-MAC with SHA-256 (Secure Hash standard) - (ASN.1 Object Identifier: id-retail-cbc-mac-sha-256).
EncapsulatedContent	NcpsltdCntt	Data to authenticate.
ContentType	CnttTp	**DATA**: Generic, non cryptographic or unqualified data content - (ASN.1 Object Identifier: id-data).
MAC	MAC	Encrypted data which authenticates the data.

Below is shown an example AcceptorAuthorisationRequest with a full security trailer, which includes the “Recipient” section containing the info about the KEK (key encryption key).

Example

Basic merchant info

Merchant type	Men’s, Women’s Clothing Store (**5691**)
Terminal Id	990001

Payment card info

PAN	1234 1234 1234 1234
Expiration date	December 2014 (**2014-12**)
Type	Contactless card (**CTLS**)

Transaction info

Start time	January 24, 2012 @ 9 am (**2012-01-24T09:00:00.00**)
Type	Card payment (**CRDP**)
Currency	Euro (**Eur**)
Amount	20.00

The resulting XML file is shown in

Type of data protection	AuthenticatedData/MAC (**AUTH**)
KeyIdentification	SpecV1TestKey[#]
KeyVersion	2010060715[#]
DerivationIdentification	`398725A501` (**OYclpQE=**)[#]
Key Encryption Algorithm	DUKPT (**DKPT**)
EncryptedKey	`E290200017` (**4pAgABc=**)[#]
MAC Algorithm	RetailSHA256MAC (**MCCS**)
Type of data	PlainData (**DATA**)
MAC	`15 20 4F 17 68 48 5B 13` (**FSBPF2hIWxM=**)[£]

#: Values taken directly from example “8.5.3.2 MAC Computation” in *EPAS Usage guide*, page 306 & 307.

£: Key used to compute MAC is the derived key found on page 306 in *EPAS Usage guide*.
Key = `5E 64 F1 AB F2 5D 3B A1 7F 62 9E C2 B3 02 F8 EA`.

Table 1.

Type of data protection	AuthenticatedData/MAC (**AUTH**)
KeyIdentification	SpecV1TestKey[#]
KeyVersion	2010060715[#]
DerivationIdentification	398725A501 (**OYclpQE=**)[#]
Key Encryption Algorithm	DUKPT (**DKPT**)
EncryptedKey	E290200017 (**4pAgABc=**)[#]
MAC Algorithm	RetailSHA256MAC (**MCCS**)
Type of data	PlainData (**DATA**)
MAC	15 20 4F 17 68 48 5B 13 (**FSBPF2hIWxM=**)[£]

#: Values taken directly from example “8.5.3.2 MAC Computation” in *EPAS Usage guide*, page 306 & 307.

£: Key used to compute MAC is the derived key found on page 306 in *EPAS Usage guide*.
Key = 5E 64 F1 AB F2 5D 3B A1 7F 62 9E C2 B3 02 F8 EA.

