IEC 62351-3

Edition 2.0 2023-06

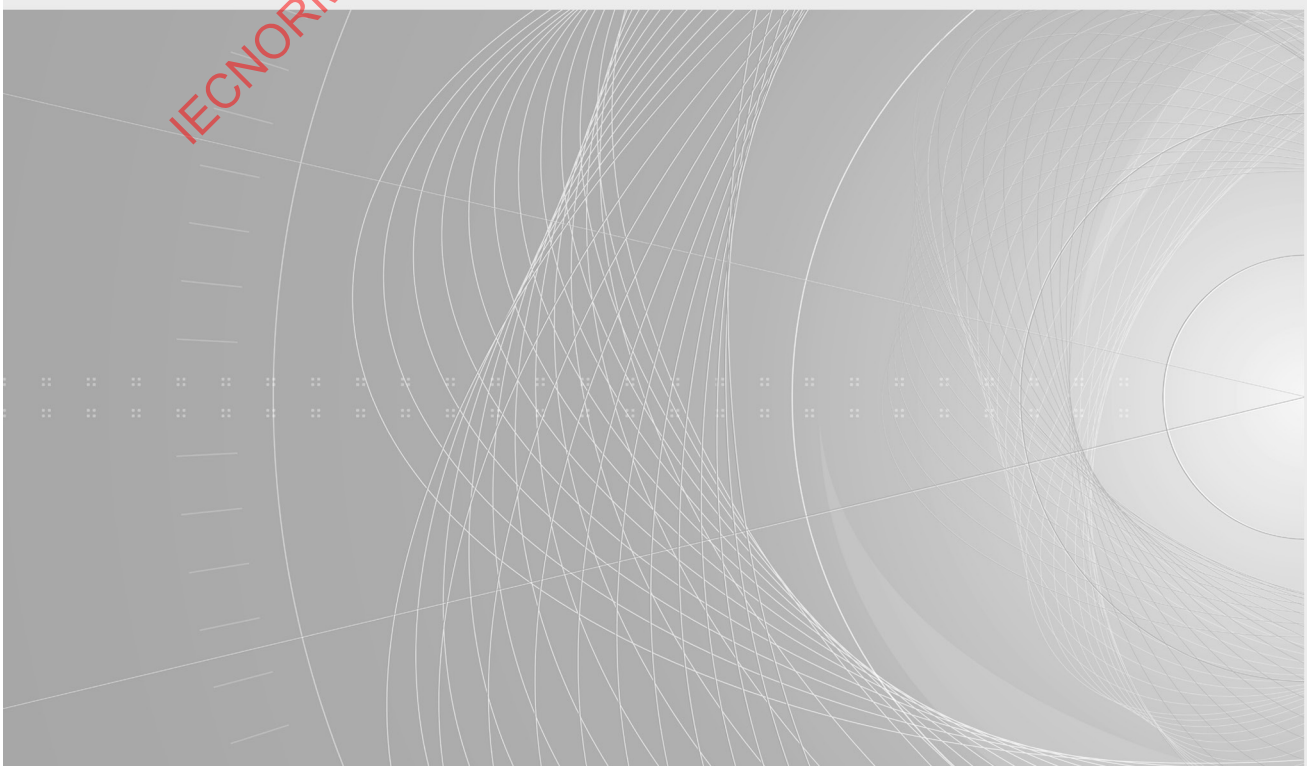
INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Power systems management and associated information exchange – Data and communications security –
Part 3: Communication network and system security – Profiles including TCP/IP

Gestion des systèmes de puissance et échanges d'informations associés – Sécurité des communications et des données –
Partie 3: Sécurité des réseaux et des systèmes de communication – Profils comprenant TCP/IP



IEC 62351-3:2023-06(en-fr)

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

IEC 62351-3

Edition 2.0 2023-06

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Power systems management and associated information exchange – Data and communications security – Part 3: Communication network and system security – Profiles including TCP/IP

Gestion des systèmes de puissance et échanges d'informations associés – Sécurité des communications et des données – Partie 3: Sécurité des réseaux et des systèmes de communication – Profils comprenant TCP/IP

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 33.200

ISBN 978-2-8322-6935-0

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

POWER SYSTEMS MANAGEMENT AND ASSOCIATED INFORMATION EXCHANGE – DATA AND COMMUNICATIONS SECURITY –

Part 3: Communication network and system security – Profiles including TCP/IP

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 62351-3 has been prepared by IEC technical committee 57: Power systems management and associated information exchange. It is an International Standard.

This second edition cancels and replaces the first edition published in 2014, Amendment 1:2018 and Amendment 2:2020. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) Inclusion of the TLSv1.2 related parameter required in IEC 62351-3 Ed.1.2 to be specified by the referencing standard. This comprises the following parameter:
   - Mandatory TLSv1.2 cipher suites to be supported.
   - Specification of session resumption parameters.
   - Specification of session renegotiation parameters.

- Revocation handling using CRL and OCSP.
- Handling of security events.

b) Inclusion of a TLSv1.3 profile to be applicable for the power system domain in a similar way as for TLSv1.2 session.

The text of this International Standard is based on the following documents:

Draft	Report on voting
57/2578/FDIS	57/2593/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

NOTE The following print types are used:

- Abstract Syntax Notation One (ASN.1) are presented in `courier new` and **`bold courier new`**

A list of all parts in the IEC 62351 series, published under the general title *Power systems management and associated information exchange – Data and communications security*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This edition of IEC 62351-3 is a self-contained document profiling the usage of TLS for to secure power system communication. It is recommended to refer to this edition of this document rather than any previous edition, because this edition updates the utilized cryptographic algorithms (ciphersuites), provides enhanced functionality, and covers different TLS versions. In contrast to previous editions, this document specifies all necessary TLS specific settings and does not require the referencing standard to define specific settings for TLS.

Note that the recommendation to use this edition, potentially also with older referencing standards, requires technical support by implementations of the TLS settings specified in this edition of the document.

POWER SYSTEMS MANAGEMENT AND ASSOCIATED INFORMATION EXCHANGE – DATA AND COMMUNICATIONS SECURITY –

Part 3: Communication network and system security – Profiles including TCP/IP

1 Scope

1.1 Scope

This part of IEC 62351 specifies how to provide confidentiality, integrity protection, and message level authentication for protocols that make use of TCP/IP as a message transport layer and utilize Transport Layer Security when cyber-security is required. This may relate to SCADA/telecontrol, protection, automation and control protocols.

IEC 62351-3 specifies how to secure TCP/IP-based protocols through constraints on the specification of the messages, procedures, and algorithms of Transport Layer Security (TLS) (TLSv1.2 defined in RFC 5246, TLSv1.3 defined in RFC 8446). In the specific clauses, there will be subclauses to note the differences and commonalities in the application depending on the target TLS version. The use and specification of intervening external security devices (e.g., "bump-in-the-wire") are considered out-of-scope.

In contrast to previous editions of this document, this edition is self-contained in terms of completely defining a profile of TLS. Hence, it can be applied directly, without the need to specify further TLS parameters, except the port number, over which the communication will be performed. Therefore, this part can be directly utilized from a referencing standard and can be combined with further security measures on other layers. Providing the profiling of TLS without the need for further specifying TLS parameters allows declaring conformity to the described functionality without the need to involve further IEC 62351 documents.

This document is intended to be referenced as a normative part of other IEC standards that have the need for providing security for their TCP/IP-based protocol exchanges under similar boundary conditions. However, it is up to the individual protocol security initiatives to decide if this document is to be referenced.

The document also defines security events for specific conditions, which support error handling, security audit trails, intrusion detection, and conformance testing. Any action of an organization in response to events to an error condition described in this document are beyond the scope of this document and are expected to be defined by the organization's security policy.

This document reflects the security requirements of the IEC power systems management protocols. Should other standards bring forward new requirements, this document may need to be revised.

1.2 Intended audience

The initial audience for this document is intended to be experts developing or making use of protocols in the field of power systems management and associated information exchange. For the measures described in this document to take effect, they must be accepted and referenced by the specifications of protocols making use of TCP/IP security by applying TLS. This document is written to enable that process.

The subsequent audience for this document is intended to be the developers of products that implement these protocols.

Portions of this document may also be of use to managers and executives in order to understand the purpose and requirements of the work.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC TS 62351-1:2007, *Power systems management and associated information exchange – Data and communications security – Part 1: Communication network and system security – Introduction to security issues*

IEC TS 62351-2:2008, *Power systems management and associated information exchange – Data and communications security – Part 2: Glossary of terms*

IEC 62351-9, *Power systems management and associated information exchange – Data and communications security – Part 9: Cyber security key management for power system equipment*

ISO/IEC 9594-8:2020 | Rec. ITU-T X.509 (2019), *Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks*

RFC 5246:2008, *The TLS Protocol Version 1.2*[1]

RFC 5280:2008, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 5288:2008, *AES Galois Counter Mode (GCM) Cipher Suites for TLS*

RFC 5289:2008, *TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)*

RFC 5746:2010, *Transport Layer Security (TLS) Renegotiation Indication Extension*

RFC 6066:2011, *Transport Layer Security Extensions*

RFC 6176:2011, *Prohibiting Secure Sockets Layer (SSL) Version 2.0*

RFC 8422:2018, *ECC Cipher Suites for TLSv1.2 and earlier*

RFC 8446:2018, *The TLS Protocol Version 1.3*

RFC 9150:2021, *TLS 1.3 Authentication and Integrity only Cipher Suites*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC TS 62351-2 and the following apply.

[1] This is typically referred to as SSL/TLS.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.2 Abbreviated terms

ACSE	Association Control Service Element
AEAD	Authenticated Encryption with Associated Data
BCP	Best Current Practice
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules
DH(E)	Diffie Hellman (Ephemeral) Key Agreement (see also IEC 62351-9)
ECDH(E)	Elliptic Curve based Diffie Hellman (Ephemeral) Key Agreement (see also IEC 62351-9)
ECDSA	Elliptic Curve Digital Signature Algorithm
IV	Initialization Vector
MAC	Message Authentication Code
MitM	Man-in-the-Middle (type of attack)
OCSP	Online Certificate Status Protocol (see RFC 6960)
PICS	Protocol Implementation Conformance Statement
PIXIT	Protocol Implementation eXtra Information for Testing
PSK	Pre-shared key
TLS	Transport Layer Security

4 Security issues addressed by this document

4.1 General

The IEC power system environment has different operational requirements from many Information Technology (IT) applications that make use of TLS to protect the transport of information over a TCP/IP connection. This requires on one hand the management of certificates used to authenticate entities, which is handled in IEC 62351-9. On the other hand, specific TLS related security parameters such as selected cipher suites, session management parameter and utilized extensions need to be defined, which is the focus of this document. One main difference in the application of TLS in the power system domain is the duration of the TCP/IP connection for which security needs to be maintained. Besides this it also has to be considered that the equipment in power system automation is long lasting. This may also require supporting older versions of TLS or cipher suites to retain backward compatibility.

Many IT protocols have short duration connections, which allow the encryption algorithms to be renegotiated at connection re-establishment. However, the connections within a telecontrol environment tend to have longer durations, often "permanent". It is the longevity of connections in the field of power systems management and associated information exchange that give rise to the need for specific consideration. In this regard, in order to provide protection for the "permanent" connections, a mechanism for updating the session key is selected within this document, based upon existing TLS features. Note that TLSv1.2 and TLSv1.3 support different approaches for rekeying.

TLS allows for a wide variety of settings such as selected cipher suites to protect the handshake and the record layer protocol, which are negotiated during connection establishment. It also supports session management mechanisms to setup new sessions or renew existing sessions. To ensure interoperability between different implementations, this document specifies a common set of TLS features to be supported by conforming implementations.

TLS has evolved and is available in version 1.3. In version 1.3 several parts of the protocol have been reworked, resulting in a protocol, which is functionally not backward compatible to TLSv1.2. As the first message exchange is defined in a backward compatible way, the TLS server may be able to switch to either version, depending on the security policy.

Additionally, this document specifies the use of particular TLS capabilities that allow for specific security threats to be countered (see also 4.2). Specifically, TLS allows the definition of extensions to the protocol to mitigate potential protocol vulnerabilities or to enhance protocol functionality. This specification makes use of already-defined extensions.

Note that TLS utilizes X.509 certificates (see also ISO/IEC 9594-8 or RFC 5280) for authentication and key agreement. In the context of this document the term certificate always relates to public-key certificates (in contrast to attribute certificates).

NOTE It is assumed that the certificate management necessary to operate TLS is addressed in accordance with IEC 62351-9.

4.2 Security threats countered

See IEC TS 62351-1 for a discussion of security threats and attack methods.

TCP/IP and the security specifications in this document cover only the communication transport layers (OSI layers 4 and lower). Specifically, TLS protects the transported messages from OSI layer 5 and above in a transparent way. This document does not cover security functionality specific for the communication application layers (OSI layers 5 and above) or application-to-application security. This is defined in other parts of IEC 62351, e.g., IEC 62351-4 for MMS, IEC 62351-5 for serial communication and telecontrol, and IEC 62351-6 for (R)GOOSE and (R)SV.

NOTE The application of TLS as profiled in this document supports the protection of information sent over a TLS protected connection.

The specific threats countered in this document for the transport layer include:

– Tampering (unauthorized modification or insertion of messages) is addressed by mutual, certificate-based authentication of the communication participants, and TLS packet-level integrity protection.

Additionally, when the information has been identified as requiring confidentiality protection:

– Unauthorized access to information through TLS packet-level encryption of the messages.

4.3 Attack methods countered

The following security attack methods are countered through the appropriate implementation of the specifications and recommendations in this document.

– Impersonation: This threat is countered through the use of mutual authentication based on X.509 certificates during the TLS handshake and digitally signed information as revocation information.
– Man-in-the-middle (MitM): This threat is countered for the TLS record layer through the use of cryptographic checksums (MAC) based on mutually authenticated negotiation of session keys. These keys are set up during the TLS handshake. Protection against MitM attacks during the TLS handshake phase itself is provided by the `Finish` message, ending the handshake phase. This message is encrypted and contains a hash value over all messages

exchanged in the handshake. In addition, TLSv1.3 already provides cryptographic protection of the handshake messages directly (except the `ClientHello`)..

– Replay: This threat is countered by the application of a MAC for integrity protection on a per packet bases including a sequence number to detect an actual replay.

– Eavesdropping: This threat is countered through the use of encryption cipher suites, when negotiating TLS record layer security.

NOTE The actual performance characteristics of an implementation claiming conformance to this part of the IEC 62351 series are out-of-scope of this document.

4.4 Handling of security events

Throughout the document security events are defined. These security events are intended to support error handling and thus to increase system resilience. Implementations should provide a mechanism for announcing security events to an evaluation system. A recommended standard is IEC 62351-4.

The information about security events and potential detailed information can only be provided by the entity based on the availability of this information through the underlying platform or utilized components.

It is recommended that the security events defined throughout this document are made available to the operational infrastructure by cyber security events as specified in IEC 62351-14 or by monitoring objects as specified in IEC 62351-7. Annex A provides a mapping of the defined events in this document to the notion of IEC 62351-14.

Note that notice, warnings, errors, and alarms are used to indicate the severity of an event from a security point of view. The following notion from IEC 62351-14 is used:

– A notice refers to a cyber security related activity during the routine use or maintenance of an entity. It does not relate to a cyber security breach or attack or deviation from the normal operating condition of an entity.

– A warning is a deviation from the normal operating condition of an entity but not necessary a cyber-attack.

– An error describes an unforeseen condition, which may indicate unauthorized activity. It may not require immediate action.

– An alarm is an indication of a serious problem, which may indicate unauthorized activity. Action is recommended to be taken immediately.

In any case, it is expected that an organization's security policy determines the final handling of events based on the operational environment. For instance, the assessment of one or more alarms could rise to the level of an incident.

5 Overview of differences in TLS versions

5.1 General

This clause provides a short overview about the main differences between the TLS version 1.2 and TLS version 1.3. Specifically addressed is the naming of cipher suites, backward compatibility, and extensions, which have been used in previous version of this standard.

