ENTSOG AS4 Profile

Draft Version 4.0 –2024-02-07

Disclaimer

This document provides only specific technical information given for indicative purposes and, as such, it can be subject to further modifications. The information contained in the document is non-exhaustive as well as non-contractual in nature and closely connected with the completion of the applicable process foreseen by the relevant provisions of Commission Regulation (EU) 2015/703 of 30 April 2015 establishing a network code on interoperability and data exchange rules.

No warranty is given by ENTSOG in respect of any information so provided, including its further modifications. ENTSOG shall not be liable for any costs, damages and/or other losses that are suffered or incurred by any third party in consequence of any use of -or reliance on- the information hereby provided.

Table of contents

1 Introduction

COMMISSION REGULATION (EU) 2015/703 of 30 April 2015 establishing a network code on interoperability and data exchange rules published on 30 April 2015 by the European Commission (EC) specifies that "*The following common data exchange solutions shall be used [for the communication] protocol: AS4*" [CR2015/703] for document-based exchanges. This document defines an ENTSOG AS4 Profile that aims to support cross-enterprise collaboration in the gas sector using secure and reliable exchange of business documents based on the AS4 standard [AS4], now also standardized internationally as part two of the ISO 15000 series [ISO 15000-2]. This is done by providing an ENTSOG AS4 ebHandler profile and a usage profile for the AS4 communication protocol that allow actors in the gas sector to deploy AS4 communication platforms in a consistent and interoperable way. This document also specifies a mechanism to manage certificate exchanges and updates for AS4 using ebCore Agreement Update [AU].

The main goals of this profile are to:

- Support exchange of EDIG@S XML documents and other payloads [EDIG@S].
- Support business processes of Transmission System Operators for gas, as well as future business processes.
- Leverage previous experience with AS2 as described in the EASEE-gas implementation guide [EGMTP].
- Provide security guidance based on state-of-the-art best practices.
- Provide suppliers of AS4-enabled B2B communication solutions with guidance regarding the required AS4 functionality.
- Align with similar profiles of AS4 developed by other user communities, in particular the eDelivery AS4 Building Block [eDeliveryAS4].
- Facilitate management and exchange of certificates for AS4 by users deploying the profile.

This version 4.0 is the first major update of the ENTSOG AS4 profile since 2016. It retains all the core functionality of the last version 3.6 which was published in 2018. The only changes relate to the message layer security section where some selected algorithms have been replaced by more state-of-the-art secure algorithms. These changes intend to provide continued secure use of ENTSOG AS4 in the coming years. These changes also provide continued alignment of ENTSOG AS4 with the upcoming version of the European Commission's eDelivery AS4 profile.

This profile adopts document conventions common in technical specifications for Internet protocols and data formats. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL", "NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2 AS4 Profile

This specification defines the ENTSOG AS4 profile as the selection of a specific conformance profile of the AS4 standard [AS4], which is profiled further for increased consistency and ease of configuration, and an AS4 Usage Profile that defines how to use a compliant implementation for gas industry document exchange. Section 2.1 describes the AS4 ebHandler Conformance Profile, of which this profile is an extended subset. Section 2.2 describes the feature set that conformant products are REQUIRED to support. Section 2.3 is a usage guide that describes configuration and deployment options for conformant products. Section 2.4 describes how certificates for use with AS4 configurations for this profile can be exchanged and managed using ebCore Agreement Update [AU].

2.1 AS4 and Conformance Profiles

2.1.1 AS4 Standard

This ENTSOG AS4 profile is based on the AS4 Profile of ebMS 3.0 Version 1.0. OASIS Standard [AS4]. AS4 itself is based on other standards, in particular on OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features OASIS Standard [EBMS3], which in turn is based on various Web Services specifications. AS4 is also part 2 of the ISO 15000 series [ISO 15000-2].

The OASIS Technical Committee responsible for maintaining the AS4, ebMS 3.0 Core and other related specifications is tracking and resolving issues in the specifications, which it intends to publish as a consolidated Specification Errata. Implementations of the ENTSOG AS4 Profile SHOULD track and implement resolutions at https://tools.oasis-open.org/issues/browse/EBXMLMSG.

2.1.2 AS4 ebHandler Conformance Profile

The AS4 standard [AS4] defines multiple conformance profiles, which define specific functional subsets of the version 3.0 ebXML Messaging, Core Specification [EBMS3]. A conformance profile corresponds to a class of compliant applications. This version of the ENTSOG AS4 Profile is based on an extended subset of the **AS4 ebHandler Conformance Profile** and a Usage Profile. It aims to support gas business processes such as Capacity Allocation Mechanism and Nomination, in which documents are to be transmitted securely and reliably to Receivers with a minimal delay.

2.2 ENTSOG AS4 ebHandler Feature Set

The ENTSOG AS4 feature set is, with some exceptions, a subset of the feature set of the AS4 ebHandler Conformance Profile. This section selects specific options in situations where the AS4 ebHandler provides more than one option. This section is addressed to providers of AS4 products and can be used as a checklist of features to be provided in AS4 products. The structure of this chapter mirrors the structure of the ebMS3 Core Specification [EBMS3].

Compared to the AS4 ebHandler Conformance Profile, this profile adds, or updates, some functionality:

- There is an added recommendation to support the Two Way Message Exchange Pattern (MEP) (cf. section 2.2.1).
- Transport Layer Security processing, if handled in the AS4 handler, is profiled (cf. section 2.2.6.1).
- Algorithms specified for securing messages at the Message Layer are updated to current guidelines (cf. section 2.2.6.2).

It also relaxes some requirements:

- Support for **Pull** mode in AS4 will only be REQUIRED when business processes determine that **Pull** mode exchanges are necessary (cf. section 2.2.2).
- All payloads are exchanged in separate MIME parts (cf. section 2.2.3.2).
- Asynchronous reporting of receipts and errors is not REQUIRED (cf. sections 2.2.4, 2.2.5).
- WS-Security support is limited to the X.509 Token Profile (cf. section 2.2.6.2).

2.2.1 Messaging Model

This profile constrains the channel bindings of message exchanges between two AS4 Message Service Handlers (MSHs), one of which acts as Sending MSH and the other as the Receiving MSH. The following diagram (from [EBMS3]) shows the various actors and operations in message exchange:

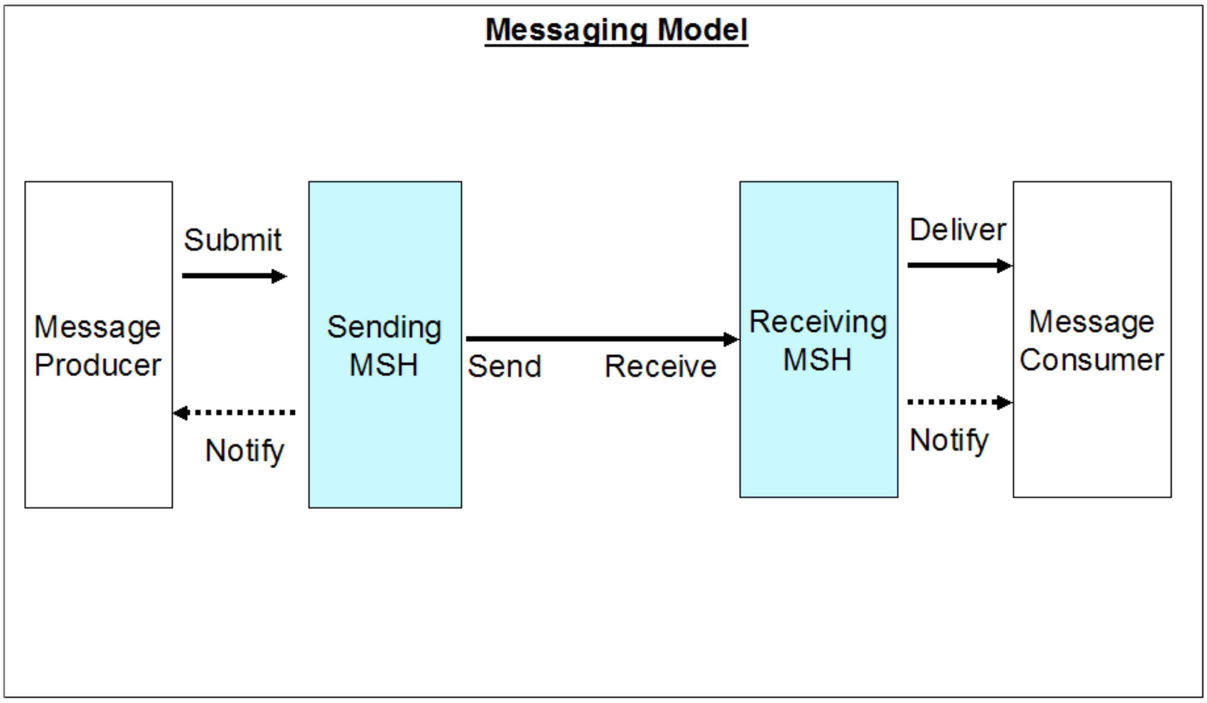

Figure 1 AS4 Messaging Model

Business applications or middleware, acting as *Producer*, *Submit* message content and metadata to the Sending MSH, which packages this content and sends it to the Receiving MSH of the business partner, which in turn *Delivers* the message to another business application that *Consumes* the message content and metadata. Subject to configuration, Sending and Receiving MSH may *Notify Producer* or *Consumer* of particular events. Note that there is a difference between *Sender* and *Initiator*. For **Push** exchanges, the Sending MSH initiates the transmission of the message. For **Pull** exchanges, the transmission is initiated by the Receiving MSH.

The AS4 ebHandler Conformance Profile is the AS4 conformance profile that provides support for Sending and Receiving roles using **Push** channel bindings. Support is REQUIRED for the following Message Exchange Pattern:

- *One Way / Push*

For **PMode.MEP**, support is therefore REQUIRED for the following values:

- *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay*

While the AS4 ebHandler does not require support for the Two-Way MEP, support for this MEP may be added in future versions of this ENTSOG AS4 profile (see section 2.3.1.3). A message handler that supports Two Way MEPs allows the Producer submitting a message unit to set the optional *RefToMessageId* element in the *MessageInfo* section in support of request-response exchanges. For **PMode.MEP**, support is therefore RECOMMENDED for the following value:

- *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/twoWay*

For **PMode.MEPbinding,** support is REQUIRED for:

- *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push*

Note that these values are identifiers only and do not resolve to content on the OASIS site.

2.2.2 Message Pulling and Partitioning

Business processes currently under consideration for this version of this profile are time-critical and considered only supported by the **Push** channel binding, because it allows the *Sender* to control the timing of transmission of the message. Future versions of this profile MAY also support business processes with less time-critical timing requirements. These future uses could benefit from the ebMS3 **Pull** feature. For **PMode.MEPbinding,** applications SHOULD therefore also support:

- *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/pull*

This allows implementations of this profile to also support the following Message Exchange Patterns:

- *One Way / Pull*
- *Two Way / Push-and-Pull*

- *Two Way / Pull-and-Push*
- *Two Way / Pull-and-Pull*

Note that any compliant AS4 ebHandler is REQUIRED to support the first of these options. That requirement is relaxed in this profile. The other three options combine Two Way exchanges (see section 2.2.1) with the **Pull** feature.

2.2.3 Message Packaging

The AS4 message structure (see Figure 2) provides a standard message header that addresses B2B requirements and offers a flexible packaging mechanism based on SOAP and MIME enveloping. Dashed line style is used for optional message components.

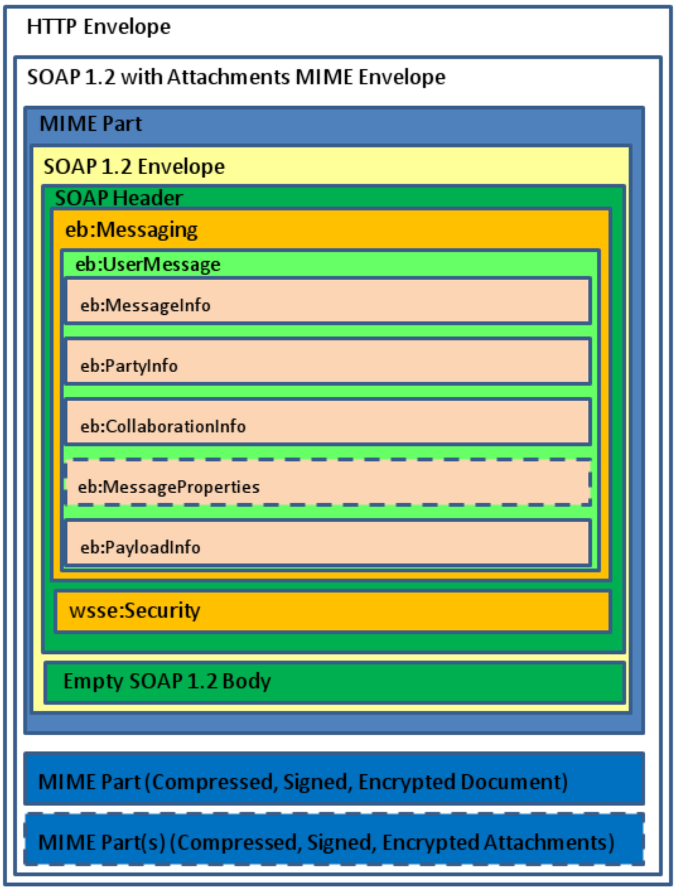

Figure 2 AS4 Message Structure

The SOAP envelope SHOULD be encoded as UTF-8 (see [EBMS3], section 5.1.2.5). If the SOAP envelope is correctly encoded in UTF-8 and the character set header is set to UTF-8, receivers MUST support the presence of the Unicode Byte Order Mark (BOM; see [BP20], section 3.1.2).

