FICOM'S (THE FINNISH FEDERATION FOR TELECOMMUNICATIONS AND TELEINFORMATICS) APPLICATION GUIDELINE FOR ETSI'S MSS STANDARDS:

V2.1

2012-01-14

Version	Description
1.0	Original version.
1.1	The MSS_Signature element required by WSDL has been added to the message format description and message examples (also missing from the ETSI TS 102 204 examples). WSDL added to the references. Support for organisation certificates added to SignatureProfile.
1.2	Added support for messaging mode ”asynchronous client-server”. Added support for operation MSS_StatusQuery. Added support for test certificates in SignatureProfile. Hash signs (#) omitted from the user experience. UserIdentifier element format corrected. The encoding of the DataToBeSigned element was corrected. It was emphasised that the event "number" begins either with a letter or an underscore (NCName).
2.0	Signature profiles were completely renewed. Position for value added service response messages was reserved in the StatusDetail element of signature responses. New value added services AE validation and PersonIdentity. Value added service SessionID renamed EventID. Support for the UserIdentifier element was omitted (replaced by the PersonIdentity service). The supported MSS_Format/MimeType/Encoding/SignatureProfile combinations were written out. New MSS_Format PKCS1. 505 status code added for test identities. Added status code extensions (Appendix C). Synchronous messaging mode is not recommended.
2.1	Due to roadmap delays in MSSP implementations, some features from 2.0 have been tagged as “not supported in the current version”. Several errors corrected. Parameter min and max lengths defined more clearly. Excessive MimeType / Encoding combinations dropped.

Contents

1 Introduction

This document specifies the FiCom Mobile Signature Service (hereinafter "FiCom recommendation") recommendation created by FiCom ry.

The FiCom recommendation is an application guideline for ETSI's Mobile Signature Service standards. A mobile certificate can be realised between different application providers in Finland in accordance with the techniques, practices, limitations and extensions described in the document, comprising:

- ETSI TS 102 204-compliant application provider interface for signatures with a mobile phone and mobile user identification
- ETSI TS 102 207-compliant signature roaming between different mobile phone networks (mobile signature roaming)
- harmonised user experience regardless of the operator
- additional services that improve safety and availability, supplementing the ETSI standards

The application guideline does not set limitations on offering additional functionalities not specified in the guideline. The application guideline does not describe the technical implementation of the MSS service, business models or commercial terms.

From the point of view of understanding the application guideline, it is recommendable to also review the original standard specifications and the SOAP 1.2 and WSDL 1.1 specification.

The application guideline is updated by FiCom.

Note: This document version describes some features which, although in the immediate roadmap, are not necessarily supported by all MSSPs at the time of writing. Such features have been indicated by the phrase “not supported in current version”.

2 References

The FiCom recommendation is based on the following techniques.

ETSI

TS 102 204; TR 102 206; TS 102 207:
http://portal.etsi.org/docbox/EC_Files/ EC_Files/ts_102204v010104p.pdf
http://portal.etsi.org/docbox/EC_Files/ EC_Files/tr_102206v010103p.pdf
http://portal.etsi.org/docbox/EC_Files/ EC_Files/ts_102207v010103p.pdf

W3C

XML Schema Part 1; Part 2:
http://www.w3.org/TR/xmlschema-1/
http://www.w3.org/TR/xmlschema-2/
SOAP Version 1.2 Part 0: Primer; Part 1: Messaging Framework; Part 2: Adjuncts:
http://www.w3.org/TR/2003/REC-soap12-part0-20030624/
http://www.w3.org/TR/2003/REC-soap12-part1-20030624/
http://www.w3.org/TR/2003/REC-soap12-part2-20030624/
XMLSignature:
http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/
WSDL 1.1:
http://www.w3.org/TR/wsdl

RSA Laboratories

PKCS#7: Cryptographic Message Syntax Standard:
http://www.rsasecurity.com/rsalabs
RFC 5652: Cryptographic Message Syntax: fourth revision of PKCS#7 specification

OASIS

Security Assertion Markup Language (SAML) v2.0:
http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

3 Abbreviations and definitions

MSS standard	Mobile Signature Service standard. The standard described in ETSI TS 102 204, specifying the Application Provider's interface for the mobile signature service. The term partly also refers to the signature roaming standard described in ETSI TS 102 207.
FiCom recommendation	See Introduction.
User	The holder of a signing device. A customer of an application provider and a home operator, i.e. an HMSSP.
AP	Service provider (Application Provider). An actor needing the User’s signature. AE's customer.
AP_ID	Application Provider's contact information in MSSP systems.
AP_PWD	Application Provider's password in AE's system.
MSSP	Mobile Signature Service Provider. In this document, MSSP refers to an actor providing HMSSP services to Users and potentially AE services to Application Providers and/or RE services to AEs. Typically, such an actor is an operator providing mobile certificate services.
AE	Acquiring Entity. An actor offering a web service interface to an Application Provider for a mobile signature service complying with the FiCom recommendation. Communicates with the User’s home operator (HMSSP) as necessary using signature service roaming.
RE	Routing Entity. An entity that routes traffic between an AE and an HMSSP. The RE may be a component of AE or HMSSP systems or a separate system of a third party.
HMSSP	Home MSSP. The User’s home operator.
VE	Verifying Entity. A Routing Entity (RE) that is also responsible for the validation of an event.
Mobile signature	A digital signature made using a mobile phone or mobile terminal device based on the public key method (PKI), which can be used for various verification services, such as digital signature, electronic identification of a person and strong authentication of a user.

4 FiCom recommendation in brief

For a person who is familiar with ETSI's MSS standards, the enclosed list offers a concise presentation of the techniques, practices, limitations and extensions chosen in the FiCom recommendation. The terms used are also presented in this document.

1. The supported messaging modes are synchronous client-server (not recommended) and asynchronous client-server. The asynchronous server-server messaging mode is not currently supported.
2. Strong mutual authentication and encryption between all entities that participate in the routing of the message.
3. AP identifier (AP_ID) and the name of AP, as displayed on the terminal device, are created in the service agreement between the AP and the AE. Upon making the agreement, also a password is created for the AP (AP_PWD). After this, the AP owns the created AP_ID and it remains even if the AP was to migrate to using another AE's interface subsequently. The AE transmits the AP_ID and the AP's name to all HMSSPs.
4. The AP name displayed on the terminal device is not the same as AP_ID.
5. Strong mutual authentication and encryption of communications between the AP and AE, in addition to the password (AP_PWD).
6. The supported message formats are MSS_SignatureReq, MSS_SignatureResp, MSS_StatusReq, MSS_StatusResp, MSS_ReceiptReq and MSS_ReceiptResp.
7. The recommendation does not address MSS service registration messages in this version. The user registration process is left as an internal matter for each HMSSP.
8. XML-signed service messages are not currently supported.
9. The user and the HMSSP can be found exclusively on the basis of the MSISDN, utilising the number transferrability resources.
10. The user can alternatively be found on the basis of a UserIdentifier whose mandatory postfix additionally identifies the HMSSP **(not supported in the current version)**.
11. The character sets supported in service requests are UTF-8, GSM and UCS2, character sets supported on the terminal device are GSM 03.38 and UCS2. Only UTF-8 characters included in the GSM 03.38 character set are available.
12. The HMSSP offer six different signature services:
    - anonymous authentication **(not supported in the current version)**
    - authentication
    - signature of plain text content
    - signature of digest content **(not supported in the current version)**
    - issuing consent
    - operator service for authentication
13. There is a separate signature profile for each offered signature service. The signature profile is directly used for indicating the desired service.
14. The user can prohibit the use of any signature profile with his or her own mobile subscription.
15. Added value services as expansions of the MSS standard (AdditionalServices):
    - mobile phone spam prevention
    - an event identifier that connects the business channel session to the authentication event
    - validation by the AE **(not supported in the current version)**
    - User's language preference **(not supported in the current version)**
    - PersonIdentity service
16. Uniform user experience: format of signature requests has been standardised.
17. The format of the digital signature is base64-encoded PKCS#7 or PKCS#1 supplemented with the user's certificate **(not supported in the current version)**.
18. Synchronizing the system clock with the NTP service is mandatory for AE, RE and HMSSP. It is recommended for AP.

5 Course of the signature event

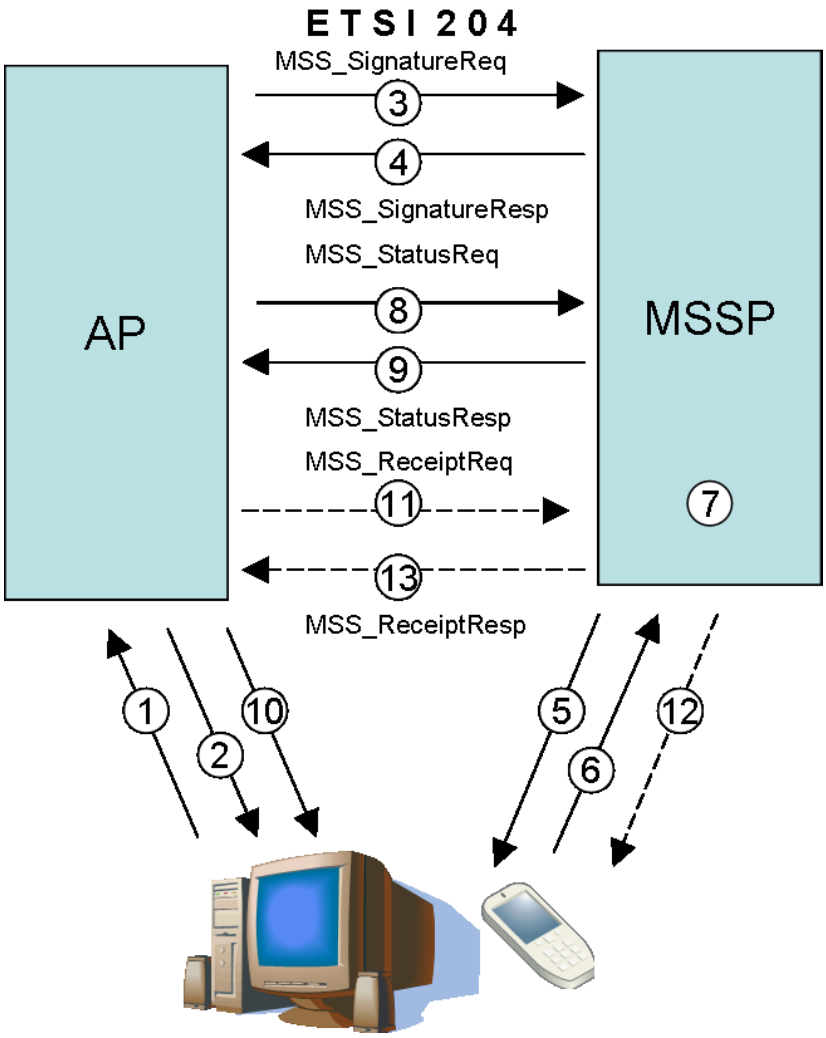
The ETSI TS 102 204 specification describes three alternative messaging modes: synchronous, asynchronous client-server and asynchronous server-server.

The FiCom recommendation supports the asynchronous client-server messaging mode. Asynchronous server-server communication is not currently supported, and synchronous messaging mode is not recommended for use as it consumes the system resources of the AP and MSSP unnecessarily. The following describes the course of an event in the supported messaging mode. In this chapter, the concept "MSSP" generally refers to the system utilising signature service roaming, formed by all operators. One of the main objectives of the ETSI MSS standards is to hide the complexity of signature service roaming from the Application Provider so that, from the point of view of the AP's systems, communication seems to take place with a single MSSP that covers the mobile subscriptions of all Users.

5.1 Asynchronous client-server messaging mode

In asynchronous client-server communication, the signature event is made up of the following service messages: **signature request**, a **status request** inquiring about the state of processing the request and a related **status response,** and an optional **receipt request** and a related **receipt response**. The event is made up of a number of HTTP sessions following each other, which are always opened from the AP's system. The messaging mode is useful compared to synchronous communication in that the AP obtains reference information for the signature event at an early phase and the service system can use all of its means to achieve high service reliability.

In order to avoid unnecessary traffic, it is recommended that the AP make its first status request 20 seconds after the signature request was sent and repeat the inquiry once every 5 seconds until the HMSSP returns a status response to the AP, indicating the end of the signature event.



1. The User opens up a session to the AP's server. A need for login or signing an electronic document appears. The User inputs his or her mobile phone number and spam prevention code into the application provider's server.
2. The AP shows a help to the User in the business channel, guiding the User to monitor the signature channel (the mobile phone). The help includes the event identifier of the signature event.
3. The AP sends a signature request (MSS_SignatureReq) to the AE to whose web service interface the AP has integrated. The AE authenticates the AP strongly. In the value-added part of the request, the AP may ask the HMSSP to deliver additional information related to the User's identity.
4. After receiving the signature request and generating an MSSP event number for it, the AE returns the event number to the AP. In the response message (MSS_SignatureResp), the AE indicates that the event is still incomplete.
5. The AE ensures that the signature service requested by the AP and the value added services are in accordance with the service agreement. The AE processes the signature request and directs it to the User's HMSSP. (In the figure, the entity is described as a single MSSP as, from the point of view of the AP, the entire world behind the AE is transparent.) The HMSSP checks the User's spam prevention code and ensures that the signature service requested by the AP and the value-added services are allowed by the User. The HMSSP delivers the signature request to the User’s mobile phone.
6. The User ensures that the event identifier shown in the introduction of the event on the mobile phone matches the event identifier indicated in the business channel. Then the User signs the event by inputting the SPIN code requested. As a result, the User's digital signature is generated, which is delivered to the HMSSP.

7. From the digital signature, the HMSSP compiles an digital signature message according to the PKCS#7 standard, which is attached as a part of the signature response (MSS_StatusResp). The HMSSP processes the value-added services requested by the AP and attaches to the digital signature message the value-added service responses for the AP (for example, additional information related to the User's identity).
8. After sending the signature request, the AP inquires about the completion of the signature response at specified intervals (MSS_StatusReq).
9. The HMSSP validates the digital signature. The HMSSP informs the AP of the status of the signature request in the status response (MSS_StatusResp). (The AE can also validate. See AE validation.) When the signature is ready, it is delivered to the AP as part of the status response.
10. The AP processes the signature response it received. The AP connects the User identified in the digital signature and the user in the AP's own user database. The User has been authenticated. If the AP uses, for example, a self-updating web page in the business channel, the AP may now change the contents of the page to be such that the User's browser is directed to the contents that required authentication.
11. If it so desires, the AP may send the User an receipt on the success of the event in the authentication channel (MSS_ReceiptReq).
12. The AP's receipt message is delivered to the User in the same way as the signature request.
13. The AP receives a confirmation on the delivery of the receipt message (MSS_ReceiptResp).