5.2 Main differences between TLSv1.2 and TLSv1.3

In the following the main differences between TLSv1.2 and TLSv1.3 for the different subprotocols (TLS Handshake, TLS Record Layer) are listed (note that RFC 8446 provides a complete list of changes from TLSv1.2 to TLSv1.3), which have a relation to features used in TLSv1.2 in IEC 62351-3:

- Fewer handshake roundtrips in TLSv1.3 (one roundtrip for unilateral authenticated sessions, 1.5 for mutual authentication).
- TLSv1.3 handshake messages are encrypted, except the `ClientHello` and parts of the `ServerHello` are send in plain. Note that there is ongoing work in the IETF to define extensions to allow to encrypt part of the `ClientHello`, like the Server Name Indication (SNI) resulting in an encrypted SNI (ESNI) or encrypting the whole `ClientHello` as encrypted `ClientHello` (ECH).
- TLSv1.3 removes support for obsolete and vulnerable cryptographic algorithms: Export ciphers, DES, 3DES, AES-CBC, RC4, MD5, SHA-1 are no longer supported
- TLSv1.3 changes in key establishment
  - Removal of arbitrary Diffie-Hellman groups
  - Key establishment is based on ephemeral Diffie-Hellman
  - No support for RSA key encryption
- TLSv1.3 removes session renegotiation but provides post-handshake messages for key and IV update as well as client authentication.
- TLSv1.3 simplifies session resumption by using PSK and Session Tickets
- TLSv1.3 introduces post handshake messages for
  - Session Ticket
  - Client authentication
  - Key and IV Update
- TLSv1.3 removes the multistapling support of OCSP responses according to RFC 6961 (status_request_v2). It has been replaced with a direct variant allowing the stapling of OCSP response to certificates in a chain by an extension of a `CertificateEntry`.
- TLSv1.3 introduces OCSP stapling for client certificates, which may be important specifically for industrial use cases, in which the server often resides on the field device with potential limited capability of checking revocation state of received certificates
- TLSv1.3 removes non-AEAD cipher suites for TLS record layer protection. Note that integrity-only cipher suites have been defined in RFC 9051.
- TLSv1.3 provides the option to have already encrypted application data in the first roundtrip (0-RTT) with some reductions in the security (option for replay by an attacker). Note that this option is explicitly not used in this document.
- Note that `ChangeCipherSpec` is an independent TLSv1.2 protocol content type and is not a TLS handshake message. `ChangeCipherSpec` was removed in TLSv1.3.

5.3 Cipher suite naming

The changes in the TLS handshake and record layer handling in TLSv1.3 are also reflected in the cipher suite definition.

A cipher suite in TLSv1.2 and before combines algorithms for authentication and key agreement during the TLS handshake as well as algorithms for encryption and message authentication (integrity) for the record layer as shown in Figure 1. The cipher suite therefore considered both, the TLS handshake as well the record layer. Typically, each cipher suite has a name, which consists of a set of mnemonics separated by underscores and has the form shown in Figure 1.

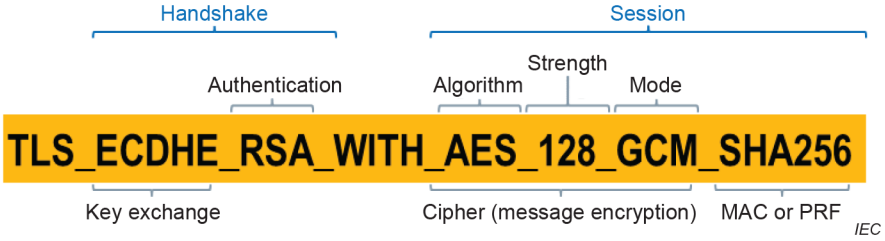


Figure 1 – Definition of cipher suites according to TLSv1.2 (RFC 5246)

In TLSv1.3 the cipher suite naming was changed to only relate to the record layer. Hence, they contain the information about the utilized encryption scheme (always authenticated encryption) and the hash utilized in the hash-based key derivation function as seen in Figure 2.

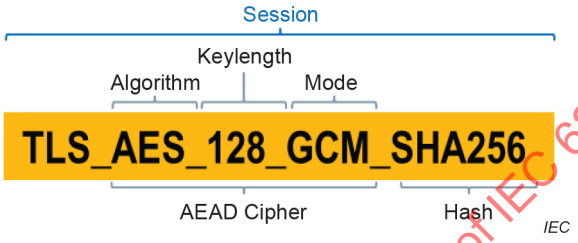


Figure 2 – Definition of cipher suites according to TLSv1.3 (RFC 8446)

The algorithms used in the TLS handshake are explicitly signalled as part of the `ClientHello` and the `ServerHello` exchange.

5.4 Backward compatibility

TLS ensures backward compatibility by signalling the TLS version in the `Client.Hello` message, which is compatible throughout the version 1.x messages. Note that the TLSv1.2 version negotiation mechanism has been deprecated in favour of a version list in an extension. Specifically, a `supported_versions` extension can be used in the `Client.Hello` to negotiate the version of TLS to use, in preference to the `legacy_version` field of the `Client.Hello`. This is intended to increase compatibility with existing servers that implemented version negotiation incorrectly.

5.5 Extensions

Extensions provide an option to include additional information in some of the handshake messages of TLS. As stated in 5.2, some of the features available in TLSv1.2 are not supported in TLSv1.3 or handled differently. Consequently, some of the extensions are not applicable or not necessary to support in TLSv1.3.

Extensions mandatorily used in Edition 1 of this document (2007) comprise:

– `renegotiation_info`: Renegotiation extension defined in RFC 5746 to bind the renegotiated session to the previous session.

– `trusted_ca_keys`: Allows a client to indicate locally available CA certificates, which can be used to validate received server certificates (defined in RFC 6066).

– `signature_algorithms`: Allows the client to signal signature algorithms (as combination of hash and signature algorithm) it supports (defined in both RFC 5426 (TLSv1.2) and RFC 8446 (TLSv1.3)).

Of these extensions, only the `signature_algorithm` extension is supported in both TLS versions. Instead of the `trusted_ca_keys` extension, TLSv1.3 specifies the "`certificate_authorities`" extension which can be used by either endpoint to signal the supported CA as guidance for the receiving endpoint. The `renegotiation_info` extension is not supported in TLSv1.3 as session renegotiation is not supported.

In addition, TLSv1.3 defines mandatory to support extensions, which will be profiled in this document. The list of extensions to be supported will be provided in the specific clauses for the different TLS versions.

6 Generic requirements

6.1 General

This clause defines requirements which are applicable to all versions of TLS considered in this document.

If the handshake to set up a TLS session performs successfully (meets the requirements outlined in Clause 6, Clause 7 for TLSv1.2, or Clause 8 for TLSv1.3 a security event shall be raised ("notice: TLS handshake successfully performed"). If errors occurred during the TLS channel establishment and the TLS session cannot be established, security events defined in Clause 6, Clause 7 for TLSv1.2, and Clause 8 for TLSv1.3, will provide detailed information.

Referencing standards of this document shall specify a port number, which is used to perform the TLS protected communication.

6.2 Signalling of supported TLS versions

TLSv1.2 as defined in RFC 5246 is the default version that shall be supported. TLSv1.3 is the most current version of TLS and is recommended to be supported.

It is recommended that the TLS client initiating a TLS connection indicates the highest TLS version supported in the `ClientHello` message of the TLS handshake. The receiving TLS server may accept higher versions if functionally supported and allowed by the security policy of the operating environment.

TLSv1.2 and TLSv1.3 handle the version signalling differently. In TLSv1.2 the version number is provided as part of the `ProtocolVersion` in the `CientHello` message and signals the highest TLS version supported by the client.

The client shall include the TLSv1.2 identifier `0x0303` in the `ProtocolVersion` to signal TLSv1.2 support. If the client supports TLSv1.3 additionally, it indicates its version preferences in the `supported_versions` extension of the `ClientHello` message. The `ProtocolVersion` in the `supported_versions` extension in this case shall be set to the TLSv1.3 identifier `0x0304` according to RFC 8446. Note that a server supporting TLSv1.3 receiving a `ClientHello` message with the `supported_versions` extension will not use the TLSv1.2 version signalled in the `ProtocolVersion` field of the `ClientHello` message.

To ensure backward compatibility implementations may optionally support TLS version 1.0 and 1.1 (sometimes referred to as SSL v3.1 and v3.2). The TLS handshake provides a built-in mechanism that shall be used to support version negotiation. The peer initiating a TLS connection shall always indicate the highest TLS version supported during the TLS handshake message. The application of TLS versions other than v1.2 is a matter of the local security policy. Proposal of versions prior to TLSv1.0 shall result in no secure connection being established (see also RFC 6176).

NOTE 1 For TLSv1.0 and TLSv1.1 certain vulnerabilities are known, Moreover, the IETF released a BCP for deprecation of TLSv1.0 and TLSv1.1 in RFC 8996. The optional support is only intended for backward compatibility.

Support for only TLSv1.0 or only TLSv1.1 or only both shall be disabled by default and requires distinct enabling authorized by an organization's security policy. If support is disabled, the proposal of only TLSv1.0 or only TLSv1.1 or only both shall raise a security event ("alarm: unsecure communication: deprecated TLS version proposed") and the session establishment shall be terminated by the receiver. If support is enabled the proposal of versions TLSv1.0 or TLSv1.1 shall raise a security event ("warning: insecure TLS version").

TLS versions prior to TLSv1.0 shall not be used. The proposal of versions prior to TLSv1.0 or SSLv3.1 only shall raise a security event ("alarm: unsecure communication: disallowed TLS version proposed") and the session establishment shall be terminated by the receiver.

NOTE 2 The IETF has deprecated SSLv3.0 in RFC 7568.

The TLS session shall be terminated if the negotiated TLS version changes from the initial TLS handshake:

- TLSv.1.2 and lower: during a TLS session renegotiation or a session resumption handshake;
- TLSv1.3: session resumption based on a provided PSK identity in TLSv1.3 from either side.

The termination of the session shall raise a security event ("alarm: requested TLS Version change (potential downgrade) in ongoing communication detected").

NOTE 3 RFC 8446 specifically forbids version changes during a TLSv1.2 renegotiation to upgrade to TLSv1.3.

6.3 Usage of non-encrypting cipher suites

Any cipher suite that specifies NULL for encryption shall not be used for communication outside the administrative domain if the encryption of this communication connection by other means cannot be guaranteed.

NOTE 1 This document does not exclude the use of encrypted communications through the use of cryptographic VPN tunnels. The use of such VPNs is out-of-scope of this standard.

Within an administrative domain, the use of non-encrypting cipher suites may be allowed, but its use is in the responsibility of the operator.

NOTE 2 The application of no-encrypting cipher suites allows for traffic inspection, while still retaining an end-to-end authentication and integrity protection of the traffic.

If the communication connection is intended to be only secured regarding the integrity of transmitted information, the following cipher suites may be used with TLSv1.2 and below:

– TLS_RSA_WITH_NULL_SHA256 (defined in RFC 5246)

NOTE 3 IEC 62351-3:2014+AMD1:2018+AMD2:2020 had support for the cipher suite TLS_RSA_WITH_NULL_SHA. This optional support has been removed due to the deprecation of SHA-1.

If the communication connection is intended to be only secured regarding the integrity of transmitted information, the following cipher suites may be used with TLSv1.3:

– TLS_SHA256_SHA256 (RFC 9150)

– TLS_SHA384_SHA384 (RFC 9150)

Implementations allowing TLS cipher suites with NULL encryption claiming conformance to this part shall provide a mechanism to explicitly enable those TLS cipher suites. Per default, non-encrypting TLS cipher suites shall be disabled.

6.4 Certificate support

6.4.1 Support of multiple trust anchors

A CA-related trust anchor provides the base for certificate validation, specifically the certificate path validation. As entities are expected to have interactions with entities belonging to a different domain, different trust anchors need to be supported. The actual number depends on the target use case.

An implementation claiming conformance to this document shall support at least 5 trust anchors (typically referrd to as root CA certificates) as the minimum number.

The criteria and selection of a CA are out-of-scope of this document.

In scenarios where more than one X.509 certificate (and corresponding private key) is available on an IED, it may be desirable to enable the requester to choose a certificate on the server side that matches the trusted anchor (root CA) certificates available at the requester side.

6.4.2 Certificate size

An implementation claiming conformance to this standard shall be able to handle certificates up to a maximum size of at least 8 192 octets. The actual supported certificates size shall be documented.

NOTE 1 The certificate may also carry role information according to IEC 62351-8, which influences its final size.

NOTE 2 The certificate size may be influenced by the careful selection of names in issuer and subject field and supported extensions, etc.

Exceeding the maximum supported size of a certificate during a TLS handshake shall raise a security event ("alarm: TLS certificate size exceeded") and termination of the TLS session establishment.

Note that the limitation of the certificate size relates to potential requirements from upper layer protocols like MMS. For MMS, in order to achieve interoperability of public-key certificates, it is necessary to set a maximum allowed size for the public-key certificates exchanged by ACSE. This size limitation is described in IEC 62351-4. If the same certificate is used in the context of MMS and TLS, this boundary condition needs to be obeyed.

Manufacturer may support larger certificate sizes for TLS if the certificates do not underlie higher layer certificate size restrictions.

6.4.3 Certificate exchange

The certificate exchange and validation shall be bi-directional to achieve mutual authentication. If either entity does not provide its certificate, the connection shall be terminated.

NOTE The server certificate is conveyed in the `ServerHello` message. The client certificate is conveyed in the `Certificate` message.

The connection termination due to the lack of a peer certificate during the TLS handshake shall raise a security event ("alarm: peer certificate unavailable").

The inability to access the local endpoint certificate and corresponding private key shall raise a security event ("alarm: endpoint certificate unavailable").

6.4.4 Public-key certificate validation

6.4.4.1 General

Certificates shall be verified by both the calling and called side. In addition, each side shall validate its own certificate as outlined in this clause and more specifically also regarding the included certificate extensions as defined in IEC 62351-9 prior to the usage of the certificate.

There are two mechanisms that shall be configurable for certificate validation.

– Acceptance of any certificate from an authorized CA
– Acceptance of individual certificates from an authorized CA (limit scope of issuing CA)

6.4.4.2 Verification based upon selected issuing CAs

An implementation claiming conformance to this standard shall be capable of being configured to accept certificates from one or more Certificate Authorities without the configuration of individual certificates.

The failure of finding a matching CA certificate during a TLS handshake shall raise a security event ("alarm: certificate validation: CA certificate not available") and termination of the TLS session establishment.

6.4.4.3 Verification based upon individual certificates from selected issuing CAs

An implementation claiming conformance to this standard should be capable of being configured to accept individual certificates (list of trusted certificates) from one or more authorized Certificate Authorities. Configuration of individual certificates may be based on the subject name or the serial number of the certificate together with the issuing CA information.

NOTE 1 The verification based upon individual certificates may be influenced by automated certificate update procedures, which typically change the serial number of certificates.

NOTE 2 The verification based upon individual certificates may be supported by certificate authorization and validation lists as outlined in IEC 62351-9.

The failure of finding a matching individual certificate during a TLS handshake shall raise a security event ("alarm: certificate validation: trusted individual certificate from authorized CA not available") and termination of the TLS session establishment.

6.4.4.4 Certificate revocation check

6.4.4.4.1 General

Certificate revocation shall follow the mandatory parameters and procedures specified in ISO/IEC 9594-8. The management of a CRL is a local implementation issue. Discussion of the management issues regarding CRLs can be found in IEC TS 62351-1. The application of CRLs is outlined in IEC 62351-9.

Alternatively, to local CRLs, OCSP may be used to check the revocation state of applied certificates. The application of OCSP interactions with an OCSP responder is outlined in IEC 62351-9.

If OCSP is applied, TLS offers support to transport OCSP response messages as so-called OCSP stapling. This is possible for the server certificate in TLSv1.2 and is described in 7.5.5. TLSv1.3 supports the stapling of OCSP responses for client certificates and server certificates as described in 8.8.10.