2.2.3.1 UserMessage

AS4 defines the ebMS3 **Messaging** SOAP header, which envelopes **UserMessage** XML structures, which provide business metadata to exchanged payloads. In AS4, ebMS3 messages other than receipts or errors carry a single **UserMessage**. The ENTSOG AS4 profile follows the AS4 ebHandler Conformance Profile in requiring full configurability for “General” and “BusinessInfo” P-Mode parameters as per sections 2.1.3.1 and 2.1.3.3 of [AS4].

A compliant product MUST allow the Producer, when submitting messages, to set a value for **AgreementRef**, to select a particular P-Mode. A compliant product, acting as Receiver, MUST take the value of the AS4 **AgreementRef** header into account when selecting the applicable P-Mode. It MUST be able to send and receive messages in which the optional *pmode* attribute of **AgreementRef** is not set.

The ebMS3 and AS4 specifications do not constrain the value of **MessageId** beyond conformance to the Internet Message Format [RFC2822], which requires the value to be unique. Products can do this by including a UUID string in the *id-left* part of the identifier set using randomly (or pseudo-randomly) chosen values.

As in the AS4 ebHandler profile, support for **MessageProperties** is REQUIRED in this profile.

2.2.3.2 Payloads

Section 5.1.1 of the ebMS3 Core Specification [EBMS3] requires implementations to process both non-multipart (simple SOAP) messages and multipart (SOAP-with-attachments) messages, and this is a requirement for the AS4 ebHandler Conformance Profile. Due to the mandatory use of the AS4 compression feature in this profile (see section 2.2.3.3), XML payloads MAY be converted to binary data, which is carried in separate MIME parts and not in the SOAP Body. AS4 messages based on this profile always have an empty SOAP Body.

The ebMS3 mechanism of supporting “external” payloads via hyperlink references (as mentioned in section 5.2.2.12 of [EBMS3]) MUST NOT be used.

2.2.3.3 Message Compression

The AS4 specification defines payload compression as one of its additional features. Payload compression is a useful feature for many content types, including XML content.

- The parameter **PMode[1].PayloadService.CompressionType** MUST be set to the value *application/gzip.* (Note that GZIP is the only compression type currently supported in AS4).

Mandatory use of the AS4 compression feature is consistent with current practices for gas B2B data exchange, such as the EASEE-gas AS2 profile [EGMTP]. Compressed payloads are in separate MIME parts.

2.2.4 Error Handling

This profile specifies that errors MUST be reported and transmitted synchronously to the Sender and SHOULD be reported to the Consumer.

- The parameter **PMode[1].ErrorHandling.Report.AsResponse** MUST be set to the value *true*.
- The parameter **PMode[1].ErrorHandling.Report.ProcessErrorNotifyConsumer** SHOULD be set to the value *true*.

2.2.5 Reliable Messaging and Reception Awareness

This profile specifies that non-repudiation receipts MUST be sent synchronously for each message type.

- The parameter **PMode[1].Security.SendReceipt.NonRepudiation** MUST be set to the value *true*.
- The parameter **PMode[1].Security.SendReceipt.ReplyPattern** MUST be set to the value *Response*.

This profile requires the use of the AS4 Reception Awareness feature. This feature provides a built-in *Retry* mechanism that can help overcome temporary network or other issues and detection of message duplicates.

- The parameter **PMode[1].ReceptionAwareness** MUST be set to *true*.
- The parameter **PMode[1].ReceptionAwareness.Retry** MUST be set to *true*.
- The parameter **PMode[1].ReceptionAwareness.DuplicateDetection** MUST be set to *true*.

The parameters **PMode[1].ReceptionAwareness.Retry.Parameters** and related **PMode[1].ReceptionAwareness.DuplicateDetection.Parameters** are sets of parameters configuring retries and duplicate detection. These parameters are not fully specified in [AS4] and implementation-dependent. Products MUST support configuration of parameters for retries and duplicate detection.

Reception awareness errors generated by the Sender MUST be reported to the Submitting application:

- The parameter **PMode[1].ErrorHandling.Report.MissingReceiptNotifyProducer** MUST be set to *true*.
- The parameter **PMode[1].ErrorHandling.Report.SenderErrorsTo** MUST NOT be set. There is no support for reporting sender errors to a third party.

2.2.6 Security

AS4 message exchanges can be secured at multiple communication layers: the network layer, the transport layer, the message layer and the payload layer. The first and last of these are not normally handled by B2B communication software and therefore out of scope for this section. Transport layer security is addressed, even though its functionality MAY be offloaded to another infrastructure component.

This section provides parameter settings based on multiple published sets of best practices. It is noted that after publication of this document, vulnerabilities may be discovered in the security algorithms, formats and exchange protocols specified in this section. Such discoveries MUST lead to revisions of this specification.

2.2.6.1 Transport Layer Security

2.2.6.1.1 Use of TLS

When using AS4, Transport Layer Security (TLS) provides content confidentiality and authentication. Server authentication, using a server certificate, allows the client to make sure the HTTPS connection is set up with the right server. When a message is pushed, the Sending MSH authenticates the HTTPS server of the Receiving MSH.

TLS can be directly handled by the AS4 message handler or be off-loaded to some infrastructure component. In the following, we refer to the TLS processing component as TLS implementation. For every TLS implementation conformant with this profile, the following rules shall apply:

- TLS versions and cipher suites MUST follow international and national minimum standard requirements and best practices such as [ECRYPT CSA], [NIST 800-52r2], [BSI TR-02102-2] and [RFC9325]. The decision which, if any, of these publications to follow is not specified in this profile as it may depend on other international, national and/or sectorial regulation or other factors.
- It MUST be possible to configure the accepted TLS version(s) in the TLS implementation.
- It MUST be possible to configure accepted TLS cipher suites in the TLS implementation. Note that naming conventions and recommendations for suites are specific to TLS versions.

2.2.6.1.2 TLS Versions

Implementations conformant with this profile:

- MUST NOT use SSL 3.0, TLS 1.0 and 1.1.
- MUST therefore at a minimum support TLS 1.2 [RFC5246]. TLS 1.2 is considered sufficient and offers good cryptographic primitives. With proper configuration of cipher suites it is considered sufficient for many years.
- SHOULD support the use of TLS 1.3 [RFC8446]. Note that [NIST 800-52r2] requires support for TLS 1.3 as from January 1, 2024.

2.2.6.1.3 TLS Cipher Suites

Implementations conformant with this profile SHOULD support the following TLS 1.3 cipher suites:

- TLS_AES_128_GCM_SHA256

- TLS_AES_256_GCM_SHA384
- TLS_AES_128_CCM_SHA256

These cipher suites are recommended by [BSI TR-02102-2] and [NIST 800-52r2]. Note that [ECRYPT CSA] does not make any explicit restrictions regarding TLS 1.3 cipher suites. [RFC9325] recommends to follow the recommendations from [RFC8446].

In addition, TLS_CHACHA20_POLY1305_SHA256 may be used [RFC8446].

For TLS 1.2, this profile recommends the usage of Perfect Forward Secure (PFS) cipher suites. Implementations conformant with this profile SHOULD support the following TLS 1.2 cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CCM
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

These cipher suites are compatible with the recommendations of [BSI TR-02102-2], [NIST 800-52r2], [ECRYPT CSA]and [RFC9325].

Further cipher suites may be used when following specific regulations. For example, [ECRYPT CSA]recommends the usage of Camellia for record layer encryption. [BSI TR-02102-2], [NIST 800-52r2], and [ECRYPT CSA] recommend the usage of TLS_DHE_* cipher suites.

2.2.6.1.4 Supported Groups for (EC)DH Key Exchange

Implementations conformant with this profile SHOULD support the following elliptic curves:

- secp256r1
- secp384r1
- secp521r1
- x25519
- x448

When using Finite Field Diffie Hellman, at least ffdhe3072 should be used.

2.2.6.1.5 Certificate Key Lengths

Implementations conformant with this profile MUST use RSA, ECDSA, or EdDSA X.509 certificates. For RSA certificates, keys larger than 3000 bits are mandatory. For ECDSA, keys larger than 250 bits are REQUIRED.

2.2.6.1.6 TLS Client Authentication

Transport Layer client authentication authenticates the Sender (when used with the Push MEP binding) or Receiver (when used with Pull). Since this profile uses WS-Security for message authentication, the use of client authentication at the Transport Layer can be considered redundant. Whether or not client authentication is to be used depends on the deployment environment. To support deployments that do require client authentication, implementations MUST allow Transport Layer client authentication to be configured for an AS4 HTTPS endpoint. Mutual Authentication or “two way” TLS Authentication is a combination of client and server authentication.

2.2.6.2 Message Layer Security

2.2.6.2.1 Use of WS-Security

To provide message layer protection for AS4 messages, this profile REQUIRES the use of the following Web Services Security version 1.1.1 OASIS specifications, profiled in ebMS3.0 [EBMS3] and AS4 [AS4]:

- Web Services Security SOAP Message Security [WSSSMS].
- Web Services Security X.509 Certificate Token Profile [WSSX509].
- Web Services Security SOAP Message with Attachments (SwA) Profile [WSSSWA].

The X.509 Certificate Token Profile supports the signing and encryption of AS4 messages. This profile REQUIRES the use of X.509 tokens for message signing and encryption, for all AS4 exchanges. The AS4 option of using Username Tokens, which is supported in the AS4 ebHandler Conformance Profile, MUST NOT be used. The AS4 message MUST be signed prior to being encrypted (see section 7.6 of [EBMS3]).

2.2.6.2.2 Message Signing

AS4 message signing is based on the W3C XML Signature recommendation used by WS-Security. AS4 can be configured to use specific digest and signature algorithms based on identifiers defined in this recommendation. At the time of publication of the AS4 specification [AS4], the current version of W3C XML Signature was the June 2008, XML Signature, Second Edition specification [XMLDSIG]. The current version is the April 2013, Version 1.1 specification [XMLDSIG1] defines important new algorithm identifiers. In addition, the Ed25519 algorithm is available based on [RFC8410] and [RFC9231].

This AS4 profile uses the following AS4 parameters and values:

- The **PMode[].Security.X509.Sign** parameter MUST be set in accordance with section 5.1.4 and 5.1.5 of [AS4].
- The **PMode[].Security.X509.Signature.HashFunction** parameter MUST be set to http://www.w3.org/2001/04/xmlenc#sha256.

- The **PMode[].Security.X509.Signature.Algorithm** parameter MUST be set to http://www.w3.org/2021/04/xmldsig-more#eddsa-ed25519.

This AS4 profile anticipates an update to the OASIS AS4 specification to reference this newer version of the XML Signature specification.

The use of XML Signature in AS4 provides Non Repudiation of Origin (NRO) at Message Exchange level.

A sending AS4 MSH performs security processing and constructs the **ds:Signature** header as follows:

1. The message parts that are to be signed (header, empty body and MIME parts) are selected in accordance with AS4.
2. Message digests are computed for all parts following [WSSSWA] using http://www.w3.org/2001/04/xmlenc#sha256. A **ds:SignedInfo** section is created that contains a **ds:Reference** element for each signed message part containing the respective message digest value.
3. The message is signed using sender's signing key, determined from the applicable P-Mode using the http://www.w3.org/2021/04/xmldsig-more#eddsa-ed25519 algorithm.
4. The signature related security headers are placed under a **ds:Signature** element.

The receiving AS4 MSH processes the secured message containing this security header as follows:

1. Once the message parts have been decrypted successfully, the recipient processes the **ds:Reference** elements. It recalculates the digests for the signed parts and validates that their digest values match the specified values.
2. It then validates the signature value by using the public key from the sender certificate.

Note that the usage of the Ed25519 curve implies that the message signer has an EdDSA certificate using the Ed25519 curve to sign AS4 messages. This certificate is signed by a CA that might use a different signing algorithm (RSA or ECDSA). This profile does not prescribe any algorithms for CAs. When issuing certificates, the CA uses its key to sign the certificate data for the party that requests the certificate. The signed data in the certificate includes the public key of the requesting party. Interoperability is not an issue as the type of public key of the requesting party is not relevant for the signing of the certificate as for the CA signature, because that signed public key is just data.