6 Application Provider's interface

6.1 General message structure

The message interface between the Application Provider (AP) and AE is realised in the form of a group of MSS service messages specified by ETSI. In the supported messaging modes, in practice each phase of communication proceeds so that the AP sends a service request to the AE and receives a service response to the request from the AE. The AP's service request is always HTTP POST Request and the AE's response message is HTTP Response.

MSS service messages are wrapped into SOAP envelopes transmitted as HTTP message content. The SOAP envelops are comprised of the header element (env:Header) and content element (env:Body).

6.1.1 SOAP Header

The SOAP Header element is optional. It is primarily useful in implementing XML signatures, and the FiCom recommendation does not address XML signatures.

6.1.2 SOAP Body

The SOAP Body element is mandatory, and it includes one of the following elements that define the message type:

- MSS_SignatureReq (operation: MSS_Signature)
- MSS_SignatureResp (operation: MSS_Signature)
- MSS_SignatureReq (operation: MSS_Status)
- MSS_StatusResp (operation: MSS_Status)
- MSS_ReceiptReq (operation: MSS_Receipt)
- MSS_ReceiptResp (operation: MSS_Receipt)

Each of these further includes the attributes and sub-elements specific to the message type in question. In accordance with the WSDL 1.1 specification, the actual message element is wrapped inside the element specifying the name of the "operation." (See the example below.)

6.1.3 Namespaces

The element specifying the SOAP Envelope reserves its own namespace:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

The element specifying the message type of the content element reserves namespaces for the ETSI MSS standard specifications and if necessary the specifications of the XML signature and value added services of signature requests specified by the FiCom recommendation:

```
<MSS_ReceiptReq xmlns=http://uri.etsi.org/TS102204/v1.1.2#
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:fi="http://mss.ficom.fi/TS102204/v1.0.0#" ...>
```

6.1.4 Message structure example

An example of the general message structure is presented below. Separate examples of each MSS service message without the optional header element will be presented below. The W3C does not recommend the use of XML comments in actual message traffic.

```
POST /MSS_Signature HTTP/1.0
Host: mss.teliasonera.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <!- Optional header -->
   <env:Header>
   </env:Header>
   <!- Mandatory body -->
   <env:Body>
       <-! WDSL op name -->
      <MSS_Signature>
         <!- MSS service message -->
         <MSS_SignatureReq ...>
            . . .
         </MSS_SignatureReq>
      </MSS_Signature>
   </env:Body>
</env:Envelope>
```

6.1.5 Message types falling outside the recommendation

The FiCom recommendation *does not* address the following message types specified by the MSS standard:

- MSS_RegistrationReq
- MSS_RegistrationResp
- MSS_ProfileReq
- MSS_ProfileResp
- MSS_HandshakeReq
- MSS_HandshakeResp

6.1.6 Error communication

If any error is detected during the event, the status code is returned to the Application Provider as a SOAP FAULT message. The structure of the error notification message is described below after the actual service messages.

6.2 Signature request (MSS_SignatureReq)

The service request message description includes a more detailed review of the service message attributes and elements, and the descriptions of other MSS service requests refer to some of these descriptions.

A signature request sent by the Application Provider to the AE includes the following information:

- Messaging mode (asynchronous client-server communication)
- Allowed duration of the event, i.e. "timeout"
- Application provider identifier (AP_ID)
- User contact information (MSISDN or UserIdentifier)
- Timestamp
- Signed content (DTBS)
- Signature profile, i.e. the signing service, requested by the AP
- Additional services, such as prevention of malicious use and person identity information

6.2.1 MSS_SignatureReq: attributes

Name	Value	Description	Required
MajorVersion	”1”	Interface main version, currently 1.	Yes
MinorVersion	”1”	Interface sub-version, currently 1.	Yes
MessagingMode	”asynchClient Server” or ”synch”	Messaging mode, currently always either ”asynchClientServer” (recommendation) or “synch”.	Yes
ValidityDate	DateTime	Time limit specified by the AP, after which an incomplete event must be rejected. Expressed as an absolute timestamp. For example, ”2003-06-25T21:32:00Z”.	No
TimeOut	Integer	Time limit specified by the AP, after which an incomplete event must be rejected. Expressed as seconds after the beginning of the event.	No

6.2.1.1 MajorVersion and MinorVersion

The version of the MSS interface used is currently 1.1. If the version number specified by the AP differs from it, the AE returns status code 108 (INCOMPATIBLE_INTERFACE) to the AP.

6.2.1.2 MessagingMode

The FiCom recommendation currently guarantees support for asynchronous client-server communication. Support for asynchronous server-server communication is not guaranteed.

If the AE or the HMSSP does not support the messaging mode requested by the Application Provider, the MSSP in question returns status code 101 (WRONG_PARAM) to the AP.

6.2.1.3 ValidityDate and TimeOut

The Application Provider can choose to specify a time limit for an event started by the Application Provider after which the AE must interrupt the event. If there is no time limit, the AE's default TimeOut of 5 minutes will be applied. The time limit can be specified either as an absolute timestamp (ValidityDate) or as seconds after the beginning of the event (TimeOut). (See http://www.w3.org/TR/xmlschema-2/#dateTime) After the default or AP-set time limit expires, the HMSSP interrupts the event and sends status code 208 (EXPIRED_TRANSACTION) to the AP.

The FiCom recommendation requires all AEs and HMSSPs to use the NTP service for synchronizing their system clocks. The AP is also recommended to synchronize its clock using NTP. NTP (Network Time Protocol) is described in the RFC 5905 document.

6.2.1.4 Example: MSS_SignatureReq element attributes

```
<MSS_SignatureReq xmlns="http://uri.etsi.org/TS102204/v1.1.2#" MajorVersion="1"
MinorVersion="1" MessagingMode="synch" ValidityDate="2003-06-25T21:32:00Z">
```

6.2.2 MSS_SignatureReq: elements

Name	Description	Required
AP_Info	The Application Provider's contact information and event number and timestamp issued by the AP for the event.	Yes
MSSP_Info	The element value is normally left blank.	Yes
MobileUser	The end user's contact information, either in format <MSISDN>+358123456789</MSISDN> in which the prefix +358 (in Finland) of the mobile phone number is mandatory, i.e. the international number format is applied OR **(not supported in the current version)** <UserIdentifier>userid42@dna</UserIdentifier> in which the postfix @operatorid is mandatory	Yes
DataToBeSigned	Text to be signed, max. 160 characters.	Yes
DataToBeDisplayed	Text to be displayed to the user, max. 110 characters.	No
MSS_Format	Format of the requested signature, for example <mssURI>http://uri.etsi.org/TS102204/v1.1.2#PKCS7</ mssURI>	No
SignatureProfile	The signature profile to be used, i.e. the selected service.	Yes*
AdditionalServices	Additional services connected to the signature event. FiCom has specified certain additional services that increase the End User's data security and ease of use.	Yes*

*Requirement that is stricter than required by the MSS standard

6.2.2.1 AP_Info

The element AP_Info contains the following attributes:

Name	Value	Description	Required
AP_ID	anyURI	The Application Provider's unique URI-type identifier with which the AP is registered to use the AE's services. For example, ”http://mss.teliasonera.com/mssURI/OyCompanyAb”	Yes
AP_TransID	NCName	A unique event ID created by the Application Provider. This ID is not usually required in synchronous message transmission, but it must nevertheless be generated in order to comply with the standard. The first character of the ID is a letter or an underscore ('_').	Yes
AP_PWD	String	Password used in authenticating the Application Provider.	Yes
Instant	DateTime	Timestamp for the time of submitting the signature request, created by the Application Provider.	Yes

AP_ID

The FiCom recommendation requires that the AE and the Application Provider strongly authenticate each other, such as using TLS handshake. In addition to this, a mechanism is needed for the AE to introduce the AP to the home operator (HMSSP) in connection with signature roaming. The AP_ID is a unique URI that the AE discloses to the AP as part of the service agreement and with which also other parties besides the AE uniquely identify the Application Provider that initiated the event. The AP_ID *is not* displayed to the User; it is information intended for communications between the MSSPs. In order to ensure uniqueness, the AP_ID complies with the format ”http://<AE-specific URI prefix>/<AP's name>”. **The length of AP_ID is 3 to 128 characters.** For example, "http://mss.teliasonera.com/mssURI/OyCompanyAb”, where ”teliasonera.com/mssURI” specifies the AE's namespace and ”OyCompanyAb” is the AP's name in the namespace administered by the AE.

If the AP_ID is not specified to comply with the AE's instructions, the AE returns status code 104 (UNAUTHORIZED_ACCESS) to the AP. If the HMSSP does not identify the Application Provider on the basis of the AP_ID, the HMSSP returns status code 104 (UNAUTHORIZED_ACCESS) to the Application Provider.

AP_TransID

The Application Provider must attach a transaction ID to every signature request it submits. Good policy requires that each generated event number is unique from the point of view of the Application Provider's own systems within the time frame the last month. The AE monitors this uniqueness. Letters, among others, are allowed in event numbers, but not colons. (See http://www.w3.org/TR/xmlschema-2/#NCName). The length of a transaction ID is 1 to 32 characters.

If the transaction ID is missing from the request, the AE returns status code 102 (MISSING_PARAM) to the AP.

AP_PWD

The FiCom recommendation requires separate strong authentication of the Application Provider, so the Application Provider password described in the MSS standard is no longer required as such. However, the password must be used in order to comply with the requirements of the standard.

If the password is missing from the signature request or the AP does not present the password required by the AE, the AE returns status code 104 (UNAUTHORIZED_ACCESS) to the AP.

Instant

The Application Provider must attach a time stamp to the service request, indicating the time when the request was submitted. (See http://www.w3.org/TR/xmlschema-2/#dateTime) Correspondingly, the HMSSP/AE will enter a time stamp in the response message.

If the AP's time stamp is missing from the request, the AE returns status code 102 (MISSING_PARAM) to the AP.

AP_URL (not in use)

The MSS standard defines optional information AP_URL used in the messaging mode ”asynchronous server-server” in routing the response message to the Application Provider. The FiCom recommendation does not currently take this parameter into account.

Example: AP_Info element attributes

```
<AP_Info AP_ID="http://mss.teliasonera.com/mssURI/oycompanyab" AP_TransID="A1203"
AP_PWD="1AP-PWD2" Instant="2003-06-24T21:32:00Z"/>
```

6.2.2.2 MSSP_Info

According to the MSS standard, the Application Provider must enter the element MSSP_Info and the element MSSP_ID for it in the signature request. However, the latter element can normally be left blank in a signature request as the AP does not need to know the HMSSP to which the signature request is directed.

```
<MSSP_Info>
    <MSSP_ID/>
</MSSP_Info>
```

If the element MSSP_Info or the element MSSP_ID contained by it are missing from a signature request, the AE returns status code 102 (MISSING_PARAM) to the AP. If the AP has specified a value for MSSP_ID and the AE does not accept this value for some reason, the AE returns status code 101 (WRONG_PARAM) to the AP. Even if it is suitable for the AE, the value of the parameter may cause subsequent routing problems, indicated with status codes 750-780.

6.2.2.3 MobileUser

The End User's contact information is either the User's mobile phone number (MSISDN) or a UserIdentifier **(not supported in the current version)**.

The User's home operator (HMSSP) can be solved in Finland on the basis of the mobile phone number using the number portabilility service. The phone number is required in the international number format (in Finland, this means the prefix +358), and it must be from 3 to 16 characters.

```
<MobileUser>
    <MSISDN>+358123456789</MSISDN>
</MobileUser>
```

A UserIdentifier is in the format <userid>@<operatorid>, where <operatorid> is an identifier used by the AE for routing purposes. <userid> consists of letters (a-zA-Z, case insensitive), numbers, hyphen ('-'), underscore ('_'), and period ('.'). <userid> is 3 to 64 characters in length.

```
<MobileUser>
    <UserIdentifier>userid42@dna</UserIdentifier>
</MobileUser>
```

If the User's contact information is missing, the AE is unable to determine the HMSSP on the basis of the contact information or the HMSSP does not identify the User on the basis of the contact information, the AE or the HMSSP returns correspondingly status code 105 (UNKNOWN_CLIENT) to the AP. If the HMSSP identifies the User but does not reach the User for some reason, the HMSSP returns status code 209 (OTA_ERROR) to the AP.

6.2.2.4 DataToBeSigned

The element transmits the content designated as to be signed by the User. The content is either:

- plain text, agreement text that is to be read by a human
- digest of arbitrary binary data
- long random figure, i.e. authentication challenge

It is recommended that a time stamp or other identifier specifying the event in plain text be included in the agreement content to be signed.

The authentication challenge is not displayed to the User at all; instead of the challenge, a standard description of the authentication event is shown.

The maximum length of the content to be signed is 160 characters. If the content of the signature request is longer than this, the HMSSP returns status code 103 (WRONG_DATA_LENGTH) to the AP. In practice, digests that are to be signed are 20-64 octets long.

In addition to the actual content to be signed, the element has two attributes that indicate the format of the content:

Name	Value	Description	Required
MimeType	string	MIME type of the content to be signed	No
Encoding	string	Used character set.	No

MimeType

The MIME types of content to be signed are listed in the table below.

If the HMSSP does not accept the MIME type reported by the AP, the HMSSP returns status code 107 (INAPPROPRIATE_DATA) to the AP.

Encoding

Encoding of the content to be signed. If an authentication challenge or digest is signed, "base64" can be set as the Encoding type. In this case, the value of the challenge or digest (as raw bytes) with base64 encoding can be set as the element value. Encodings that are supported with plain text include UTF-8.

If the HMSSP does not accept the character set reported by the AP, the HMSSP returns status code 107 (INAPPROPRIATE_DATA) to the AP.

The table below describes the MimeType/Encoding combinations for which the FiCom recommendation guarantees support. Other combinations are not allowed.

MimeType	Encoding	Description	MSS_ Format	SignaturePro files*
text/plain	UTF-8	Generic plain text signature. The AP sends the text in the UTF-8 character set, the HMSSP converts it into the GSM 03.38 character set.	PKCS1 PKCS7	A,S
application/octet-stream	base64	Signature of an authentication challenge generated by the AP. The AP generates the authentication challenge (for example, a binary presentation of AP_TransID) and sends it base64-encoded in the DTBS element. The HMSSP base64-decodes the challenge and generates a PKCS#7 signature from it.	PKCS7	A
application/octet-stream	base64	Signature of a MessageDigest generated by the AP. The AP digests the content to be signed using a strong hash algorithm and generates an RSAES-PKCS1-v1_5-compliant DER-encoded DigestInfo structure (MessageDigest). The AP submits the structure base64-encoded in the DTBS element. The HMSSP base64-decodes the MessageDigest and generates a PKCS#1 signature without authenticated attributes from the MessageDigest.	PKCS1	D
application/x-sha1	base64	Signature of SHA1-digested binary data. The AP digests the content to be signed using the SHA1 hash algorithm and sends the structure base64-encoded in the DTBS element. The HMSSP base64-decodes the digested content, adds ”random nonce” as an authenticated attribute to the content, digests the supplemented content and generates a MessageDigest as the final result. Furthermore, the HMSSP generates a signature from the MessageDigest in accordance with the format requested by the AP.	PKCS1	D
application/x-sha256	base64	As above, but using the SHA256 hash algorithm instead of SHA1.	PKCS1	D

*A=authentication, anonymous authentication, operator authentication; S=signature of a plain-text message, consent; D=digest signature **(not supported in the current version)**

Note: If a plain-text message is signed using the UTF-8 character set, the HMSSP performs the conversion of the content to be signed from the UTF-8 character set in the XML format to the GSM 03.38 character set of the terminal device. The character conversion is performed for the purpose of efficiency. The Application Provider must take into account the character conversion performed by the HMSSP when validating the signature itself. The message to be signed does not change in terms of content in the conversion if the message does not include characters not included in the GSM 03.38 character set. Appendix D lists the symbols of the GSM 03.38 character set and the corresponding UTF-8 character set symbols. The AP must naturally restrict the UTF-8 characters used by it to the characters listed in Appendix D. If characters not included in Appendix D occur, the HMSSP returns status code 107 (INAPPROPRIATE_DATA) to the AP.