Table 1: AcceptorAuthorisationRequest Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:caaa.003.001.01">
    <AccptrAuthstnReq>
        <Hdr>
            <MsgFctn>FAUQ</MsgFctn>
            <PrtcolVrsn>1.0</PrtcolVrsn>
            <XchgId>0</XchgId>
            <CreDtTm>2012-01-24T09:00:05.00+01:00</CreDtTm>
            <InitgPty>
                <Id>990001</Id>
            </InitgPty>
        </Hdr>
        <AuthstnReq>
            <Envt>
                <POI>
                    <Id>
                        <Id>990001</Id>
                    </Id>
                </POI>
                <Card>
                    <PlainCardData>
                        <PAN>1234123412341234</PAN>
                        <XpryDt>2014-12</XpryDt>
                    </PlainCardData>
                </Card>
            </Envt>
            <Cntxt>
                <PmtCntxt>
                    <CardDataNtryMd>CTLS</CardDataNtryMd>
                </PmtCntxt>
            </Cntxt>
            <Tx>
                <TxCaptr>true</TxCaptr>
                <TxTp>CRDP</TxTp>
                <MrchntCtgyCd>5691</MrchntCtgyCd>
                <TxId>
                    <TxDtTm>2012-01-24T09:00:00.00+01:00</TxDtTm>
                    <TxRef>000001</TxRef>
                </TxId>
                <TxDtls>
                    <Ccy>EUR</Ccy>
                    <TtlAmt>20.00</TtlAmt>
                </TxDtls>
            </Tx>
        </AuthstnReq>
        <SctyTrlr>
            <CnttTp>AUTH</CnttTp>
            <AuthntcdData>
                 <Rcpt>
                    <KEK>
                        <KEKId>
                            <KeyId>SpecV1TestKey</KeyId>
                            <KeyVrsn>2010060715</KeyVrsn>
                            <DerivtnId>OYclpQE=</DerivtnId>
                        </KEKId>
                        <KeyNcrptnAlgo>
                            <Algo>DKPT</Algo>
                        </KeyNcrptnAlgo>
                        <NcrptdKey>4pAgABc=</NcrptdKey>
                    </KEK>
                </Rcpt>
                <MACAlgo>
                    <Algo>MCCS</Algo>
                </MACAlgo>
                <NcpsltdCntt>
                    <CnttTp>DATA</CnttTp>
                </NcpsltdCntt>
                <MAC>FSBPF2hIWxM=</MAC>
            </AuthntcdData>
        </SctyTrlr>
    </AccptrAuthstnReq>
</Document>
```

AcceptorAuthorisationResponse

The AcceptorAuthorisationResponse message is sent by the acquirer to inform the card acceptor of the outcome of the authorisation process.

The AcceptorAuthorisationResponse message is used to indicate one of the possible outcomes of an authorisation process:

- A successful authorisation
- A decline from the acquirer for financial reasons
- A decline from the acquirer for technical reasons (for instance, a timeout).

Message Item	XML Tag	Description
AcceptorAuthorisationResponse	AccptrAuthstnRspn	

Message Item	XML Tag	Description
Header	Hdr	
MessageFunction	MsgFctn	**AUTP**: Response for authorisation without financial capture. (*TransactionCapture*=FALSE) **FAUP**: Response for authorisation with financial capture. (*TransactionCapture*=TRUE)
ProtocolVersion	PrtcolVrsn	Copy from AcceptorAuthorisationRequest.
ExchangeIdentification	XchgId	Copy from AcceptorAuthorisationRequest.
CreationDateTime	CreDtTm	Date and time of the creation of the message response. Time accuracy has to be at least tenth of a second. (ISO 8601 format)
InitiatingParty	InitgPty	Copy from AcceptorAuthorisationRequest.
Identification	Id	

Message Item	XML Tag	Description
AuthorisationResponse	AuthstnRspn	
Environment	Envt	
POI Identification	POIId	
Identification	Id	Can be different from the request.
Transaction	Tx	
TransactionIdentification	TxId	Copy from AcceptorAuthorisationRequest.
TransactionDateTime	TxDtTm	
TransactionReference	TxRef	
TransactionDetails	TxDtls	Copy from AcceptorAuthorisationRequest.
Currency	Ccy	
TotalAmount	TtlAmt	
TransactionResponse	TxRspn	
AuthorisationResult	AuthstnRslt	
ResponseToAuthorisation	RspnToAuthstn	
Response	Rspn	**APPR**: (Approved) Authorisation is approved for the full amount requested, including capture if requested. **DECL**: (Declined) Authorisation is declined or the requested capture is not performed. **TECH**: (TechnicalError) Service cannot be provided for technical reason (e.g. timeout contacting the Issuer, security problem).
Balance	Bal	Balance of the account, related to the payment. (Optional)

Table 2: AcceptorAuthorisationResponse Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns="urn:iso:std:iso:20022:tech:xsd:caaa.003.001.01">
    <AccptrAuthstnRspn>
        <Hdr>
            <MsgFctn>FAUP</MsgFctn>
            <PrtcolVrsn>1.0</PrtcolVrsn>
            <XchgId>0</XchgId>
            <CreDtTm>2012-01-24T09:00:05.00+01:00</CreDtTm>
            <InitgPty>
                <Id>990001</Id>
            </InitgPty>
        </Hdr>
        <AuthstnRspn>
            <Envt>
                <POIId>
                    <Id>990001</Id>
                </POIId>
            </Envt>
            <Tx>
                <TxId>
                    <TxDtTm>2012-01-24T09:00:00.00+01:00</TxDtTm>
                    <TxRef>000001</TxRef>
                </TxId>
                <TxDtls>
                    <Ccy>EUR</Ccy>
                    <TtlAmt>20.00</TtlAmt>
                </TxDtls>
            </Tx>
            <TxRspn>
                <AuthstnRslt>
                    <RspnToAuthstn>
                        <Rspn>APPR</Rspn>
                    </RspnToAuthstn>
                </AuthstnRslt>
            </TxRspn>
        </AuthstnRspn>
    </AccptrAuthstnRspn>
</Document>
```

AcceptorCompletionAdvice

The AcceptorCompletionAdvice message is sent by a card acceptor to notify an acquirer about the completion and final outcome of a card payment transaction.

The AcceptorCompletionAdvice message is used either to:

- Reverse a transaction which was not successfully completed (for example, cancellation of transaction by the cardholder), but where an authorisation had been previously given. A reversal **must always** be approved!