An implementation claiming conformance to this standard shall be capable of checking the revocation state of received certificates at a configurable time interval.

The certificate revocation check may be invoked based on TLS protocol features like session renegotiation for TLSv1.2 or by the application monitoring the utilized certificates.

In case of application monitoring of utilized TLS certificates, the application is required to keep track of peer certificates used in TLS session establishments and invoke the certificate revocation check based on configurable time. To allow for operational actions in case of revoked certificates (e.g., provisioning of a valid credential or performing a security check of the related IED) the application may provide a grace period (in hours) in which these actions may be performed without immediate termination of the current session. If no grace period is provided, the application is required to terminate the associated connection. The default value for the grace period is 0 hours (i.e., no grace period). Note that the configuration of the grace period for the application is a local implementation issue and may be subject to approval according to the organization's security policy.

Revoked certificates shall not be used in the establishment of a TLS session (TLSv1.2 and TLSv1.3). An entity receiving a revoked certificate during session establishment shall refuse the connection. An entity receiving a revoked certificate during a TLSv1.2 session renegotiation shall terminate the connection.

The detection of a revoked certificate in a session establishment or session renegotiation shall raise a security event ("alarm: certificate validation: revoked certificate").

Termination of a TLS connection due to a revoked certificate shall raise a security event ("alarm: session terminated due to revoked certificate").

Note that revoked certificates may lead to TLS session termination. Therefore, system administrators should develop certificate management procedures to mitigate this situation (see also IEC 62351-9). Also, a security management process is expected to be in place to evaluate the CRL or OCSP response before distribution to avoid an undesired teardown of a communication connection, which may influence the reliability of the system. Implementations may also support multiple certificates per device to allow a switch on-the-fly to another (valid) certificate. Note that this latter option is a local implementation issue.

6.4.4.4.2 Certificate revocation check using CRLs

The process of checking the CRL shall not cause an established session to be terminated.

For TLSv1.2 the teardown of a TLS session due to a revoked certificate will become effective with the next session renegotiation as outlined in 7.4.5. This may leave a grace period (between certificate revocation and detection of the revoked certificate in use) for an operator to handle a revoked certificate. The alternative is the handling of the certificate revocation check by the application as outlined in 6.4.4.4.1.

NOTE 1 As TLSv1.2 allows session renegotiation, it is typically executed by the TLS stack and can invoke the application to perform the certificate validation and specifically the revocation check.

As TLSv1.3 does not support session renegotiation, it becomes the responsibility of the application to invoke the certificate validation and act upon detected revoked certificates, specifically in long-lasting sessions. This requires the application to keep track of utilized certificates during the connection establishment. This is outlined in 6.4.4.4.1

The recommended period of refreshing a local CRL is 24 hours. The CRL itself also contains information when the CRL will be updated in the `nextUpdate` field. The refreshing of the CRL should be aligned with whatever time comes first. For fetching a new CRL the CRL distribution point of the issuing CA should be contacted. An inability to access the CRL distribution point shall not cause the session to be terminated. The unavailability of a local CRL shall raise a security event ("warning: local CRL not accessible"). An inability to access a local CRL shall not cause the session to be terminated.

NOTE 2 The CRL can be distributed in different ways (manual as file, fetched from CRL distribution point, etc.).

NOTE 3 If there are no revoked certificates, the CRL is an empty list, but would still be available.

The expiry of a local CRL shall raise a security event ("warning: CRL expired") but shall not cause the session to be terminated.

6.4.4.4.3 Certificate revocation check using OCSP

An implementation claiming conformance to this document should be capable of interacting with an OCSP responder to fetch revocation information.

An implementation claiming conformance to this document shall be capable of processing OCSP response messages, when provided as part of the TLS handshake as stapled OCSP response.

NOTE The interaction with an OSCP responder also provides an OCSP response message.

The expiry of an OCSP response shall raise a security event ("warning: OCSP response expired")."

OCSP responses have an expiry time indicated by the `nextUpdate` value. This allows caching an OCSP response. If OCSP is applied, the caching of OCSP responses shall be supported with a maximum of 24 hours.

If OCSP is applied for certificate revocation checks, the inaccessibility of the OCSP responder shall raise a security event ("warning: OCSP responder not accessible").

For TLSv1.2, RFC 6066 defines a certificate status request (`status_request`) and response extension allowing TLSv1.2 to provide an OSCP response for the server certificate along with its certificate. As this extension only allows to provide a single OCSP response, a further extension is defined in RFC 6961 allowing to request (`status_requestv2`) and staple OCSP responses also for intermediate CA certificates contained in the certificate list of the server certificate message. More details are provided in 7.5.5.

TLSv1.3 provides support for requesting and stapling OCSP responses as described in RFC 6066 for all certificates in the certificate list provided by the client or the server side. More details are provided in 8.8.10.

6.4.4.5 Expired certificates

Expired certificates shall not be used in the establishment (TLSv1.2 and TLSv1.3) or renegotiation (only TLSv1.2) of a TLS session. An entity receiving an expired certificate during session establishment shall refuse the connection (TLSv1.2 and TLSv1.3). An entity receiving an expired certificate during session renegotiation in case of TLSv1.2 shall terminate the connection.

Note that it is expected that there is a security management process in place to initiate a timely certificate renewal procedure (see also IEC 62351-9). An example time frame may be a month. There may also be support for multiple certificates per device to permit the switch to another (valid) certificate.

The refusal of an initial session establishment or session renegotiation due to an expired certificate shall raise a security event ("alarm: certificate validation: expired certificate").

6.4.4.6 Signature algorithms

The support of cryptographic algorithms for signatures during the key exchange is defined for TLSv1.2 in 7.4 and for TLSv1.3 in 8.3.

The failure of finding a matching signature algorithm to the certificate components during a TLS handshake shall raise a security event ("alarm: certificate validation: signature algorithms not supported").

The failure of validating the signature of received certificates during a TLS handshake shall raise a security event ("alarm: certificate validation: certificate signature could not be verified"). Note that the indication for supported signature algorithms must be done using the appropriate extensions as defined in 7.5.4 for TLSv1.2 and in 8.8.4 (signature algorithm) and 8.8.4.2 (signature algorithms certificate) for TLSv1.3.

6.5 Co-existence with non-secure protocol traffic

Referencing standards requiring support for non-secure protocol traffic may specify how the co-existence of secure and non-secure communication is handled. This may be done by providing a separate TCP/IP port for TLS-secured traffic.

7 Requirements specific to TLSv1.2

7.1 General

In earlier versions of this standard, a referencing standard was required to provide additional definitions for handling TLS or for supporting of TLS protocol features.

Starting with this edition, these definitions have been integrated into this document. Moreover, the description for TLSv1.2 has been aligned with the specification of TLS related parameters provided in the referencing standards to IEC 62351-3:2014+AMD1:2018+AMD2:2020, which are IEC 62351-4:2018+ AMD1:2018 and IEC 62351-8:2020.

7.2 Supported cipher suites

For TLSv1.2 (and below) this document specifies support of the mandatory and optionally supported cipher suites in Table 1.

Table 1 – Support of cipher suites for TLSv1.2

Key exchange		Encryption	Hash	IANA Value	Source	Support
Algorithm	Signature					
TLS_RSA_		WITH_NULL	SHA256	0x00,0x3B	RFC 5246	c1
TLS_RSA		WITH_AES_128_CBC_	SHA256	0x00,0x3C	RFC 5246	m
TLS_DHE_	RSA_	WITH_AES_128_GCM_	SHA256	0xC0,0x9E	RFC 5288	m
TLS_DHE_	RSA_	WITH_AES_256_GCM_	SHA384	0x00,0xA1	RFC 5288	o
TLS_ECDHE_	RSA_	WITH_AES_128_GCM_	SHA256	0xC0,0x2F	RFC 5289	m
TLS_ECDHE_	RSA_	WITH_AES_256_GCM_	SHA384	0xC0,0x30	RFC 5289	o
TLS_ECDHE_	ECDSA_	WITH_AES_128_GCM_	SHA256	0xC0,0x2B	RFC 5289	m
TLS_ECDHE_	ECDSA_	WITH_AES_256_GCM_	SHA384	0xC0,0x2C	RFC 5289	o
c1: may be supported if integrity only protection is desired. These cipher suites shall be disabled by default and require distinct enabling authorized by an organization's security policy.						

Additional cipher suites may be supported. For the selection of these cipher suites the recommendations from IETF should be followed. The interoperability for additional cipher suites depends on the peer support of these cipher suites.

The final selection of cipher suites used in the operational environment depends on the organization's security policy. As this document support multiple mandatory cipher suites, an operator may opt out of cipher suites not providing PFS, if no backward compatibility is desired.

The handling of cipher suites supporting NULL for encryption is described in 6.3.

7.3 Disallowed cipher suites

The list of disallowed cipher suites includes, but is not limited to:

– TLS_NULL_WITH_NULL_NULL
– Cipher suites containing TLS_*_*_MD5 (disallowing MD5 for authentication for any combination) and
– Cipher suites containing TLS_*_DES_* (disallowing DES for encryption for any combination)

The signalling of a disallowed cipher suite during the TLS handshake is handled differently on client and server side:

If only disallowed cipher suites are proposed in the `ClientHello`, a security event shall be raised ("alarm: disallowed TLSv1.2 cipher suite proposed") on the TLS server.

If a disallowed cipher suite is signalled in the `ServerHello` message, the TLS client shall raise a security event ("alarm: disallowed TLSv1.2 cipher suite proposed").

If in a set of cipher suites, a disallowed cipher suite is proposed in the `ClientHello`, the TLS Server shall ignore the disallowed proposal.

The session establishment shall be terminated if only disallowed cipher suites are signaled by either side.

TLS client/server are expected to utilize the TLS cipher suites listed in Table 1. It is allowed to also utilize further cipher suites but with no guarantee for interoperability, except the disallowed. Using other cipher suites should be done based on the risk assessment of the operator.

If the list of cipher suites excludes the stated cipher suites in 7.2 and is overwritten by local security policies, it then results in non-conformance to part 3 (negative test case in 100-3 PIXIT).

7.4 Key exchange

7.4.1 General

The key exchange mechanisms are determined by the supported cipher suites as stated in 7.2.

7.4.2 Key exchange mechanisms

The key exchange mechanisms to be supported are aligned with the supported cipher suites listed in 7.2 and comprise:

– Public key encryption using RSA
– Ephemeral Diffie-Hellman key agreement.
– Elliptic curve based ephemeral Diffie Hellman key agreement.

NOTE There are cipher suites defined for TLSv1.2 also supporting a mixture of both RSA based signatures and elliptic curve based Diffie Hellman.

7.4.3 Cryptographic algorithms

Signing through the use of RSA and ECDSA algorithms shall be supported.

For RSA-based signatures, the following key length shall be supported:

– Mandatory: Signature-operation: RSA with a key length of at least 2 048 Bits.

RSA with 2 048 bit key length shall be supported. The key length of 2 048 is the minimum key length to be supported for RSA signatures. Based on recommendations NIST SP800-57 or BSI TR01202-1 it is strongly recommended to also support RSA key length of 3 072 bit and higher to address advances in cryptography. The selection of the signature algorithm depends on the organization's security policy.

The support of RSA with 1 024 bit keys is deprecated. Its use is limited to backward compatibility. Support for 1 024 bit RSA keys shall be disabled by default and requires distinct enabling, authorized by an organization's security policy. If the use of 1 024 bit RSA key is disabled, the detection of RSA keys with less than 2 048 bit RSA shall raise a security event ("alarm: insufficient RSA key length (<2 048 bit)") and the session establishment shall be terminated by the receiver. If the use of 1 024 bit RSA keys is enabled the detection of a RSA key in the interval of 1024 <= RSA key length <2048 shall raise a security event ("warning: minimum key length").

If larger keys than 1 024 bits cannot be used, it is also recommended that additional security measures be taken.

RSA keys with less than 1 024 bits key shall not be used. The detection of RSA keys with less than 1 024 bits shall raise a security event ("alarm: insufficient key length") and the session establishment shall be terminated by the receiver.

The following key length shall be supported for the ECDSA public key algorithm:

– Mandatory: Signature-operation: ECDSA key length of at least 256 bit

NOTE 1 Recommendations regarding required key length for signature algorithms are reviewed constantly and can be found in NIST SP800-57, BSI TR 02102-1, or the NSA Suite B.

For signatures based on elliptic curves defined over finite prime fields with signature algorithm ECDSA. The minimum key length is 256 bits (in combination with SHA-256). The OID for `ecdsa-with-SHA256` to be used is: iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) ecdsa-with-SHA256(2).

Implementations claiming conformance to this standard shall support ECDSA with the curve: {OID}:

`secp256r1`: {iso(1) member-body(2) us(840) ansi-X9-62(10045) curves(3) prime(1) prime256v1(7)}.

Optionally the following curves {OID} may be supported for ECDSA:

– `secp384r1`: {iso(1) identified-organization(3) certicom(132) curve(0) ansip384r1(34)}
– `secp521r1`: {iso(1) identified-organization(3) certicom(132) curve(0) ansip521r1(35)}
– `brainpoolP256r1`: {iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP256r1(7)}
– `brainpoolP384r1`: {iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP384r1(11)}
– `brainpoolP512r1`: {iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP512r1(13)}

SHA-256 shall be supported and is the preferred hash algorithm to be used.

The support of SHA-1 is deprecated. Its use is limited to backward compatibility. SHA-1 is no longer recognized as secure with respect to collision resistance and it is therefore strongly

recommended to perform a risk assessment before using this algorithm. If SHA-256 cannot be used, it is also recommended that additional security measures be taken.

NOTE 2 IETF released RFC 9155 deprecating MD5 and SHA-1 signature hashes in TLSv1.2.

The detection of SHA-1 usage shall raise a security event ("warning: deprecated hash algorithm used").

Security policies of entities shall be able to disallow or prevent any optional hashing algorithms being supported.

NOTE 3 Recommendations regarding hash signature algorithms are reviewed constantly and can be found in NIST SP800-57, BSI TR 02102-1, or the NSA Suite B.

To signal the supported signature algorithms, the "`signature_algorithm`" extension as described in 7.5.4 shall be used.

7.4.4 Session resumption

Session resumption in TLSv1.2 allows resuming or rekeying a session based on the session ID connected with a dedicated (existing) pre-master secret, which will result in a new session key. This minimizes the performance impact of asymmetric handshakes and can be done during a running session or after a session has ended within a defined time period.

TLS suggests not more than 24 hours in RFC 5246 for `sessionID` lifetime. This document follows this suggestion. Session resumption shall be performed in regular intervals for active sessions. For ended sessions, the session may resume not later than 24 hours (depending on the `sessionID` lifetime). The actual parameters should be defined based on risk assessment by the operator.

If session resumption is performed based on the session identifier (as specified in RFC 5246) and the `sessionID` has expired, the TLS client and server perform a full handshake as per section 7.3 of RFC 5246. In this case a security event shall be raised ("notice: Session could not be resumed `sessionID` lifetime expired. Performing a full TLS handshake instead)").

Session resumption of active sessions to update the session key is expected to be more frequent than session renegotiation. For this reason, the session resumption interval is smaller than the session renegotiation interval. The following formula provides a recommendation for setting the respective values: 0 < session resumption interval < session renegotiation interval <= 24h.

Implementations claiming conformance to this document shall support the configuration of session resumption intervals for active sessions. A default value for session resumption interval should be 6 hours.

NOTE 1 For reasoning of the default values see also the informative example in 7.4.5.

Clients shall initiate session resumption using the `ClientHello` message.

NOTE 2 Former editions of this document allowed the server to send a `HelloRequest` message to trigger a server-initiated session resumption in TLSv1.2. This is not possible according to RFC 5246 as the `HelloRequest` is only intended to trigger a session renegotiation, and has therefore been deleted.