Example of an authentication challenge:

```
<DataToBeSigned MimeType="application/octet-stream" Encoding="base64">
   TWFuIGlzIGRpc3R=
</DataToBeSigned>
```

Example of an undigested agreement:

```
<DataToBeSigned MimeType="text/plain" Encoding="UTF-8">
   Vahvistan muutokset OyCompanyAb:n käyttäjäprofiilissani: uusi postitusosoite
x ; uusi puhelinnumero y.
</DataToBeSigned>
```

Example of a digest:

```
<DataToBeSigned MimeType="application/octet-stream" Encoding="base64">
   TWFuIGlzIGRpc3Rpbmd1aXNoQA9=
</DataToBeSigned>
```

6.2.2.5 DataToBeDisplayed

The FiCom recommendation does not allow this element (see the exception below). When a digest (bit string) is signed with a designated SignatureProfile, the digest displayed to the User is edited into a hexadecimal presentation as described in the document Service Description for the Mobile Certificate Service; however, the HMSSP performs this formatting automatically on behalf of the AP, and the DTBD element is thus not used.

If the DTBD element is included in the AP's request, the HMSSP returns status code 101 (WRONG_PARAM) to the AP.

Note: In the operator authentication service (see SignatureProfile), DataToBeDisplayed is a required element and not a prohibited element. If the DTBD element is missing in this case, the HMSSP returns status code 102 (MISSING_PARAM) to the AP.

6.2.2.6 MSS_Format

The format of the signature is either PKCS#7 or PKCS#1 **(not supported in the current version)**. The latter one is a format specified by FiCom, and it includes the PKCS#1 signature with the user's mobile certificate. (For additional information on the PKCS#1 signature format, see the description of the message type MSS_SignatureResp element MSS_Signature.)

The FiCom recommendation *does not* guarantee support for other signature formats.

PKCS#7:

```
<MSS_Format>
      <mssURI>http://uri.etsi.org/TS102204/v1.1.2#PKCS7</mssURI>
</MSS_Format>
```

PKCS#1 **(not supported in the current version)**:

```
<MSS_Format>
      <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#PKCS1</mssURI>
</MSS_Format>
```

6.2.2.7 SignatureProfile

The SignatureProfile element indicates which signature service the Application Provider is requesting from the HMSSP. Currently, the FiCom recommendation takes the signature profile information into account as follows: FiCom specifies six different services: anonymous authentication, user authentication, signature of plain-text content, signature of digested content,

issuing consent. FiCom has specified four signature profiles whose URIs directly indicate the service required by the AP.

The signature profiles that the FiCom recommendation takes into account are:

1. anonymous authentication **(not supported in the current version)**

```
<SignatureProfile>
   <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/anonymous-profile.xml</mssURI>
</SignatureProfile>
```

2. authentication

```
<SignatureProfile>
   <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/authentication-
profile.xml</mssURI>
</SignatureProfile>
```

3. signature of plain text content

```
<SignatureProfile>
   <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/signature-profile.xml</mssURI>
</SignatureProfile>
```

4. signature of digested content **(not supported in the current version)**

```
<SignatureProfile>
   <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/digestive-signature-
profile.xml</mssURI>
</SignatureProfile>
```

5. issuing consent

```
<SignatureProfile>
   <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/consent-profile.xml</mssURI>
</SignatureProfile>
```

6. operator authentication service (operator's internal service)

Note: In the operator authentication service, DataToBeDisplayed is a required element and not a prohibited element.

```
<SignatureProfile>
   <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/operauth-profile.xml</mssURI>
</SignatureProfile>
```

If the Application Provider does not specify a signature profile in the service request, the HMSSP assumes signature profile 2 (authentication).

If the signature profile required by the Application Provider is something else, the HMSSP generates status code 109 (UNSUPPORTED_PROFILE). If the User has prohibited service requests with the signature profile required by the Application Provider, the HMSSP returns status code 109 (UNSUPPORTED_PROFILE) to the AP.

6.2.2.8 AdditionalServices

This element is described below under Value added services.

6.2.2.9 SignatureProfileComparison (not in use)

Within the signature profiles covered by the FiCom recommendation, in practice the only sensible value for this information, which is optional in accordance with the MSS standard, is "exact." Since this value is also the default value specified by the standard, the FiCom recommendation ignores the element.

If the Application Provider has included this information in the signature request and specified a value other than "exact" for it, the FiCom recommendation does not specify how the HMSSP will react. However, if the HMSSP generates an error message to the AP, its status code is 109 (UNSUPPORTED_PROFILE).

6.2.2.10 KeyReference (not in use)

The FiCom recommendation does not take this optional element of the MSS standard into account. The choice of the signing device or the choice of a certain certificate of the device is determined directly according to the signature profile (SignatureProfile) required by the Application Provider. FiCom prefers this policy due to the fact that it will be easier to create and update detailed signature profiles if each signing device and device service (certificate) has a separate profile of its own.

If the Application Provider has included KeyReference information, the FiCom recommendation does not specify how the HMSSP will react to this information. However, if the HMSSP generates an error message to the AP, its status code is 404 (NO_KEY_FOUND).

6.2.3 Value added services

ETSI's MSS standard makes it possible to specify added value services which as elements of the signature request are in the following format:

```
<AdditionalServices>
   <Service>
      <Description>
         <mssURI>(service 1 URI)</mssURI>
      </Description>
      <(param1)>(parameter 1 value)</(param1)>
      <(param2)>(parameter 2 value)</(param2)>
      . . .
   </Service>
   <Service>
      <Description>
         <mssURI>(service 2 URI)</mssURI>
      </Description>
      <(param1)>(parameter 1 value)</(param1)>
      <(param2)>(parameter 2 value)</(param2)>
      . . .
   </Service>
    . . .
</AdditionalServices>
```

Each added value service therefore has a unique URI and optionally one or more parameters that can be freely specified. In addition, ETSI allows the Entity element for each added value service, allowing the AP to require a certain party to be responsible for implementing the added value service. However, the FiCom recommendation does not take the Entity element into account.

The FiCom recommendation covers the added value services specified in the table below:

Name	URI	Required
Event identifier	http://mss.ficom.fi/TS102204/v1.0.0#eventId	Yes
Spam prevention code	http://mss.ficom.fi/TS102204/v1.0.0#noSpam	Yes*
AE validation	http://mss.ficom.fi/TS102204/v1.0.0#validate	No
PersonIdentity	http://mss.ficom.fi/TS102204/v1.0.0#personIdentity	No
Language preference	http://mss.ficom.fi/TS102204/v1.0.0#userLang	No

* Can be replaced by strong or weak authentication implemented by the AP, but the descriptive element of the actual added value service must always be present.

The purpose of the added value services is described in more detail in the document Service Description for the Mobile Certificate Service edited by Finnish telecom operators.

6.2.3.1 Event identifier

The purpose of the event identifier is to make it easier to match a mobile certification event with the corresponding business channel event. The event identifier is displayed to the User simultaneously in both channels (if it can be displayed in the business channel).

An added value service always has one parameter, an element named EventID. The value of the element is an arbitrary alphanumeric string from 4 to 8 characters in length.

```
<Service>
   <Description>
      <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#eventId</mssURI>
   </Description>
   <EventID xmlns="http://mss.ficom.fi/TS102204/v1.0.0#>14521412</EventID>
</Service>
```

If the event number is missing from the request, the HMSSP returns status code 102 (MISSING_PARAM) to the AP.

6.2.3.2 Spam prevention code

The purpose of the spam prevention code is to prevent inappropriate interference with the User's mobile phone. The User informs the AP of his or her spam prevention code in the business channel, and the AP delivers it further to the HMSSP. If the code does not match the code saved in the User's profile, the HMSSP refuses to serve the service request.

An added value service always has one parameter, an element named NoSpamCode. The value of the element is a string of numbers and letters from 3 to 16 characters in length. The first character must be a letter. It also has an optional element named verify, with the possible values "yes" and "no."

```
<Service>
   <Description>
      <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#noSpam</mssURI>
   </Description>
   <NoSpamCode xmlns="http://mss.ficom.fi/TS102204/v1.0.0#">A1B2</NoSpamCode>
</Service>
```

The value of the element can also be blank as a sign of the User choosing not to use spam prevention. (In this case, the User's spam prevention code must similarly be inactivated in the HMSSP user profile.)

```
<Service>
```

```
    <Description>
       <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#noSpam</mssURI>
    </Description>
    <NoSpamCode xmlns="http://mss.ficom.fi/TS102204/v1.0.0#"></NoSpamCode>
</Service>
```

If weak or strong authentication always precedes mobile signatures in the Application Provider's system or the spam prevention code has been inquired once, or the User is identified using a UserIdentifier instead of MSISDN, the code need not be asked from the User. **In this case, the AP must include the NoSpamCode element attribute "verify" and its value "no"**, which requests the HMSSP to bypass the spam prevention code check. In this case, the value of the element is irrelevant.

```
<Service>
    <Description>
       <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#noSpam</mssURI>
    </Description>
    <NoSpamCode verify="no"
xmlns="http://mss.ficom.fi/TS102204/v1.0.0#"></NoSpamCode>
</Service>
```

If this added value service is missing from the signature request, the HMSSP returns status code 102 (MISSING_PARAM) to the AP. If the service request prevention code does not match with the code in the HMSSP user profile, the HMSSP returns status code 101 (WRONG_PARAM) to the AP.

6.2.3.3 AE validation (not supported in the current version)

With the Validate service, the Application Provider can request the AE to perform separate validation for the event in addition to validation by the HMSSP. The AE's validation result is recorded in a separate added value response element (see AE's validation response below).

```
<Service>
 <Description>
  <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#validate</mssURI>
 </Description>
</Service>
```

6.2.3.4 PersonIdentity

With the PersonIdentity service, the Application Provider can request the HMSSP to submit personal information not found in the User's mobile certificate in the signature response, and possibly to also submit "ready-parsed" mobile certificate information. The requested personal information is identified using URI names. In the example below, the Application Provider requests personal ID and postal address from the HMSSP.

6.2.3.4.1 Personal information inquiry

Personal information inquiry is a SAML2 attribute inquiry with the following instructions.

Samlp:AttributeQuery always includes the following required field [SAML2 Core, chapter 3].

Field name	Instructions
ID	Required [Saml2Core, 3.2.1]. The value is, for example, a copy of AP_TransID.
Version	Required [Saml2Core, 3.2.1]. The value is "2.0".
IssueInstant	Required [Saml2Core, 3.2.1]. The value is, for example, a copy of the MSS message Instant.

Subject ja Subject.NameID	Required [SAML2Core, 3.3.2.1]. Subject is the person whose personal information is being inquired. As a rule, the AP knows nothing about the Subject besides the telephone number, and the telephone number is not an actual search criterion for personal information. The AP writes the Subject field so that it contains a blank NameID field.

The actual content of the personal information inquiry is a list of the person's attributes that the Application Provider wants to know in the response to the authentication request. The table below lists the URI-named attributes for which the FiCom recommendation guarantees support. The Application Provider describes the attributes only with their technical URI names.

Name of the attribute	Description
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#hetu	The response message contains the User's Finnish personal identity code (in Finnish "henkilötunnus", HETU). The service requires a separate agreement. Data type xs:string
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#satu	The User's electronic transaction identifier in the User's certificate. Data type xs:string
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#address	The User's address saved in the operator's system. Semantics: JHS 106 Data type fi:PostalAddress **(not supported in the current version)**
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#age	The User's age in years. Data type xs:integer
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#ageClass	Is the User over 18 years of age? The value set is true/false. Data type xs:boolean
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#email	The User's e-mail address Data type xs:string **(not supported in the current version)**
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#gender	The User's gender. M=male, F=female. Data type xs:string
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#givenName	The User's given name information in the User's certificate. (The User's all given names, for example, Timothy Tyler.) Data type xs:string
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#surName	The User's surname information in the User's certificate. (for example: Smith) Data type xs:string
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#subject	The value of the Subject field in the User's certificate. Data type xs:string
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#validUntil	The last validity of the User's certificate. Data type xs:dateTime

The personal information is returned in the way described for element Status of message type MSS_SignatureResp.

6.2.3.4.2 Restrictions on PersonIdentity based on selected SignatureProfile

The SigatureProfile for issuing consent does support PersonID additional service. The SignatureProfile for anonymous authentication does not support PersonID queries for any other attribute except: age, ageClass, or gender.

6.2.3.4.3 Example of an SAML2 attribute inquiry of the PersonIdentity service

```
<Service>
   <Description>
     <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#personIdentity</mssURI>
   </Description>
   <samlp:AttributeQuery ID="id-20090817105401849"
```

```
       Version="2.0"
       IssueInstant="2009-08-17T10:54:01.849+03:00"
       xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
       xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
     <saml:Subject>
        <saml:NameID/>
     </saml:Subject>
     <saml:Attribute
        Name="http://mss.ficom.fi/TS102204/v1.0.0/PersonID#hetu"/>
     <saml:Attribute
        Name="http://mss.ficom.fi/TS102204/v1.0.0/PersonID#address"/>
   </samlp:AttributeQuery>
</Service>
```

6.2.3.5 Language preference (not supported in the current version)

The language preference informs the HMSSP of the language in which the AP wants to offer service.