2.2.6.2.3 Message Encryption

For encryption, WS-Security leverages the W3C XML Encryption recommendation used by WS-Security. The following AS4 parameters configure this feature:

- The **PMode[].Security. X509.Encryption.Encrypt** parameter MUST be set in accordance with section 5.1.6 and 5.1.7 of [AS4].
- The parameter **PMode[].Security.X509.Encryption.Algorithm** MUST be set to http://www.w3.org/2009/xmlenc11#aes128-gcm. This is the algorithm used as value for the Algorithm attribute of **xenc:EncryptionMethod** on **xenc:EncryptedData**. This means that in this profile, AES MUST NOT be used in CBC mode.

As specified in section 5.1.6 of [AS4] and in https://issues.oasis-open.org/browse/EBXMLMSG-111, when XML Encryption is used, all and only payload MIME parts MUST be encrypted. The **eb:Messaging header** and any of its sub-elements MUST NOT be encrypted at message layer. Note that this header remains encrypted at transport layer.

In WS-Security, there are three mechanisms to reference a security token (see section 3.2 in [WSSX509]). The ebMS3 and AS4 specifications do not constrain this; neither do they provide a P-Mode parameter to select a specific option. For interoperability, implementations SHOULD therefore implement all three options. It is RECOMMENDED that implementations allow configuration of security token reference type, so that a compatible type can be selected for a communication partner. Note that as BinarySecurityToken is the most widely implemented option for security token references in AS4 implementations, implementations SHOULD implement this option. To allow certificate chain validation, the ValueType attribute SHOULD be set to the X509PKIPathv1 URI.

In this version of this AS4 profile, message encryption is based on the X25519 key agreement algorithm as specified in section 5.6 of [XMLENC1].

- For the key agreement method http://www.w3.org/2021/04/xmldsig-more#x25519 MUST be used. This is the algorithm used as value for the Algorithm attribute of **xenc:AgreementMethod** in **ds:KeyInfo**.
- When using X25519 public keys, the originator key info is included as a **dsig11:DEREncodedKeyValue** element. The ASN.1 content of that element references the OID 1.3.101.110 for X25519.
- To derive the AES 128 data encryption key, the http://www.w3.org/2021/04/xmldsig-more#hkdf algorithm defined in [RFC9231] is used on the agreed shared secret. This identifier is used as a value for the Algorithm attribute of **xenc11:KeyDerivationMethod** in **xenc:AgreementMethod**.

A sending AS4 MSH performs security processing and message encryption as follows:

1. For key agreement related information, an **xenc:AgreementMethod** element is created.
2. The sender generates an ephemeral X25519 key pair. The public key MUST be DER-encoded and placed in a **dsig11:DEREncodedKeyValue** element in the **xenc:OriginatorKeyInfo** sub-element of **xenc:AgreementMethod**.

3. The recipient's static public key information is determined from the applicable P-Mode. It is identified in a ds:KeyValue element placed in the **xenc:RecipientKeyInfo** sub-element of **xenc:AgreementMethod**.
4. A shared secret is constructed from the sender and recipient keys using X25519 key agreement.
5. The sender uses HKDF, http://www.w3.org/2021/04/xmldsig-more#hkdf, to derive an encryption key from the shared secret, a Salt, and an Info value. For hashing it uses the http://www.w3.org/2001/04/xmldsig-more#hmac-sha256 algorithm. The length of the key is 16 bytes. The HKDF parameter information is placed under **xenc:AgreementMethod** in a **dsig-more:HKDFParams** sub-element.
6. A random AES symmetric key is generated and used to encrypt the MIME payload parts using the **http://www.w3.org/2009/xmlenc11#aes128-gcm** algorithm following [WSSSWA].
7. The AES key created in step 6 is wrapped using the derived key created in step 5 using the http://www.w3.org/2001/04/xmlenc#kw-aes128 algorithm.
8. The constructed **xenc:AgreementMethod** element is placed under a **ds:KeyInfo** element under an **xenc:EncryptedKey** element.
9. An **xenc:EncryptedData** element is added for each encrypted part as a child of the **wsse:Security** element.
10. In each of these **xenc:EncryptedData** elements the encrypted key is referenced by using its identifier as the value of the URI attribute of a **wsse:Reference** in a **wsse:SecurityTokenReference** sub-element.
11. An **xenc:ReferenceList** is added under the **xenc:EncryptedKey** element listing the encrypted parts using their identifiers.
12. The **xenc:EncryptedKey** element is in turn placed as a child of the **wsse:Security** element.

Note that this eDelivery AS4 profile anticipates the **dsig-more:HKDFParams** element proposed in [RFC9231bis].

After message encryption, the **xenc:EncryptedKey** element representing the encryption key data and the **xenc:EncryptedData** elements representing the encrypted data are available for processing in the **wsse:Security** header and the MIME part content is encrypted.

The receiving AS4 MSH processes the secured message containing these two encryption related security headers as follows:

1. It identifies the **xenc:ReferenceList** in the **xenc:EncryptedKey** element and the **xenc:EncryptedData** elements to find the parts that are to be decrypted.
2. For each **xenc:EncryptedData** element, using the **wsse:SecurityTokenReference**, it finds the encryption key reference information.

3. In the referenced **xenc:EncryptedKey** element it processes the **xenc:AgreementMethod** element in the **ds:KeyInfo**. Using the **xenc:OriginatorKeyInfo** public key value and the private key identified by **xenc:RecipientKeyInfo**, it performs the ephemeral-static X25519 key agreement to obtain the X25519 shared secret key.
4. Using the shared secret key and the HKDF parameters specified on the **dsig-more:HKDFParams** element, it can unwrap the AES symmetric encryption key needed to decrypt the data.
5. With this key, it uses AES-GCM to decrypt data referenced in **xenc:EncryptedData**.

In the base implementation, ECDH is used in so-called ephemeral-static mode (ECDH-ES) in which the sender creates a shared secret based on a short-lived sender key agreement key in combination with a long-lived recipient key agreement key. The shared secret key is used to wrap a randomly generated key that is used for the symmetric encryption of the payload.

Alternatively, optionally, sender or recipient may use ebCore Certificate Update to update the static key frequently, as explained below in section 2.4 below.

2.2.6.2.4 Sample Security Header

The resulting WS-Security header might look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsse:Security xmlns:env="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:wsse11="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:dsig-more="http://www.w3.org/2021/04/xmldsig-more#"
    xmlns:dsig11="http://www.w3.org/2009/xmldsig11#"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:xenc11="http://www.w3.org/2009/xmlenc11#"
    env:mustUnderstand="true">

    <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
        wsu:Id="EK-6263cc2e-e01a-4bd2-a2f3-39f9c74e82ab">
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
        <ds:KeyInfo>
            <xenc:AgreementMethod Algorithm="http://www.w3.org/2021/04/xmldsig-more#x25519">
                <xenc11:KeyDerivationMethod Algorithm="http://www.w3.org/2021/04/xmldsig-more#hkdf">
                    <dsig-more:HKDFParams>
                        <dsig-more:PRF
                            Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"/>
                        <dsig-more:Salt>xWdTey4T6awUJkp0NPZNVTa2JQkWukC0Uk+qaeEpn4Y=</dsig-
more:Salt>
                        <dsig-more:Info>dGVzdC1pbmZvLWRhdGE=</dsig-more:Info>
                        <dsig-more:KeyLength>16</dsig-more:KeyLength>
                    </dsig-more:HKDFParams>
                </xenc11:KeyDerivationMethod>
                <xenc:OriginatorKeyInfo>
                    <dsig11:DEREncodedKeyValue>
                        MCwwBwYDK2VuBQADIQBf3vfsPjIizIMXS0Z5ombgWtKPLXpTMpV1QQW2ytMLLw==
                    </dsig11:DEREncodedKeyValue>
                </xenc:OriginatorKeyInfo>
                <xenc:RecipientKeyInfo>
                    <ds:KeyValue>
                        <!-- Assumes the recipient key is has been shared as a certificate and can
be
                            referenced using its SKI. -->
                        <wsse:SecurityTokenReference
                            xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd">
```

```
                            <wsse:KeyIdentifier
                                EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-soap-message-security-1.0#Base64Binary"
                                ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
x509-token-profile-1.0#X509SubjectKeyIdentifier"
                                > ENCODED </wsse:KeyIdentifier>
                        </wsse:SecurityTokenReference>
                    </ds:KeyValue>
                </xenc:RecipientKeyInfo>
            </xenc:AgreementMethod>
        </ds:KeyInfo>
        <xenc:CipherData>
            <xenc:CipherValue>1OygswQnDMJi8AUWzoMhIuyyE/GjfHY3</xenc:CipherValue>
        </xenc:CipherData>
        <xenc:ReferenceList>
            <xenc:DataReference URI="#ED-ad394cf3-a2c0-442e-9943-f01cea6782cb"/>
        </xenc:ReferenceList>
    </xenc:EncryptedKey>

    <xenc:EncryptedData
        Id="ED-ad394cf3-a2c0-442e-9943-f01cea6782cb" MimeType="application/gzip"
        Type="http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Content-Only">
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2009/xmlenc11#aes128-gcm"/>
        <ds:KeyInfo >
            <wsse:SecurityTokenReference
                wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-soap-message-security-
1.1#EncryptedKey">
                <wsse:Reference URI="#EK-6263cc2e-e01a-4bd2-a2f3-39f9c74e82ab"/>
            </wsse:SecurityTokenReference>
        </ds:KeyInfo>
        <xenc:CipherData>
            <xenc:CipherReference URI="cid:1400668830234@seller.eu">
                <xenc:Transforms>
                    <ds:Transform xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                        Algorithm="http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-
1.1#Attachment-Ciphertext-Transform"
                    />
                </xenc:Transforms>
            </xenc:CipherReference>
        </xenc:CipherData>
    </xenc:EncryptedData>

    <wsse:BinarySecurityToken
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary"
        ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509v3"
        wsu:Id="X509-48b6d459-777b-4226-81bd-df327f37b30c"
        > ENCODED
    </wsse:BinarySecurityToken>

    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        Id="SIG-adcdc058-ddac-4437-8902-ab37cf037ca4">
        <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
                    PrefixList="env"/>
            </ds:CanonicalizationMethod>
            <ds:SignatureMethod Algorithm="http://www.w3.org/2021/04/xmldsig-more#eddsa-ed25519"/>
            <ds:Reference URI="#_840b593a-a40f-40d8-a8fd-89591478e5df">
                <!-- The (empty) SOAP body -->
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                <ds:DigestValue>jyTXyVrh+cX3iJzgmxqiHdnnJQxcX6kTGHPES1YUYEs=</ds:DigestValue>
            </ds:Reference>
            <ds:Reference URI="#_210bca51-e9b3-4ee1-81e7-226949ab6ff6">
                <!-- the AS4 eb:Messaging header -->
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                <ds:DigestValue>5RMz5/mSIFTI1+amk+XLHsLR2yE7h5KFgAsLrHrya98=</ds:DigestValue>
            </ds:Reference>
            <ds:Reference URI="cid:1400668830234@seller.eu">
                <!-- A message payload in a MIME attachment -->
                <ds:Transforms>
                    <ds:Transform
                        Algorithm="http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-
1.1#Attachment-Content-Signature-Transform"
                    />
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                <ds:DigestValue>wVgT8wKEsJlO0O5OjjQB/vw9mGsxi1n/0dc9qeRqFM4=</ds:DigestValue>
            </ds:Reference>
        </ds:SignedInfo>

<ds:SignatureValue>CyVaSr9BLh7m4KC7xNszOsmJNM6aNJPKwQwNNqY5cvu3GgSIYBQWecg==</ds:SignatureValue>
        <ds:KeyInfo Id="KI-29066baf-2595-444f-9d27-58667dc40da3">
            <wsse:SecurityTokenReference wsu:Id="STR-a54b721a-0d19-4112-b1cf-06752cd826fa">
                <wsse:Reference URI="#X509-48b6d459-777b-4226-81bd-df327f37b30c"
                    ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509v3"
                />
            </wsse:SecurityTokenReference>
        </ds:KeyInfo>
    </ds:Signature>
</wsse:Security>
```

2.2.6.2.5 Elliptic Curve Cryptography Option

In order to provide a fall-back for the (highly unlikely) situation in which vulnerabilities are found in the algorithms for signing (based on Ed25519) or encryption (based on X25519), or for reasons of constraints relating to capabilities of issuing PKI Certification Authorities, AS4 products supporting this profile SHOULD also support an alterative signing and encryption option based on Elliptic Curve Cryptography. This section profiles this option.

2.2.6.2.5.1 Signature using ECDSA

As a variant alternative to the specification in section 2.2.6.2.2, the signature algorithm MAY be set to http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256.

2.2.6.2.5.2 Encryption using ECDH-ES

As a variant alternative to the specification in section 2.2.6.2.3, the ECDH-ES algorithm MAY be used. In this variant:

- The key agreement algorithm used is http://www.w3.org/2009/xmlenc11#ECDH-ES.
- The originator key is encoded as a **dsig11:ECKeyValue** element instead of a **dsig11:DEREncodedKeyValue** element.
- Implementations MUST support at least the secp256r1, secp384r1, secp521r1, BrainpoolP256r1 curves but MAY also support other ECC curves.
- When including public keys based on BrainpoolP256r1 curves, the value of the URI attribute on NamedCurve is to be set to urn:oid:1.3.36.3.3.2.8.1.1.7.

The http://www.w3.org/2009/xmlenc11#ECDH-ES algorithm is also used in [BDEW AS4]. That specification still differs from the ENTSOG profile as follows:

- In [BDEW AS4] the older http://www.w3.org/2009/xmlenc11#ConcatKDF is used whereas this ENTSOG profile uses http://www.w3.org/2021/04/xmldsig-more#hkdf.

The following XML snippet shows an **xenc:AgreementMethod** based on ECDH-ES instead of X25519. The 1.3.36.3.3.2.8.1.1.7 OID indicates that the BrainpoolP256r1 curve is used.