Message Item	XML Tag	Description
AcceptorCompletionAdvice	AccptrCmpltnAdvc	

Message Item	XML Tag	Description
Header	Hdr	
MessageFunction	MsgFctn	**FRVA**: Advice for reversal with financial capture. (*TransactionCapture*=TRUE, *Reversal*=TRUE) **RVRA**: Advice for reversal without financial capture. (*TransactionCapture*=FALSE, *Reversal*=TRUE)
ProtocolVersion	PrtcolVrsn	MM.mm (assigned by EPASOrg). Current version is 1.0.
ExchangeIdentification	XchgId	Used in combination with *CreationDateTime* to allow the Recipient to identify retransmissions. It is a cyclic counter that increments by one with each new message, starting at 0.
CreationDateTime	CreDtTm	Time accuracy has to be at least tenth of a second. (ISO 8601 format)
InitiatingParty	InitgPty	
Identification	Id	Terminal id.

Message Item	XML Tag	Description
CompletionAdvice	CmpltnAdvc	
Environment	Envt	
POI	POI	
Identification	Id	
Identification	Id	Terminal id.
Card	Card	
PlainCardData	PlainCardData	
PAN	PAN	n-digit PAN without spaces.
ExpiryDate	XpryDt	Format: YYYY-MM
Transaction	Tx	
TransactionCapture	TxCaptr	TRUE/FALSE based on *MessageFunction*.
MerchantCategoryCode	MrchntCtgyCd	Category code conform to ISO 18245, related to the type of services or goods the merchant provides for the transaction. List of MCC codes: http://www.irs.gov/irb/2004-31_IRB/ar17.html
TransactionIdentification	TxId	
TransactionDateTime	TxDtTm	UTC date time with offset or local date time. (ISO 8601 format)
TransactionReference	TxRef	Identification of the transaction that has to be unique for a time period. Max 35 characters.
TransactionSuccess	TxSucss	TRUE/FALSE.

Reversal	Rvsl	TRUE/FALSE based on *MessageFunction*.
TransactionDetails	TxDtls	
Currency	Ccy	
TotalAmount	TtlAmt	Use a “.” (dot) as decimal point.

Table 3: AcceptorCompletionAdvice Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns="urn:iso:std:iso:20022:tech:xsd:caaa.003.001.01">
    <AccptrCmpltnAdvc>
        <Hdr>
            <MsgFctn>FRVA</MsgFctn>
            <PrtcolVrsn>1.0</PrtcolVrsn>
            <XchgId>0</XchgId>
            <CreDtTm>2012-01-24T09:00:05.00+01:00</CreDtTm>
            <InitgPty>
                <Id>990001</Id>
            </InitgPty>
        </Hdr>
        <CmpltnAdvc>
            <Envt>
                <POI>
                    <Id>
                        <Id>990001</Id>
                    </Id>
                </POI>
                <Card>
                    <PlainCardData>
                        <PAN>1234123412341234</PAN>
                        <XpryDt>2014-12</XpryDt>
                    </PlainCardData>
                </Card>
            </Envt>
            <Tx>
                <TxCaptr>true</TxCaptr>
                <MrchntCtgyCd>5691</MrchntCtgyCd>
                <TxId>
                    <TxDtTm>2012-01-24T09:00:00.00+01:00</TxDtTm>
                    <TxRef>000001</TxRef>
                </TxId>
                <TxSucss>false</TxSucss>
                <Rvsl>true</Rvsl>
                <TxDtls>
                    <Ccy>EUR</Ccy>
                    <TtlAmt>20.00</TtlAmt>
                </TxDtls>
            </Tx>
        </CmpltnAdvc>
    </AccptrCmpltnAdvc>
</Document>
```

AcceptorCompletionAdviceResponse

The AcceptorCompletionAdviceResponse message is sent by the acquirer to acknowledge the proper receipt of an AcceptorCompletionAdvice.

Message Item	XML Tag	Description
AcceptorCompletionAdviceResponse	AccptrCmpltnAdvcRspn	

Message Item	XML Tag	Description
Header	Hdr	
MessageFunction	MsgFctn	**CMPK**: Advice response for completion without financial capture. **FCMK**: Advice response for completion with financial capture. **FRVR**: Advice response for reversal with financial capture. **RVRR**: Advice response for reversal without financial capture.
ProtocolVersion	PrtcolVrsn	MM.mm (assigned by EPASOrg). Current version is 1.0.
ExchangeIdentification	XchgId	Copy from AcceptorCompletionAdvice.
CreationDateTime	CreDtTm	Time accuracy has to be at least tenth of a second. (ISO 8601 format)
InitiatingParty	InitgPty	Copy from AcceptorCompletionAdvice.
Identification	Id	

Message Item	XML Tag	Description
CompletionAdviceResponse	CmpltnAdvcRspn	
Environment	Envt	
POIIdentification	POIId	
Identification	Id	Copy from AcceptorCompletionAdvice.
Transaction	Tx	
TransactionIdentification	TxId	Copy from AcceptorCompletionAdvice.
TransactionDateTime	TxDtTm	
TransactionReference	TxRef	
Response	Rspn	**APPR**: (Approved) Service has been successfully provided.

Table 4: AcceptorCompletionAdviceResponse Example

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xmlns="urn:iso:std:iso:20022:tech:xsd:caaa.003.001.01">
    <AccptrCmpltnAdvcRspn>
        <Hdr>
            <MsgFctn>FRVR</MsgFctn>
            <PrtcolVrsn>1.0</PrtcolVrsn>
            <XchgId>0</XchgId>
            <CreDtTm>2012-01-24T09:00:05.00+01:00</CreDtTm>
            <InitgPty>
                <Id>990001</Id>
            </InitgPty>
        </Hdr>
        <CmpltnAdvcRspn>
            <Envt>
                <POIId>
                    <Id>990001</Id>
                </POIId>
            </Envt>
            <Tx>
                <TxId>
                    <TxDtTm>2012-01-24T09:00:00.00+01:00</TxDtTm>
                    <TxRef>000001</TxRef>
                </TxId>
                <Rspn>APPR</Rspn>
            </Tx>
        </CmpltnAdvcRspn>
    </AccptrCmpltnAdvcRspn>
</Document>
```