An added value service always has one parameter, an element named UserLang. The value of the element is an ISO-639-1 compliant lower case two-letter language code, such as ”fi”, ”sv”, ”en”. The FiCom recommendation guarantees that this added value service is syntactically accepted, but the implementation of the service and the more detailed instructions are HMSSP-specific.

```
<Service>
   <Description>
      <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#userLang</mssURI>
   </Description>
   <UserLang xmlns=”http://mss.ficom.fi/TS102204/v1.0.0#”>fi</UserLang>
</Service>
```

6.2.4 Example: signature request

Below is an unabbreviated and commented example that merges the described signature request attributes and elements into a single message.

```
POST /MSS_Signature HTTP/1.0
Host: mss.teliasonera.com
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Body>
   <!- WSDL op name -->
   <MSS_Signature>

   <!- Signature request begins, notice namespace declarations -->
   <MSS_SignatureReq xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
xmlns:fi="http://mss.ficom.fi/TS102204/v1.0.0#" MajorVersion="1" MinorVersion="1"
MessagingMode="synch">

      <!- Application Provider ID, transaction ID and timestamp -->
      <AP_Info AP_ID="http://mss.teliasonera.com/mssURI/oycompanyab"
AP_TransID="A1203" AP_PWD="ssl" Instant="2003-06-24T21:32:00Z"/>

      <!- Include, but fill with blank values -->
      <MSSP_Info>
         <MSSP_ID/>
      </MSSP_Info>

      <!- End user identifier -->
      <MobileUser>
         <MSISDN>+358123456789</MSISDN>
      </MobileUser>

      <!- Authentication challenge or content to be signed -->
      <DataToBeSigned MimeType="text/plain" Encoding="UTF-8">
         24F56B879D6ADF71027E65A7095D1162EAF17C7A
      </DataToBeSigned>

      <!- Signature profile, i.e. the selected signing service -->
      <SignatureProfile>
         <mssURI>http://mss.ficom.fi/TS102206/v1.0.0/authentication-profile.xml
</mssURI>
      </SignatureProfile>

      <!- Value added services -->
      <AdditionalServices>
         <Service>
            <Description>
               <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#eventId</mssURI>
            </Description>
            <fi:EventID>A1B2</fi:EventID>
         </Service>
         <Service>
            <Description>
               <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#noSpam</mssURI>
            </Description>
            <fi:NoSpamCode>A1B2</fi:NoSpamCode>
         </Service>
      </AdditionalServices>
```

```
    </MSS_SignatureReq>
    </MSS_Signature>
</env:Body>
</env:Envelope>
```

6.3 Signature response (MSS_SignatureResp)

The signature response is generated by the HMSSP. The response is routed to the Application Provider via the AE. The Application Provider receives the response in the same HTTP session in which it submitted the signature request (synchronous communication) or the most recent status query (asynchronous client-server communication). If asynchronous client-server communication is used, the signature response is returned immediately, and the status information indicates that the event is incomplete: 100 (REQUEST_OK) in MSS_SignatureResp and 504 (OUTSTANDING_TRANSACTION) in subsequent MSS_StatusResp until completion. In case of failures where the AE is not able to forward the signature request to the HMSSP, the signature response and its error codes is generated by the AE.

The signature response includes the following data:

- Application Provider's contact information
- Contact information of the MSSP that generated the signature response
- User's contact information
- Timestamp
- End status of the event
- any added value service response messages

When signature is successful, additionally:

- Signature profile complied with by the HMSSP
- Digital signature

6.3.1 MSS_SignatureResp: attributes

Name	Value	Description	Required
MajorVersion	”1”	Interface main version, currently 1.	Yes
MinorVersion	”1”	Interface sub-version, currently 1.	Yes
MSSP_TransID	NCName	Event number generated by the MSSP.	Yes

6.3.1.1 MajorVersion and MinorVersion

The version of the MSS interface used is currently 1.1.

6.3.1.2 MSSP_TransID

The HMSSP must include an event number that is unique from *its* point of view to the response. (For the permitted character set, see http://www.w3.org/TR/xmlschema-2/#NCName) The length of the transaction ID is 1 to 32 characters.

Example: MSS_SignatureResp element attributes

```
<MSS_SignatureResp xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" MajorVersion="1"
MinorVersion="1" MSSP_TransID="B653">
```

6.3.2 MSS_SignatureResp: elements

Name	Description	Required
AP_Info	Copied directly from the request (MSS_SignatureReq).	Yes

MSSP_Info	MSSP's contact information. The URI format is used.	Yes
MobileUser	Copied directly from the request (MSS_SignatureReq).	Yes
Status	End status of the event.	Yes
SignatureProfile	Signature profile used.	No*
MSS_Signature	Digital signature.	No*

*in case of errors, cannot usually be included

6.3.2.1 AP_Info

The element is a direct copy of the element with the same name in the signature request submitted by the AP (MSS_SignatureReq).

6.3.2.2 MSSP_Info

In the signature response, MSSP_Info identifies the HMSSP. The MSSP must use the URI-format presentation in recording its identity (in the sub-element MSSP_ID). The maximum length of the presentation is 64 characters. In addition, the MSSP must specify the attribute "Instant" for the MSSP_Info element. A time stamp indicating the time when the signature response was sent is specified as the value of the attribute. (See http://www.w3.org/TR/xmlschema-2/#dateTime) Example of a URI-type identity:

```
<MSSP_Info Instant="2003-06-24T21:33:00Z">
   <MSSP_ID>
      <URI>http://mss.elisa.fi</URI>
   </MSSP_ID>
</MSSP_Info>
```

6.3.2.3 MobileUser

The element is a direct copy of the element with the same name in the signature request submitted by the AP (MSS_SignatureReq).

6.3.2.4 Status

The element indicates the status of the event (end status in the case of synchronous communication). The element consists of an MSS-compliant status code (StatusCode) that has a required integer-type attribute Value, an optional status message (StatusMessage) and an optional additional element (StatusDetail).

The value of the status message is always the name corresponding to the status code specified in the MSS standard; these names are listed in the appendices to the document.

StatusDetail includes the element ServiceResponse, which is divided into sub-elements named ServiceResponse according to the added value services used. The response messages of the additional value services are described below.

```
<Status>
   <StatusCode Value="500"/>
   <StatusMessage>SIGNATURE</StatusMessage>
   <StatusDetail>
      <fi:ServiceResponses>
         <fi:ServiceResponse>
            <fi:Description>
               <mssURI>http://uri.etsi.org/TS102204/v1.1.2#validate</mssURI>
            </fi:Description>
            <fi:Entity>
               <mssURI>http://ae.elisa.fi</mssURI>
            </fi:Entity>
            <fi:Status>
               <mss:StatusCode Value="502"/>
               <mss:StatusMessage>VALID_SIGNATURE</mss:StatusMessage>
            </fi:Status>
         </fi:ServiceResponse>
         <fi:ServiceResponse>
            <fi:Description>
               <mssURI>
                  http://mss.ficom.fi/TS102204/v1.0.0#personIdentity
               </mssURI>
            </fi:Description>
            <samlp:Response
               ID="id-20090817105401849"
               ...
            </samlp:Response>
         </fi:ServiceResponse>
         ...
      </fi:ServiceResponses>
   </StatusDetail>
</Status>
```

The following rules of the MSS standard are complied with in the values of the status codes:

In the asynchronous client-server messaging mode, the HMSSP acknowledges the event as received (but not yet finished) with the status code 100 (REQUEST_OK). When the HMSSP has succeeded in composing the digital signature, the HMSSP validates the event (in addition to the HMSSP, the AE can also validate the event if the Application Provider requests such added value service. See AE validation response below.)

If the event validator

- finds that the digital signature valid, the status code is 502 (VALID_SIGNATURE)
- finds the digital signature valid otherwise but the data content of the User's certificate does not match the additional requirements specified in the signature profile, the status code is 505 (CONSTRAINT_MISMATCH)
  - the additional criterion error could be, for example, a missing certificate policy identifier or non-allowed value of the identifier
- notices that the signature is in the blacklist, the status code is 501 (REVOKED_CERTIFICATE)
- notices that the digital signature is incorrect or the certificate is expired, the status code is 503 (INVALID_SIGNATURE)

6.3.2.5 AE's validation response (not supported in the current version)

Normally, the HMSSP is responsible for validating the signature events. In addition to the HMSSP, also the AE can choose to validate the signature event. The AE records the end status provided by the validation as ServiceResponse included in the StatusDetail element, where the added value service identifier is:

http://uri.etsi.org/TS102204/v1.1.2#validate

(See the example above.)

If the HMSSP has validated the signature event and recorded the status code 501 (REVOKED_SIGNATURE) or 503 (INVALID_SIGNATURE), the AE may leave the event non-validated, and the AE must repeat the status code reported by the HMSSP.

If the HMSSP has validated the event and recorded the status code 502 (VALID_SIGNATURE), 424 (CRL_EXPIRED) or 425 (ERROR_CERTIFICATE), the AE validates the event and records the status according to its own validation as the status code.

6.3.2.6 Response of the PersonIdentity value added service

The response message of the PersonIdentity added value service is recorded as a ServiceResponse included in the StatusDetail element, where the added value service identifier is:

http://mss.ficom.fi/TS102204/v1.0.0#personIdentity

(See the example above.)

The PersonIdentity added value service response is an SAML2 response message with the following application instructions.

Samlp:Response always includes the following required fields [SAML2 Core, chapter 3].

Field name	Instructions
ID	Required. The value is a copy of the ID field of the personal information inquiry.
Version	Required. The value is "2.0".
IssueInstant	Required. The value is the time when the SAML2 service created the response. This may differ from the timestamp of the MSS response message.
Status and Status.StatusCode	Required. The StatusCode value is required content in the Status field. Its value set is specified in the table below.

SAMLP status code	Description
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#ok	Personal information search successful.
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#partialFailure	Personal information search failed partly. The response may nevertheless include the part of personal information that the service succeeded in resolving.
http://mss.ficom.fi/TS102204/v1.0.0/PersonID#failure	Personal information search failed completely.

In addition to the required information, the personal information response includes the personal information as attributes, presented as specified in the table under Personal information inquiry.

The selected SignatureProfile may impose restrictions on what personal information can be inquired. Violations of these restrictions in queries made by the AP simply result in ignoring the violating attributes in the resoponse (but adjusting SAML status code accordinglyd).

The samlp:Response element of the response is signed by the HMSSP with the internal XML signature of the response [XML Signature, SAML2 Core chapter 5]. The signature includes the certificate used in the signature in the ds:X509Certificate elements and any certificate path up to the HMSSP root CA.

6.3.2.7 Example of an SAML2 assertion of the PersonIdentity added value service

```
<fi:ServiceResponse>
   <Description>
     <mssURI>http://mss.ficom.fi/TS102204/v1.0.0#personIdentity</mssURI>
   </Description>
   <samlp:Response ID="id-20090817105442102"
       Version="2.0"
       IssueInstant="2009-08-17T10:54:42.102+03:00"
       xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
       xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
       <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <ds:SignedInfo>
             <ds:CanonicalizationMethod
                 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
             <ds:SignatureMethod
                 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
             <ds:Reference URI="#RnlmcdL2pmTIvZ7c">
                 <ds:Transforms>
                     <ds:Transform
               Algorithm="http://www.w3.org/2000/09/xmldsig#envelopedsignature"/>
                     <ds:Transform
                             Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                         <InclusiveNamespaces
                             PrefixList="#default samlp saml ds"
                             xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                     </ds:Transform>
                 </ds:Transforms>
                 <ds:DigestMethod
                     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                 <ds:DigestValue>TliIDRiYWMgOGIwNyAwNTIxCg==</ds:DigestValue>
             </ds:Reference>
         </ds:SignedInfo>
         <ds:SignatureValue>
             OTYyOCAw...
         </ds:SignatureValue>
         <ds:KeyInfo>
             <ds:X509Data>
                 <ds:X509Certificate>
                     AwYzA2IGQ...
                 </ds:X509Certificate>
                 <ds:X509Certificate>
                     AwYzAOTli...
                 </ds:X509Certificate>
             </ds:X509Data>
         </ds:KeyInfo>
     </ds:Signature>
     <samlp:Status>
        <samlp:StatusCode
           Value="http://mss.ficom.fi/TS102204/v1.0.0/PersonID#ok">
        </samlp:StatusCode>
     </samlp:Status>
     <saml:Subject>
       <saml:NameID/>
     </saml:Subject>
     <saml:Attribute
        Name="http://mss.ficom.fi/TS102204/v1.0.0/PersonID#hetu">
       <saml:AttributeValue type="xs:string">141175-112P</saml:AttributeValue>
     </saml:Attribute>
     <saml:Attribute
        Name="http://mss.ficom.fi/TS102204/v1.0.0/PersonID#address">
       <saml:AttributeValue>
```

```
          <fi:PostalAddress>
            <fi:Name>Sanna Perkiö</fi:Name>
            <fi:StreetAddress>Malminrinne 5 A 12</fi:StreetAddress>
            <fi:PostalCode>00100</fi:PostalCode>
            <fi:Town>Helsinki</fi:Town>
            <fi:Country>Finland</fi:Country>
          </fi:PostalAddress>
        </saml:AttributeValue>
      </saml:Attribute>
    </samlp:Response>
</fi:ServiceResponse>
```

6.3.2.8 SignatureProfile

With the element, the HMSSP reports the signature profile used to the Application Provider. In the FiCom recommendation, this element is a direct copy of the information with the same name in the signature request submitted by the AP.

6.3.2.9 MSS_Signature

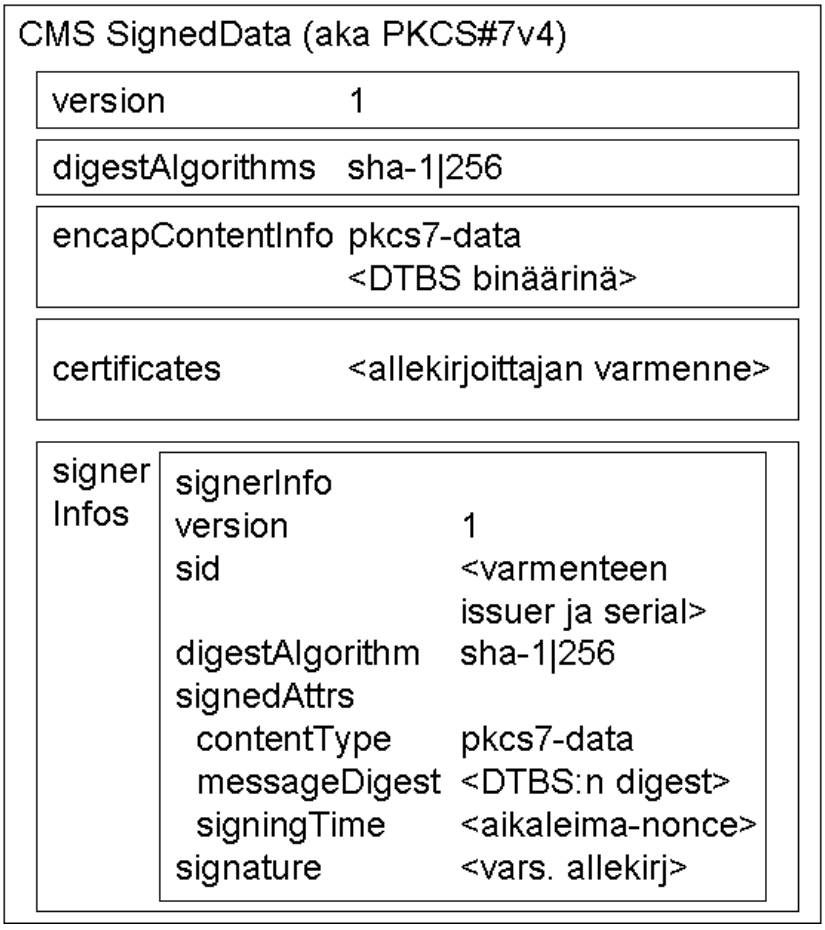
The User's digital signature compiled by the HMSSP is one of the following:

- **PKCS#7**-compliant signature message that is **base64**-encoded
- **PKCS#1**-compliant signature with the user's mobile certificate, **base64**-encoded into separate elements (see the example below)

The FiCom recommendation *does not* guarantee that the HMSSP will support other signature formats.