```
<?xml version="1.0" encoding="UTF-8"?>
<xenc:AgreementMethod Algorithm="http://www.w3.org/2009/xmlenc11#ECDH-ES"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:dsig-more="http://www.w3.org/2021/04/xmldsig-more#"
    xmlns:dsig11="http://www.w3.org/2009/xmldsig11#"
    xmlns:xenc11="http://www.w3.org/2009/xmlenc11#"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
    <xenc11:KeyDerivationMethod
        Algorithm="http://www.w3.org/2021/04/xmldsig-more#hkdf"
        xmlns:xenc11="http://www.w3.org/2009/xmlenc11#">
        <dsig-more:HKDFParams
            xmlns:dsig-more="http://www.w3.org/2021/04/xmldsig-more#">
            <dsig-more:PRF
                Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256"/>
            <dsig-more:Salt>DXitIRbhMjQaOT3WXgi8Nj1iNaiy5UPCpdjwXwun8Mk=</dsig-more:Salt>
            <dsig-more:Info>dGVzdC1pbmZvLWRhdGE=</dsig-more:Info>
            <dsig-more:KeyLength>16</dsig-more:KeyLength>
        </dsig-more:HKDFParams>
    </xenc11:KeyDerivationMethod>
    <xenc:OriginatorKeyInfo>
        <ds:KeyValue>
            <dsig11:ECKeyValue xmlns:dsig11="http://www.w3.org/2009/xmldsig11#">
                <dsig11:NamedCurve URI="urn:oid:1.3.36.3.3.2.8.1.1.7"/>
                <dsig11:PublicKey>
                    BAHQXIjLoPO4LBehXFzOveAzouszXfs3aTmkFiwPrsXwTgaV7lBy5B7mPRLYCB7NgPlWD/Yhx1Oq
                    JmSkrU+HjugU6AFPPrUmNARHk7x+JKK+V5v8ErNO1+GSnB25X6N9y08rIHeYaazT5Rc9YpdwEFBG
                    mPOciWlDJCOfRVLJtcRF2X6L0Q==
                </dsig11:PublicKey>
            </dsig11:ECKeyValue>
        </ds:KeyValue>
    </xenc:OriginatorKeyInfo>
    <xenc:RecipientKeyInfo>
        <ds:KeyValue>
            <!-- Assumes the recipient key is has been shared as a certificate and can be
                        referenced using its SKI. -->
            <wsse:SecurityTokenReference>
                <wsse:KeyIdentifier
                    EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary"
                    ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-
profile-1.0#X509SubjectKeyIdentifier"
                    > ENCODED </wsse:KeyIdentifier>
            </wsse:SecurityTokenReference>
        </ds:KeyValue>
    </xenc:RecipientKeyInfo>
</xenc:AgreementMethod>
```

2.2.7 Networking

AS4 communication products compliant with this profile MUST support both IPv4 and IPv6 and MUST be able to connect using either IP4 or IPv6. To support transition from IPv4 to IPv6, products SHOULD support the “happy eyeballs” requirements defined in [RFC8305].

2.2.8 Configuration Management

ENTSOG has identified a requirement for automated or semi-automated exchange and management of AS4 configuration data in order to allow parties to negotiate and automate updates to AS4 configurations using the exchange of AS4 messages. The main initial requirement is the automated exchange of X.509 certificates.

AS4 products compliant with this specification MUST provide an Application Programming Interface (API) to manage (i.e. create, read, update and delete) AS4 configuration data, including Processing Mode definitions and X.509 certificates used for AS4 message exchanges. This API MUST provide all functionality required to create and process ebCore Agreement Update messages (see section 2.4).

2.3 Usage Profile

This section contains implementation guidelines that specify how products that comply with the requirements of the ENTSOG AS4 ebHandler (section 2.2) SHOULD be configured and deployed. This is similar to the concept of Usage Agreements in section 5 of [AS4] as it does not constrain how AS4 products are implemented, but rather how they are configured and used. The audience for this section are operators/administrators of AS4 products and B2B integration project teams. The structure of this chapter also partly mirrors the structure of [EBMS3], and furthermore covers some aspects outside core pure B2B messaging functionality.

2.3.1 Message Packaging

This usage profile constrains values for several elements in the AS4 message header.

2.3.1.1 Party Identification

When exchanging messages in compliance with this profile, parties registered in the ENTSOG Energy Identification Coding Scheme (EIC) for natural gas transmission MUST be identified using the appropriate EIC Code [EIC]. Entities that do not have an EIC code and need to use this profile MUST contact ENTSOG or their Local Issuing Office (LIO) and request an EIC code. This value MUST be used as the content for the **PMode.Initiator.Party** and **PMode.Responder.Party** processing mode parameters, which AS4 message handlers use to populate the **UserMessage/PartyInfo/{From|to}/PartyId** elements.

The *type* attribute on the **PartyId** element MUST be present and set to the fixed value *http://www.entsoe.eu/eic-codes/eic-party-codes-x* which indicates that the value of the element is to be interpreted as an EIC code. This value is a URI used as an identifier only. It is not a URL that resolves to content on the ENTSOE web site.Note that AS4 party identifiers identify the communication partner. The communication partner may be:

1. The entity involved in the business transaction

2. A third party providing B2B communication services for other entities

In the second case, there are two options for setting the P-Mode parameters:

1. The communication partner may *impersonate* the business entity. In this case the AS4 **Party** identifier is the identifier of the business entity.

2. The business entity may explicitly *delegate* message processing to the communication partner. In this case the AS4 **Party** identifier is the identifier of the communication partner. Note that, when used to exchange EDIG@S documents, in this case the AS4 party identifier will differ from the value of the EDIG@S *{issuer/recipient}_MarketParticipant.identification* elements, as the latter refer to the business partner.

Parties MAY use third party communication providers for AS4 communication. Such providers MAY use either the impersonation or delegation model, subject to approval by the business transaction partner.

The AS4 processing layer will validate the identifiers of Sender and Receiver specified in the ebMS3 headers against P-Mode configurations. This involves the validation of message signatures against configured X.509 certificates. In case of delegation, the X.509 certificates used at the AS4 level relate to the communication partners rather than to business partners on whose behalf the messages are exchanged. The exchanged payloads (EDIG@S or other) typically also reference sending and receiving business entities. The responsibility of determining the validity of implied delegation relations between business document layer entities and entities at the AS4 layer is not in scope for the AS4 message handler, but MUST be addressed in business applications or integration middleware.

2.3.1.2 Business Process Alignment

Several mandatory headers in AS4 serve to carry metadata to align a message exchange to a business process or to a technical service.

2.3.1.2.1 Service

The **Service** and **Action** header elements in the **UserMessage/ CollaborationInfo** group relate a message to the business process the message relates to and the roles that sender and receiver perform, or to a technical service. This Usage Profile is intended to be used with business processes that are currently being modelled by ENTSOG and EASEE-gas as well as future, possibly not yet identified, business processes. For current and future gas business processes, ENTSOG maintains and publishes, on its public Web site, a link to a table of **Service** and **Action** values to be used in AS4 messages compliant to this Usage Profile (see section 2.3.1.2.4).

The value of the **Service** element content MUST set as follows:

- For gas business processes covered by EDIG@S, the value content of **Service** is specified in the ENTSOG AS4 Mapping Table (section 2.3.1.2.4) which MUST be used for AS4 messages carrying specified messages. These values are taken from an EDIG@S process area code list. As not all EDIG@S message exchanges concern TSOs, it may be that not all **Service** values that are needed to fully cover the EDIG@S processes are in the table. The example message in section 3.1 uses the value *A06*, which is an EDIG@S code representing Nomination and Matching Processes.

- For the pre-defined test service (see section 2.3.6), the absolute **Service** URI value *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/service* defined in [EBMS3] MUST be used. This value is a URI used as an identifier only. It does not resolve to content on the OASIS web site.
- For ebCore Agreement Update messages used for certificate exchange (see section 2.4), the absolute **Service** URI value *http://docs.oasis-open.org/ebcore/ns/CertificateUpdate/v1.0* defined in [AU], section 4.1, MUST be used. This value is a URI used as an identifier only. It is not a URL that resolves to content on the OASIS web site.
- For other services not related to gas business processes, or not related to gas business processes covered by EDIG@S, no convention is defined in or imposed by this Usage Profile. The ENTSOG list (or future versions of it) MAY specify other non-gas business services.

The value of the *type* attribute of the **Service** element MUST comply with the following:

- For gas business processes covered by EDIG@S, the value MUST be the fixed value *http://edigas.org/service*. This value is a URI used as an identifier only. It does not resolve to a URL on the EDIGAS web sites
- For other services, the use (or non-use) of the *type* attribute on **Service** is not constrained by this Usage Profile.

In situations where the data exchange has not been classified, the service value *http://docs.oasis-open.org/ebxml-msg/as4/200902/service* MAY be used. This is the default P-Mode value for this parameter specified in section 5.2.5 of [AS4]. With this value, the *type* attribute MUST NOT be used. The non-normative example in section 3.1 uses the value “A06” for the **Service** header element, which is an EDIG@S service code. The other non-normative example in section 3.2 uses the AS4 default P-Mode parameter value.

2.3.1.2.2 Action

The **Action** header identifies an operation or activity in a **Service**.

- For gas business processes covered by EDIG@S in which EDIG@S XML documents are exchanged, ENTSOG provides a value table listing actions (section 2.3.1.2.4). The value for **Action** in that table for a particular exchange MUST be used in AS4 messages. The example messages in section 3.1 use the *http://docs.oasis-open.org/ebxml-msg/as4/200902/action* value, which is the default action defined in section 5.2.5 of the AS4 standard [AS4]. As not all EDIG@S message exchanges concern TSOs, it may be that not all **Action** values that are needed to fully cover the EDIG@S business processes are in the service metadata table.
- For the pre-defined test service (see section 2.3.6) the absolute **Action** URI value *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/test* defined in [EBMS3] MUST be used. This value is a URI used as an identifier only. It is not a URL that resolves to content on the OASIS web site.

- For ebCore Agreement Update messages used for certificate exchange, the **Action** values *UpdateCertificate*, *ConfirmCertificateUpdate* and *RejectCertificateUpdate* defined in [AU], section 4.1, MUST be used.
- For other services not related to gas business processes, and for any (hypothetical future) gas business processes not covered by EDIG@S, no convention is defined in or imposed by this Usage Profile.

2.3.1.2.3 Role

The mandatory AS4 headers **UserMessage/PartyInfo/ {From|To}/Role** elements define the role of the entities sending and receiving the AS4 message for the specified **Service** and **Action**.

- For gas business processes covered by EDIG@S, the values MUST be set to values specified in the ENTSOG AS4 Mapping Table (section 2.3.1.2.4). For gas business processes, that table will relate to information in the EDIG@S document content. In EDIG@S, the sender and receiver role are expressed as EDIG@S header elements. For example, in an EDIG@S v5.1 Nomination document, these are called *issuer_Marketparticipant_marketRole.code* of type *IssuerRoleType* and *recipient_Marketparticipant_marketRole.code* of type *PartyType*.
- For the ebMS3 test service and for ebCore Agreement Update, the default initiator and responder roles *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator* and *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder* defined in section 5.2.5 of [AS4] MUST be used. These URI values are used as identifiers only. They are not URLs that resolve to content on the OASIS web site.
- For services not related to gas business processes, or services not covered by EDIG@S, no convention is defined in or imposed by this Usage Profile.

In situations where the data exchange has not been classified, the role values *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator* MAY be used for the initiator role and *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder* for the responder role. These are the default P-Mode values for this parameter specified in section 5.2.5 of [AS4].

The non-normative example in section 3.1 uses the value “ZSH” for the initiating role header element (EDIG@S code for Shipper) and “ZSO” (EDIG@S code for Transmission System Operator) for the responding role header element. The other non-normative example in section 3.2 uses the AS4 default P-Mode parameter values.

2.3.1.2.4 ENTSOG AS4 Mapping Table

ENTSOG maintains and publishes, in a machine-processable format, in collaboration with EASEE-gas, the ENTSOG AS4 Mapping Table containing columns for the following values:

- EDIG@S process category (e.g. *A06 Nomination and Matching*).

- EDIG@S XML document schema (e.g. NOMINT).
- Document type element code for the **type** child element of the EDIG@S document root element (e.g. *ANC*).
- Document type value defined for the document type element code in the EDIG@S XML schema (e.g. *Forwarded single sided nomination*).
- **Service** value to use in an AS4 message carrying the EDIG@S document (configured as the **PMode[1].BusinessInfo.Service** P-Mode parameter). For gas industry exchanges, the values identify the gas business services that TSOs provide to each other and to other communication partners.
- **Action** value to use in an AS4 message carrying the EDIG@S document (configured as the **PMode[1].BusinessInfo.Action** P-Mode parameter). For exchanges that are modelled in a service-oriented approach, the values identify the operations or activities in a service. For exchanges that are not modelled in a service-oriented approach, the default action *http://docs.oasis-open.org/ebxml-msg/as4/200902/action* specified in the AS4 standard [AS4] will be used.
- **From/Role** to use in an AS4 message carrying the EDIG@S document (configured as the AS4 **PMode.Initiator.Role** P-Mode parameter). This value matches the EDIG@S *recipient_Marketparticipant_marketRole.code* (e.g. *ZSH*). Corresponding sender role code value (e.g. *Shipper*)
- **To/Role** to use in an AS4 message carrying the EDIG@S document (configured as the AS4 **PMode.Responder.Role** P-Mode parameter). This value matches the EDIG@S *issuer_Marketparticipant_marketRole.code* (e.g. *ZSO*). Corresponding receiver role code value (e.g. *Transit System Operator*)

Implementations of this profile MUST use the **Service**, **Action**, **From/Role** and **To/Role** values to use specified in this table for the data exchanges covered by the table.

For business services, AS4 **Role** values MUST indicate business roles. If a Service Provider sends or receives messages on behalf of some other organisation (whether in a delegation or impersonation mode), the AS4 role values used relates to the business role of that other organisation. There is no separate role value for Service Providers.

2.3.1.3 Message Correlation

AS4 provides multiple mechanisms to correlate messages within a particular flow.

1. **UserMessage/MessageInfo/RefToMessageId** provides a way to express that a message is a response to a single specific previous message. The **RefToMessageId** element is used in response messages in Two Way message exchanges. Whether two exchanges in a business process are modelled as a Two Way exchange or as two One Way exchanges is a decision made in the Business Requirements Specification for the business process. In this version of this Usage Profile, all exchanges are considered One Way.

2. **UserMessage/CollaborationInfo/ConversationId** provides a more general way to associate a message with an ongoing conversation, without requiring a message to be a response to a single specific previous message, but allowing update messages to existing conversations from both Sender and Receiver of the original message.

In this version of this Usage Profile, the following rules shall apply:

1. **UserMessage/MessageInfo/RefToMessageId** MUST NOT be used. The default exchange is the One Way exchange.
2. **UserMessage/CollaborationInfo/ ConversationId** MUST be included in any AS4 message (as it is a mandatory element) with as content the empty string.

The **RefToMessageId** and **ConversationId** elements may be used in future versions of this Usage Profile, for example to support request-response interactions.

2.3.2 Agreements

The **AgreementRef** element is profiled as follows:

- The element MUST be present in every AS4 message.
- Its value MUST be agreed between each pair of gas industry parties exchanging AS4 messages conforming to this profile.
- In ebMS3, in principle, any value will do as long as, between two parties, the selected identifier is unique and therefore distinguishes messaging using one agreement from messages using another. For consistency, it is RECOMMENDED to use the following URI naming convention:
  http://entsog.eu/communication/agreements/<EIC_CODE_Party_A>/<EIC_CODE_Party_B>/<version>
  where **EIC_CODE_Party_A** is the EIC code of the party that alphabetically precedes **EIC_CODE_Party_B** of the other party, the version number is initially 1 and increments for any update.
- Its value MUST unambiguously identify each party's X.509 signing certificate and X.509 encryption certificate. In other words, if two AS4 messages from P1 to P2 compliant with this Usage Profile have the same value for this element, they are signed using the same mutually known and agreed signing certificate (for P1) and their payloads are encrypted using the same mutually known and agreed encryption certificate (for P2). This is a deployment constraint on P-Mode configurations, in support of the introduction of the ebCore Agreement Update protocol [AU].
- The attributes *pmode* and *type* MUST NOT be set.

Furthermore:

- It is REQUIRED that for every tuple of <**From/PartyId**, **From/Role**, **To/PartyId**, **To/Role**, **Service**, **Action**, **AgreementRef**> values, a unique processing mode is configured. This is another deployment constraint on P-Mode configurations.

- For a tuple of <**From/PartyId**, **From/Role**, **To/PartyId**, **To/Role**, **Service**, **Action**> values, organisations MAY agree to configure multiple processing modes differing on other P-Mode parameters such as certificates used, or the URL of endpoints, for different values of **AgreementRef**. This includes the AS4 test service (see section 2.3.6), meaning two parties can verify that they have consistent and properly configured P-Modes and firewalls for a particular agreement by sending each other AS4 test service messages using the corresponding **AgreementRef**.
- Parties MAY also use different values for **AgreementRef** to target AS4 gateways in different environments (see section 2.3.7), each having a different gateway endpoint URL and possibly certificates.

2.3.3 MPC