Session resumption may be initiated by the client, as long as the security policies for both the client and the server permit this. In case of failures to resume a session, the failure handling described in TLSv1.2 shall be followed. Session resumption may be done based on the session identifier (native TLS according to RFC 5246). Alternatively, session resumption may be done based on session tickets (RFC 5077). The latter option allows for avoiding server-side state for sessions, which can be resumed. This option may apply for constrained devices to avoid a larger session cache.

In the ticket-based TLS session resumption approach, TLS server generates a session ticket. This ticket contains session context information in an encrypted form, allowing the TLS server to rebuild the formerly closed session or to establish a new session based on key material of the existing session. The session context is encrypted under a ticket key known only to the TLS server. This ticket key should be renewed regularly based on the organization's security policy. This is an operation related recommendation that has no influence on interoperability. To align with the allowed session resumption time the maximum lifetime of issued tickets shall be 24 hours.

NOTE 3 Application of session tickets to avoid the session specific storage on the server side provides the benefit in environments that tear down a connection and reconnect after a specific time. If session resumption is used to update the session key of an ongoing session, there may be no benefit.

If TLS session resumption is utilized for the same TLS connection, the TLS session renegotiation extension shall be used as outlined in 7.5.2.

Note that if session resumption is performed to re-establish a previously closed TLS session, the certificate, used in the establishment of the original session, is not validated during the session resumption. This is handled in a similar approach to the regular validation of certificates after the defined period as outlined in 6.4.4. Note that an implementation may need to store the peer certificates to be able to do the certificate validation.

7.4.5 Session renegotiation

Session renegotiation in TLSv1.2 initiates a complete TLS handshake where all asymmetric operations during the key exchange, including the certificate validation, must be performed. As outlined in 6.4.4.4 and 6.4.4.5, revoked or expired certificates shall not be used in session renegotiation.

NOTE 1 If an application verifies the certificate during an ongoing session resulting in the detection of an expired or a revoked certificate, the time to the next TLS session renegotiation may provide a grace period to take actions (e.g., provisioning of a valid credential or performing a security check of the related IED). This is typically handled by an organization's security policy.

Session renegotiation will result in a new session based upon both a freshly negotiated master key and new session keys. As this involves specifically the certificate validation the timeframe for session renegotiation should be chosen in accordance with the refresh of the revocation state information (CRL) as described in 6.4.4.4.

Session renegotiation intervals shall be configurable so long as they are within the specified maximum time period and shall be aligned with the CRL update period. If OCSP is used for certificate revocation checks, session renegotiation shall be aligned with the OCSP response cache time. In any case, for long lasting connections, renegotiation shall be performed at least every 24 hours to enforce a certificate validity check. Shorter intervals may be defined by a referencing standard.

NOTE 2 An example alignment is ½ CRL refresh time or ½ OCSP response caching time to limit the possibility of undetected revoked certificates.

Implementations claiming conformance to this standard shall support the configuration of session renegotiation intervals. TLS session shall be renegotiated within a maximum time period of 24 hours. A default value for session renegotiation should be 12 hours.

The lower bound for the session renegotiation supported by an implementation shall be 10 minutes.

NOTE 3 This requirement protects client and server performance by preventing too many full TLS handshakes and associated public-key certificate validations.

Session renegotiation may be initiated by either side, so long as both the TLS client and TLS server are allowed to use this feature by their security policy. In case of failures to renegotiate a session, the failure handling described in TLSv1.2 shall be followed.

The calling entity (TLS client) and the called entity (TLS server) are responsible for verifying that the TLS session renegotiation takes place at the expected intervals. If the calling entity does not receive a TLS session renegotiation request (`HelloRequest`) from the called entity at the expected interval, then the calling entity shall initiate the TLS renegotiation itself using a `ClientHello`. If the called entity does not receive a TLS renegotiation (`ClientHello`) in response to a `HelloRequest`, the called entity shall terminate the connection. The termination of a connection due to a missed TLS session renegotiation should raise a security event ("alarm: session renegotiation interval expired").

Implementations claiming conformance to this document shall support the processing of the `HelloRequest`.

NOTE 4 According to RFC 5246 the `HelloRequest` is an optional message that the server may sent to a client.

NOTE 5 It is expected that client and server are configured with the same TLS security policy.

To avoid weaknesses in session renegotiation, the session renegotiation extension defined in RFC 5746 shall be used as outlined in 7.5.1.

Note, the following informative example is provided to configure session renegotiation and session resumption:

– The assumed CRL refresh time (or OCSP response validity) is 24h.
– Session renegotiation involves the validation of the peer certificates including the revocation check. Session renegotiation involving the peer certificates for authentication may be performed at least every 12 hours.
– To allow for a session key update during the 12-hour session renegotiation interval session resumption is performed every 6 hours during the session. The maximum time to resume a previously ended session is 24 hours.

7.5 Support of extensions

7.5.1 General

This subclause defines extensions to the TLS handshake, which are mandatory or optional to be supported by conformant implementations. Each subclause references the specification for the considered extension.

7.5.2 TLS session renegotiation extension

The definition of TLSv1.2 and below did not consider providing a cryptographic binding between an initial session and a renegotiated session. This missing binding may allow an attacker to inject content in an ongoing TLS connection establishment (also known as triple handshake attack). To address this attack, RFC 5746 defines a renegotiation extension, which binds the renegotiated TLS session to the TLS session, in which the renegotiation is performed. Conformant implementations must support the session renegotiation extension.

According to RFC 5746 a TLS client includes the `"renegotiation_info"` extension in the initial `ClientHello` message. During the initial handshake this extension is empty and will signal the client's support for secure renegotiation. Likewise, the TLS server includes an empty `"renegotiation_info"` extension in the `ServerHello` message if it detected the information from the client and if it supports session renegotiation. In the context of this standard, the support of this extension is mandatory. If either side detects the absence of the `renegotiation_info` in the initial handshake, it shall raise a security event ("warning: secure session renegotiation not supported (initial handshake)"). Session renegotiation without using this extension must not be performed.

According to RFC 5746 a TLS client includes the client verification data `"client_verify_data"` sent in the `Finish` message of the previous handshake into the

renegotiation_info extension contained in the renegotiated ClientHello message. The TLS server will include the "client_verify_data" and in addition the "server_verify_data" in the extension, when sending the corresponding ServerHello message. If either side detects the absence of the renegotiation_info or one of its components in the renegotiated handshake, for which the support was signalled in the initial handshake, it shall raise a security event ("alarm: secure session renegotiation not supported (renegotiated handshake)"). The session shall be aborted. Implementations claiming conformance to this specification shall support this extension according to RFC 5746.

According to RFC 5746 section 3.1 session renegotiation extension is applicable for full handshakes and resumed handshakes in the same TLS connection. Thus, the parameter client_verify_data and the server_verify_data are connection specific and are not part of the TLS session cache. If TLS session resumption is done for a different TLS connection the TLS session renegotiation extension cannot be used. For TLS session resumption of the same TLS connection, it shall be used.

NOTE The usage of extensions is typically done by the TLS stack.

7.5.3 Signalling of client supported CA certificates via Trusted CA

The Trusted CA Indication extension specified in RFC 6066 (section 6) allows a TLS client to provide information about locally supported CA certificates since the root CA of the utilities may not be public. The extension allows the requesting party to influence the selection of the X.509 certificate on the IED side for the server-side authentication to enable the verification of the used X.509 certificate on the requestor side.

The Trusted CA Indication is contained in the ClientHello message. The "extension_data" field of this extension shall be empty. A TLS server receiving a Trusted CA Indication may use this information to guide its selection of an appropriate certificate chain to return to the client. According to RFC 6066 in this event, the server shall include an extension of type "trusted_ca_keys" in the (extended) ServerHello message.

Implementations claiming conformance to this standard should support this extension.

The support of this extension may be applicable in scenarios where IEDs are accessed by different administrative domains, e.g., two utilities with a separated public key infrastructure. If different administrative domains are to be supported, the TLS Trusted CA Indication extension should be used.

Implementations claiming conformance to this standard using this extension may specify the selection of the requested CA issued certificates on the TLS server side.

The failure of selecting a matching CA issued certificate shall raise a security event ("alarm: matching CA certificate supported on TLSv1.2 client not found").

7.5.4 Signalling of supported signature algorithms

RFC 5246 defines the signature algorithm extension to allow a client to indicate which hash and signature combinations may be used in digital signatures for certificates and TLS handshake operations.

Clients using TLSv1.2 shall include the signature_algorithm extension with at least the combination of {SHA-1, RSA; SHA-256, RSA; SHA-256, ECDSA} in the ClientHello message. The combination SHA-1 and RSA is for backward compatibility only (see also below) and depends on an organization's security policy (see also 7.4.3). Other combinations may be supported.

NOTE If the signature_algorithm is not included, TLSv1.2 server will fall back to default, which results in always applying SHA-1 for hashing.

The detection of a missing `signature_algorithm` extension shall raise a security event ("warning: Signature algorithm extension missing").

SHA-256 shall be supported and is the preferred hash algorithm to be used.

The support of SHA-1 is deprecated. Its use is limited to backward compatibility. SHA-1 is no longer recognized as secure with respect to collision resistance and it is therefore strongly recommended to perform a risk assessment before using this algorithm. If SHA-256 cannot be used, it is also recommended that additional security measures be taken. The usage of SHA-1 will be disallowed in the next edition of this standard.

The detection of a `signature_algorithm` combination {SHA-1, RSA} shall raise a security event ("warning: deprecated signature algorithm signalled").

Security policies of entities shall be able to disallow or prevent any optional hashing algorithms being supported.

NOTE Recommendations regarding hash signature algorithms are reviewed constantly and can be found in NIST SP800-57, BSI TR 02102-1, or the NSA Suite B.

Implementation claiming conformance to this specification shall support this extension according to RFC 6066.

7.5.5 Stapling of OCSP response messages

7.5.5.1 General

OCSP response stapling allows a server to provide a client with the revocation information of the server side certificates as part of the TLS handshake. The usage of this mechanism may be helpful if the client is not able to retrieve revocation information via CRL or OCSP.

For TLSv1.2 there exist two different extensions, which may be optionally supported and are outlined in 7.5.5.2 and 7.5.5.3. For the specification of the mechanisms the corresponding documents are referenced.

7.5.5.2 OCSP Stapling

RFC 6066 (section 8) defines stapling of the OCSP response for the server-side certificate.

If a TLS client wants to receive an OCSP response it may include the "`status_request`" extension in the `ExtensionType` of the `ClientHello` message. The data in the extension is the "`CertificateStatusRequest`", containing the status type as "`OCSPStatusRequest`" and a list of OCSP responders that the client trusts. If this list is empty, the OCSP responder is implicitly known by the server. There are also further options for the TLS client like including a nonce in the extension, which is used by the OCSP client to query the revocation state.

A TLS server receiving a `ClientHello` message with a "`status_request`" extension may return the OCSP response as "`OCSPResponse`" in the `CertificateStatus` message, which is a new handshake message defined by RFC 6066. This message immediately follows the "`Certificate`" message send by the TLS server.

An implementation claiming conformance to this document shall support the status_request_v2 extension according to RFC 6066.

Note that this does not necessarily require support of OCSP interaction with an OCSP responder.

7.5.5.3 OCSP Multistapling

While the OCSP stapling defined in RFC 6066 provides an option to provide an OCSP response for the server certificate, RFC 6961 defines multistapling of OCSP responses. This also supports the provisioning of OCSP responses for intermediate CAs in the certificate chain. This is achieved by providing additional information structures as well as enhancements to the `ExtensionTypes` and `CertificateStatusTypes` originally defined in RFC 6066.

If a TLS client wants to receive multiple OCSP responses it may include the "`status_request_v2`" extension in the `ExtensionType` of the `ClientHello` message. The data in the extension is the "`CertificateStatusRequestItemV2`", containing the status type as "`OCSPStatusRequest`" and a list of OCSP responders that the client trusts. If this list is empty, the OCSP responder is implicitly known by the server. There are also further options for the client like including a nonce in the extension, which is used by the OCSP client to query the revocation state.

A TLS server receiving a `ClientHello` message with a "`status_request_v2`" extension may return the OCSP responses as ""`OCSPResponseList`" in the `CertificateStatus` message, which was originally defined by RFC 6066. This message directly follows the "`Certificate`" message send by the TLS server.

An implementation should support this extension according to RFC 6961.

Note that this does not necessarily require support of OCSP interaction with an OCSP responder.

7.5.6 Signalling of intended target TLS server via Server Name Indication

RFC 6066 (section 3) defines the "`server_name`" extension, which can be used by a TLS client in a `ClientHello` message. This extension may be used by the TLS client to facilitate situations, in which multiple (virtual) TLS servers are hosted at a single network address. Signalling the intended target server name allows the server to select the appropriate certificate to return to the client or to handle other aspects concerned by a security policy.

An implementation claiming conformance to this document should support this extension according to RFC 6066.

7.5.7 Support of encryption before authentication

In TLSv1.2 the default approach for record layer security is MAC-then-encrypt. As attacks on CBC cipher suites are known (e.g., LUCKY13, POODLE), which leverage the order of the operations, an extension has been defined to reverse the order to first encrypt and then authenticate.

RFC 7366 defines the "`encrypt_then_mac`" extension, which can be used by a TLS client in a `ClientHello` message to signal the support. Likewise, the TLS server also includes the extension into the `ServerHello` message.

As this document also specifies support of CBC-based cipher suites in 7.2, implementations shall support this extension as specified in RFC 7366.

If the "`encrypt_then_mac`" extension is not included in the `ClientHello` or in the `ServerHello` messages, while a TLS cipher suite (`CipherSuite`) containing an encryption algorithm in CBC mode is signalled in the supported `CipherSuite` (`ClientHello`) or the selected cipher suite (`ServerHello`), a security event shall be provided ("alarm: TLS session with MAC-then-encrypt detected.") and the session shall be terminated.

8 Requirements specific to TLSv1.3

8.1 General

This clause describes the mandatory to support cipher suites, key exchange approaches and settings as profile for TLSv1.3 to secure power system automation communication.

8.2 Supported cipher suites

For TLSv1.3 this document specifies support of the mandatory and optionally supported cipher suites in Table 2.

Table 2 – Support of cipher suites for TLSv1.3

Cipher suite	IANA Value	Source	Support (Client/Server)
TLS_AES_128_GCM_SHA256	0x13, 0x01	RFC8446	m
TLS_AES_256_GCM_SHA384	0x13, 0x02	RFC8446	m
TLS_CHACHA20_POLY1305_SHA256	0x13, 0x03	RFC8446	o
TLS_AES_128_CCM_SHA256	0x13, 0x04	RFC8446	m
TLS_AES_128_CCM_8_SHA256	0x13, 0x05	RFC8446	o
TLS_SHA256_SHA256	0xC0, 0xB4	RFC 9150	c1
TLS_SHA384_SHA384	0xC0, 0xB5	RFC 9150	c1
c1: may be supported if integrity only protection is desired. These cipher suites shall be disabled by default and require distinct enabling authorized by an organization's security policy.			

Any cipher suite that does not specify an AEAD cipher for encryption shall not be used for communication outside the administrative domain if the encryption of this communication connection by other means cannot be guaranteed.

NOTE 1 This document does not exclude the use of encrypted communications through the use of cryptographic VPN tunnels. The use of such VPNs is out-of-scope of this standard.

Within an administrative domain, the use of non-encrypting cipher suites may be allowed, but its use is in the responsibility of the operator.

NOTE 2 The application of non-encrypting cipher suites allows for traffic inspection, while still retaining an end-to-end authentication and integrity protection of the traffic.

Implementations allowing TLS cipher suites without encryption claiming conformance to this document shall provide a mechanism to explicitly enable those TLS cipher suites. By default, non-encrypting TLS cipher suites shall be disabled.

8.3 Key exchange

8.3.1 General

The requirements from RFC 8446 regarding the mandatory support of signature algorithms for handshake messages and certificates have been followed here. This relates specifically to

– Mandatory support of digital signatures with rsa_pkcs1_sha256 (for certificates), rsa_pss_rsae_sha256 (for `CertificateVerify` message and certificates), and ecdsa_secp256r1_sha256.