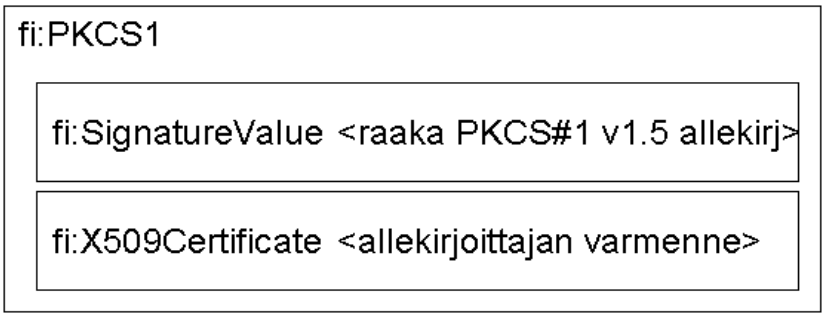
PKCS#7:

```
<MSS_Signature>
   <Base64Signature>ExTOCTrERKqKs+HY1lZfC5Xwd4SqjIXhPwWpHL6TPw2Fu7LtjMkxdEv42jgu
CMBYEGM97sdn23Ewz0NtG7RGRrVaU6Do5B5XfnRr827+bCoZ+Ll8Jgj1ft6PmZXzecDUzTC17QM6tS4+L
WDzTIWq/Qhdeie5b9k6U/EOvqd0wek=</Base64Signature>
</MSS_Signature>
```


CMS SignedData (aka PKCS#7v4)
version 1
digestAlgorithms sha-1|256
encapContentInfo pkcs7-data
<DTBS binäärinä>
certificates <allekirjoittajan varmenne>
signer Infos
signerInfo
version 1
sid <varmenteen issuer ja serial>
digestAlgorithm sha-1|256
signedAttrs
contentType pkcs7-data
messageDigest <DTBS:n digest>
signingTime <aikaleima-nonce>
signature <vars. allekirj>

PKCS#1:

```
<MSS_Signature>
  <fi:PKCS1 >
<fi:SignatureValue>Jbzj3d2HGmAVeFmcTSEfoZEmPk9uSZs65+gtFdzTP0SuazBJ6ym9SqDKcuqRvH
4WuvurUCgiuyo81bpK0w5pxhIHSBNz...</fi:SignatureValue >
<fi:X509Certificate>MIIDmDCCAoCgAwIBAgIDAI2rMA0GCSqGSIb3DQEBBQUAMEwxCzAJBgNVBAYTA
kZJ…</fi:X509Certificate>
  </ fi:PKCS1>
</MSS_Signature>
```



PKCS#1 is specified in the document RFC 2437: RSA Cryptography Specifications Version 2.0

PKCS#7 is specified in the document RFC 5652: Cryptographic Message Syntax.

Base64-encoding is specified, among others, in the W3C recommendation:
http://www.w3.org/TR/xmlschema-2/#base64Binary

6.3.3 Example: signature response

```
HTTP/1.0 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Body>
      <MSS_Signature>

      <!- Signature response begins -->
      <MSS_SignatureResp xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
MajorVersion="1" MinorVersion="1" MSSP_TransID="B653">

         <!- Application Provider ID, transaction ID, and timestamp -->
         <AP_Info AP_ID=" http://mss.teliasonera.com/mssURI/oycompanyab"
AP_TransID="A1203" AP_PWD="ssl" Instant="2003-06-24T21:32:00Z"/>

         <!- HMSSP identifier -->
         <MSSP_Info Instant="2003-06-24T21:32:28Z">
            <MSSP_ID>
               <URI>http://mss.elisa.fi</URI>
            </MSSP_ID>
         </MSSP_Info>

         <!- End user's identifier -->
         <MobileUser>
            <MSISDN>+358123456789</MSISDN>
         </MobileUser>

         <!- Digital signature -->
         <MSS_Signature>
            <Base64Signature>
ExTOCTrERKqKs+HY1lZfC5Xwd4SqjIXhPwWpHL6TPw2Fu7LtjMkxdEv42jguCMBYEGM97sdn23Ewz0NtG
7RGRrVaU6Do5B5XfnRr827+bCoZ+Ll8Jgj1ft6PmZXzecDUzTC17QM6tS4+LWDzTIWq/Qhdeie5b9k6U/
EOvqd0wek=
            </Base64Signature>
         </MSS_Signature>

         <!- Trsansaction end status -->
         <Status>
            <StatusCode Value="502"/>
            <StatusMessage>VALID_SIGNATURE</StatusMessage>
            <StatusDetail>
               <fi:ServiceResponses>

               <!-- Optional validation response from AE -->
                  <fi:ServiceResponse>
                     <fi:Description>
                        http://uri.etsi.org/TS102204/v1.1.2#validate
                     </fi:Description>
                     <fi:Entity>
                        <mss:URI>http://askonae.methics.fi</mss:URI>
                     </fi:Entity>
                     <fi:Status>
                        <mss:StatusCode Value="502"/>
                        <mss:StatusMessage>VALID_SIGNATURE</mss:StatusMessage>
                     </fi:Status>
                  </fi:ServiceResponse>
               </fi:ServiceResponses>
            </mss:StatusDetail>
```

```
          </Status>

       </MSS_SignatureResp>
       </MSS_Signature>
    </env:Body>
</env:Envelope>
```

6.4 Status request (MSS_StatusReq)

With a status request, the Application Provider inquires about the completion of a previously submitted service request from the HMSSP in asynchronous client-server communication. The status request is repeated from the Application Provider's system until the status of the event is other than ”outstanding” (504 OUTSTANDING_TRANSACTION). A separate HTTP session is opened for each status request, but at the level of the TCP protocol, the Application Provider is recommended to use persistent connections with an "idle timeout" of 5 minutes, for example, by the AP's systems if there is no traffic in the TCP channel for 5 minutes.

In order to avoid unnecessary load on the HMSSP systems, the Application Provider is recommended to make the first status request only 20 seconds after having received the HMSSP's acknowledgement of the receipt of the signature request (MSS_SignatureResp). In addition, it is recommended that inquiries concerning the same event not be repeated more frequently than once every five seconds. The latter restriction is based on the status inquiries causing unnecessary load when the event ends due to a timeout attributable to the User.

6.4.1 MSS_StatusReq: attributes

Name	Value	Description	Required
MajorVersion	”1”	Interface main version, currently 1.	Yes
MinorVersion	”1”	Interface sub-version, currently 1.	Yes
MSSP_TransID	NCName	Event number generated by the HMSSP.	Yes

6.4.1.1 MajorVersion and MinorVersion

The version of the MSS interface used is currently 1.1. If the version number specified by the AP differs from it, the AE returns status code 108 (INCOMPATIBLE_INTERFACE) to the AP.

6.4.1.2 MSSP_TransID

The attribute contains the event number that the Application Provider previously received in the signature response for the event from the HMSSP, i.e. a copy of the attribute MSSP_TransID of the element MSS_SignatureResp. The Application Provider uses this number to allocate the status request to the initiated signature event.

If the event number is missing or the HMSSP is not able to match the status request with a signature event for some reason, the HMSSP returns status code 101 (WRONG_PARAM) to the AP.

Example: MSS_StatusReq element attributes

The second line of the example contains a reference to the XMLSignature specifications that make XML-signed receipts possible.

```
<MSS_StatusReq xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" MajorVersion="1" MinorVersion="1"
MSSP_Trans_ID="B653">
```

6.4.2 MSS_StatusReq: elements

Name	Description	Required
AP_Info	The Application Provider's contact information and event number and timestamp issued by the AP for the event.	Yes
MSSP_Info	To the MSSP_ID field, a copy of the MSSP_ID field of the MSS_SignatureResp message.	Yes

6.4.2.1 AP_Info

The element is a direct copy of the element with the same name in the signature request previously submitted by the AP (MSS_SignatureReq). However, a timestamp must be specified for the status request, i.e. the attribute Instant of the element AP_Info must be renewed to correspond to the time when the status request was sent.

6.4.2.2 MSSP_Info

The element contains a copy of the MSSP_ID element of the MSS_SignatureResp message.

6.4.3 Example: status request

Note that the timestamp of the receipt request has been renewed and the name of the WSDL operation has been changed compared to the signature request example.

```
HTTP/1.0 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Body>

      <MSS_StatusQuery>

      <!- Status request begins -->
      <MSS_StatusReq xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
MajorVersion="1" MinorVersion="1" MSSP_TransID="B653">

         <!- Application Provider ID, transaction ID, and timestamp -->
         <AP_Info AP_ID="http://mss.teliasonera.com/mssURI/oycompanyab"
AP_TransID="A204" AP_PWD="ssl" Instant="2003-06-24T21:32:31Z"/>

         <!- HMSSP identifier -->
         <MSSP_Info>
            <MSSP_ID>
               <URI>http://mss.teliasonera.com</URI>
            </MSSP_ID>
         </MSSP_Info>
      </MSS_StatusReq>

      </MSS_StatusQuery>
   </env:Body>
</env:Envelope>
```

6.5 Status response (MSS_StatusResp)

In the status response, the HMSSP reports the status of the signature event initiated by the Application Provider.

6.5.1 MSS_StatusResp: attributes

Name	Value	Description	Required
MajorVersion	”1”	Interface main version, currently 1.	Yes
MinorVersion	”1”	Interface sub-version, currently 1.	Yes

6.5.1.1 MajorVersion and MinorVersion

The version of the MSS interface used is currently 1.1.

6.5.2 MSS_StatusResp: elements

Name	Description	Required
AP_Info	The Application Provider's contact information and event number and timestamp issued by the AP for the event.	Yes
MSSP_Info	AE's contact information.	Yes
MobileUser	The end user's contact information, currently always an element in the format: <MSISDN>+358x01234567</MSISDN> in which the prefix +358 (in Finland) of the mobile phone number is mandatory, i.e. the international number format is applied.	Yes
MSS_Signature	Digital signature.	No
Status	End status of the event.	Yes

6.5.2.1 AP_Info and MSSP_Info

The elements are identical with the elements with the same names in the signature request (MSS_SignatureReq).

6.5.2.2 MobileUser

The element is identical with the element with the same name in the signature request (MSS_SignatureReq).

6.5.2.3 MSS_Signature

The element is identical with the element with the same name in the signature response (MSS_SignatureResp).

6.5.2.4 Status

The element indicates the current status of the signature event. If the processing of the event is still incomplete, the HMSSP returns status code 504 (OUTSTANDING_TRANSACTION) to the AP. If the processing of the event is complete, the element is identical with the element with the same name in the signature response (MSS_SignatureResp).

6.5.3 Example: status response

```
HTTP/1.0 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Body>
      <MSS_StatusQuery>

      <!- Status reseponse begins -->
      <MSS_StatusResp xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
MajorVersion="1" MinorVersion="1" MSSP_TransID="B653">

         <!- Application Provider ID, transaction ID, and timestamp -->
         <AP_Info AP_ID=" http://mss.teliasonera.com/mssURI/oycompanyab"
AP_TransID="A1203" AP_PWD="ssl" Instant="2003-06-24T21:32:31Z"/>

         <!- HMSSP identifier -->
         <MSSP_Info Instant="2003-06-24T21:32:36Z>
            <MSSP_ID>
               <URI>http://mss.elisa.fi</URI>
            </MSSP_ID>
         </MSSP_Info>

         <!- End user's identifier -->
         <MobileUser>
            <MSISDN>+358123456789</MSISDN>
         </MobileUser>

         <!- Transaction status -->
         <Status>
            <StatusCode Value="504"/>
            <StatusMessage>
               OUTSTANDING_TRANSACTION
            </StatusMessage>
         </Status>
      </MSS_StatusResp>

      </MSS_StatusQuery>
   </env:Body>
</env:Envelope>
```

6.6 Receipt request (MSS_ReceiptReq)

With an receipt request, the Application Provider can choose to send an acknowledgement of the success or failure of the event to the User's mobile phone after the signature event.

6.6.1 MSS_ReceiptReq: attributes

Name	Value	Description	Required
MajorVersion	”1”	Interface main version, currently 1.	Yes
MinorVersion	”1”	Interface sub-version, currently 1.	Yes
MSSP_TransID	NCName	Event number generated by the MSSP.	Yes

6.6.1.1 MajorVersion and MinorVersion

The version of the MSS interface used is currently 1.1. If the version number specified by the AP differs from it, the AE returns status code 108 (INCOMPATIBLE_INTERFACE) to the AP.

6.6.1.2 MSSP_TransID

The attribute contains the event number that the Application Provider previously received in the signature response for the event from the HMSSP, i.e. a copy of the attribute MSSP_TransID of the element MSS_SignatureResp. With this number, the Application Provider matches the receipt request with a previous signature event.

If the event number is missing or the HMSSP is not able to match the receipt request with a signature event for some reason, the HMSSP returns status code 101 (WRONG_PARAM) to the AP.

Example: MSS_ReceiptReq element attributes

The second line of the example contains a reference to the XMLSignature specifications that make XML-signed receipts possible.

```
<MSS_ReceiptReq xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" MajorVersion="1" MinorVersion="1"
MSSP_Trans_ID="B653">
```

6.6.2 MSS_ReceiptReq: elements

Name	Description	Required
AP_Info	The Application Provider's contact information and event number and timestamp issued by the AP for the event.	Yes
MSSP_Info	AE's contact information.	Yes
MobileUser	The end user's contact information, currently always an element in the format: <MSISDN>+358123456789</MSISDN> in which the prefix +358 (in Finland) of the mobile phone number is mandatory, i.e. the international number format is applied.	Yes
Status	End status of the event.	No
Message	Application Provider's receipt message to the User.	No
SignedReceipt	Receipt electronically signed by the Application	No

	Provider.	

6.6.2.1 AP_Info and MobileUser

The elements are direct copies of the elements with the same names in the signature request previously submitted by the AP (MSS_SignatureReq). However, a timestamp must be specified for the receipt request, i.e. the attribute Instant of the element AP_Info must be renewed to correspond to the time when the receipt request was sent.

6.6.2.2 MSSP_Info

The element contains a copy of the MSSP_ID element of the MSS_SignatureResp message.

6.6.2.3 Status

End status of the event. The element is a direct copy of the element with the same name submitted to the Application Provider in the signature response (MSS_SignatureResp).