The ebMS3 optional attribute *mpc* on UserMessage is mainly used to support the Pull feature, which is not used in the current value of this Usage Profile. Therefore, the use of *mpc* is profiled. The attribute:

- MAY be present in the AS4 UserMessage. If this is the case, it MUST be set to the value *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC*, which identifies the default MPC, and therefore MUST NOT be set to some other value
- MAY be omitted from the AS4 UserMessage. This is equivalent to it being present with the default MPC value

2.3.4 Security

This section describes configuration and deployment considerations in the area of security.

2.3.4.1 Network Layer Security

Commission Regulation 2015/703 states that the Internet shall be used to exchange AS4 messages [CR2015/703]. When using the public Internet, each organisation is individually responsible to implement security measures to protect access to its IT infrastructure.

Organisations use firewalls to restrict incoming or outgoing message flows to specific IP addresses, or address ranges. This prevents unauthorised hosts from connecting to the AS4 communication server. Organisations therefore:

- MUST use static IP addresses (or IP address ranges) for inbound and outbound AS4 HTTPS connections.
- MUST communicate all IP addresses (or IP address ranges) used for outgoing and incoming connections to their trading partners, also covering addresses of any passive nodes in active-passive clusters. Note that the address of the HTTPS endpoint which an AS4 server is to push messages to or pull messages from MAY differ from the address (or addresses) used for outbound connections.

- MUST notify their trading partners about any IP address changes sufficiently in advance to allow firewall and other configuration changes to be applied.

2.3.4.2 Transport Layer Security

The Transport Layer Security settings defined in section 2.2.6.1 MAY be implemented in the AS4 communication server but TLS MAY also be offloaded to a separate infrastructure component (such as a firewall, proxy server or router). In that case, the recommendations on TLS version and cipher suites of 2.2.6.1 MUST be addressed by that component.

The X.509 certificate used by such a separate component MAY follow the requirements of section 2.3.4.4 and 2.3.4.5, but this is NOT REQUIRED.

The TLS cipher suites recommended in section 2.2.6.1 are supported in recent versions of TLS toolkits and which therefore are available for use. Support for these suites is RECOMMENDED. Whether or not less secure cipher suites (which are only recommended for legacy applications) are allowed is a local policy decision.

This profile does NOT REQUIRE the use of client authentication. Client authentication MAY be a requirement in the networking policy of individual organisations that the AS4 deployment needs to meet, but is NOT RECOMMENDED.

2.3.4.3 Message Layer Security

The following parameters control configuration of security at the message layer:

- The **PMode[1].Security.X509.Signature.Certificate** parameter MUST be set to a value matching the requirements specified in section 2.3.4.4.
- The **PMode[1].Security.X509.Encryption.Certificate** parameter MUST be set to a value matching the requirements specified in section 2.3.4.4.
- If a product allows selection of the type of security token reference, it MUST be set to a type supported by the counterparty.

2.3.4.4 Certificates and Public Key Infrastructure

In this Usage Profile, X.509 certificates are used to secure both Transport Layer and Message Layer communication. Requirements on certificates can be sub-divided into three groups:

- General requirements;
- Requirements for Transport Layer Security;
- Requirements for Message Layer Security.

The following general requirements apply to all certificates:

- A maximum three year validity period for leaf certificates is RECOMMENDED.
- A certificate for use in a production environment MUST be issued by a Certification Authority (CA).

- The choice of Certification Authority issuing the certificate is left to implementations but is subject to review by ENTSOG.
- The signature algorithm used by the CA to sign public keys SHOULD be based on EdDSA as used in this profile. RSA or ECDSA signing keys MAY be used. As noted, the type of key used to sign the certificate and the type of the key that is included in the certificate data.
- The issuing CA SHOULD, at a minimum, meet the Normalised Certificate Policy (NCP) requirements specified in [**Error! Reference source not found.**].

The following additional requirements apply for certificates for Transport Layer Security:

- A TLS server certificate SHOULD comply with the certificate profile defined in [EN 319 412-4].
- If a single TLS server certificate is needed to secure host names on different base domains, or to host multiple virtual HTTPS servers using a single IP address, it is RECOMMENDED to use a Multi-Domain (Subject Alternative Name) certificate. Alternatively, wild card certificates MAY be used.
- No additional requirements are placed on TLS client certificates.

The following additional requirements apply for certificates for Message Layer Security:

- Organisations MAY use a certificate issued by EASEE-gas.
- The type of certificate MUST be certificates for organisations, for which proof of identity is required.
- The issued certificate SHOULD comply with the certificate profile defined in [EN 319 412-3].

Section 2.3.4.5 references the EASEE-gas certificate profile. For certificates used for Message Layer Security it follows the EASEE-gas convention of including the party EIC code (see section 2.3.1.1) as recommended value for the Common Name. Alternatively, the EIC code MAY be used as the Subject SerialNumber or as the Subject OrganisationIdentifier.

B2B document exchange typically occurs in a community of known entities, where communication between parties and counterparties is secured using pre-agreed certificates. Such an environment is different from open environments, where certificates establish identities for (possibly previously unknown) entities and Certification Authorities play an essential role to establish trust. Entities MUST proactively notify all communication partners of any updates to certificates used, and in turn MUST process any certificate updates from their communication partners. This concerns both regular renewals of certificates at their expiration dates and replacements for revoked certificates. See section 2.4 for a description of the use of ebCore Agreement Update to exchange certificates.

Organisations MAY also use Certificate Revocation Lists (CRL) or the Online Certificate Status Protocol (OCSP). Individual companies should assess the potential impact on the availability of the AS4 service when using such mechanisms, as their use may cause a certificate to be revoked automatically and messages to be rejected.

2.3.4.5 EASEE-gas Certificate Profile

X.509 certificates used to secure AS4 communication MAY use EASEE-gas certificates that follow the EASEE-gas certificate profile.

2.3.5 Message Payload and Flow Profile

A single AS4 UserMessage MUST reference, via the *PayloadInfo* header, a single structured business document and MAY reference one or more other (structured or unstructured) payload parts. The business document is considered the “leading” payload part for business processing. Any payload parts other than the business document are not to be processed in isolation but only as adjuncts to the business document. Business document, attachments and metadata MUST be submitted and delivered as a logical unit. The format of the business document SHOULD be XML, but other datatypes MAY be supported in specific business processes or contexts.

For each business process, the Business Requirement Specification specifies the XML schema definition (XSD) that the business document is expected to conform to.

- For gas business processes covered by EDIG@S, in which the value content of **Service** is specified in the ENTSOG AS4 Mapping Table, the **Action** is set to the default action and the exchanged business document is an EDIG@S XML document (section 2.3.1.2.4), for the business document part a **Property** SHOULD be included in the **PartProperties** with a name *EDIGASDocumentType* set to the same value as the top-level **type** element in the EDIG@S XML document, which is of type *DocumentType*. The mapping from a combination of **From/PartyId** element, **To/PartyId** and *EDIGASDocumentType* property values to XSDs MUST be agreed and unique, allowing Receivers to validate XML documents using a specific (version of an) XML schema for a particular sender, receiver and document type.
- The part property *EDIGASDocumentType* MUST NOT be used with payloads that are not EDIG@S XML business documents.
- When using the ebMS3 test service (see section 2.3.6), no XML schema constraints apply to any of the included payloads.
- For certificate exchange (see section 2.4), the XML schemas specified in the ebCore Agreement Update [AU] specification for certificate update request, update acceptance and update exception MUST be used with, respectively, the *UpdateCertificate*, *ConfirmCertificateUpdate* and *RejectCertificateUpdate* values for **Action**.
- For other services, in case the **Action** is not set to the AS4 default action, the mapping from **Service** and **Action** value pairs to XSDs MUST be unique, allowing Receivers to validate XML documents using a specific XML schema.

Some gas data exchanges are traditional batch-scheduled exchanges that can involve very large payloads. The trend in the industry towards service-oriented and event-driven exchanges is leading to more, and more frequent, exchanges, with smaller payloads per exchange. It is expected that the vast majority of payloads will be less than 1 MB in size (prior to compression), with rare exceptions up to 10 MB. The number of messages exchanged over a period, their distribution over time and the peak load/average load ratio, are dependent on business process and other factors. Parties MUST take peak message volumes and maximum message size into account when initially deploying AS4. Parties SHOULD also monitor trends in message traffic for existing processes and anticipate any new business processes being deployed (and the expected increases in message and data volumes), and adjust their deployments accordingly in a timely manner.

In practice, there are limitations on the maximum size of payloads that business partners can accept. These limitations may be caused by capabilities of the AS4 message product, or by constraints of the business application, internal middleware, storage or other software or hardware. When designing business processes and document schemas, and when generating content based on those schemas, these requirements SHOULD be taken into account. In particular, business processes in which large amounts of data are exchanged and the business applications supporting these processes SHOULD be designed such that data can be exchanged as a series of related messages, the payload size of each of which does not exceed 10 MB, rather than as a single message carrying a single large payload that could potentially be much larger.

2.3.6 Test Service

Section 5.2.2 of [EBMS3] defines a server test feature that allows an organisation to “Ping” a communication partner. The feature is based on messages with the values of:

- **UserMessage/CollaborationInfo/Service** set to *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/service*
- **UserMessage/CollaborationInfo/Action** set to *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/test*.

This feature MUST be supported so that parties can perform a basic test of the communication configuration (including security at network, transport and message layer, and reliability) in any environment, including the production environment, with any of their communication partners. This functionality MAY be supported as a built-in feature of the AS4 product. If not, a P-Mode MUST be configured with these values. The AS4 product MUST be configured so that messages with these values are not delivered to any business application.

2.3.7 Environments

B2B data exchange solutions are part of the overall IT service lifecycle, in which different environments are operated (typically in parallel) for development, test, pre-production (in some companies referred to as “acceptance environments” or “QA environments”) and production. Development and test are typically internal environments in which trading partners are simulated using stubs. When exchanging messages between organisations (in either pre-production or production environments), they must target the appropriate environment. In order to prevent a configuration error from causing non-production messages to be delivered to production environments or vice versa, organisations SHOULD configure processing modes at message handlers so that messages from one type of environment cannot be accepted inadvertently in a different type of environment.

2.4 ebCore Agreement Update

Based on ENTSOG and other community requirements, an XML schema and exchange protocol for Agreement Updates [AU] was developed in the OASIS ebCore Technical Committee. This specification is currently an OASIS Committee Specification (CS). A Committee Specification is an OASIS Standards Final Deliverable that is stable and suited for implementation. The Agreement Update specification is similar to, but not to be confused with, earlier work in the IETF defining a Certificate Exchange Message for EDIINT [CEM].

2.4.1 Mandatory Support

As from 01.07.2017, implementers of the ENTSOG AS4 Usage Profile MUST be able to support ebCore Agreement Update for Certificate Exchange with their communication partners. Prior to that date, partners MAY use the mechanism, subject to bilateral agreement.

Support for ebCore Agreement Update requirement entails the following:

- AS4 products MUST be able to exchange ebCore Agreement Update AS4 messages. As AS4 is payload-agnostic, this imposes no special requirements on products. The only requirement on implementers deploying AS4 products is that these messages MUST use the **Service** and **Action** values specified in sections 2.3.1.2.1 and 2.3.1.2.2, respectively.

- Mechanisms to create an ebCore AU document; use it to submit an update to an AS4 configuration; convert the success/failure of such an update to a positive/negative ebCore response document; provide an interface to the AS4 MSH for submission and delivery of ebCore documents exchanged with communication partners.

The AS4 configuration management API (see section 2.2.8) MUST provide all functionality to implement ebCore Agreement Update. However, direct integration of any functionality to process ebCore Agreement Update within the AS4 gateway is NOT REQUIRED. The functionality MAY be implemented in some add-on component or in an application that both uses the AS4 gateway for partner communication and is able to manipulate its configuration.

It is NOT REQUIRED to implement a fully automated process to process certificate updates. Organizations MAY implement a process that involves approval or other manual steps to process certificate updates.

Note that Agreement Update is also an EASEE-gas Common Business Practice [EGAU].

2.4.2 Implementation Guidelines

When using Agreement Update for Certificate Update, the following guidelines apply:

- A party MUST obtain the new certificate that it intends to replace an existing certificate with significantly in advance of the expiration date of the certificate to be replaced.
- Once a party has obtained the new certificate, parties MUST determine the communication partners and agreements that are using the old certificate. To each of these partners, and for all agreements, the party SHOULD send a Certificate Update Request as soon as possible.
- The **ActivateBy** value in the update requests MUST be set such that the period in which the request is to be processed is sufficiently long. The definition of “sufficiently long” is partner-dependent, but should take into account that the process on the partner side may be a (partly) manual process. Therefore, time for validation of the request, including validation of the certificate and the issuing Certification Authority; time to create and perform a change request within the partner organization SHOULD be taken into account.
- The specific **ActivateBy** value MUST be set to a date and time acceptable to the receiving organization. This MAY depend on working hours and staff availability, release schedules etc.
- When an updated agreement has been created and agreed, it MUST first be tested using the test service, as described in section 2.3.6 of this document and section 3.5 of [AU]. These tests MUST cover test messages in both directions.
- The **ActivateBy** value SHOULD be set to a date and time sufficiently in advance to the expiration data and time of the old agreement, such that a fall-back to the old agreement, and any necessary troubleshooting, is possible in case any blocking issue occurs during tests.
- If the updated agreement has been tested successfully, the regular message flow that used the old agreement SHOULD be re-deployed to the new agreement. The old agreement SHOULD NOT be used any more for new exchanges.
- The ebCore Agreement also provides an explicit Agreement Termination feature. Use of this feature is NOT REQUIRED, but may be agreed bilaterally.
- Even in case of successful deployment of the new agreement, the old agreement SHOULD NOT be deactivated immediately. This is to allow any in-process messages that use to old agreement to still be processed. For example, a message that was not successfully sent and is being retransmitted due to AS4 reliable messaging may be received at a time when the new agreement has already been deployed. In this case, the configuration for the old agreement SHOULD still be available to successfully receive, acknowledge and deliver the message.

2.4.3 Use for Encryption Key Updates

In addition to supporting updating the certificate used for AS4 message signing, ebCore Certificate Update MAY be used to update the static key of the recipient used in the ephemeral-static key exchange used for AS4 message encryption. In ideal cryptographic protocols, ephemeral keys are only used once for establishing symmetric keys. It is RECOMMENDED to change ephemeral keys as frequently as possible, giving potential attackers less chance to break previous messages. Therefore, it is RECOMMENDED to use ebCore Certificate Update to update keys such that keys are replaced within 7 days. The 7 day limit is the maximum lifetime TLS 1.3 [RFC8446] uses for session tickets which effectively break forward secrecy of TLS connections.

Automatic processing of ebCore Certificate Update messages (i.e. processing of update requests not requiring intervention by a human operator or non-immediate service management process) allows low-overhead, frequent updates of the static key contained in the certificate for the recipient for key exchange. The static key in practice approximates an ephemeral key.

While ebCore Certificate Update packages keys using certificates, the certificates containing ECDH public keys do not need to be signed by a certification authority. As they are issued using signed ebCore Agreement Update messages, their authenticity is established.

3 Examples

3.1 Message with EDIG@S Payload

The following non-normative example is included to illustrate the structure of an AS4 message conforming to this profile, for a hypothetical http://docs.oasis-open.org/ebxml-msg/as4/200902/action action invoked by a hypothetical shipper 21X-EU-A-X0A0Y-Z on a hypothetical service *A06* exposed by a hypothetical transmission system operator 21X-EU-B-P0Q0R-S. The detailed contents of the *wsse:Security* header is omitted.