– Mandatory support of key exchange based on the elliptic curve secp256r1.

To renew session keys, TLSv1.3 supports different new mechanisms in comparison with TLSv1.2:

– A new message `KeyUpdate` has been defined as post handshake message for updating the sender's cryptographic key material (keys and initialization vectors). This functionality is utilized to update the session key from either side during the session. It was accomplished in TLSv1.2 using session resumption (to only update the session keys) and with session renegotiation (to update the session keys and also to validate the certificates of both peers).
– New pre-shared key handshake modes allow the utilization of PSKs established during the last TLS session or provided by external means to negotiate new session keys without involving certificates in the context of session setup. This is comparable to the session resumption without server-side state (based on tickets) in TLSv1.2.
– Session renegotiation is forbidden according to RFC 8446. For TLSv1.2 session renegotiation is used in IEC 62351-3 to enforce a certificate-based authentication of both sides after a certain time, which is to be aligned with the update of certificate revocation information. In contrast to using TLSv1.2, the invocation of the certificate validation has to be handled by the application specifically for long lasting sessions (longer than the CRL refresh period) as outlined in 6.4.4.4.

8.3.2 Handshake modes

TLSv1.3 supports different handshake modes. Diffie Hellman Key Exchanges (DHE, ECDHE) are mandatory to support and are used by default.

TLSv1.3 also supports PSK modes as listed in Table 3 to enable a faster session setup by supporting session resumption and/or the establishment of parallel TLS connections.

Independent of the utilized handshake mode, the provisioning of encrypted application data during the TLS handshake (until the TLS client finished message is received by the TLS server) shall not be used. This also aligns with RFC 8446, section 2.

Note that for TLSv1.2, IEC 62351 utilizes session resumption to enforce a session key update or to establish a second TLS connection, which is bound to the full handshake of the initial connection. In TLSv1.3 there exists a distinct mechanism for the session key update during an ongoing TLS session, which can be initiated by either side. Therefore, session resumption with TLSv1.3 is only intended for establishing another secured connection, which is bound to the initial TLS session between the same peers.

Table 3 – Support of PSK-based handshake modes for TLSv1.3

Handshake modes	Reference	IANA Value	Support (Client/Server)
psk_ke	RFC8446	0	x
psk_dhe_ke	RFC8446	1	m
NOTE when using psk_ke, no perfect forward secrecy is provided.			

Based on the PSK, TLSv1.3 enables a 0-RTT handshake, which allows the transport of encrypted data (indicated by the `early_data` extension) with the first message (`ClientHello`). As the PSK is provided either out-of-band or is the result of a previous session, perfect forward secrecy cannot be provided. In the context of this specification 0-RTT shall not be used.

To ensure perfect forward secrecy for the session key to be setup `psk_dhe_ke` shall be supported by implementations claiming conformance to this document.

The option to only use `psk_ke` does not require the server to provide a key share and does not result in perfect forward secrecy properties for the agreed session key. In the context of this specification the handshake mode `psk_ke` shall not be used.

If a TLS client signals support of `psk_ke` only, a security event shall be provided ("alarm: support of non-ephemeral PSK mode only"). The session shall be terminated.

8.3.3 Diffie-Hellman Groups

TLSv1.3 allows the communication peers to specify the supported groups for (EC)DHE. In the context of this document, the Diffie-Hellman groups marked as mandatory in Table 4 shall be supported. Optional groups may be supported.

Table 4 – Support of Diffie Hellman Groups for TLSv1.3

Diffie Hellman Group	Reference	IANA Value	Support (Client/Server)
secp256r1	RFC8422	23	m
secp384r1	RFC8422	24	o
brainpoolP256r1	RFC8734	31	o
brainpoolP384r1	RFC8734	32	o
brainpoolP512r1	RFC8734	33	o
ffdhe2048	RFC7919	256	m
ffdhe3072	RFC7919	257	o
ffdhe4096	RFC7919	258	o

8.3.4 Signature algorithms

8.3.4.1 Signature algorithms for handshake messages

When compared with TLSv1.2, TLSv1.3 allows to indicate the desired signature algorithms more specifically than TLSv1.2. Here two extensions are offered (see also 8.3.4.2). The `signature_algorithm` extension specifies which algorithms are supported and to be used in the `CertificateVerify` message. The algorithms given in Table 5 are to be supported mandatorily and optionally for use with `signature_algorithm` in the context of this document.

Table 5 – Supported signature algorithms for the handshake in TLSv1.3

Signature algorithms	Reference	IANA Value	Support (Client/Server))
rsa_pss_rsae_sha256	RFC8446	0x0804	m
rsa_pss_rsae_sha384	RFC8446	0x0805	o
rsa_pss_rsae_sha512	RFC8446	0x0806	o
rsa_pss_pss_sha256	RFC8446	0x0809	m
rsa_pss_pss_sha384	RFC8446	0x080A	o
rsa_pss_pss_sha512	RFC8446	0x080B	o
ecdsa_secp256r1_sha256	RFC8446	0x0403	m
ecdsa_secp384r1_sha384	RFC8446	0x0503	o
ecdsa_brainpoolP256r1_sha256	RFC8734	0x081A	o
ecdsa_brainpoolP384r1_sha384	RFC8734	0x081B	o
ecdsa_brainpoolP512r1_sha512	RFC8734	0x081C	o

8.3.4.2 Signature algorithms for certificates

TLS1.3 allows to indicate the desired signature algorithms to be used for signatures in certificates in the `signature_algorithms_cert` extension. According to RFC 8446, if no "`signature_algorithms_cert`" extension is included, the "`signature_algorithms`" extension (see 8.3.4.1) also applies to signatures in certificates. The algorithms shown in Table 6 shall be supported mandatorily and optionally for use with `signature_algorithms_cert`:

Table 6 – Supported signature algorithms for the certificates in TLSv1.3

Signature algorithms	Reference	IANA Value	Support (Client/Server)
rsa_pkcs1_sha256	RFC8446	0x0401	m
rsa_pkcs1_sha384	RFC8446	0x0501	o
rsa_pkcs1_sha512	RFC8446	0x0601	o
rsa_pss_rsae_sha256	RFC8446	0x0804	m
rsa_pss_rsae_sha384	RFC8446	0x0805	o
rsa_pss_rsae_sha512	RFC8446	0x0806	o
rsa_pss_pss_sha256	RFC8446	0x0809	m
rsa_pss_pss_sha384	RFC8446	0x080A	o
rsa_pss_pss_sha512	RFC8446	0x080B	o
ecdsa_secp256r1_sha256	RFC8446	0x0403	m
ecdsa_secp384r1_sha384	RFC8446	0x0503	o
ecdsa_brainpoolP256r1_sha256	RFC8734	0x081A	o
ecdsa_brainpoolP384r1_sha384	RFC8734	0x081B	o
ecdsa_brainpoolP512r1_sha512	RFC8734	0x081C	o
NOTE The German BSI recommends using RSA (IANA values 0x0401, 0x0501, 0x0601) only until 2025 as problems with the padding in the utilized PKCS#1 are known.			

Subclauses 8.8.4 and 8.8.4.2 define the usage of the related extensions.

8.4 Session key update (post-handshake message)

TLSv1.3 allows the update of the sending key material (keys and initialization vectors) by either side of the connection through the use of a post-handshake message.

Implementations claiming conformance to this document shall support the configuration of key update intervals for active sessions. A default value for key update intervals should be 6 hours.

The update of the keys of the sending side shall be performed using the `KeyUpdate` post handshake message as described in RFC 8446. As this only updates the key material for the sending side, the sender shall include an "`update_requested`" into the `KeyUpdate` structure to force the receiver to also update its sending key material. This will result in a complete renewal of the session key material. The receiver of the `KeyUpdate` shall renew his sending key material in the same way and include an "`update_not_requested`" in his `KeyUpdate` message to avoid immediate rekeying of the initiator.

Both entities (TLS client, TLS server) are responsible for verifying that the TLS session key update is performed as defined by the security policy. If either side does not receive a TLS `KeyUpdate` from the called entity in response to an own `KeyUpdate` the calling entity shall terminate the connection. The termination of a connection due to a missed TLS session key update shall raise a security event ("alarm: session key update interval expired").

8.5 New session ticket (post-handshake message)

Another post handshake message, `NewSessionTicket` (see also RFC 8446, section 4.6.1) supports session resumption by providing a session-ticket.

Implementations conforming to this document shall support the post-handshake message `NewSessionTicket` as specified in RFC 8446.

The client shall use this session-ticket for future handshakes by including the ticket value in the `pre_shared_key` extension (8.8.9) in the `ClientHello` for a resumed session (see 8.6).

In the ticket-based TLS session resumption approach, TLS server generates a session ticket, where this ticket contains session context information in an encrypted form, allowing the TLS server to rebuild the formerly closed session. The session context is encrypted under a ticket key known only to the TLS server. This ticket key should be renewed regularly based on the organization's security policy. This is an operation related recommendation that has no influence on interoperability.

8.6 Session resumption

Session resumption in TLSv1.3 uses the session-ticket established after the initial handshake using the `NewSessionTicket` message (see 8.5) to enable either the resumption of a previous session or to establish an additional session between the connected entities. Session resumption may be performed during an active session to establish additional connections.

NOTE According to RFC 8446, section 4.1.2 session resumption cannot be performed in the context of an active session as a `ClientHello` message after the TLS handshake will lead to a session termination.

Session tickets for session resumption shall be issued with a maximum lifetime of 24 hours. Session resumption for already terminated connections may be performed based on the remaining validity time of the session ticket. The used parameter for resuming terminated sessions shall be defined based on risk assessment by the operator.

Implementations conforming to this document shall support session resumption.

If session resumption is used it shall be performed using the `psk_dhe_ke` handshake mode as outlined in RFC 8446, section 4.2.9.

Note that if session resumption is performed to re-establish a previously closed TLS session, the certificate, used in the establishment of the original session, is not validated during the session resumption. This is handled in a similar approach to the regular validation of certificates after the defined period as outlined in 6.4.4. Note that an implementation may need to store the peer certificates to be able to do the certificate verification.

8.7 Certificate validation

Session renegotiation is forbidden in TLSv1.3. RFC 8446 section 4.1.2 explicitly states that if a server receives a `ClientHello` in an already established TLSv1.3 connection, it must terminate this connection. Session renegotiation is used in TLSv1.2 (see also 7.4.5) to update the session key and to invoke the certificate validation of both sides.

As stated in 6.4.4.4, support is required to check the revocation state of received certificates at a configurable time interval. In contrast to using TLSv1.2, the invocation of the certificate validation has to be handled by the application specifically for long lasting sessions (longer than the CRL refresh period).

8.8 Support of extensions

8.8.1 General

This clause defines extensions to the TLS handshake. TLSv1.3 specifies the mandatory to implement extensions if no application profile standard is available in section 4.2. IEC 62351-3 specifies such a TLS application profile and defines the mandatory to support extensions by taking the TLSv1.3 required extensions as well as additional extensions into account.

This is described in 8.8.2, 8.8.3, 8.8.4, 8.8.4.2, 8.8.5, 8.8.6, 8.8.7, 8.8.8, 8.8.9, and 8.8.10 for the context of power system application. Each subclause references the specification for the considered extension.

8.8.2 Signalling of supported TLS versions

RFC 8446 section 4.2.1 specifies that the "`supported_versions`" extension in the `ClientHello` indicates which versions of TLS the client supports. In the `ServerHello` message it indicates which version the server will use. The extension contains a list of TLS versions supported in the order of preference.

Implementations claiming conformance to this document and supporting TLSv1.3 shall send the TLSv1.3 identifier `0x0304` in this extension. They may also include lower TLS version identifiers if the security policy also permits the usage of previous versions of TLS as outlined in 6.2.

According to RFC 8446 the "`supported_versions`" extension is required in all `ClientHello`, `ServerHello`, and `HelloRetryRequest` messages.

8.8.3 Cookie

RFC 8446 section 4.2.2 specifies the "`cookie`" extension in the absence of an application profile as mandatory to be supported by implementations. It is used by the TLS server in the `HelloRetryRequest` message and reflected by the TLS client in the `ClientHello` message. It serves for two main purposes:

– Showing that a TLS client is reachable under a given network address, which is useful for non-connection-oriented transports.

– An option to offload state to the TLS client.

In the context of this specification the usage of the cookie extension is not mandated, as the protection targets connection-oriented transports and the offloading of state is addressed by supporting session-tickets during session resumption. The session-tickets are provided to the TLS client through the use of the post handshake message `NewSessionTicket` as described in 8.5.

8.8.4 Signalling of supported signature algorithms

8.8.4.1 Signalling of supported signature algorithms for handshake messages

RFC 8446 section 4.2.3 specifies the "`signature_algorithm`" extension. It applies to signatures in `CertificateVerify` messages.

Implementations claiming conformance to this document shall support this extension as defined in RFC 8446. TLS clients using TLSv1.3 shall include the `signature_algorithm` extension with at least the mandatory algorithms listed in Table 5.

8.8.4.2 Signalling of supported signature algorithms for certificates

RFC 8446 specifies in section 4.2.3 the "`signature_algorithm_cert`" extension. It applies to signatures in certificates. While RFC 8446 states that if this extension is not provided by the client, the listed algorithms from the "`signature_algorithm`" extension are used. As the `signature_algorithm_cert` extension also supports PKCS1 versions, this specification requires support of this extension.

Implementations claiming conformance to this specification shall support this extension as defined in RFC 8446. TLS clients using TLSv1.3 shall include the `signature_algorithm_cert` extension with at least the mandatory algorithms listed in Table 6.

8.8.5 Signalling of supported groups

RFC 8446 specifies in section 4.2.7 the "`supported_groups`" extension. It is used by the TLS client to indicate the named groups (for elliptic curves and for finite fields), which the client supports for key exchange. They are ordered according to the preferences.

Implementations claiming conformance to this document shall support this extension as defined in RFC 8446. TLS clients using TLSv1.3 shall include the `supported_groups` extension with the mandatory and the chosen optionally supported groups listed in Table 4.

8.8.6 Signalling of key share

RFC 8446 specifies in section 4.2.8 the "`key_share`" extension. This extension transports the cryptographic parameters for the key exchange (ECDHE and DHE proposed by the endpoint). The extension allows to support different parameter sets. If multiple parameter sets are offered, the TLS server will determine the final set to be used.

The `key_share` extension is a structure containing

– A named group, which shall be at least one of the mandatory groups listed in Table 4.

– The key exchange parameter corresponding to the selected group. RFC 8446 describes the parameter sets to be provided for finite-field Diffie Hellman groups (RFC 8446, section 4.2.8.1) as well as for elliptic-curve Diffie Hellman groups (RFC 8446, section 4.2.8.2).

Implementations claiming conformance to this document shall support this extension as defined in RFC 8446. TLS clients using TLSv1.3 shall include the `key_share` extension with at least on selected mandatory group listed in Table 4. They may list multiple mandatory or optional groups.

8.8.7 Signalling of intended target TLS server via Server Name Indication

RFC 8446 relies for the "`server_name`" extension on the specification in RFC 6066. As for TLSv1.2 the extension can be used by a TLS client in a `ClientHello` message. It may be used by the TLS client to facilitate situations, in which multiple (virtual) TLS servers are hosted at a single network address. Signalling the intended target server name allows the server to select the appropriate certificate to return to the client or to handle other aspects concerned by a security policy.

In the absences of an application profile, RFC 8446 requires this extension to be mandatory supported by implementations.

An implementation claiming conformance to this document should support this extension according to RFC 6066.

8.8.8 Signalling of supported certificate authorities

In contrast to the `trusted_CA` extension described for TLSv1.2 clients as outlined in 7.5.3, TLSv1.3 specifies the "`certificate_authorities`" extension in RFC 8446 section 4.2.4. This extension can be used by either endpoint to signal the supported CA as guidance for the receiving endpoint.

Implementations claiming conformance to this document should support this extension. Implementations using this extension may specify the selection of the requested CA issued certificates on the TLS endpoint.