6.6.2.4 Message

Receipt message to the User. The message can be a free-form receipt of the transaction or a more general receipt message indicating the end status of the event. The element content is text/plain UTF-8. Maximum length is 160 characters.

6.6.2.5 SignedReceipt

Receipt message signed by the AP addressed to the User. The FiCom recommendation does not specify the use of this information. The AP provides instructions to the User via the business channel if this information is used. Usually, it is an electronically signed Message digest; the AP's public key required for checking the signature can be reported in the business channel.

6.6.3 Example: receipt request

Please note that the timestamp of the receipt request is renewed compared to the signature request example.

```
HTTP/1.0 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Body>

      <MSS_Receipt>

      <!- Receipt request begins -->
      <MSS_ReceiptReq xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
MajorVersion="1" MinorVersion="1" MSSP_TransID="B653">

         <!- Application Provider ID, transaction ID, and timestamp -->
         <AP_Info AP_ID=" http://mss.teliasonera.com/mssURI/oycompanyab"
AP_TransID="A204" AP_PWD="ssl" Instant="2003-06-24T21:32:31Z"/>

         <!- HMSSP identifier -->
         <MSSP_Info>
            <MSSP_ID>
               <URI>http://mss.teliasonera.com</URI>
            </MSSP_ID>
         </MSSP_Info>

         <!- End user's identifier -->
         <MobileUser>
            <MSISDN>+358123456789</MSISDN>
         </MobileUser>

         <!- Transaction end status -->
         <Status>
            <StatusCode Value="502"/>
            <StatusMessage>
               VALID_SIGNATURE
            </StatusMessage>
         </Status>

         <!- Message for end user's signing device -->
         <Message>
            Welcome to Company Ltd.
         </Message>

      </MSS_ReceiptReq>
      </MSS_Receipt>
   </env:Body>
</env:Envelope>
```

6.7 Receipt response (MSS_ReceiptResp)

6.7.1 MSS_ReceiptResp: attributes

Name	Value	Description	Required
MajorVersion	”1”	Interface main version, currently 1.	Yes
MinorVersion	”1”	Interface sub-version, currently 1.	Yes

6.7.1.1 MajorVersion and MinorVersion

The version of the MSS interface used is currently 1.1.

6.7.2 MSS_ReceiptResp: elements

Name	Description	Required
AP_Info	The Application Provider's contact information and event number and timestamp issued by the AP for the event.	Yes
MSSP_Info	AE's contact information.	Yes
Status	End status of the event.	No

6.7.2.1 AP_Info and MSSP_Info

The elements are identical with the elements with the same names in the signature request (MSS_SignatureReq).

6.7.2.2 Status

The element indicates the success or failure of the receipt request. If the receipt request was sent to the User's terminal device, the HMSSP returns status code 100 (REQUEST_OK) to the AP. If the AE or the HMSSP does not know the User, status code 105 (UNKNOWN_CLIENT) is returned to the AP. If the HMSSP cannot reach the User, the HMSSP can return status code 209 (OTA_ERROR) to the AP. If the HMSSP does not identify the MSSP_TransID, the HMSSP returns status code (WRONG_PARAM) to the AP.

6.7.3 Example: receipt response

```
HTTP/1.0 200 OK
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: ...

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Body>
      <MSS_Receipt>

      <!- Receipt response begins -->
      <MSS_ReceiptResp xmlns="http://uri.etsi.org/TS102204/v1.1.2#"
MajorVersion="1" MinorVersion="1" MSSP_TransID="B653">

         <!- Application Provider ID, transaction ID, and timestamp -->
         <AP_Info AP_ID=" http://mss.teliasonera.com/mssURI/oycompanyab"
AP_TransID="A1203" AP_PWD="ssl" Instant="2003-06-24T21:32:31Z"/>

         <!- HMSSP identifier -->
         <MSSP_Info Instant="2003-06-24T21:32:36Z>
            <MSSP_ID>
               <URI>http://mss.elisa.fi</URI>
            </MSSP_ID>
         </MSSP_Info>

         <!- Transaction end status -->
         <Status>
            <StatusCode Value="100"/>
            <StatusMessage>
               REQUEST_OK
            </StatusMessage>
         </Status>

      </MSS_ReceiptResp>
      </MSS_Receipt>
   </env:Body>
</env:Envelope>
```

6.8 Fault report message (SOAP FAULT)

In accordance with ETSI TS 102 204, the FiCom recommendation uses the standard fault report mechanism of the SOAP 1.2 specification. The entity that observes a fault during the event (AE, RE or HMSSP) interrupts the event and routes a SOAP FAULT message instead of the MSS service response message to the Application Provider. The structure of the message is described below.

Additional information on SOAP FAULT messages:
http://www.w3.org/TR/2003/REC-soap12-part0-20030624/#L11549

Element	Sub-element		Value	Description
Code	Value		”env:Receiver” tai ”env:Sender”	SOAP 1.2 -compliant fault code.
	Subcode	Value	Integer	Fault code of the MSS service
Reason	Text		reasontext	MSS service fault code explanation.
Node			anyURI	URI of the entity that detected the fault.
Role			anyURI	Role of the entity that detected the fault.

6.8.1 Code

The element contains the sub-elements Value and Subcode.

The Value sub-element includes a SOAP 1.2-compliant status code (main code). In practice, the code is either env:Sender or env:Receiver. Appendix C indicates which code is used in each fault situation. SOAP 1.2 also specifies codes env:VersionMismatch, env:MustUnderstand and env:DataEncodingUnknown. The FiCom recommendation does not include these codes and they should not be needed if all parties comply with the SOAP version and service message structures of the recommendation and use the UTF-8 character set throughout the message structure. From the point of view of the Application Provider, the actual reason for the fault is indicated in the MSS service status code, not the SOAP-level main code.

The Subcode sub-element consists of the sub-element Value that includes the MSS service status code (subcode). The subcodes and their explanations are described in appendix C. The concrete application guidelines for the subcodes are indicated above in the descriptions of MSS message structures. SOAP 1.2 allows the embedding of subcodes. In the FiCom recommendation, the top-level Subcode element is used for indicating the subcode specified by the ETSI 204 standard and the more detailed subcode specified by FiCom is indicated with the Subcode element of the next level.

The value of the sub-element Value is in the format

```
<env:Value>fi:_(virhekoodi)</env:Value>
```

6.8.2 Reason

The only sub-element of the element, Text, includes an explanation corresponding to the MSS service status code.

6.8.3 Detail

The element indicates the explanation specifying the fault, either free-form text or a structure with sub-elements.

6.8.4 Node

The element indicates the URI of the entity that detected the fault.

6.8.5 Role

The element indicates the role of the entity that detected the fault. The role is one of the following:

http://uri.etsi.org/TS102207/v1.1.3#role_AcquiringEntity
http://uri.etsi.org/TS102207/v1.1.3#role_HomeMSSP
http://uri.etsi.org/TS102207/v1.1.3#role_IdentityIssuer
http://uri.etsi.org/TS102207/v1.1.3#role_RoutingEntity
http://uri.etsi.org/TS102207/v1.1.3#role_VerifyingEntity

6.8.6 Example: SOAP FAULT

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Receiver</env:Value>
        <env:Subcode>
          <env:Value>fi:_208</env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en">EXPIRED_TRANSACTION</env:Text>
      </env:Reason>
      <env:Node>mss.elisa.fi</env:Node>
      <env:Role>http://uri.etsi.org/TS102207/v1.1.3#role_HomeHMSSP</env:Role>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

Appendix A: XML Schema (ETSI)