Communication Security

Security Trailer

The following description is taken from the *EPAS Usage Guide* from page 288 onwards.

Key Management

Test key identification is distinguished from production key by a name including the suffix "TestKey".

The DUKPT key management mechanism uses 10 bytes of information (Key Serial Number or KSN) sent by the *InitiatingParty* in the message to uniquely identified the derived key at the *RecipientParty*.

This KSN contains the following information:

- Issuer Identification Number (3 bytes): a collision free 6 digit number which will ensure the uniqueness of the KSN.
- Merchant ID (1 byte): can be used by an acquirer or manufacturer to differentiate merchants from each other.
- Group ID (1 byte): can be used by an acquirer or manufacturer to classify devices for a given merchants.
- Device ID (19 bits): can be used to identify a device inside a specific group ID.
- Transaction Counter (21 bits): the counter value can be used to detect message replay.

The 3 first elements (5 bytes) are sent in the *Recipient.KEK.KEKIdentification.DerivationIdentification* item of the EnvelopedData component, the last 2 elements (5 bytes) are sent in the *Recipient.KEK.EncryptedKey* of the EnvelopedData component.

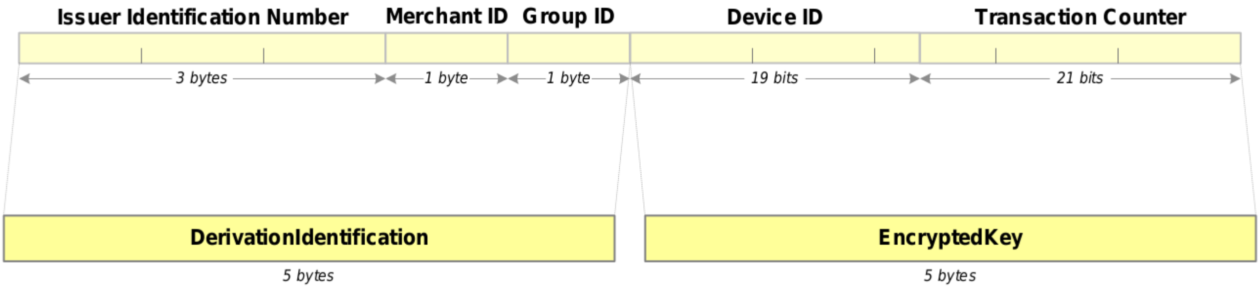


Figure 1: Key Serial Number Details

After derivation of the resultant key, a XOR with the hexadecimal value `00000000 0000FF00 00000000 0000FF00` (MACmask) is applied to the resultant key in order to use a variant of the key for MAC computation.

The same key is used for the MAC of a message request and its corresponding message response, i.e. the Base Derivation Key (as the Terminal Initial Key) and the KSN are the same.

MAC Computation

The following explanation is taken from the *EPAS Usage guide*, page 296.

MAC computation uses Triple DES algorithm with double length key (112 Bit), using the retail CBC (Chaining Block Cipher) mode as defined in ISO 9807 and ANSI X9.19, on the result of the SHA-256 digest of the message body as defined in FIPS 180-1 and 2. Before encryption, the digest is padded according to the ISO/DIS 7816-4.

MAC computation and MAC verification use the same algorithm presented below.

<u>MAC Computation Process:</u>

(i) Compute the SHA-256 digest D on the body of the message, including the XML envelope, and as transmitted by the transport level.

- For the MAC verification of a received message, the digest is computed on the body as received by the transport level.
- For the MAC generation of a message to send, the body shall have no transformation after the computation of the digest.

(ii) Padding of the data to encrypt D: the hexadecimal byte 80 is added to D. If the new length is not a multiple of 8, D is extended by null bytes (hexadecimal 00), to reach a length multiple of 8.

(iii) The result D of the padded data is split in blocks of 8 bytes $D_1 ... D_n$.