```
POST /as4handler HTTP/1.1
Host: receiver.example.com:8893
User-Agent: Turia
Content-Type: multipart/related; start="<f8df1904-a6b9-422b-8239-6a971838503f@sender.example.com>";
boundary= "c5bae1842d1e"; type="application/soap+xml"
Content-Length: 472639

--c5bae1842d1e
Content-Id: <f8df1904-a6b9-422b-8239-6a971838503f@sender.example.com>
Content-Type: application/soap+xml; charset="UTF-8"

<S12:Envelope xmlns:S12="http://www.w3.org/2003/05/soap-envelope"
 xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
 xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
 xmlns:eb3="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/">
  <S12:Header>
    <eb3:Messaging wsu:Id="_18f85fc2-a956-431e-a80e-09a10364871b">
      <eb3:UserMessage>
         <eb3:MessageInfo>
            <eb3:Timestamp>2016-04-03T14:49:28.886Z</eb3:Timestamp>
            <eb3:MessageId>2016-921@5209999001264@example.com</eb3:MessageId>
         </eb3:MessageInfo>
         <eb3:PartyInfo>
            <eb3:From>
               <eb3:PartyId
```

```
                   type="http://www.entsoe.eu/eic-codes/eic-party-codes-x">21X-EU-A-X0A0Y-Z</eb3:PartyId>
               <eb3:Role>ZSH</eb3:Role>
            </eb3:From>
            <eb3:To>
               <eb3:PartyId
                   type="http://www.entsoe.eu/eic-codes/eic-party-codes-x">21X-EU-B-P0Q0R-S</eb3:PartyId>
               <eb3:Role>ZSO</eb3:Role>
            </eb3:To>
         </eb3:PartyInfo>
         <eb3:CollaborationInfo>
               <eb3:AgreementRef
             >http://entsog.eu/communication/agreements/21X-EU-A-X0A0Y-Z/21X-EU-B-P0Q0R-S/3</eb3:AgreementRef>
            <eb3:Service type="http://edigas.org/service">A06</eb3:Service>
            <eb3:Action> http://docs.oasis-open.org/ebxml-msg/as4/200902/action</eb3:Action>
            <eb3:ConversationId></eb3:ConversationId>
         </eb3:CollaborationInfo>
         <eb3:PayloadInfo>
          <eb3:PartInfo href="cid:0b960692-a3c6-4e85-80da-36009d3ae043@sender.example.com">
            <eb3:PartProperties>
              <eb3:Property name="MimeType">application/xml</eb3:Property>
              <eb3:Property name="CharacterSet">utf-8</eb3:Property>
              <eb3:Property name="CompressionType">application/gzip</eb3:Property>
              <eb3:Property name="EDIGASDocumentType">01G</eb3:Property>
            </eb3:PartProperties>
          </eb3:PartInfo>
        </eb3:PayloadInfo>
      </eb3:UserMessage>
    </eb3:Messaging>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd"
      xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
      <!-- details omitted -->
    </wsse:Security>
  </S12:Header>
  <S12:Body wsu:Id="_b656ef2c-516"/>
</S12:Envelope>

--c5bae1842d1e
Content-Id: <0b960692-a3c6-4e85-80da-36009d3ae043@sender.example.com>
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

BINARY CIPHER DATA

--c5bae1842d1e—
```

3.2 *Alternative Using Defaults*

The following example fragment is a variant of the sample message shown in section 3.1. for a data exchange that has not been classified using EDIG@S code values for **Service** and **Role**. Instead of an EDIG@S service code, it uses the default service value, as described in section 2.3.1.2.1. Instead of EDIG@S role codes, it uses the default initiator and responder roles, as described in section 2.3.1.2.3.