The failure of selecting a matching CA issued certificate shall raise a security event ("alarm: matching CA certificate supported on TLSv1.3 peer not found").

8.8.9 Support of PSK based key agreement

RFC 8446 also supports PSK based key agreement, which is useful to either resume a terminated session or to establish a second TLS session based on an existing session. RFC 8446 defines different modes for the PSK key agreement.

According to RFC 8446, if implementations support PSK key agreement, they must support the following two extensions:

– "`pre_shared_key`" to negotiate the identity of the pre-shared key to be used with a given handshake in association with PSK key establishment

– "`psk_key_exchange_modes`" to signal the PSK mode to be used. The modes are outlined in 8.3.2.

According to RFC 8446, all implementations must support these extensions when offering PSK based key agreement. Implementations claiming conformance to this document shall support PSK-based key agreement by the two extensions stated above. Implementations using this extension shall offer only the `psk_dhe_ke` key exchange mode as outlined in RFC 8446, section 4.2.9, to ensure perfect forward secrecy for the session key to be negotiated.

8.8.10 Stapling of OCSP response messages

RFC 8446 (in contrast to TLSv1.2) defines in section 4.4.2.1 OCSP response stapling for the endpoint certificate and also for intermediate CA certificates for both endpoints (TLS client and TLS server). This allows both endpoints to provide the revocation information of the utilized certificates (and chains) as part of the TLS handshake. The usage of this mechanism may be helpful if one endpoint is not able to retrieve fresh revocation information.

In TLSv1.3 the handling is as follows:

– The OCSP information about the TLS server certificate is carried in a `status_request` extension (as defined in RFC 6066) of the `CertificateEntry`. The extension must contain the `CertificateStatus` structure.

– If the TLS server requests the client to send OCSP response information, it provides an empty `status_request` extension in its `CertificateRequest` message. If the client is willing (and able) to provide this information, it includes the `status_request` extension (as defined in RFC 6066) containing the `CertificateStatus` structure in its response

An implementation claiming conformance to this document shall support this extension according to the requirements in RFC 8446 (section 4.4.2.1) relying on RFC 6066.

Note that this does not necessarily require support of OCSP interaction with an OCSP responder.

8.8.11 Signalling of early data

RFC 8446 section 4.2.10 defines an early data indication, which allows a client using a PSK to send already encrypted data in the handshake messages. As this document does not allow 0-RTT (see also 8.3.2), this extension shall not be used by implementations conforming to this document.

The TLS server shall ignore the received early data until the handshake has finished. The detection of early data in handshake messages shall raise a security event ("warning: Early data in handshake detected but ignored").

9 Optional security measure support

In certain deployments, additional support is necessary to further restrict the usage of certificates based on their serial numbers and issuers. This restriction is known as certificate authorization list or certificate pinning. Certificate authorization list may optionally be supported. If an implementation supports certificate authorization list, the list shall be built by stating the serial number and the issuer of the authorized certificates. As this approach is not restricted to the usage of certificates in TLS, it is further specified in IEC 62351-9 in the context of Certificate Authorization and Validation lists (CertAVL, as defined in ISO/IEC 9594-8:2020 | Rec. ITU-T X.509 (2019)).

10 Conformance

10.1 General

Static conformance requirements specify what shall be implemented, what may be implemented and what shall not be implemented for an implementation claiming compliance to this document.

Note that this clause refers to settings defined in Clauses 6 and 7 for TLSv1.2 and Clauses 6 and 8 for TLSv1.3.

10.2 Notation

The following notations are used for specifying conformance requirements:

- m: Mandatory support. The item shall be implemented.
- o: Optional support. The item may be implemented.
- c: Conditional support. The item shall be implemented as specified by the condition.
- x: Excluded. The item shall not be supported.
- F/S: Functional Standard

10.3 Conformance to selected TLS versions

Table 7 states the conformance requirements for TLS versions, which are defined in 6.2.

Table 7 – Conformance to TLS versions

TLS Version	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
Prior 1.0	x		x		
1.0	c		c		Weaknesses known, only for backward compatibility
1.1	c		c		Weaknesses known, only for backward compatibility
1.2	m		m		
1.3	o		o		
c – the use of TLS versions prior to version 1.2 is deprecated.					

10.4 Conformance to certificate handling

Table 8 states the conformance requirements for certificate handling support, which are defined in Clause 6.

Table 8 – Conformance to certificate support

	Client		Server		Value/Comment	Reference
	F/S	Declared	F/S	Declared		
Support of multiple CA (root certificates)	m		m		Minimum to support 5 root CA certificates.	6.4.1
Support of certificates handling up to a maximum certificate size of 8 192 octets.	m		m			6.4.2
Follow certificate validation rules according to RFC 5280 (validity, CA signature, revocation state, etc.)	m		m			6.4.4
Certificate revocation state validation using CRL	m		m		Evaluation period at least every 24 hours	6.4.4.4.2
Certificate revocation state validation using OCSP response messages	o1		o1		Caching period at most 24 hours	6.4.4.4.3
Certificate authorization lists according to IEC 62351-9	o		o			9
o1: An implementation shall be able to validate OCSP responses.						

10.5 Conformance to TLSv1.2 specifics

10.5.1 Conformance to selected cipher suites

Table 9 states the conformance requirements for cipher suites usable with TLSv1.2, which are defined in 7.2 and 7.3.

Table 9 – Conformance to TLSv1.2 usable cipher suites

Cipher suite	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
TLS_NULL_WITH_NULL_NULL	x		x		disallowed
TLS_RSA_WITH_NULL_MD5	x		x		disallowed
TLS_*_*_MD5	x		x		disallowed
TLS_*_DES_*	x		x		disallowed
TLS_RSA_WITH_NULL_SHA256	c		c		
TLS_RSA_WITH_AES_128_CBC_SHA256	m		m		
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	m		m		
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	o		o		
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	m		m		
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	o		o		
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	m		m		
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	o		o		
c: may be supported if integrity only protection is desired. These cipher suites shall be disabled by default and require distinct enabling authorized by an organization's security policy. The usage of cipher suites containing SHA-1 as hash function is deprecated and requires explicit authorization by an organization's security policy.					

10.5.2 Conformance to cryptographic algorithm support

Table 10 states the conformance requirements for cryptographic algorithm support for certificates and handshake, which are defined in 7.4.3.

Table 10 – Conformance to cryptographic algorithm support

Signature and Hash algorithms	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
Support of RSA 2048	m		m		
Support of RSA 1024	c		c		deprecated
Support of ECDSA	m		m		
Support of curve secp256r1	m		m		
Support of curve secp384r1	o		o		
Support of curve secp521r1	o		o		
Support of curve brainpoolP256r1	o		o		
Support of curve brainpoolP384r1	o		o		
Support of curve brainpoolP512r1	o		o		
Support of SHA-256	m		m		
Support of SHA-1	c		c		deprecated
Support of MD5	x		x		disallowed
c: deprecated. Use is only intended for backward compatibility and should be authorized by an organization's security policy.					

10.5.3 Conformance to TLSv1.2 session management features

Table 11 states the conformance requirements for TLSv1.2, which are defined in 7.4.4. and 7.4.5

Table 11 – Conformance to TLSv1.2 session management features

TLSv1.2 feature	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
TLS Session resumption	m		m		
TLS Session resumption initiation using `ClientHello`	m		x		Only refers to the initiation of resumption
TLS Session resumption using session tickets	o		o		according to RFC 5077
TLS Session renegotiation	m		m		
TLS Session renegotiation initiation using `ClientHello`	m		x		Only refers to the initiation of renegotiation
TLS Session renegotiation initiation using `HelloRequest`	x		m		Only refers to the initiation of renegotiation

10.5.4 Conformance to selected TLSv1.2 extensions

Table 12 states the conformance requirements for TLS version, which are defined in 7.5.

Table 12 – Conformance to TLSv1.2 handshake extensions

TLSv1.2 extension	Client		Server		Value/Comment	Reference
	F/S	Declared	F/S	Declared		
TLS Session renegotiation extension	m		m		According to RFC 5746	7.5.2
Trusted CA indication	o		o		According to RFC 6066	7.5.3
Supported Signature Algorithms	m		m		According to RFC 5246	7.5.4
OCSP Stapling (`status_request`)	m		m		According to RFC 6066	7.5.5.2
OCSP Multi Stapling (`status_request_v2`)	o		o		According to RFC 6961	7.5.5.3
Server Name Indication	o		o		According to RFC 6066	7.5.6
Encrypt-then-MAC	m		m		According to RFC 7366	7.5.7

10.6 Conformance to TLSv1.3 specifics

10.6.1 Conformance to selected TLSv1.3 cipher suites

Table 13 states the conformance requirements for TLSv1.3 cipher suites, which are defined in 8.2.

Table 13 – Conformance to TLSv1.3 cipher suites

Cipher suite	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
TLS_AES_128_GCM_SHA256	m		m		
TLS_AES_256_GCM_SHA384	m		m		
TLS_CHACHA20_POLY1305_SHA256	o		o		
TLS_AES_128_CCM_SHA256	m		m		
TLS_AES_128_CCM_8_SHA256	o		o		
TLS_SHA256_SHA256	c1		c1		
TLS_SHA384_SHA384	c1		c1		
c1: may be supported if integrity only protection is desired. These cipher suites shall be disabled by default and require distinct enabling authorized by an organization's security policy.					

10.6.2 Conformance to selected TLSv1.3 session management features

This subclause states the conformance requirements for TLSv1.3 session management features, which are defined in 8.3, 8.4, and 8.6 and reflected in Table 14, Table 15, Table 16, Table 17 and Table 18.

Handshake Modes are defined in 8.3.2.

Table 14 – Conformance to handshake modes of TLSv1.3

TLSv1.3 handshake modes	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
psk_ke	x		x		
psk_dhe_ke	m		m		

The use of the early data option of TLSv1.3 is defined in 8.3.2.

Table 15 – Conformance to early data feature (0-RTT) of TLSv1.3

TLSv1.3 handshake modes	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
0-RTT (early data)	x		x		

Diffie-Hellman Group support is defined in 8.3.3.

Table 16 – Conformance to supported Diffie Hellman Groups in TLSv1.3

TLSv1.3 supported Diffie Hellman groups	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
secp256r1	m		m		
secp384r1	o		o		
brainpoolP256r1	o		o		
brainpoolP384r1	o		o		
brainpoolP512r1	o		o		
ffdhe2048	m		m		
ffdhe3072	o		o		
ffdhe4096	o		o		

Signature algorithms support is defined in 8.3.4.

Table 17 – Conformance to supported signature algorithms for the handshake in TLSv1.3

TLSv1.3 supported signature algorithms	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
rsa_pss_rsae_sha256	m		m		
rsa_pss_rsae_sha384	o		o		
rsa_pss_rsae_sha512	o		o		
rsa_pss_pss_sha256	m		m		
rsa_pss_pss_sha384	o		o		
rsa_pss_pss_sha512	o		o		
ecdsa_secp256r1_sha256	m		m		
ecdsa_secp384r1_sha384	o		o		
ecdsa_brainpoolP256r1_sha256	o		o		
ecdsa_brainpoolP384r1_sha384	o		o		
ecdsa_brainpoolP512r1_sha512	o		o		

Signature algorithms for certificates are defined in 8.3.4.2.

Table 18 – Conformance to supported signature algorithms for the certificates in TLSv1.3

TLSv1.3 supported signature algorithms	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
rsa_pkcs1_sha256	m		m		
rsa_pkcs1_sha384	o		o		
rsa_pkcs1_sha512	o		o		
rsa_pss_rsae_sha256	m		m		
rsa_pss_rsae_sha384	o		o		
rsa_pss_rsae_sha512	o		o		
rsa_pss_pss_sha256	m		m		
rsa_pss_pss_sha384	o		o		
rsa_pss_pss_sha512	o		o		
ecdsa_secp256r1_sha256	m		m		
ecdsa_secp384r1_sha384	o		o		
ecdsa_brainpoolP256r1_sha256	o		o		
ecdsa_brainpoolP384r1_sha384	o		o		
ecdsa_brainpoolP512r1_sha512	o		o		
NOTE The German BSI recommends using RSA (IANA values 0x0401, 0x0501, 0x0601) only until 2025 as problems with the padding in the utilized PKCS#1 are known.					

10.6.3 Conformance to selected TLSv1.3 extensions

Table 19 states the conformance requirements for TLSv1.3, which are defined in 8.8.

Table 19 – Conformance to TLSv1.3 extensions

TLSv1.3 extension	Client		Server		Value/Comment	Reference
	F/S	Declared	F/S	Declared		
Supported Versions	m		m			8.8.2
Cookie	o		o			8.8.3
Signature algorithms	m		m			8.8.4.1
Signature algorithms certificate	m		m			8.8.4.2
supported groups	m		m			8.8.5
Key share	m		m			8.8.6
Certificate Authorities	o		o			8.8.8
pre-shared-key	m		m			8.8.9
psk_key_exchange_modes (PSK mode)	m		m			8.8.9
Server Name Indication	o		o			8.8.7
status_request (OCSP Stapling)	m		m			8.8.10
early_data	x		x			8.8.11

10.6.4 Conformance to selected TLSv1.3 post-handshake messages

Post-handshake messages are defined in 8.4 and shown in Table 20.

Table 20 – Conformance to post-handshake messages of TLSv1.3

TLSv1.3 post-handshake message	Client		Server		Value/Comment
	F/S	Declared	F/S	Declared	
KeyUpdate	m		m		
NewSessionTicket	m		m		

Annex A
(informative)

Security Events

A.1 Security event logs

This annex contains the mapping of the security events defined in this document to IEC 62351-14.

The information about the security events and potential detailed information to be contained in the ExtraInfo can only be provided by the entity based on the availability of this information through the underlying platform or utilized components.

The IEC Version of all security events shall be "1".

For the security events, the following grouping is used:

– Value "1" relates to TLS handshake specific security events
– Value "2" relates to certificate handling specific security events

A.2 Mapping of TLS events related to the TLS handshake

Table A.1 – Security event logs related to TLS handshake defined in IEC 62351-3:— (Ed.2) mapped to IEC 62351-14

Security Event	Subclause (IEC 62351-3:— (Ed.2))	MNEMONIC	Severity	Event identifier	Text	Extra Info
TLS handshake successfully performed	6.1	TLS_HS_SUCCESS	notice	IEC 62351-3:1.1	TLS session successfully established.	
unsecure communication: deprecated TLS version proposed	6.2	TLS_DEPRECATED_VERSION	alarm	IEC 62351-3:1.2	Deprecated TLS version proposed and support of TLS versions prior to TLSv1.2 disabled.	Proposed TLS version number.
insecure TLS version	6.2	TLS_WEAK_VERSION	warning	IEC 62351-3:1.3	Deprecated TLS version proposed and support of TLS versions prior to TLSv1.2 enabled.	Proposed TLS version number.
unsecure communication: disallowed TLS version proposed	6.2	TLS_DISALLOWED_VERSION	alarm	IEC 62351-3:1.4	Disallowed TLS version (prior to TLSv1.0) proposed.	Proposed TLS version number.
requested TLS Version change in ongoing communication detected	6.2	TLS_VERSION_CHANGE	alarm	IEC 62351-3:1.5	TLS version change (potential downgrade) in ongoing session detected.	Proposed TLS version number.
peer certificate unavailable	6.4.3	TLS_NO_PEER_CERT	alarm	IEC 62351-3:1.6	Peer did not provide endpoint certificate during the TLS handshake.	
endpoint certificate unavailable	6.4.3	TLS_NO_LOCAL_CERT	alarm	IEC 62351-3:1.7	Local endpoint certificate not available.	
session terminated due to revoked certificate	6.4.4.4	TLS_SESSION_CLOSED-REV	alarm	IEC 62351-3:1.8	TLS session closure due to received revoked endpoint certificate.	Revocation reason.