(iv) With the left part K_L of key K, and initialising C_0 by 8 null bytes, compute the sequence $C_1 ... C_{n-1}$, where

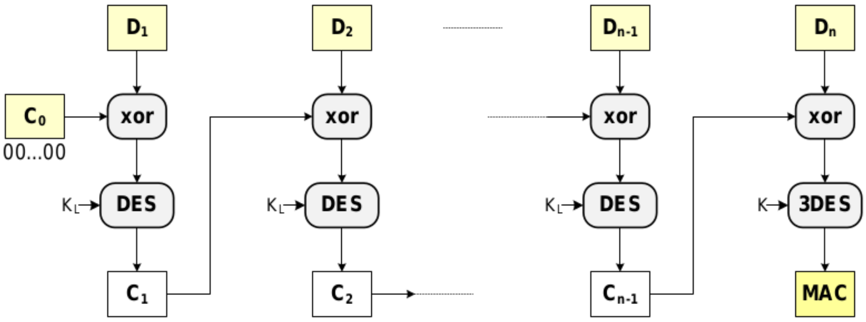
$$C_i = E_{KL} (C_{i-1} \text{ XOR } D_i)$$

E_{KL} being the DES encryption with K_L.

(v) The MAC is the result of:

$$MAC = E_K (C_{n-1} \text{ XOR } D_n)$$

E_K being the Triple-DES encryption with K.



Figur 1: MAC Computation Process

Example

We use a small example to show how to compute the MAC. The example is from the *EPAS Usage guide*.

The XML message is:

```
<DgnstcReq>
    <Envt>
        <AcqrrParamsVrsn>2010-01-01T08:00:00</AcqrrParamsVrsn>
        <MrchntId>
            <Id>EPASMER001</Id>
            <Tp>MERC</Tp>
        </MrchntId>
        <POIId>
            <Id>66000001</Id>
            <Tp>OPOI</Tp>
        <Issr>ACQR</Issr>
        </POIId>
    </Envt>
</DgnstcReq>
```

Derived key: `5E 64 F1 AB F2 5D 3B A1 7F 62 9E C2 B3 02 F8 EA`.

The SHA256 digest of the *DiagnosticRequest* message body is:

```
0000  C4 11 A9 4F 56 97 8E A1 8B 9D CA F4 A0 DE 5B 44
0010  09 BE A9 93 87 58 1A CA E5 01 3D 4A 55 38 AF B0
```

This message is then with the byte 0x80 followed by 7 null bytes:

```
0000  C4 11 A9 4F 56 97 8E A1 8B 9D CA F4 A0 DE 5B 44
0010  09 BE A9 93 87 58 1A CA E5 01 3D 4A 55 38 AF B0
0020  80 00 00 00 00 00 00 00
```

Now we encrypt the first 32 bytes using DES CBC and the left half of the shared key:

```
0000  0C 39 D3 CF 05 F9 F4 97 E0 1E 69 DE 5F 23 F8 72
0010  81 EC 98 C5 B4 12 CD A4 19 E8 06 D6 F2 03 9F B3
```

We encrypt the final 8 bytes using 3DES CBC and get:

```
0000  0C 39 D3 CF 05 F9 F4 97 E0 1E 69 DE 5F 23 F8 72
0010  81 EC 98 C5 B4 12 CD A4 19 E8 06 D6 F2 03 9F B3
0020  21 86 58 17 8E B7 E8 F6
```

The MAC is the last 8 bytes: `21 86 58 17 8E B7 E8 F6`

The base64 conversion of this value is: `IYZYF4636PY=`

The resulting security trailer with no info about the KEK looks like:

```
<SctyTrlr>
    <CnttTp>AUTH</CnttTp>
    <AuthntcdData>
        <MACAlgo>
            <Algo>MCCS</Algo>
        </MACAlgo>
        <NcpsltdCntt>
            <CnttTp>DATA</CnttTp>
        </NcpsltdCntt>
        <MAC>IYZYF4636PY=</MAC>
    </AuthntcdData>
</SctyTrlr>
```

Encrypting the Communication

The communication between the terminal and the server can be secured using SSL encryption. If this shall be enabled, the integrator must send the CA certificate to Point. The terminal does not use client certificate. The terminal will validate the server certificate and ensure that the server certificate is signed by the CA and that the Common Name (CN) is the same as the IP address of the server.

The terminal supports the cipher “AES256-SHA”. It is up to the integrator to make sure that the latest requirements from PAN Nordic and PCI regarding key sizes are met.

Please note that you should create a new certificate for each server installation. All of these certificates should be signed by the same CA certificate, which is the certificate that you sent to Point.

Flow of Transactions (Sequence Diagrams)

All transactions can be put into one of two boxes, depending on whether the request was successful or not. In a successful case we send the AcceptorAuthorisationRequest and receive an approved AcceptorAuthorisationResponse from the Acquirer. In all other cases, we finish the message exchange by sending an AcceptorCompletionAdvice, telling the Acquirer to abort the transaction. The possible different scenarios are shown below.