The FiCom recommendation is compliant with the XML Schema specifying the ETSI 102 204 standard message structures presented here.

```
<xs:schema targetNamespace="http://uri.etsi.org/TS102204/v1.1.2#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:mss="http://uri.etsi.org/TS102204/v1.1.2#"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:env="http://www.w3.org/2003/05/soap-envelope"
elementFormDefault="qualified">
   <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"/>
   <xs:import namespace="http://www.w3.org/2001/04/xmlenc#"/>
   <xs:import namespace="http://www.w3.org/2003/05/soap-envelope"/>

   <xs:complexType name="MessageAbstractType" abstract="true">
      <xs:sequence>
         <xs:element name="AP_Info">
            <xs:complexType>
               <xs:attribute name="AP_ID" type="xs:anyURI" use="required"/>
               <xs:attribute name="AP_TransID" type="xs:NCName"
use="required"/>
               <xs:attribute name="AP_PWD" type="xs:string" use="required"/>
               <xs:attribute name="Instant" type="xs:dateTime" use="required"/>
               <xs:attribute name="AP_URL" type="xs:anyURI" use="optional"/>
            </xs:complexType>
         </xs:element>
         <xs:element name="MSSP_Info">
            <xs:complexType>
               <xs:sequence>
                  <xs:element name="MSSP_ID" type="mss:MeshMemberType"/>
               </xs:sequence>
               <xs:attribute name="Instant" type="xs:dateTime" use="optional"/>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
      <xs:attribute name="MajorVersion" type="xs:integer" use="required"/>
      <xs:attribute name="MinorVersion" type="xs:integer" use="required"/>
   </xs:complexType>

   <xs:element name="MSS_SignatureReq" type="mss:MSS_SignatureReqType"/>
   <xs:complexType name="MSS_SignatureReqType">
      <xs:complexContent>
         <xs:extension base="mss:MessageAbstractType">
            <xs:sequence>
               <xs:element name="MobileUser" type="mss:MobileUserType"/>
               <xs:element name="DataToBeSigned" type="mss:DataType"/>
               <xs:element name="DataToBeDisplayed" type="mss:DataType"
minOccurs="0"/>
               <xs:element name="SignatureProfile" type="mss:mssURIType"
minOccurs="0"/>
               <xs:element name="AdditionalServices" minOccurs="0">
                  <xs:complexType>
                     <xs:sequence>
                        <xs:element name="Service"
type="mss:AdditionalServiceType" maxOccurs="unbounded"/>
                     </xs:sequence>
                  </xs:complexType>
               </xs:element>
               <xs:element name="MSS_Format" type="mss:mssURIType"
minOccurs="0"/>
```

```
                <xs:element name="KeyReference" type="mss:KeyReferenceType"
minOccurs="0"/>
                <xs:element name="SignatureProfileComparison"
type="mss:SignatureProfileComparisonType" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="ValidityDate" type="xs:dateTime"
use="optional"/>
            <xs:attribute name="TimeOut" type="xs:positiveInteger"
use="optional"/>
            <xs:attribute name="MessagingMode" type="mss:MessagingModeType"
use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_SignatureResp" type="mss:MSS_SignatureRespType"/>
   <xs:complexType name="MSS_SignatureRespType">
      <xs:complexContent>
         <xs:extension base="mss:MessageAbstractType">
            <xs:sequence>
               <xs:element name="MobileUser" type="mss:MobileUserType"/>
               <xs:element name="MSS_Signature" type="mss:SignatureType"
minOccurs="0"/>
               <xs:element name="SignatureProfile" type="mss:mssURIType"
minOccurs="0"/>
               <xs:element name="Status" type="mss:StatusType"/>
            </xs:sequence>
            <xs:attribute name="MSSP_TransID" type="xs:NCName" use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_StatusReq" type="mss:MSS_StatusReqType"/>
   <xs:complexType name="MSS_StatusReqType">
      <xs:complexContent>
         <xs:extension base="mss:MessageAbstractType">
            <xs:attribute name="MSSP_TransID" type="xs:NCName" use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS StatusResp" type="mss:MSS StatusRespType"/>
   <xs:complexType name="MSS_StatusRespType">
      <xs:complexContent>
         <xs:extension base="mss:MessageAbstractType">
            <xs:sequence>
               <xs:element name="MobileUser" type="mss:MobileUserType"/>
               <xs:element name="MSS_Signature" type="mss:SignatureType"
minOccurs="0"/>
               <xs:element name="Status" type="mss:StatusType"/>
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_RegistrationReq" type="mss:MSS_RegistrationReqType"/>
   <xs:complexType name="MSS_RegistrationReqType">
      <xs:complexContent>
         <xs:extension base="mss:MessageAbstractType">
            <xs:sequence>
               <xs:element name="MobileUser" type="mss:MobileUserType"/>
               <xs:element name="EncryptedData" type="xenc:EncryptedType"
minOccurs="0"/>
```

```
                <xs:element name="EncryptResponseBy" type="xs:anyURI"
minOccurs="0"/>
                <xs:element name="CertificateURI" type="xs:anyURI"
minOccurs="0"/>
                <xs:element name="X509Certificate" type="xs:base64Binary"
minOccurs="0"/>
                <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_RegistrationResp" type="mss:MSS_RegistrationRespType"/>
   <xs:complexType name="MSS_RegistrationRespType">
       <xs:complexContent>
          <xs:extension base="mss:MessageAbstractType">
             <xs:sequence>
                <xs:element name="Status" type="mss:StatusType"/>
                <xs:element name="EncryptedData" type="xenc:EncryptedType"
minOccurs="0"/>
                <xs:element name="CertificateURI" type="xs:anyURI"
minOccurs="0"/>
                <xs:element name="X509Certificate" type="xs:base64Binary"
minOccurs="0"/>
                <xs:element name="PublicKey" type="xs:base64Binary"
minOccurs="0"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_ProfileReq" type="mss:MSS_ProfileReqType"/>
   <xs:complexType name="MSS_ProfileReqType">
       <xs:complexContent>
          <xs:extension base="mss:MessageAbstractType">
             <xs:sequence>
                <xs:element name="MobileUser" type="mss:MobileUserType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_ProfileResp" type="mss:MSS_ProfileRespType"/>
   <xs:complexType name="MSS_ProfileRespType">
       <xs:complexContent>
          <xs:extension base="mss:MessageAbstractType">
             <xs:sequence>
                <xs:element name="SignatureProfile" type="mss:mssURIType"
minOccurs="0" maxOccurs="unbounded"/>
                <xs:element name="Status" type="mss:StatusType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_ReceiptReq" type="mss:MSS_ReceiptReqType"/>
   <xs:complexType name="MSS_ReceiptReqType">
       <xs:complexContent>
          <xs:extension base="mss:MessageAbstractType">
             <xs:sequence>
                <xs:element name="MobileUser" type="mss:MobileUserType"/>
                <xs:element name="Status" type="mss:StatusType" minOccurs="0"/>
                <xs:element name="Message" type="mss:DataType" minOccurs="0"/>
```

```
                <xs:element name="SignedReceipt" type="ds:SignatureType"
minOccurs="0"/>
             </xs:sequence>
             <xs:attribute name="MSSP_TransID" type="xs:NCName" use="required"/>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>

    <xs:element name="MSS_ReceiptResp" type="mss:MSS_ReceiptRespType"/>
    <xs:complexType name="MSS_ReceiptRespType">
       <xs:complexContent>
          <xs:extension base="mss:MessageAbstractType">
             <xs:sequence>
                <xs:element name="Status" type="mss:StatusType"/>
             </xs:sequence>
          </xs:extension>
       </xs:complexContent>
    </xs:complexType>

    <xs:element name="MSS_HandshakeReq" type="mss:MSS_HandshakeReqType"/>
    <xs:complexType name="MSS_HandshakeReqType">
       <xs:complexContent>
          <xs:extension base="mss:MessageAbstractType">
             <xs:sequence>
                <xs:element name="SecureMethods">
                   <xs:complexType>
                      <xs:attribute name="MSS_Signature" type="xs:boolean"
use="required"/>
                      <xs:attribute name="MSS_Registration" type="xs:boolean"
use="required"/>
                      <xs:attribute name="MSS_Notification" type="xs:boolean"
use="required"/>
                      <xs:attribute name="MSS_ProfileQuery" type="xs:boolean"
use="required"/>
                      <xs:attribute name="MSS_Receipt" type="xs:boolean"
use="required"/>
                      <xs:attribute name="MSS_Status" type="xs:boolean"
use="required"/>
                   </xs:complexType>
                </xs:element>
                <xs:element name="Certificates">
                   <xs:complexType>
                      <xs:sequence>
                         <xs:element name="Certificate" type="xs:base64Binary"
minOccurs="0" maxOccurs="unbounded"/>
                      </xs:sequence>
                   </xs:complexType>
                </xs:element>
                <xs:element name="RootCAs">
                   <xs:complexType>
                      <xs:sequence>
                         <xs:element name="DN" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
                      </xs:sequence>
                   </xs:complexType>
                </xs:element>
                <xs:element name="SignatureAlgList">
                   <xs:complexType>
                      <xs:sequence>
                         <xs:element name="Algorithm" type="mss:mssURIType"
minOccurs="0" maxOccurs="unbounded"/>
                      </xs:sequence>
                   </xs:complexType>
                </xs:element>
```

```
            </xs:sequence>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>

   <xs:element name="MSS_HandshakeResp" type="mss:MSS_HandshakeRespType"/>
   <xs:complexType name="MSS_HandshakeRespType">
      <xs:complexContent>
         <xs:extension base="mss:MessageAbstractType">
            <xs:sequence>
               <xs:element name="SecureMethods">
                  <xs:complexType>
                     <xs:attribute name="MSS_Signature" type="xs:boolean"
use="required"/>
                     <xs:attribute name="MSS_Registration" type="xs:boolean"
use="required"/>
                     <xs:attribute name="MSS_Notification" type="xs:boolean"
use="required"/>
                     <xs:attribute name="MSS_ProfileQuery" type="xs:boolean"
use="required"/>
                     <xs:attribute name="MSS_Receipt" type="xs:boolean"
use="required"/>
                     <xs:attribute name="MSS_Status" type="xs:boolean"
use="required"/>
                  </xs:complexType>
               </xs:element>
               <xs:element name="MatchingMSSPCertificates">
                  <xs:complexType>
                     <xs:sequence>
                        <xs:element name="Certificate" type="xs:base64Binary"
minOccurs="0" maxOccurs="unbounded"/>
                     </xs:sequence>
                  </xs:complexType>
               </xs:element>
               <xs:element name="MatchingAPCertificates">
                  <xs:complexType>
                     <xs:sequence>
                        <xs:element name="Certificate" type="xs:base64Binary"
minOccurs="0" maxOccurs="unbounded"/>
                     </xs:sequence>
                  </xs:complexType>
               </xs:element>
               <xs:element name="MatchingSigAlgList">
                  <xs:complexType>
                     <xs:sequence>
                        <xs:element name="Algorithm" type="mss:mssURIType"
minOccurs="0" maxOccurs="unbounded"/>
                     </xs:sequence>
                  </xs:complexType>
               </xs:element>
            </xs:sequence>
            <xs:attribute name="MSSP_TransID" type="xs:NCName" use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>

   <xs:complexType name="MobileUserType">
      <xs:sequence>
         <xs:element name="IdentityIssuer" type="mss:MeshMemberType"
minOccurs="0"/>
         <xs:element name="UserIdentifier" type="xs:string" minOccurs="0"/>
         <xs:element name="HomeMSSP" type="mss:MeshMemberType" minOccurs="0"/>
         <xs:element name="MSISDN" type="xs:string" minOccurs="0"/>
      </xs:sequence>
```

```
   </xs:complexType>

   <xs:complexType name="DataType">
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="MimeType" type="xs:string" use="optional"/>
            <xs:attribute name="Encoding" type="xs:string" use="optional"/>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>

   <xs:simpleType name="SignatureProfileComparisonType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="exact"/>
         <xs:enumeration value="minimum"/>
         <xs:enumeration value="better"/>
      </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="MessagingModeType">
      <xs:restriction base="xs:string">
         <xs:enumeration value="synch"/>
         <xs:enumeration value="asynchClientServer"/>
         <xs:enumeration value="asynchServerServer"/>
      </xs:restriction>
   </xs:simpleType>

   <xs:complexType name="DigestAlgAndValueType">
      <xs:sequence>
         <xs:element name="DigestMethod" type="ds:DigestMethodType"
minOccurs="0"/>
         <xs:element name="DigestValue" type="ds:DigestValueType"/>
      </xs:sequence>
   </xs:complexType>

   <xs:complexType name="mssURIType">
      <xs:sequence>
         <xs:element name="mssURI" type="xs:anyURI"/>
         <xs:element name="DigestAlgAndValue" type="mss:DigestAlgAndValueType"
minOccurs="0"/>
         <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
   </xs:complexType>

   <xs:complexType name="MeshMemberType">
      <xs:sequence>
         <xs:element name="DNSName" type="xs:string" minOccurs="0"/>
         <xs:element name="IPAddress" type="xs:string" minOccurs="0"/>
         <xs:element name="URI" type="xs:anyURI" minOccurs="0"/>
         <xs:element name="IdentifierString" type="xs:string" minOccurs="0"/>
      </xs:sequence>
   </xs:complexType>

   <xs:complexType name="KeyReferenceType">
      <xs:sequence>
         <xs:element name="CertificateURL" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="CertificateIssuerDN" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
         <xs:element name="HashOfUsersPublicKey"
type="mss:DigestAlgAndValueType" minOccurs="0" maxOccurs="unbounded"/>
         <xs:element name="HashOfCAPublicKey" type="mss:DigestAlgAndValueType"
minOccurs="0" maxOccurs="unbounded"/>
```

```
            <xs:any namespace="##other" processContents="lax"/>
         </xs:sequence>
      </xs:complexType>

      <xs:complexType name="SignatureType">
         <xs:choice>
            <xs:element name="XMLSignature" type="ds:SignatureType"/>
            <xs:element name="Base64Signature" type="xs:base64Binary"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
            <!-- this can also be an advanced XML Signature-->
         </xs:choice>
      </xs:complexType>

      <xs:element name="MSS_MessageSignature">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="ds:Signature"/>
         </xs:sequence>
         <xs:attribute ref="env:role" use="required"/>
         <xs:attribute ref="env:mustUnderstand" use="required"/>
      </xs:complexType>
      </xs:element>

      <xs:complexType name="AdditionalServiceType">
         <xs:sequence>
            <xs:element name="Description" type="mss:mssURIType"/>
            <xs:element name="Entity" type="mss:MeshMemberType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>

      <xs:complexType name="StatusType">
         <xs:sequence>
            <xs:element name="StatusCode" type="mss:StatusCodeType"/>
            <xs:element name="StatusMessage" type="xs:string" minOccurs="0"/>
            <xs:element name="StatusDetail" type="mss:StatusDetailType"
minOccurs="0"/>
         </xs:sequence>
      </xs:complexType>

      <xs:complexType name="StatusCodeType">
         <xs:sequence>
            <xs:element name="StatusCode" type="mss:StatusCodeType" minOccurs="0"/>
         </xs:sequence>
         <xs:attribute name="Value" type="xs:integer" use="required"/>
      </xs:complexType>

      <xs:complexType name="StatusDetailType">
         <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
</xs:schema>
```

Appendix B: XML Schema (FiCom)

The added value services described in the FiCom recommendation are specified in the XML Schema below.

```
<xs:schema targetNamespace="http://mss.ficom.fi/TS102204/v1.0.0#"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:mss="http://uri.etsi.org/TS102204/v1.1.2#"
           xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
           elementFormDefault="qualified">

  <xs:import namespace="http://uri.etsi.org/TS102204/v1.1.2#"
        schemaLocation="MSS-plus.xsd"/>
  <xs:import namespace="urn:oasis:names:tc:SAML:2.0:protocol"
             schemaLocation="saml-schema-protocol-2.0.xsd"/>

  <xs:element name="PKCS1">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SignatureValue" type="xs:base64Binary"/>
        <xs:element name="X509Certificate" type="xs:base64Binary"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="NoSpamCode">
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="xs:string">
          <xs:attribute name="verify" type="xs:string" default="yes"/>
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:element name="EventID" type="xs:string"/>
  <xs:element name="SessionID" type="xs:string"/>

  <xs:element name="UserLang" type="xs:string" default="fi"/>

  <xs:element name="ServiceResponses">
    <xs:complexType>
      <xs:sequence>
   <xs:element name="ServiceResponse" minOccurs=”0” maxOccurs="unbounded">
         <xs:complexType>
           <xs:sequence>
             <xs:element name="Description" type="mss:mssURIType"/>
             <xs:element name="Entity" type="mss:MeshMemberType" minOccurs="0"/>
        <xs:element name="Status" type="mss:StatusType" minOccurs="0"/>
             <xs:element ref="samlp:Response" minOccurs="0"/>
             <xs:any namespace="##other" processContents="lax" />
           </xs:sequence>
         </xs:complexType>
       </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Appendix C: Status codes

The FiCom recommendation covers the following status codes specified by ETSI. The list does not include, for example, codes related to the asynchronous messaging mode and message types not covered by the recommendation.

The first list includes any fault codes returned to the AP in the SOAP FAULT structure. The latter list includes the status codes returned in the Status element of MSS_SignatureResp and MSS_ReceiptResp messages.

The HMSSP is not necessarily able to detect the interruption of the event on the terminal device (status codes 401-406). In this case, the HMSSP might wait until the time limit is exceeded and return status code 208 (EXPIRED_TRANSACTION). On the other hand, the HMSSP might be able to report a more detailed reason for the fault that can be reported using the status codes specified by ETSI; such an HMSSP may return a status code specifying the problem, a "specifier", in the Detail element of the SOAP FAULT message structure. The specifiers according to the FiCom recommendation are mentioned in the list, and the list is followed by a message example utilising the specifier.

Codes 423, 424 and 600-603 are not taken into account at all in the FiCom recommendation.

Codes 700-780 are error codes recorded during signature roaming. Their more detailed semantics comply with the ETSI TS 102 207 specification. This involves a system error independent of the Application Provider in almost all cases. An exception is code 720 in cases where the Application Provider has set an unreasonably short expiry period for the event.

SOAP FAULT message status codes (env:Sender)

Code		Explanation	Description
101		WRONG_PARAM	One or more of the service request parameters is invalid.
	1011	Invalid NoSpamCode	The spam prevention code transmitted by the AP does not match the HMSSP code.
	1012	Missing NoSpamCode	The blank spam prevention code transmitted by the AP does not match the HMSSP code.
	1013	Illegal MessagingMode	The AE does not support the messaging mode used by the AP.
	1014	Unknown AdditionalService	The MSSP does not recognise one or more added value service requested by the AP.
	1015	DataToBeDisplayed not supported	The HMSSP does not support the DataToBeDisplayed element.

	1016	Unsupported MimeType and/or Encoding	The HMSSP does not support the requested DTBS MimeType and/or Encoding for the requested signature profile.
	1017	MSS_Format unsupported for MimeType, Encoding and/or SignatureProfile	The HMSSP does not support the requested signature format for the requested MimeType, Encoding and/or signature profile.
102		MISSING_PARAM	One or more of the required service request parameters is missing.
	1021	DataToBeDisplayed missing	The DataToBeDisplayed element required by the HMSSP is missing.
	1022	DataToBeSigned missing	The DataToBeSigned element required by the HMSSP is missing.
103		WRONG_DATA_LENGTH	The value of the DataToBeSigned element is shorter or longer than permitted.
104		UNAUTHORIZED_ACCESS	The AP is unknown, the password provided is invalid, the AP is requesting an added value service which it is not entitled to use or the AP is requesting a SignatureProfile or added value service or PersonIdentity service attribute whose use the User has prohibited.
	1041	User-disabled SignatureProfile	The AP is requesting a SignatureProfile whose use the User has prohibited.
	1042	User disabled AdditionalService	The AP is requesting an added value service or PersonIdentity service attribute whose use the User has prohibited.
	1043	Service suspended	The User's right to use the service is blocked.
	1044	SignatureProfile not allowed	The AP is requesting a SignatureProfile whose use the AE has prohibited.

	1045	AdditionalService not allowed	The AP is requesing an added value service or PersonIdentity service attribute whose use the AE has prohibited.
105		UNKNOWN_CLIENT	The AE or the HMSSP does not know the User whom the service request concerns.
	1051	Malformatted user identifier	The user's contact information is wrongly formatted.
	1052	User identifier does not exist	Unknown User.
	1053	Unregistered user	The User is not registered.
	1054	Incompatible SIM card	The User's SIM card is not suitable for the service.
107		INAPPROPRIATE_DATA	The AP has specified a MIME type or encoding not supported by the HMSSP for the element DataToBeSigned, DataToBeDisplayed or Message.
108		INCOMPATIBLE_INTERFACE	The AE or the HMSSP does not support the interface version number reported by the AP.
701		A Roaming Header block is missing.	The event was interrupted due to a technical problem that occurred in signature roaming.
702		An Identity Issuer Header block is missing.	
703		A Home MSSP Header block is missing.	
710		Appropriate input information is missing.	
720		The validity date of the transaction has expired.	The time limit for the event has expired during the routing of the service request.

SOAP FAULT message status codes (env:Receiver)

109	UNSUPPORTED_PROFILE	The AP has specified a signature profile not supported by the HMSSP.

208		EXPIRED_TRANSACTION	The time limit for the event has been exceeded.
	2081	Server timeout	The server did not react within the time limit
	2082	User timeout	The User did not react within the time limit
209		OTA_ERROR	The HMSSP could not reach the User. The mobile phone is switched off or there is a connection problem.
	2091	Unknown OTA Error	Unknown OTA error
	2092	Card not found (OTA DB)	The card is not found in the OTA database
	2093	ME Communication Error	Error due to the phone in OTA communication
	2094	Invalid capabilities	The properties of the SIM application are not sufficient for fulfilling the request
401		USER_CANCEL	The User has cancelled the event.
	4011	User Cancel	The User has cancelled the event.
	4012	Incorrect PoP	The User entered an invalid PoP code
	4013	Postponed signature	The User has postponed the event
402		PIN_NR_BLOCKED	Signature with the terminal device has failed.
	4021	PIN blocked	
403		CARD_BLOCKED	
	4031	PUK blocked	
	4032	PIN blocked	
404		NO_KEY_FOUND	
	4041	Requested Key not found	
	4042	Incorrect key usage	
405		NO_URL_FOUND	
406		PB_SIGNATURE_PROCESS	
407		REGISTRATION_NOK	Error during the registration process
	4071	Unknown Registration Error	Unknown error during the registration process
	4072	Registration Failed on Server	Error in the server during the registration process
	4073	Registration Failed on SIM	Error in the SIM card during the registration process

422		NO_CERT_FOUND	The User's signature succeeded but no certificate corresponding to the signature could be attached.
423		CRL_PB	
424		CRL_EXPIRED	The AE's CRL copy is expired and CRL update fails.
425		ERROR_CERTIFICATE	The certificate was found to be invalid in the AE's validation.
750		Unable to provide Routing Entity services.	The event was interrupted due to a technical problem that occurred in signature roaming.
760		Unable to provide Identity Issuer services.	
770		Unable to provide Verifying Entity services.	
780		Unable to provide services.	
900		INTERNAL_ERROR	Internal system error at HMSSP
	9001	Unknown Internal error	Unknown problem.
	9002	Server Error	Server problem
	9003	SIM Application error	SIM application in error mode
	9004	SIM Configuration error	The configuration of the SIM application is invalid
	9005	DTBD missing	DTBD is missing
	9006	Invalid key length	Key length is invalid
	9007	Invalid hash type	Unknown/unsupported hash type or algorithm
	9008	Invalid Key Algorithm	Invalid key type

MSS message status codes

Code	Name	Description
100	REQUEST_OK	Receipt request submitted to the user.
500	SIGNATURE	Digital signature has been compiled. The signature has not yet been validated.

501	REVOKED_ CERTIFICATE	The electronic signature has been compiled but the User's certificate is in the blacklist. Note: the expiry of the certificate is indicated using code 503.
502	VALID_SIGNATURE	The electronic signature has been compiled and found to be valid.
503	INVALID_SIGNATURE	The electronic signature has been compiled but the signature has been found to be invalid upon validation or the certificate has expired.
504	OUTSTANDING_TRANSACTIO N	The signature response is not yet complete.
505	CONSTRAINT_MISMATCH	The event is otherwise valid but the User's certificate does not fulfil the additional conditions specified for it in the signature profile. This is probably a test certificate. (FiCom-specific status code.)

Example: SOAP FAULT (with specifier)