Security Event	Subclause (IEC 62351-3:— (Ed.2))	MNEMONIC	Severity	Event identifier	Text	Extra Info
disallowed TLSv1.2 cipher suite proposed	7.3	TLS_DISALLOWED_CIPHER	warning	IEC 62351-3:1.9	Disallowed cipher suite proposed.	Proposed cipher suite.
sessionID expired, session could not be resumed	7.4.4	TLS_SESSIONID_EXPIRED_FULL_HS	Warning	IEC 62351-3:1.10	SessionID expired, Resumption not possible, full TLS handshake done.	
session renegotiation interval expired	7.4.5	TLS_NO_RENEG	alarm	IEC 62351-3:1.11	Renegotiation interval exceeded. TLSv1.2 peer does not renegotiate.	End of renegotiation time.
secure session renegotiation not supported (initial handshake)	7.5.2	TLS_NO_RENEG_SIG	warning	IEC 62351-3:1.12	TLSv1.2: Peer does not signal secure renegotiation support (session tickets) in initial handshake.	
secure session renegotiation not supported (renegotiated handshake)	7.5.2	TLS_NO_RENEG_TICKET	alarm	IEC 62351-3:1.13	TLSv1.2: Peer does not use secure renegotiation (session tickets) capability in renegotiated handshake.	
matching CA certificate supported for TLSv1.2 client not found	7.5.3	TLS_NO_TR_CA_MATCH_S	alarm	IEC 62351-3:1.14	TLSv1.2: No server-side support of client signalled trusted CA.	SKID of Trusted CA certificate.
signature algorithm extension missing	7.5.4	TLS_NO-SIG_ALGO_EXT	warning	IEC 62351-3:1.15	TLSv1.2: No signature-algorithm extension included. Fallback to deprecated signature algorithm	
deprecated signature algorithm signalled	7.5.4	TLS_DEP_SIG_ALGO	warning	IEC 62351-3:1.16	TLSv1.2: Deprecated signature algorithm combination detected.	Signalled cipher suite.
No encrypt-then-MAC extension included while negotiating CBC cipher suites.	7.5.7	TLS_NO_EPSK_MODE	alarm	IEC 62351-3:1.17	TLSv1.2: no support for encrypt-then-MAC.	
No ephemeral PSK mode proposed	8.3.2	TLS_NO_ENCRYPT-THEN-MAC	warning	IEC 62351-3:1.18	TLSv1.3: support of non-ephemeral PSK mode only	
session key update interval expired	8.4	TLS_NO_SK_UPDATE	alarm	IEC 62351-3:1.19	TLSv1.3: Peer does not perform session key update.	
matching CA certificate supported for TLSv1.3 peer not found	8.8.8	TLS_NO_TR_CA_MATCH_SC	alarm	IEC 62351-3:1.20	TLSv1.3: No support of peer signalled trusted CA.	SKID of Trusted CA certificate.
early data in handshake neglected	8.8.11	TLS_EARLY-DATA	warning	IEC 62351-3:1.21	TLSv1.3:: Early data in handshake detected but neglected.	

A.3 Mapping of TLS events related to the certificate handling

Table A.2 – Security event logs related to certificate validation defined in IEC 62351-3 Ed.2 mapped to IEC 62351-14

Security Event	Subclause (IEC 62351-3:— (Ed.2))	MNEMONIC	Severity	Event identifier	Text	Extra Info
TLS certificate size exceeded	6.4.2	TLS_CERT_SIZE_MISMATCH	alarm	IEC 62351-3:2.1	Endpoint certificate size exceeds limits.	Certificate size of received certificate.
certificate validation: CA certificate not available	6.4.4.2	TLS_NO_CA_MATCH	alarm	IEC 62351-3:2.2	Matching RootCA certificate for endpoint certificate validation not available.	Info about expected Root CA in certificate path.
certificate validation: trusted individual certificate from authorized CA not available	6.4.4.3	TLS_NO_TRUSTED_CERT_MATCH	alarm	IEC 62351-3:2.3	Endpoint certificate not in certificate trust list.	Endpoint certificate information
certificate validation: revoked certificate	6.4.4.4	TLS_CERT_REVOKED	alarm	IEC 62351-3:2.4	Endpoint certificate is revoked.	Revocation reason.
local CRL not accessible	6.4.4.4.2	TLS_NO_CRL	warning	IEC 62351-3:2.5	Local CRL not accessible.	
CRL expired	6.4.4.4.2	TLS_CRL_EXP	warning	IEC 62351-3:2.6	CRL expired.	CRL expiry info.
OCSP response expired	6.4.4.4.3	TLS_OCSP_RES_EXP	warning	IEC 62351-3:2.7	OCSP response expired.	OCSP expiry info.
OCSP responder unavailable	6.4.4.4.3	TLS_OCSP_RESP_UNAVAIL	Warning	IEC 62351-3:2.8	OCSP responder address.	
certificate validation: expired certificate	6.4.4.5	TLS_CERT_EXP	alarm	IEC 62351-3:2.8	Endpoint certificate expired.	Endpoint certificate information , Certificate expiry info.
certificate validation: signature algorithms not supported	6.4.4.6	TLS_SIG_ALG_MISMATCH	alarm	IEC 62351-3:2.9	Signature algorithms in received endpoint certificate not supported.	Proposed signature algorithm.
certificate validation: failed verification	6.4.4.6	TLS_SIG_V_FAILED	alarm	IEC 62351-3:2.10	Certificate signature could not be verified	
insufficient RSA key length	7.4.3	TLS_SHORT_RSA_KEY	alarm	IEC 62351-3:2.11	RSA key with key length of less than 2048 bit detected	Proposed RSA key length.
minimum key length	7.4.3	TLS_MIN_KEY	warning	IEC 62351-3:2.12	RSA key with minimum key length of 1024 bit detected and allowed by configuration.	Proposed RSA key length.
insufficient key length	7.4.3	TLS_SHORT_KEY	alarm	IEC 62391-3:2.13	RSA key with insufficient key length (less than 1024 bit) detected.	Proposed RSA key length.
deprecated hash algorithm used	7.4.3	TLS_DEP_HASH	warning	IEC 62351-3:2.14	Use of deprecated hash algorithm detected	Proposed hash algorithm

Bibliography

ISO/IEC 15946-2, *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures* (withdrawn)

IEC 62351-4:2020, *Power systems management and associated information exchange – Data and communications security – Part 4: Profiles including MMS and derivatives*
IEC 62351-4:2018/AMD1:2020

IEC 62351-5, *Power systems management and associated information exchange - Data and communications security - Part 5: Security for IEC 60870-5 and derivatives*

IEC 62351-6:2020, *Power systems management and associated information exchange – Data and communications security – Part 6: Security for IEC 61850*

IEC 62351-7, *Power systems management and associated information exchange – Data and communications security – Part 7: Network and System Management (NSM) data object models*

IEC 62351-8:2020, *Power systems management and associated information exchange – Data and communications security – Part 8: Role-based access control for power system management*

IEC 62351-14, *Power systems management and associated information exchange – Data and communications security – Part 14: Cyber Security Events Logs*[2]

RFC 4492:2006, *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*

RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*

RFC 7027, *Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS)*

RFC 8734, *Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS) Version 1.3*

RFC 8744, *Issues and Requirements for Server Name Identification (SNI) Encryption*

IETF Draft: *TLS Encrypted Client Hello* (https://datatracker.ietf.org/doc/draft-ietf-tls-esni/)

NIST SP-800-57 Part 1 Rev. 5, *Recommendations for Key Management*, May 2020 (https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final)

NSA Suite B, Suite B *Cryptography / Cryptographic Interoperability*

BSI TR 02102-1, *Cryptographic Mechanisms, February 2022 (https://www.bsi.bund.de/EN/Publications/TechnicalGuidelines/tr02102/tr02102_node.html)*

RFC8996: *Deprecating TLSv1.0 and TLSv1.1 (https://datatracker.ietf.org/doc/html/rfc8996/)*

IETF recommendations on TLS cipher suites
(https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4)

IETF RFC 9155: Deprecating MD5 and SHA-1 signature hashes in TLS 1.2, December 2021, (https://datatracker.ietf.org/doc/html/rfc9155)

[2] Under preparation. Stage at the time of publication: IEC/ACDV 62351-14:2023.

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

GESTION DES SYSTÈMES DE PUISSANCE ET ÉCHANGES D'INFORMATIONS ASSOCIÉS – SÉCURITÉ DES COMMUNICATIONS ET DES DONNÉES –

Partie 3: Sécurité des réseaux et des systèmes de communication – Profils comprenant TCP/IP

AVANT-PROPOS

1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses Publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et averti de leur existence.

L'IEC 62351-3 a été établie par le comité d'études 57 de l'IEC: Gestion des systèmes de puissance et échanges d'informations associés. Il s'agit d'une Norme internationale.

Cette seconde édition annule et remplace la première édition parue en 2014, l'Amendement 1:2018 et l'Amendement 2:2020. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

a) inclusion du paramètre lié à la TLSv1.2 exigé dans l'IEC 62351-3 Éd.1.2 à spécifier par la norme de référence. Ce paramètre comprend les éléments suivants:

- les suites chiffrées TLSv1.2 obligatoires à prendre en charge;
- la spécification des paramètres de reprise de session;
- la spécification des paramètres de renégociation de session;
- la gestion des révocations à l'aide de la CRL et du protocole OCSP;
- la gestion des événements de sécurité;

b) inclusion d'un profil TLSv1.3 applicable au domaine des systèmes de puissance d'une manière similaire à celle de la session TLSv1.2.

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
57/2578/FDIS	57/2593/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

NOTE Les caractères d'imprimerie suivants sont employés:

- les caractères en notation de syntaxe abstraite numéro un (ASN.1) sont présentés en `courier new` et **`bold courier new`**.

Une liste de toutes les parties de la série IEC 62351, publiées sous le titre général *Gestion des systèmes de puissance et échanges d'informations associés – Sécurité des communications et des données*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera:

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de ce document indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La présente édition de l'IEC 62351-3 est un document autosuffisant qui décrit l'utilisation de la TLS pour sécuriser la communication des systèmes de puissance. Il est recommandé de se référer à la présente édition du présent document plutôt qu'à n'importe quelle édition précédente, car cette édition met à jour les algorithmes cryptographiques utilisés (suites chiffrées, en anglais cipher suites), fournit des fonctionnalités améliorées et couvre différentes versions TLS. Contrairement aux éditions précédentes, le présent document spécifie tous les réglages spécifiques TLS nécessaires et n'exige pas que la norme de référence définisse des réglages spécifiques pour la TLS.

Il est à noter que la recommandation d'utiliser la présente édition, éventuellement aussi avec des normes de référencement plus anciennes, exige un support technique par les mises en œuvre des paramètres TLS spécifiés dans la présente édition du document.

GESTION DES SYSTÈMES DE PUISSANCE ET ÉCHANGES D'INFORMATIONS ASSOCIÉS – SÉCURITÉ DES COMMUNICATIONS ET DES DONNÉES –

Partie 3: Sécurité des réseaux et des systèmes de communication – Profils comprenant TCP/IP

1 Domaine d'application

1.1 Domaine d'application

La présente partie de l'IEC 62351 spécifie comment assurer la confidentialité, la protection de l'intégrité et l'authentification des niveaux des messages pour les protocoles qui utilisent les protocoles TCP/IP comme couche transport des messages et utilisent la sécurité de la couche transport lorsque la cybersécurité est exigée. Ceci peut concerner les protocoles SCADA/téléconduite, protection, automation et contrôles.

L'IEC 62351-3 spécifie une méthode permettant de sécuriser les protocoles TCP/IP par l'intermédiaire de contraintes sur la spécification des messages, procédures et algorithmes de sécurité de la couche transport (TLS) (version 1.2 de TLS définie dans la RFC 5246 et version 1.3 définie dans la RFC 8446). Des articles spécifiques contiennent des paragraphes indiquant les différences et les points communs d'application en fonction de la version TLS cible. L'utilisation et la spécification des dispositifs de sécurité externe concernés (par exemple "bump-in-the-wire") sont considérées comme ne relevant pas du domaine d'application du présent document.

Contrairement aux précédentes éditions du présent document, la présente édition est autosuffisante, car elle définit entièrement un profil de TLS. De ce fait, elle peut être appliquée directement, sans nécessiter de spécifier de paramètres TLS supplémentaires, à l'exception du numéro du port par lequel la communication est effectuée. Par conséquent, la présente partie peut être directement utilisée à partir d'une norme de référence et peut être combinée avec des mesures de sécurité supplémentaires sur d'autres couches. La définition du profil de TLS sans nécessiter de spécifier de paramètres TLS supplémentaires permet de déclarer la conformité à la fonctionnalité décrite sans nécessiter de recourir à d'autres documents IEC 62351.

Le présent document est destiné à être référencé comme partie normative des autres normes IEC qui traitent de la nécessité d'assurer la sécurité de leurs échanges protocolaires basés sur TCP/IP dans des conditions limites similaires. Cependant, il revient aux initiatives individuelles en matière de sécurité des protocoles de décider si le présent document est à référencer.

Le présent document définit également des événements de sécurité pour des conditions spécifiques, qui prennent en charge la gestion des erreurs, les pistes d'audit de sécurité, la détection d'intrusion et les essais de conformité. Toute action d'un organisme en réponse à des événements dus à une condition d'erreur décrite dans le présent document ne relève pas du domaine d'application du présent document et est susceptible d'être définie par la politique de sécurité de l'organisme.

Le présent document présente les exigences de sécurité des protocoles de gestion des systèmes de puissance de l'IEC. Une révision du présent document pourrait s'avérer nécessaire en cas d'ajout de nouvelles exigences dans d'autres normes.

1.2 Public ciblé

Les premiers utilisateurs auxquels s'adresse le présent document sont les experts qui développent ou utilisent les protocoles dans le domaine de la gestion des systèmes de

puissance et des échanges d'informations associés. Pour que les mesures décrites dans le présent document prennent effet, elles doivent être acceptées et référencées par les spécifications de protocoles qui utilisent la sécurité TCP/IP en appliquant la TLS. Le présent document est rédigé afin de permettre ce processus.

Les autres utilisateurs auxquels s'adresse le présent document sont les développeurs de produits qui mettent en œuvre ces protocoles.

Des parties du présent document peuvent également être utiles aux responsables et aux dirigeants afin de comprendre l'objectif d'une activité et les exigences correspondantes.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC TS 62351-1:2007, *Power systems management and associated information exchange – Data and communications security – Part 1: Communication network and system security – Introduction to security issues* (disponible en anglais seulement)

IEC TS 62351-2:2008, *Power systems management and associated information exchange – Data and communications security – Part 2: Glossary of terms* (disponible en anglais seulement)

IEC 62351-9, *Gestion des systèmes de puissance et échanges d'informations associés – Sécurité des communications et des données – Partie 9: Gestion de clé de cybersécurité des équipements de système de puissance*

ISO/IEC 9594-8:2020, Rec. UIT-T X.509 (2019), *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire – Partie 8: Cadre général des certificats de clé publique et d'attribut*

RFC 5246:2008, *The TLS Protocol Version 1.2* [1]

RFC 5280:2008, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 5288:2008, *AES Galois Counter Mode (GCM) Cipher Suites for TLS*

RFC 5289:2008, *TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)*

RFC 5746:2010, *Transport Layer Security (TLS): Renegotiation Indication Extension*

RFC 6066:2011, *Transport Layer Security Extensions*

RFC 6176:2011, *Prohibiting Secure Sockets Layer (SSL) Version 2.0*

RFC 8422:2018, *ECC Cipher Suites for TLSv1.2 and earlier*

RFC 8446:2018, *The TLS Protocol Version 1.3*

[1] Généralement appelé SSL/TLS.