Successful Authentication

In Figure 2 is shown a successful transaction, thus no AcceptorCompletionAdvice is sent.

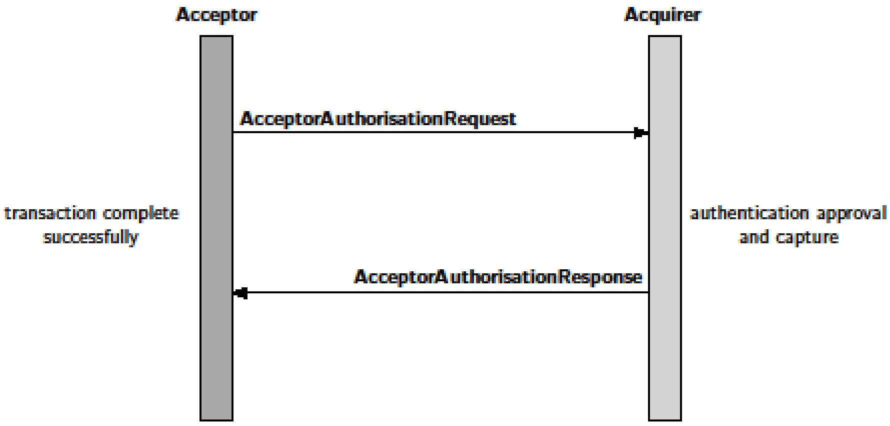

Acceptor
Acquirer
AcceptorAuthorisationRequest
transaction complete successfully
authentication approval and capture
AcceptorAuthorisationResponse

Figure 2: Successful authentication and money transfer

Failed Authentication

Figure 3 shows a transaction where the authentication fails. The Acquirer informs the Acceptor and the Acceptor acknowledges the response by sending an AcceptorCompletionAdvice.

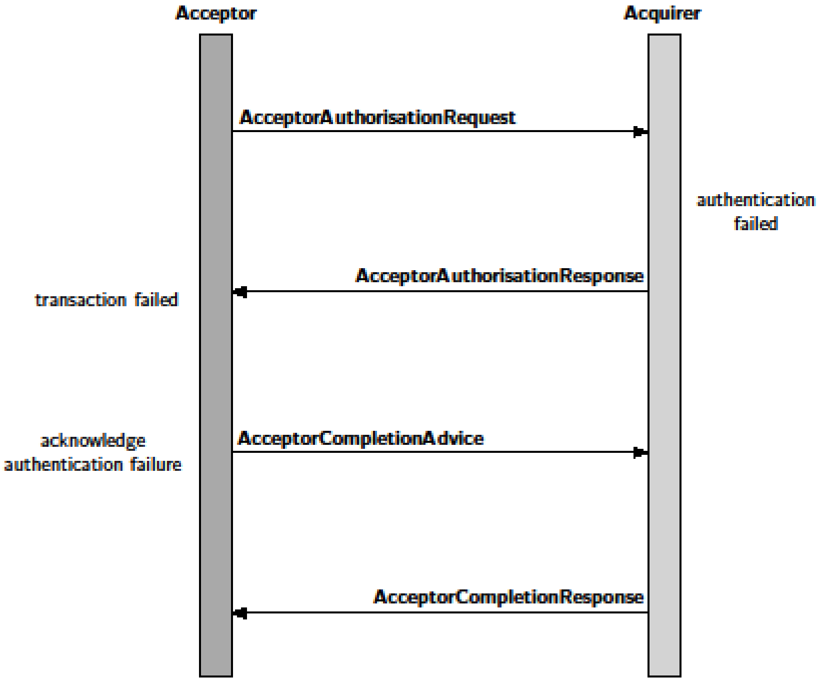

Acceptor
Acquirer
AcceptorAuthorisationRequest
authentication failed
AcceptorAuthorisationResponse
transaction failed
acknowledge authentication failure
AcceptorCompletionAdvice
AcceptorCompletionResponse

Figure 3: Authentication failure

No Authorisation Response

Figure 4 shows an example of a communication error. In this example the Acceptor never receives the AcceptorAuthorisationResponse from the Acquirer. The Acceptor handles this by sending an AcceptorCompletionAdvice to inform the Acquirer to reverse the transaction. The Acquirer must always accept this reversal and return an approval.