```
<?xml version=”1.0” encoding=”UTF-8”?>
<soapenv:Envelope xmlns:soapenv=”http://www.w3.org/2003/05/soap-envelope”
xmlns:xsd=”http://www.w3.org/2001/XMLSchema” xmlns:xsi=”http://www.w3.org/2001/XMLSchema-
instance”>
 <soapenv:Body>
  <soapenv:Fault>
   <soapenv:Code>
    <soapenv:Value>soapenv:Receiver</soapenv:Value>
    <soapenv:Subcode>
        <soapenv:Value>fi:_101</soapenv:Value>
        <soapenv:Subcode>
            <soapenv:Value>fi:1014</soapenv:Value>
        </soapenv:Subcode>
    </soapenv:Subcode>
   </soapenv:Code>
   <soapenv:Reason>
    <soapenv:Text xml:lang=”en”>User cancel</soapenv:Text>
   </soapenv:Reason>
  </soapenv:Fault>
 </soapenv:Body>
</soapenv:Envelope>
```

Appendix D: GSM 03.38 character set

The table below includes the GSM 03.38 character set used in the terminal device and the corresponding Unicode UCS16 and UTF-8 character codes.

Hex	Dec	Character name	Character	UCS16 Char	UTF-8 Hex
0x00	0	COMMERCIAL AT	@	\u0040	0x40
0x01	1	POUND SIGN	£	\u00A3	0xC2A3
0x02	2	DOLLAR SIGN	$	\u00A4	0xC2A4
0x03	3	YEN SIGN	¥	\u00A5	0xC2A5
0x04	4	LATIN SMALL LETTER E WITH GRAVE	è	\u00A8	0xC3A8
0x05	5	LATIN SMALL LETTER E WITH ACUTE	é	\u00A9	0xC3A9
0x06	6	LATIN SMALL LETTER U WITH GRAVE	ù	\u00B9	0xC3B9
0x07	7	LATIN SMALL LETTER I WITH GRAVE	ì	\u00AC	0xC3AC
0x08	8	LATIN SMALL LETTER O WITH GRAVE	ò	\u00B2	0xC3B2
0x09	9	LATIN CAPITAL LETTER C WITH CEDILLA	Ç	\u00C7	0xC387
0x0A	10	LINE FEED		\u000A	0x0A
0x0B	11	LATIN CAPITAL LETTER O WITH STROKE	Ø	\u00D8	0xC398
0x0C	12	LATIN SMALL LETTER O WITH STROKE	ø	\u00F8	0xC3B8
0x0D	13	CARRIAGE RETURN		\u000D	0x0D
0x0E	14	LATIN CAPITAL LETTER A WITH RING ABOVE	Å	\u00C5	0xC385
0x0F	15	LATIN SMALL LETTER A WITH RING ABOVE	å	\u00E5	0xC3A5
0x10	16	GREEK CAPITAL LETTER DELTA	Δ	\u0394	0xCE94
0x11	17	LOW LINE	_	\u005F	0x5F
0x12	18	GREEK CAPITAL LETTER PHI	Φ	\u03A6	0xCEA6
0x13	19	GREEK CAPITAL LETTER GAMMA	Γ	\u0393	0xCE93
0x14	20	GREEK CAPITAL LETTER LAMBDA	Λ	\u039B	0xCE9B
0x15	21	GREEK CAPITAL LETTER OMEGA	Ω	\u03A9	0xCEA9
0x16	22	GREEK CAPITAL LETTER PI	Π	\u03A0	0xCEA0
0x17	23	GREEK CAPITAL LETTER PSI	Ψ	\u03A8	0xCEA8
0x18	24	GREEK CAPITAL LETTER SIGMA	Σ	\u03A3	0xCEA3
0x19	25	GREEK CAPITAL LETTER THETA	Θ	\u0398	0xCE98
0x1A	26	GREEK CAPITAL LETTER XI	Ξ	\u039E	0xCE9E
0x1B	27	ESCAPE TO EXTENSION TABLE			
0x1B0A	27 10	FORM FEED		\u000C	0x0C
0x1B14	27 20	CIRCUMFLEX ACCENT	^	\u005E	0x5E
0x1B28	27 40	LEFT CURLY BRACKET	{	\u007B	0x7B
0x1B29	27 41	RIGHT CURLY BRACKET	}	\u007D	0x7D
0x1B2F	27 47	REVERSE SOLIDUS (BACKSLASH)	\	\u005C	0x5C
0x1B3C	27 60	LEFT SQUARE BRACKET	[	\u005B	0x5B
0x1B3D	27 61	TILDE	~	\u007E	0x7E
0x1B3E	27 62	RIGHT SQUARE BRACKET	]	\u005D	0x5D
0x1B40	27 64	VERTICAL BAR	|	\u007C	0x7C
0x1B65	27 101	EURO SIGN	€	\u20AC	0xE282AC
0x1C	28	LATIN CAPITAL LETTER AE	Æ	\u00C6	0xC386
0x1D	29	LATIN SMALL LETTER AE	æ	\u00E6	0xC3A6
0x1E	30	LATIN SMALL LETTER SHARP S (German)	ß	\u00DF	0xC39F
0x1F	31	LATIN CAPITAL LETTER E WITH ACUTE	É	\u00C9	0xC389
0x20	32	SPACE		\u0020	0x20
0x21	33	EXCLAMATION MARK	!	\u0021	0x21
0x22	34	QUOTATION MARK	"	\u0022	0x22

0x23	35	NUMBER SIGN	#	\u0023	0x23
0x24	36	CURRENCY SIGN	¤	\u00A4	0xC2A4
0x25	37	PERCENT SIGN	%	\u0025	0x25
0x26	38	AMPERSAND	&	\u0026	0x26
0x27	39	APOSTROPHE	'	\u0027	0x27
0x28	40	LEFT PARENTHESIS	(	\u0028	0x28
0x29	41	RIGHT PARENTHESIS	)	\u0029	0x29
0x2A	42	ASTERISK	*	\u002A	0x2A
0x2B	43	PLUS SIGN	+	\u002B	0x2B
0x2C	44	COMMA	,	\u002C	0x2C
0x2D	45	HYPHEN-MINUS	-	\u002D	0x2D
0x2E	46	FULL STOP	.	\u002E	0x2E
0x2F	47	SOLIDUS (SLASH)	/	\u002F	0x2F
0x30	48	DIGIT ZERO	0	\u0030	0x30
0x31	49	DIGIT ONE	1	\u0031	0x31
0x32	50	DIGIT TWO	2	\u0032	0x32
0x33	51	DIGIT THREE	3	\u0033	0x33
0x34	52	DIGIT FOUR	4	\u0034	0x34
0x35	53	DIGIT FIVE	5	\u0035	0x35
0x36	54	DIGIT SIX	6	\u0036	0x36
0x37	55	DIGIT SEVEN	7	\u0037	0x37
0x38	56	DIGIT EIGHT	8	\u0038	0x38
0x39	57	DIGIT NINE	9	\u0039	0x39
0x3A	58	COLON	:	\u003A	0x3A
0x3B	59	SEMICOLON	;	\u003B	0x3B
0x3C	60	LESS-THAN SIGN	<	\u003C	0x3C
0x3D	61	EQUALS SIGN	=	\u003D	0x3D
0x3E	62	GREATER-THAN SIGN	>	\u003E	0x3E
0x3F	63	QUESTION MARK	?	\u003F	0x3F
0x40	64	INVERTED EXCLAMATION MARK	¡	\u00A1	0xC2A1
0x41	65	LATIN CAPITAL LETTER A	A	\u0041	0x41
0x42	66	LATIN CAPITAL LETTER B	B	\u0042	0x42
0x43	67	LATIN CAPITAL LETTER C	C	\u0043	0x43
0x44	68	LATIN CAPITAL LETTER D	D	\u0044	0x44
0x45	69	LATIN CAPITAL LETTER E	E	\u0045	0x45
0x46	70	LATIN CAPITAL LETTER F	F	\u0046	0x46
0x47	71	LATIN CAPITAL LETTER G	G	\u0047	0x47
0x48	72	LATIN CAPITAL LETTER H	H	\u0048	0x48
0x49	73	LATIN CAPITAL LETTER I	I	\u0049	0x49
0x4A	74	LATIN CAPITAL LETTER J	J	\u004A	0x4A
0x4B	75	LATIN CAPITAL LETTER K	K	\u004B	0x4B
0x4C	76	LATIN CAPITAL LETTER L	L	\u004C	0x4C
0x4D	77	LATIN CAPITAL LETTER M	M	\u004D	0x4D
0x4E	78	LATIN CAPITAL LETTER N	N	\u004E	0x4E
0x4F	79	LATIN CAPITAL LETTER O	O	\u004F	0x4F
0x50	80	LATIN CAPITAL LETTER P	P	\u0050	0x50
0x51	81	LATIN CAPITAL LETTER Q	Q	\u0051	0x51
0x52	82	LATIN CAPITAL LETTER R	R	\u0052	0x52
0x53	83	LATIN CAPITAL LETTER S	S	\u0053	0x53
0x54	84	LATIN CAPITAL LETTER T	T	\u0054	0x54
0x55	85	LATIN CAPITAL LETTER U	U	\u0055	0x55
0x56	86	LATIN CAPITAL LETTER V	V	\u0056	0x56
0x57	87	LATIN CAPITAL LETTER W	W	\u0057	0x57

0x58	88	LATIN CAPITAL LETTER X	X	\u0058	0x58
0x59	89	LATIN CAPITAL LETTER Y	Y	\u0059	0x59
0x5A	90	LATIN CAPITAL LETTER Z	Z	\u005A	0x5A
0x5B	91	LATIN CAPITAL LETTER A WITH DIAERESIS	Ä	\u00C4	0xC384
0x5C	92	LATIN CAPITAL LETTER O WITH DIAERESIS	Ö	\u00D6	0xC396
0x5D	93	LATIN CAPITAL LETTER N WITH TILDE	Ñ	\u00D1	0xC391
0x5E	94	LATIN CAPITAL LETTER U WITH DIAERESIS	Ü	\u00DC	0xC39C
0x5F	95	SECTION SIGN	§	\u00A7	0xC2A7
0x60	96	INVERTED QUESTION MARK	¿	\u00BF	0xC2BF
0x61	97	LATIN SMALL LETTER A	a	\u0061	0x61
0x62	98	LATIN SMALL LETTER B	b	\u0062	0x62
0x63	99	LATIN SMALL LETTER C	c	\u0063	0x63
0x64	100	LATIN SMALL LETTER D	d	\u0064	0x64
0x65	101	LATIN SMALL LETTER E	e	\u0065	0x65
0x66	102	LATIN SMALL LETTER F	f	\u0066	0x66
0x67	103	LATIN SMALL LETTER G	g	\u0067	0x67
0x68	104	LATIN SMALL LETTER H	h	\u0068	0x68
0x69	105	LATIN SMALL LETTER I	i	\u0069	0x69
0x6A	106	LATIN SMALL LETTER J	j	\u006A	0x6A
0x6B	107	LATIN SMALL LETTER K	k	\u006B	0x6B
0x6C	108	LATIN SMALL LETTER L	l	\u006C	0x6C
0x6D	109	LATIN SMALL LETTER M	m	\u006D	0x6D
0x6E	110	LATIN SMALL LETTER N	n	\u006E	0x6E
0x6F	111	LATIN SMALL LETTER O	o	\u006F	0x6F
0x70	112	LATIN SMALL LETTER P	p	\u0070	0x70
0x71	113	LATIN SMALL LETTER Q	q	\u0071	0x71
0x72	114	LATIN SMALL LETTER R	r	\u0072	0x72
0x73	115	LATIN SMALL LETTER S	s	\u0073	0x73
0x74	116	LATIN SMALL LETTER T	t	\u0074	0x74
0x75	117	LATIN SMALL LETTER U	u	\u0075	0x75
0x76	118	LATIN SMALL LETTER V	v	\u0076	0x76
0x77	119	LATIN SMALL LETTER W	w	\u0077	0x77
0x78	120	LATIN SMALL LETTER X	x	\u0078	0x78
0x79	121	LATIN SMALL LETTER Y	y	\u0079	0x79
0x7A	122	LATIN SMALL LETTER Z	z	\u007A	0x7A
0x7B	123	LATIN SMALL LETTER A WITH DIAERESIS	ä	\u00E4	0xC3A4
0x7C	124	LATIN SMALL LETTER O WITH DIAERESIS	ö	\u00F6	0xC3B6
0x7D	125	LATIN SMALL LETTER N WITH TILDE	ñ	\u00F1	0xC3B1
0x7E	126	LATIN SMALL LETTER U WITH DIAERESIS	ü	\u00FC	0xC3BC
0x7F	127	LATIN SMALL LETTER A WITH GRAVE	à	\u00E0	0xC3A0