```
…
 <eb3:PartyInfo>
   <eb3:From>
       <eb3:PartyId
           type="http://www.entsoe.eu/eic-codes/eic-party-codes-x">21X-EU-A-X0A0Y-Z</eb3:PartyId>
      <eb3:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</eb3:Role>
   </eb3:From>
   <eb3:To>
      <eb3:PartyId
          type="http://www.entsoe.eu/eic-codes/eic-party-codes-x">21X-EU-B-P0Q0R-S</eb3:PartyId>
      <eb3:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</eb3:Role>
   </eb3:To>
 </eb3:PartyInfo>
 <eb3:CollaborationInfo>
```

```
    <eb3:AgreementRef
      >http://entsog.eu/communication/agreements/21X-EU-A-X0A0Y-Z/21X-EU-B-P0Q0R-S/3</eb3:AgreementRef>
    <eb3:Service>http://docs.oasis-open.org/ebxml-msg/as4/200902/service</eb3:Service>
    <eb3:Action>http://docs.oasis-open.org/ebxml-msg/as4/200902/action</eb3:Action>
    <eb3:ConversationId></eb3:ConversationId>
 </eb3:CollaborationInfo>
…
```

4 Processing Modes

P-Mode Parameter	Profile Value
PMode.ID	Not used
PMode.Agreement	http://entsog.eu/communication/agreements/<EIC_CODE_Party_A>/<EIC_CODE_Party_B>/<version> @pmode and @type attributes not used.
PMode.MEP	http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/twoWay
PMode.MEPBinding	http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/pushAndPush
PMode.Initiator.Party	Value is an EIC code. The @type attribute is required with fixed value http://www.entsoe.eu/eic-codes/eic-party-codes-x
PMode.Initiator.Role	Set in accordance with ENTSOG AS4 Mapping Table or to AS4 default for test and AU.
PMode.Initiator.Authorisation. username	Not used
PMode.Initiator.Authorisation. password	Not used
PMode.Responder.Party	Value is an EIC code. @type attribute required with value http://www.entsoe.eu/eic-codes/eic-party-codes-x
PMode.Responder.Role	Set in accordance with ENTSOG AS4 Mapping Table for business services.
PMode.Responder.Authorisation.	Not used

P-Mode Parameter	Profile Value
username	
PMode.Responder.Authorisation. password	Not used
PMode[1].Protocol.Address	Required, HTTPS URL of the receiver.
PMode[1].Protocol.SOAPVersion	1.2
PMode[1].BusinessInfo.Service	Set in accordance with ENTSOG AS4 Mapping Table, for business services. Default service for test; ebCore AU service for certificate update.
PMode[1].BusinessInfo.Action	Default values from AS4, *http://docs.oasis-open.org/ebxml-msg/as4/200902/action*, for business services. Test action for test. The ebCore AU values for AU.
PMode[1].BusinessInfo. Properties	Optional
PMode[1].BusinessInfo.MPC	Either not used or (equivalently) set to the ebMS3 default MPC.
PMode[1].ErrorHandling.Report. SenderErrorsTo	Not used
PMode[1].ErrorHandling.Report. ReceiverErrorsTo	Not used
PMode[1].ErrorHandling.Report. AsResponse	True
PMode[1].ErrorHandling.Report. ProcessErrorNotifyConsumer	True (Recommended)
PMode[1].ErrorHandling. DeliveryFailuresNotifyProducter	True (Recommended)
PMode[1].Reliability	Not used
PMode[1].Security.WSSVersion	1.1.1
PMode[1].Security.X509.Sign	True

P-Mode Parameter	Profile Value
PMode[1].Security. X509. Signature.Certificate	Signing Certificate of the Sender
PMode[1].Security. X509. Signature.HashFunction	http://www.w3.org/2001/04/xmlenc#sha256
PMode[1].Security.X509. Signature.Algorithm	http://www.w3.org/2021/04/xmldsig-more#eddsa-ed25519
PMode[1].Security.X509. Encryption.Encrypt	True
PMode[1].Security.X509. Encryption.Certificate	Encryption Certificate of the Receiver
PMode[1].Security.X509. Encryption.Algorithm	Key agreement: http://www.w3.org/2021/04/xmldsig-more#x25519 Key wrapping: http://www.w3.org/2001/04/xmlenc#kw-aes128 Key derivation: http://www.w3.org/2021/04/xmldsig-more#hkdf Content encryption: http://www.w3.org/2009/xmlenc11#aes128-gcm
PMode[1].Security.X509. Encryption.MinimalStrength	128
PMode[1].Security. UsernameToken. username	Not used
PMode[1].Security. UsernameToken. password	Not used
PMode[1].Security. UsernameToken.Digest	Not used
PMode[1].Security. UsernameToken.Nonce	Not used
PMode[1].Security. UsernameToken.Created	Not used

P-Mode Parameter	Profile Value
PMode[1].Security. PModeAuthorise	False
PMode[1].Security.SendReceipt	True
PMode[1].Security.SendReceipt. NonRepudiation	True
PMode[1].Security.SendReceipt. ReplyPattern	Response
PMode[1].PayloadService. CompressionType	application/gzip
PMode[1].ReceptionAwareness	True
PMode[1].ReceptionAwareness. Retry	True
PMode[1].ReceptionAwareness. Retry.Parameters	Not profiled
PMode[1].ReceptionAwareness. DuplicateDetection	True
PMode[1].ReceptionAwareness. DetectDuplicates.Parameters	Not profiled
PMode[1].BusinessInfo. subMPCext	Not used

1356

1357
5 Revision History

Revision	Date	Editor	Changes Made
v0r1	2013-10-29	PvdE	First Draft for discussion
V0r2	2013-11-18	PvdE	• Textual updates from discussions at F2F 2013-11-04. • Improved separation of the AS4 feature set (chapter 2.2) and the usage profile (2.3). For the feature set the audience are vendors and for the usage profile users/implementers. • Provided guidance for TLS based on ENISA and other guidelines (section 2.2.6.1). • Provided guidance on WS-Security based on ENISA guidelines, advice from XML Security experts (section 2.2.6.2). • Added test service (section 2.3.6). • Added support for CL3055 (section 2.3.1.1). • Guidance on correlation is now mentioned as an option only, leaving choice between document-oriented and service-oriented exchanges (section 2.3.1.3). • More guidance on certificates (section 2.3.4.4). • Added a section on environments (section 2.3.7). • Added an example message (section 3.1). • Values to be confirmed: five minutes for retries (section 2.2.5), 10 MB total payload size (section 2.3.5)
V0r3	2013-11-29	PvdE	• Textual updates from F2F on 2013-11-21. • Added messaging model diagram (section 2.2.1). • Add note that Pull is not required to summary (section 2.2) • Added a diagram of AS4 message structure (section 2.2.3). • All payloads are carried in separate MIME parts;

			no support for external payloads; renamed from “attachments” to “payloads” (section 2.2.3.2). • The reference to TLS cipher suites is more general (section 2.2.6.1). • Simplified party identifiers, only EIC codes are allowed (section 2.3.1.1). • ENTSOG will publish Service/Action info (section 2.3.1.2). • Guidance on correlation is left to business processes (section 2.3.1.3). • Client authentication not recommended (section 2.3.4.2). • No preferred CA; state the 3072 is for future applications (section 2.3.4.4). • The test service is now in the Usage Profile as it can be provided via configuration (section 2.3.6). • The section on separating environments is simplified (section 2.3.7). • The usage profile on reliable messaging is removed. • Fixed reference to BSI TLS document (section 6).
V0r4	2013-12-04		• Updates based on discussions at F2F, 2013-12-03 • Disclaimer added. • In 2.2.1, explained Sender-Receiver concepts are orthogonal to Initiator-Responder. • Updated guidance on payload size. • Added RFC 6176 reference. • Improved wording on environments. • Anonymous EIC codes in example.
V0r5	2013-12-06	PvdE	• Draft finalized in team teleconference.
V0r6	2014-02-14	PvdE, EJvN	• Updates based on team teleconference • Generalized title of 2.3.4.4 and updated content to reflect the new appendix on certificate

			requirements. - Added discussion on key transport algorithms. - Updated AES encryption from to *http://www.w3.org/2001/04/xmlenc#aes128-cbc* to http://www.w3.org/2001/04/xmlenc#aes128-gcm following [XMLENC1].
V0r7	2014-04-22	PvdE	ENISA comments: - In 2.3.4.1, change use of firewalls from MAY to SHOULD. - New section 2.2.7 which recommends IPv6.
V0r8	2014-07-28	PvdE	- The AES-GCM encryption URI is identified using *http://www.w3.org/2009/xmlenc11#aes128-gcm*. - Moved the certificate profile into the Usage Profile section. - Minor editorial changes.
V0r9	2014-07-30	PvdE	- Fixed header dates. Accepted all changes to fix Microsoft Word change track formatting errors.
V1r0	2014-09-22	JDK	- Remove “draft” and “not for implementation”. Add reference to PoC in introduction.
V1r1	2015-03-05	PvdE	- New draft V1r1 incorporating first updates for 2015: o Updates on Role, Service, Action based on meeting of 2015-02-17 (section 2.3.1.2). o Message identifiers to be universally unique (2.2.3.1). - Updated the example in section 3.1 accordingly. - New profiling for **AgreementRef**, in support of certificate rollover (section 2.2.3.1 and 2.3.2). - No need to be able to set MessageId, RefToMessageId and ConversationId as we’re not using them (section 2.2.3.1).

V1r2	2015-03-09	JM, PvdE	• Service and Action in example are changed to their coded values. • Corrected the current EDIG@S version to 5.1. • Various spelling corrections. • Profiling for MPC (another feature that is not used currently). • Added missing AgreementRef in message example. • Changed year in timestamps in example to 2016. • In section 2.2.1, the requirement to support Two Way MEPs no longer makes sense as it is inconsistent with the profiling of 2.3.1.3, which says that *RefToMessageId is not used.* Added a note that it may be added in the future.
V1r3	2015-03-18	PvdE	• Accepted all changes up to and including v1r2 for ease of review. • Added more clarification on Communication vs Business partners. • Changed language on mapping table to not preclude that a future version of the table may be maintained somewhere else/by someone else. • Removed the BRS reference from the mapping table column list. • Added some comments on the relation (degree of overlap) between EDIG@S process categories and ENTSOG Service/Action values. • Added some text for a change (to be confirmed) from using EDIG@S process category names instead of category numbers, and from using Document Type names instead of Document Type code, and of Role names instead of Role codes. These are marked as comments and to be processed before finalizing the document.
V1r4	2015-03-24	PvdE	• In Service example, add a prefix http://entsog.eu/services/EDIG@S/ to indicate

			that a Service is based on an EDIG@S service category.
V1r5	2015-04-02	PvdE	• Accepted all changes up to v1r4 for readability. Updates based on conference call of 2015-04-01 • In section 2.3.5, introduced the *EDIGASDocumentType* property and added further profiling of the PartInfo element. • Renamed the Service Metadata Mapping Table to ENTSOG AS4 Mapping Table. • Introduced the AS4 default action. • Changed the example in section 3.1 to use agreed values. • Clarified that roles are business roles in 2.3.1.2.4. • In 2.3.5, allowed XSDs to be agreed not just per Service/Action, but also for a partner.
V1r6	17/04/15	JM	• Accepted some formatting changes and corrected some small editorial errors.
V1r7	20/04/15	JM	• Accepted all changes
V1r8	19/05/15	PvdE	• New section 2.2.8 on configuration management.
V1r9	26/5/15	PvdE	• Update on certificate requirements
V1r10	2/6/15	PvdE	• The part property "*EDIGASDocumentType*" was replaced by an incorrect value in the message example in section 3.1.
V1r11	09/06/15	JM	• Updated Service Field in message example with EDIG@S Code
V1r12	15/06/15	PvDE/JM	• Improved discussion of ENTSOG AS4 Mapping Table • Editorial clean up • Updated reference to Network Code to the Commission Regulation 2015/703. • Removed a reference to an unpublished

			overview of certificate standards and requirements. - Updated Agreement Update reference to ebCore Working Draft.
V2r0	17/06/15	JM	- Revised to Version number to 2 for publication
V2r1	05/01/16	JM	- Added in confirmation of algorithm requirements
V2r2	09/06/16	PvdE	- Type attribute on PartyId in section 2.3.1.1 added. - Type attribute on Service in section 2.3.1.2.1 added. - In section 2.3.2, provided a URI-based naming conventions for agreements. - In section 2.3.5, the schema is fixed for sender and document type for each receiver. - In section 2.3.5, added that EDIG@S XML documents are encoded in UTF-8. - Updated example in section 3.1. - New section 4, PMode table. - Updated reference to ebCore AU to current version.
V2r3	30/06/16	PvdE	- Removed statement on UTF-8 encoding of EDIG@S - Added UTF-8 and BOM clarification to SOAP envelope encoding. - In the example in section 3.1, added a missing closing tag `</eb3:Property>` and made ConversationId an empty element as per section 2.3.1.3. - Added BP20 reference to bibliography. - Removed an obsolete duplicate comment on type attribute on PartyId. - Added discussion of security token

			references and indicated a preference for BST in 2.2.6.2. - In 2.3.4.3, indicated that parties must select a compatible option for security token references.
V2r4	19/07/16	ICT KG	- Reviewed at ITC KG meeting
V2r5	22/08/16	JM	- Updated Legal Disclaimer