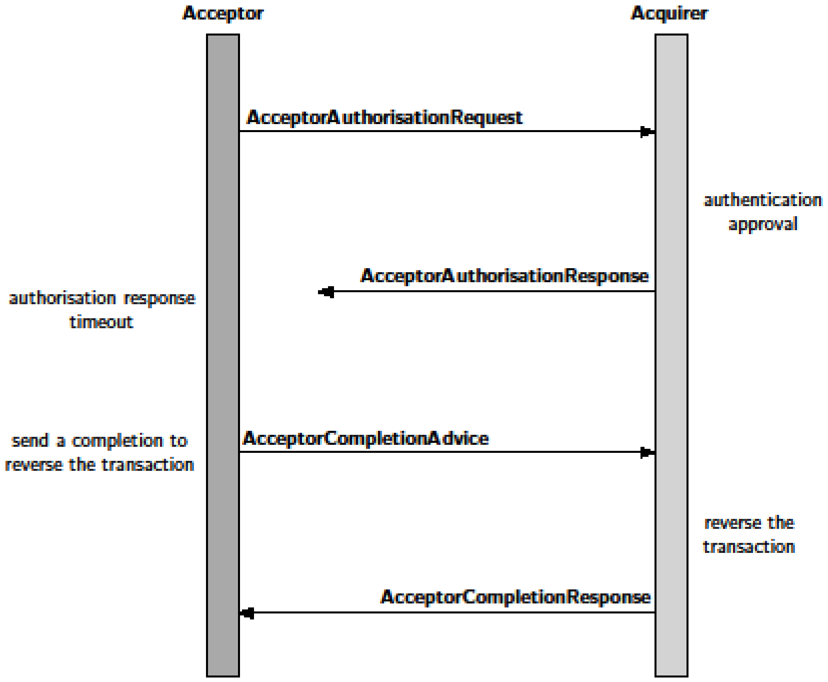
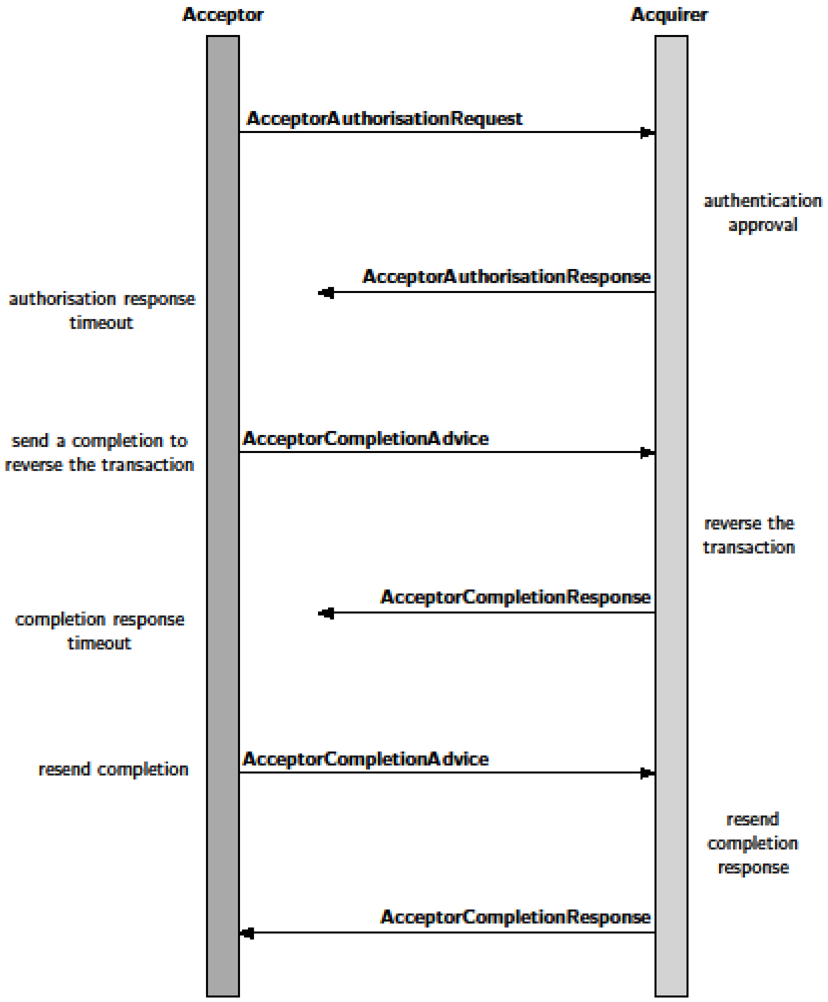

Acceptor
Acquirer
AcceptorAuthorisationRequest
authentication approval
AcceptorAuthorisationResponse
authorisation response timeout
send a completion to reverse the transaction
AcceptorCompletionAdvice
reverse the transaction
AcceptorCompletionResponse

Figure 4: Example of communication error

No Completion Response

The next example scenario is shown in Figure 5. Here the transaction fails, either because the AcceptorAuthorisationResponse never arrives or because the authentication fails for some other reason. In this example the response never arrives. As the transaction fails, the Acceptor sends an AcceptorCompletionAdvice, but never receives a response. When this happens, the Acceptor resends the AcceptorCompletionAdvice until a response is received.



Acceptor
Acquirer
AcceptorAuthorisationRequest
authentication approval
AcceptorAuthorisationResponse
authorisation response timeout
send a completion to reverse the transaction
AcceptorCompletionAdvice
reverse the transaction
AcceptorCompletionResponse
completion response timeout
resend completion
AcceptorCompletionAdvice
resend completion response
AcceptorCompletionResponse

Figure 5: Example of retransmission

Suggestions for Additional Message Items

Message Item	XML Tag	Description
AuthorisationRequest		
Environment		
POI		
Identification		
...		
SystemName	SysNm	Common name assigned by the acquirer to the POI system, i.e. “Xenta”, “Yomani”.
Card		
PlainCardData		
...		
AdditionalCardData	AddtlCardData	UUID/RFIDID

Example with no PAN and UUID

Table 5: Example Request

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:caaa.003.001.01">
    <AccptrAuthstnReq>
        <Hdr>
            <MsgFctn>FAUQ</MsgFctn>
            <PrtcolVrsn>1.0</PrtcolVrsn>
            <XchgId>0</XchgId>
            <CreDtTm>2011-06-31T00:00:00.00+01:00</CreDtTm>
            <InitgPty>
                <Id>Point</Id>
            </InitgPty>
            <RcptPty>
                <Id>Host</Id>
            </RcptPty>
        </Hdr>
        <AuthstnReq>
            <Envt>
                <POI>
                    <Id>
                        <Id>990001</Id>
                    </Id>
                    <SysNm>Yomani</SysNm>
                </POI>
                <Card>
                    <PlainCardData>
                        <PAN>99990000000001</PAN>
                        <XpryDt>2014-12</XpryDt>
                    </PlainCardData>
                    <AddtlCardData></AddtlCardData> <!-- UID -->
                </Card>
            </Envt>
            <Cntxt>
                <PmtCntxt>
                    <CardDataNtryMd>CTLS</CardDataNtryMd>
                </PmtCntxt>
            </Cntxt>
            <Tx>
                <TxCaptr>TRUE</TxCaptr>
                <TxTp>CRDP</TxTp>
                <MrchntCtgyCd>5691</MrchntCtgyCd>
                <TxId>
                    <TxDtTm>2011-06-31T00:00:00.00+01:00</TxDtTm>
                    <TxRef>1234567890</TxRef>
                </TxId>
                <TxDtls>
                    <Ccy>DKK</Ccy>
                    <TtlAmt>20.00</TtlAmt>
                </TxDtls>
            </Tx>
        </AuthstnReq>
    </AccptrAuthstnReq>
</Document>
```

Message Item	XML Tag	Description
AuthorisationResponse		
TransactionResponse		
AuthorisationResult		
ResponseToAuthorisation		
Response		
ResponseReason	RspnRsn	
Action	Actn	
ActionType	ActnTp	**DISP**: Display a message. **PRNT**: Print a message.
MessageToPresent	MsgToPres	
MessageDestination	MsgDstn	**CDSP**: Cardholder display or interface. **CRCP**: Cardholder receipt. **MDSP**: Merchant display or interface. **MRCP**: Merchant receipt.
MessageContent	MsgCntt	Text or graphic data to be display or printed to the cardholder or the cashier. Maximum 256 characters.

Table 6: Example Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:iso:std:iso:20022:tech:xsd:caaa.003.001.01">
    <AccptrAuthstnRspn>
        <Hdr>
            <MsgFctn>FAUP</MsgFctn>
            <PrtcolVrsn>1.0</PrtcolVrsn>
            <XchgId>0</XchgId>
            <CreDtTm>2011-06-31T00:00:00.00+01:00</CreDtTm>
            <InitgPty>
                <Id>Point</Id>
            </InitgPty>
            <RcptPty>
                <Id>Host</Id>
            </RcptPty>
        </Hdr>
        <AuthstnRspn>
            <Envt>
                <POIId>
                    <Id>990001</Id>
                </POIId>
            </Envt>
            <Tx>
                <TxId>
                    <TxDtTm>2011-06-31T00:00:00.00+01:00</TxDtTm>
                    <TxRef>1234567890</TxRef>
                </TxId>
                <TxDtls>
                    <Ccy>EUR</Ccy>
                    <TtlAmt>20.00</TtlAmt>
                </TxDtls>
            </Tx>
            <TxRspn>
                <AuthstnRslt>
                    <RspnToAuthstn>
                        <Rspn>APPR</Rspn> <!-- Status = OK -->
                        <RspnRsn>1</RspnRsn> <!-- StatusTextID = 1 -->
                    </RspnToAuthstn>
                </AuthstnRslt>
                <Actn>
                    <ActnTp>DISP</ActnTp>
                    <MsgToPres>
                        <MsgDstn>CDSP</MsgDstn>
                        <MsgCntt>2</MsgCntt> <!-- LogoID = 2 -->
                    </MsgToPres>
                </Actn>
            </TxRspn>
        </AuthstnRspn>
    </AccptrAuthstnRspn>
</Document>
```