V2r6	4/10/16	PvdE	- Updated status of ebCore Agreement Update, due its approval as Committee Specification in the OASIS ebCore TC - Updated Configuration Management API discussion in section 2.2.8 - New section 2.4 on Agreement Update. - Updated discussion of **Service** and **Action** also for ebCore messages. - Fixed a typo in section 3.1, message ID was not RFC 2822 compliant. - Many editorial changes, a.o. redundant white space.
V2.7	18/10/16		- Accepted all changes - In 2.2.3.2, changed to reflect that compression is not guaranteed to take place when the compression P-Mode is set. - In 2.2.6.1 changed “support TLS 1.2” to “at least support TLS 1.2”. - In 2.3.1.2.4, added “For business services,”. - In 2.3.1.3, rephrased as “as content the empty string”. - Fixed the wording in the first bullet in 2.3.5. - In section, improved definition of PMode[1].BusinessInfo.Service, Action and Role to include test and AU.
V2.8	24/10/16	JM	- Reviewed and corrected grammatical errors

			• Created Rev 3 for publication following ITC KG & INT WG approval
V2.9	2/11/16	PvdE	• Minor editorial • In section 2.2.3.1, add requirement that a Receiving MSH MUST use AgreementRef to select the P-Mode to use for a message: *"A compliant product, acting as Receiver, MUST take the value of the AS4* **AgreementRef** *header into account when selecting the applicable P-Mode."* This is needed so that the right certificates are selected. • In section 2.3.1.2.4, added the underlined eight words to the sentence *"Implementations of this profile MUST use the Service, Action, From/Role and To/Role values to use specified in this table for the data exchanges covered by the table"* to explain that for other exchanges, the profile does not apply. This is intended to help users that also want to use AS4 for other exchanges. • In section 2.3.4.5, removed "Class 2" terminology for requirements, as the term creates confusion. Some CAs have different categories and/or constraints. The reference to NCP is now the only constraint. • Renamed title of a section to include TLS as well. • In CA section, clarified that many CAs do not support the use of EIC codes as CN in certificates, and that therefore this is not mandatory. • In section certificate section, KeyAgreement requirement dropped. • In the References section, upgraded to references to the ENISA report from the 2013 to the (most recent) 2014 version.

V3.0	PvdE		• Added back in the 2013 ENISA reference as requested by ITC KG • Approved as v3.0 by ITC KG
V3r1	PvdE		• Updated the references of ETSI ESI European Norms to the current versions. • Some re-structuring of requirements on certificates, making it clear the review process applies to all certificates and CAs. • Harmonized “CA” as abbreviation for Certific**ation** Authority. • Mention that EV certificates may be used. • Mentioned options for EIC code in certificate.
V3r2	PvdE	2016-12-23	• Incorporated improvements in the sections on Certificates, TLS and IP networking from the Interactive and Integrated profiles, to create a common base and consistency with the other documents. • New minor section “Networking” in Usage Profile to cover IPv4/IPv6. • Removed reference to private networks, as the network code states that the Internet is to be used and for consistency with other profiles.
V3.3	PvdE	2017-02-13	• Specified the use of the AS4 P-Mode values for *Service* and *Role* for situations where the data exchange is not classified. (For *Action*, the default value was already specified).
V3.4	PvdE	2017-02-24	• Added an example of unclassified exchanges using default Service and Role values in section 3.2. The other example is now in the subsection 3.1.
V3.5	PvdE	2017-02-24	• In section 2.3.5, changed the requirement on presence of the **EDIGASDocumentType** part property from MUST to SHOULD.

V3.6	PvdE	2018-03-27	After feedback from implementators, ITC kernel group reviewed all "recommendations" (e.g. SHOULD instead of MUST) and checked whether they could be tightened. This version incorporates the decisions of the ITC KG. • Section 2.2.3.1, UUID in MessageId. • Section 2.2.6.2, BinarySecurityToken. • Section 2.2.6.2, Key Transport Algorithms. • Section 2.3.1.1, checking delegation relations. • Section 2.3.4.1, use of firewalls.
V4.0 internal draft	PvdE	2023-03-06	DRAFT UPDATE Major revision on security algorithm and parameters. • Added references to eDelivery in sections 1 and 6. • Added reference to ISO 15000 in 1 and 2. • 2.2.6 is completely revised for both TLS and message layer security. • Simplied the certificate profile in 2.3.4.5. The previous text was out-of-date and did not add much value compared to the referenced sources. • Removed the section on networking in the usage profile that discussed IPv4 / IPv6 transition. This profile requires AS4 products to support both as stated in 2.2.7 so no additional usage profiling is required. • Updated section 6 (references), additional and updated.
V4.0 internal draft	PvdE	2023-04-10	DRAFT UPDATE continued • Updated references for ETSI standards referenced in certificate section to their current versions.

			• Made EDIG@S reference version-neutral. • Removed obsolete references to the CA Browser forum. • Fixed URLs for some EASEE-gas links. • Updated several IETF references. • Added reference to EASEE-gas CBP on Agreement Update.
V4.0 internal draft	PvdE	2023-06-11	DRAFT UPDATE continued • Processed comments from TSWG
V4.0 internal draft	PvdE	2023-09-18	DRAFT UPDATE continued • Improved description of encryption with ECDH aligned with eDelivery • Minor editorial
V4.0 internal draft	PvdE	2024-02-07	DRAFT UPDATE continued • Improved the sections on WS-Security in particular the one on encryption based on discussion and review of all content with the EC eDelivery team. • HKDF instead of ConcatKDF aligned with the upcoming [rfc9231bis]. • Added a section 2.2.6.2.5 with alternative algorithms based on ECC, as fallback. • Added some text on the rational for 4.0 in the introduction section.

6 References

[AES] Advanced Encryption Standard. FIPS 197. NIST, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[AS4] AS4 Profile of ebMS 3.0 Version 1.0. OASIS Standard, 23 January 2013. http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/profiles/AS4-profile/v1.0/

[AU] ebCore Agreement Update Specification Version 1.0. OASIS Committee Specification. 19 September 2016. http://docs.oasis-open.org/ebcore/ebcore-au/v1.0/

[BP20] Basic Profile Version 2.0. OASIS Committee Specification. http://docs.oasis-open.org/ws-brsp/BasicProfile/v2.0/BasicProfile-v2.0.pdf

[BDEW AS4] BDEW AS4-Profile. https://www.bundesnetzagentur.de/DE/Beschlusskammern/1_GZ/BK6-GZ/2021/BK6-21-282/Mitteilung02/AS4%20Profil.pdf?__blob=publicationFile&v=1.

[BSI TR-02102-1] Cryptographic Mechanisms: Recommendations and Key Lengths. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html. Version: 2023-1.

[BSI TR-02102-2] Cryptographic Mechanisms: Recommendations and Key Lengths: Use of Transport Layer Security (TLS) Version: 2023-1. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-2.html

[CEM] Certificate Exchange Messaging for EDIINT. Expired Internet-Draft. https://tools.ietf.org/html/draft-meadors-certificate-exchange-14.

[CR2015/703] COMMISSION REGULATION (EU) 2015/703 of 30 April 2015 establishing a network code on interoperability and data exchange rules. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2015.113.01.0013.01.ENG

[EBMS3] OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features. OASIS Standard. 1 October 2007. http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/

[ECRYPT CSA] H2020-ICT-2014 – Project 645421. Algorithms, Key Size and Protocols Report (2018). https://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf.

[eDeliveryAS4] European Commission. eDelivery AS4. https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/eDelivery+AS4.

[EDIG@S] EASEE-gas EDIG@S. https://www.edigas.org/.

[EGAU] Agreement Update and Certificate Exchange. EASEE-gas Common Business Praction 2019-001/01. https://easee-gas.eu/download_file/DownloadFile/33/cbp-2019-001-01-agreement-update-and-certificate-exchange.

[EGCDN] Common Data Network. EASEE-gas Common Business Practice 2007-002/01. https://easee-gas.eu/download_file/DownloadFile/13/cbp-2007-002-01-common-data-communications-network

[EGMTP] Message Transmission Protocol. EASEE-gas Common Business Practice 2007-001/01. https://easee-gas.eu/download_file/DownloadFile/24/cbp-2007-001-02-on-message-transmission-protocol

[EIC] ENTSOG. Energy Identification Coding Scheme (EIC) for natural gas transmission. Party Codes. https://www.entsog.eu/energy-identification-codes-eic

[ETSI EN 319 411-1)] European Standard. Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements. V1.3.1 (2021-05). https://www.etsi.org/deliver/etsi_en/319400_319499/31941101/01.03.01_60/en_31941101v010301p.pdf

[EN 319 412-3] Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 3: Certificate profile for certificates issued to legal persons. V1.2.1. (2020-07). https://www.etsi.org/deliver/etsi_en/319400_319499/31941203/01.02.01_60/en_31941203v010201p.pdf.

[EN 319 412-4] Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 4: Certificate profile for web site certificates. v1.2.1. 2021-11 http://www.etsi.org/deliver/etsi_en/319400_319499/31941204/01.01.01_60/en_31941204v010101p.pdf

[ISO 15000-1] ISO 15000-1:2021. Electronic business eXtensible Markup Language (ebXML) — Part 1: Messaging service core specification. https://www.iso.org/standard/79108.html.

[ISO 15000-2] ISO 15000-2:2021. Electronic business eXtensible Markup Language (ebXML) — Part 2: Applicability Statement (AS) profile of ebXML messaging service https://www.iso.org/standard/79109.html.

[NIST 800-52r2] Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations. NIST Special Publication 800-52 Revision 2. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-52r2.pdf.

[RFC2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2119. March 1997. https://www.rfc-editor.org/rfc/rfc2119

[RFC2392] E. Levinson. Content-ID and Message-ID Uniform Resource Locators. August 1998. https://www.rfc-editor.org/rfc/rfc2392.

[RFC2822] P. Resnick. Internet Message Format.https://www.rfc-editor.org/rfc/rfc2822.

[RFC5246] T. Dierks et al. The Transport Layer Security (TLS) Protocol Version 1.2. IETF RFC 5246. August 2008. https://www.rfc-editor.org/rfc/rfc5246

[RFC6176] S. Turner et al.Prohibiting Secure Sockets Layer (SSL) Version 2.0. RFC 6176. March 2011. https://www.rfc-editor.org/rfc/rfc6176

[RFC8305] D. Schinazi and T. Pauly. Happy Eyeballs Version 2: Better Connectivity Using Concurrency. https://www.rfc-editor.org/rfc/rfc8305.

[RFC8410] S. Josefsson and J. Schaad. Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure. https://www.rfc-editor.org/rfc/rfc8410.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, https://www.rfc-editor.org/info/rfc8446.

[RFC9231] D. Eastlake 3rd. Additional XML Security Uniform Resource Identifiers (URIs). https://www.rfc-editor.org/rfc/rfc9231.html.

[RFC9231bis] D. Eastlake 3rd. Additional XML Security Uniform Resource Identifiers (URIs) draft-eastlake-rfc9231bis-xmlsec-uris-02. https://datatracker.ietf.org/doc/draft-eastlake-rfc9231bis-xmlsec-uris/.

[RFC9325] Y. Sheffer, P. Saint-Andre and T. Fossati. Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). https://www.rfc-editor.org/rfc/rfc9325.

[WSSSMS] OASIS Web Services Security: SOAP Message Security Version 1.1.1. OASIS Standard, May 2012. http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-SOAPMessageSecurity-v1.1.1.doc

[WSSSWA] OASIS Web Services Security: Web Services Security SOAP Message with Attachments (SwA) Profile Version 1.1.1. OASIS Standard, May 2012. http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-SwAProfile-v1.1.1.doc

[WSSX509] OASIS Web Services Security: Web Services Security X.509 Certificate Token Profile Version 1.1.1. OASIS Standard, May 2012. http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-x509TokenProfile-v1.1.1.doc

[XML10] T. Bray et al. Extensible Markup Language (XML) 1.0. W3C Recommendation 26 November 2008, http://www.w3.org/TR/REC-xml/

[XMLDSIG] XML Signature Syntax and Processing (Second Edition). W3C Recommendation 10 June 2008. http://www.w3.org/TR/2008/REC-xmldsig-core-20080610

[XMLDSIG1] XML Signature Syntax and Processing Version 1.1. W3C Recommendation 11 April 2013. http://www.w3.org/TR/xmldsig-core1/

[XDSIGBP] XML Signature Best Practices. W3C Working Group Note 11 April 2013. http://www.w3.org/TR/2013/NOTE-xmldsig-bestpractices-20130411/

[XMLENC] XML Encryption Syntax and Processing. W3C Recommendation 10 December 2002. http://www.w3.org/TR/xmlenc-core/

[XMLENC1] XML Encryption Syntax and Processing Version 1.1. W3C Recommendation 11 April 2013. http://www.w3.org/TR/xmlenc-core1/