RFC 9150:2021, *TLS 1.3 Authentication and Integrity only Cipher Suites*

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et les définitions de l'IEC 62351-2, ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

3.2 Abréviations

ACSE	Association Control Service Element (élément de service de contrôle d'association)
AEAD	Authenticated Encryption with Associated Data (chiffrement authentifié avec données associées)
BCP	Best Current Practice (meilleure pratique courante)
CRL	Certificate Revocation List (liste de révocation de certificat)
DER	Distinguished Encoding Rules (règles de codage distinctives)
DH(E)	Diffie Hellman (Ephemeral) key agreement (accord d'échange de clés Diffie-Hellman éphémères) (voir aussi IEC 62351-9)
ECDH(E)	Elliptic Curve based Diffie Hellman (Ephemeral) key agreement (accord d'échange de clés Diffie-Hellman éphémères basé sur des courbes elliptiques) (voir aussi IEC 62351-9)
ECDSA	Elliptic Curve Digital Signature Algorithm (algorithme de signature numérique à courbe elliptique)
IV	Initialization Vector (vecteur d'initialisation)
MAC	Message Authentication Code (code d'authentification de message)
MitM	Man-in-the-Middle (attaque de l'homme du milieu) (type d'attaque)
OCSP	Online Certificate Status Protocol (protocole OCSP) (voir RFC 6960)
PICS	Protocol Implementation Conformance Statement (protocole PICS)
PIXIT	Protocol Implementation eXtra Information for Testing (protocole PIXIT)
PSK	Pre-Shared Key (clé prépartagée)
TLS	Transport Layer Security (sécurité de la couche transport)

4 Problèmes de sécurité couverts par le présent document

4.1 Généralités

L'environnement des systèmes de puissance de l'IEC a des exigences opérationnelles différentes de nombreuses applications informatiques (IT) qui utilisent la TLS pour protéger le transport d'informations sur une connexion TCP/IP. Cela exige, d'une part, la gestion des certificats utilisés pour authentifier les entités, cette gestion étant traitée dans l'IEC 62351-9. D'autre part, il est nécessaire de définir des paramètres de sécurité spécifiques liés à la TLS, tels que les suites chiffrées sélectionnées, le paramètre de gestion de session et les extensions utilisées, cette définition faisant l'objet du présent document. Une différence majeure dans l'application de la TLS dans le domaine des systèmes de puissance est la durée de la connexion TCP/IP pour laquelle la sécurité nécessite d'être maintenue. En outre, il faut également tenir compte du fait que le matériel utilisé pour l'automatisation des systèmes de puissance a une longue durée de vie, ce qui peut exiger de prendre également en charge d'anciennes versions de la TLS ou de suites chiffrées pour maintenir la rétrocompatibilité.

De nombreux protocoles IT ont des durées de connexion courtes qui permettent la renégociation des algorithmes de chiffrement lors du rétablissement de la connexion. Cependant, les connexions dans un environnement de téléconduite ont tendance à être plus longues, et sont souvent "permanentes". En raison de leur longévité, les connexions dans le domaine de la gestion des systèmes de puissance et des échanges d'informations associés rendent nécessaire une prise en compte particulière. À cet effet, afin d'assurer la protection des connexions "permanentes", un mécanisme de mise à jour de la clé de session est choisi dans le présent document, en fonction des caractéristiques TLS existantes. Il est à noter que TLSv1.2 et TLSv1.3 prennent en charge différentes approches concernant le renouvellement de clé.

La TLS permet une grande diversité de réglages, tels que des suites chiffrées, sélectionnés pour protéger l'établissement de liaison et le protocole de couche d'enregistrement, qui sont négociés lors de l'établissement de la connexion. Elle prend également en charge les mécanismes de gestion de session pour configurer de nouvelles sessions ou renouveler les sessions existantes. Pour assurer l'interopérabilité entre différentes mises en œuvre, le présent document spécifie un ensemble commun de caractéristiques TLS à prendre en charge par les mises en œuvre conformes.

TLS a évolué et est disponible dans la version 1.3. Dans la version 1.3, plusieurs parties du protocole ont été retouchées, aboutissant à un protocole qui n'est pas fonctionnellement rétrocompatible avec la TLSv1.2. Comme le premier échange de messages est défini de manière rétrocompatible, le serveur TLS peut être en mesure de passer à l'une ou l'autre version, en fonction de la politique de sécurité.

De plus, le présent document spécifie l'utilisation de capacités TLS particulières qui permettent de contrer des menaces spécifiques à la sécurité (voir aussi 4.2). Plus précisément, la TLS permet la définition d'extensions du protocole afin d'atténuer les vulnérabilités potentielles du protocole ou d'améliorer la fonctionnalité du protocole. La présente spécification utilise des extensions qui ont déjà été définies.

Il est à noter que la TLS utilise des certificats X.509 (voir aussi l'ISO/IEC 9594-8 ou la RFC 5280) pour l'authentification et l'accord d'échange de clés. Dans le contexte du présent document, le terme "certificat" fait toujours référence à des certificats de clé publique (par opposition aux certificats d'attribut).

NOTE Il est présumé que la gestion des certificats nécessaire au fonctionnement de la TLS est traitée conformément à l'IEC 62351-9.

4.2 Menaces à la sécurité contrées

Voir l'IEC TS 62351-1 pour obtenir des informations sur les menaces à la sécurité et les méthodes d'attaque.

Les protocoles TCP/IP et les spécifications de sécurité dans le présent document couvrent uniquement les couches transport de communication (couches OSI 4 et inférieures). Plus précisément, la TLS protège de manière transparente les messages transportés de la couche OSI 5 et des couches supérieures. Le présent document ne couvre pas la fonctionnalité de sécurité spécifique aux couches application de communication (couches OSI 5 et supérieures) ou la sécurité d'application à application. Cette fonctionnalité est définie dans d'autres parties de l'IEC 62351, par exemple l'IEC 62351-4 pour les MMS, l'IEC 62351-5 pour la communication en série et la téléconduite, et l'IEC 62351-6 pour (R)GOOSE et (R)SV.

NOTE L'application de TLS décrite dans le présent document prend en charge la protection des informations envoyées sur une connexion TLS protégée.

Les menaces spécifiques contrées dans le présent document pour la couche transport comprennent:

– la falsification (modification ou insertion non autorisée de messages) est traitée par authentification mutuelle fondée sur des certificats des participants à la communication et la protection de l'intégrité du niveau des paquets TLS.

De plus, lorsque les informations ont été identifiées comme exigeant une protection de la confidentialité:

– l'accès non autorisé aux informations par le biais du chiffrement des messages au niveau des paquets TLS.

4.3 Méthodes d'attaque contrées

La mise en œuvre appropriée des spécifications et des recommandations traitées dans le présent document permet de contrer les méthodes d'attaque suivantes:

– usurpation d'identité: cette menace est contrée par l'utilisation d'une authentification mutuelle basée sur des certificats X.509 pendant l'établissement de liaison TLS et d'informations signées numériquement comme des informations de révocation;
– attaque de l'homme du milieu (MitM): cette menace est contrée pour la couche d'enregistrement TLS par l'utilisation de sommes de contrôle cryptographiques (MAC) basées sur la négociation mutuellement authentifiée des clés de session. Ces clés sont mises en place pendant l'établissement de liaison TLS. La protection contre les attaques MitM pendant la phase d'établissement de liaison TLS elle-même est assurée par le message `Finish` qui met fin à la phase d'établissement de liaison. Ce message est chiffré et contient une valeur de hachage pendant tous les messages échangés lors de l'établissement de liaison. En outre, TLSv1.3 fournit déjà une protection cryptographique des messages d'établissement de liaison directement (à l'exception de `ClientHello`);
– relecture: cette menace est contrée par l'application d'un MAC pour la protection de l'intégrité sur une base par paquet incluant un numéro de séquence afin de détecter une relecture réelle;
– écoute clandestine: cette menace est contrée par l'utilisation de suites chiffrées lors de la négociation de la sécurité de la couche d'enregistrement TLS.

NOTE Les caractéristiques actuelles de performance d'une mise en œuvre revendiquant la conformité à la présente partie de la série IEC 62351 ne relèvent pas du domaine d'application du présent document.

4.4 Gestion des événements de sécurité

Des événements de sécurité sont définis tout au long du présent document. Ces événements de sécurité sont destinés à prendre en charge la gestion des erreurs, et donc à accroître la résilience du système. Il convient que les mises en œuvre fournissent un mécanisme pour annoncer les événements de sécurité auprès d'un système d'évaluation. L'IEC 62351-4 est recommandé.

Les informations concernant les événements de sécurité et les informations détaillées potentielles ne peuvent être fournies par l'entité que sur la base de la disponibilité de ces informations par l'intermédiaire de la plate-forme sous-jacente ou des composants utilisés.

Il est recommandé que les événements de sécurité définis tout au long du présent document soient tenus à la disposition de l'infrastructure opérationnelle par les événements de sécurité spécifiés dans l'IEC 62351-14 ou par les objets de surveillance spécifiés dans l'IEC 62351-7. L'Annex A fournit une mise en correspondance des événements définis dans le présent document et le concept de l'IEC 62351-14.

Il est à noter que les avis, les avertissements, les erreurs et les alarmes servent à indiquer la gravité d'un événement en matière de sécurité. Le concept suivant est repris de l'IEC 62351-14:

– un avis fait référence à une activité liée à la cybersécurité pendant l'utilisation ou la maintenance de routine d'une entité. Il ne se rapporte pas à une brèche de cybersécurité, une cyberattaque ou un écart par rapport à la condition de fonctionnement normale d'une entité;
– un avertissement est un écart par rapport à la condition de fonctionnement normale d'une entité, mais il ne s'agit pas nécessairement d'une cyberattaque;
– une erreur décrit une condition imprévue, qui peut indiquer une activité non autorisée. Elle peut ne pas exiger d'action immédiate;
– une alarme est une indication d'un problème grave, qui peut indiquer une activité non autorisée. Il est recommandé de mener immédiatement une action.

Dans tous les cas, la politique de sécurité d'un organisme est censée déterminer le traitement final des événements en fonction de l'environnement opérationnel. Par exemple, l'évaluation d'une ou plusieurs alarmes peut indiquer le niveau d'un incident.

5 Vue d'ensemble des différences entre les versions TLS

5.1 Généralités

Le présent article donne un bref aperçu des principales différences entre les versions 1.2 et 1.3 de TLS. Les différences spécifiquement traitées sont le nommage des suites chiffrées, la rétrocompatibilité et les extensions qui ont été utilisées dans la version précédente de la présente norme.

5.2 Principales différences entre TLSv1.2 et TLSv1.3

Dans les paragraphes suivants, les principales différences entre TLSv1.2 et TLSv1.3 pour les différents sous-protocoles (établissement de liaison TLS, couche d'enregistrement de TLS) sont énumérées (à noter que la RFC 8446 fournit une liste complète des modifications apportées de TLSv1.2 à TLSv1.3), qui ont une relation avec les caractéristiques utilisées dans TLSv1.2 dans l'IEC 62351-3:

– moins d'allers-retours pendant l'établissement de liaison dans TLSv1.3 (1 aller-retour pour les sessions authentifiées unilatérales, 1,5 pour une authentification mutuelle);
– les messages d'établissement de liaison TLSv1.3 sont chiffrés, à l'exception de `ClientHello` et de parties de `ServerHello`, qui sont envoyés en clair. Il est à noter que les travaux en cours de l'IETF visent à définir des extensions permettant de chiffrer une partie du `ClientHello`, comme l'indication de nom de serveur (SNI) entraînant un SNI chiffré (ESNI) ou un chiffrement de la totalité du `ClientHello` comme un `ClientHello` chiffré (ECH);
– TLSv1.3 supprime la prise en charge des algorithmes cryptographiques obsolètes et vulnérables: les chiffrements d'exportation, DES, 3DES, AES-CBC, RC4, MD5, SHA-1, ne sont plus pris en charge;
– modifications de l'établissement des clés dans TLSv1.3
  - suppression des groupes Diffie-Hellman arbitraires
  - établissement de clés Diffie-Hellman éphémères
  - aucune prise en charge du chiffrement des clés RSA

– TLSv1.3 supprime la renégociation de session, mais fournit des messages de post-établissement de liaison pour la mise à jour des clés et des IV, ainsi que l'authentification des clients;

– TLSv1.3 simplifie la reprise de session en utilisant des PSK et des tickets de session;

– TLSv1.3 introduit des messages de post-établissement de liaison pour

  • les tickets de session

  • l'authentification des clients

  • la mise à jour des clés et des IV

– TLSv1.3 supprime la prise en charge du multi-agrafage des réponses OCSP conformément à la RFC 6961 (status_request_v2). Il a été remplacé par une variante directe permettant l'agrafage de la réponse OCSP aux certificats d'une chaîne par une extension d'un `CertificateEntry`;

– TLSv1.3 introduit l'agrafage OCSP pour les certificats de clients, qui peuvent être importants en particulier pour les cas d'utilisation industrielle, où le serveur réside souvent sur le dispositif de terrain ayant une capacité potentielle limitée de vérifier l'état de révocation des certificats reçus;

– TLSv1.3 supprime les suites chiffrées non AEAD pour la protection de la couche d'enregistrement de TLS. Il est à noter que les suites chiffrées assurant uniquement l'intégrité ont été définies dans la RFC 9051;

– TLSv1.3 fournit l'option d'utiliser des données d'application déjà chiffrées pendant le premier aller-retour (0-RTT) avec certaines réductions de la sécurité (possibilité de relecture par un attaquant). Il est à noter que cette option est explicitement inutilisée dans le présent document;

– il est aussi à noter que `ChangeCipherSpec` est un type de contenu de protocole TLSv1.2 indépendant et qu'il ne s'agit pas d'un message d'établissement de liaison TLS. `ChangeCipherSpec` a été supprimé dans TLSv1.3.

5.3 Nommage d'une suite chiffrée

Les modifications de l'établissement de liaison TLS et de la gestion de la couche d'enregistrement dans TLSv1.3 sont également reflétées dans la définition de la suite chiffrée.

Une suite chiffrée dans TLSv1.2 et les versions antérieures combine des algorithmes d'authentification et d'accord d'échange de clés pendant l'établissement de liaison TLS, ainsi que des algorithmes pour le chiffrement et l'authentification des messages (intégrité) pour la couche d'enregistrement, comme représenté à la Figure 1. La suite chiffrée a donc considéré à la fois l'établissement de liaison TLS ainsi que la couche d'enregistrement. En général, chaque suite chiffrée a un nom qui se compose d'un ensemble de mnémoniques séparés par des traits de soulignement et qui se présente sous la forme représentée à la Figure 1.

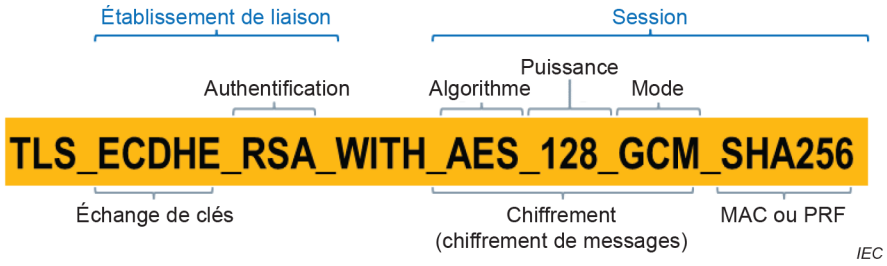


Figure 1 – Définition de suites chiffrées selon TLSv1.2 (RFC 5246)

Dans TLSv1.3, le nommage de la suite chiffrée a été changé pour ne se rapporter qu'à la couche d'enregistrement. Par conséquent, il contient les informations relatives au système de chiffrement utilisé (chiffrement toujours authentifié) et le hachage utilisé dans la fonction de dérivation de clés basée sur le hachage, comme représenté à la Figure 2.

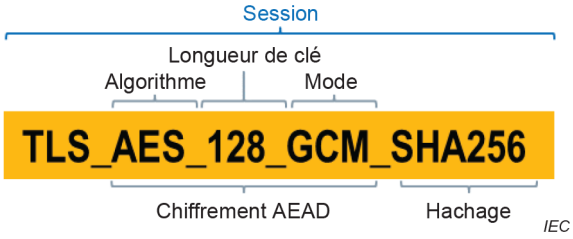


Figure 2 – Définition des suites chiffrées selon TLSv1.3 (RFC 8446)

Les algorithmes utilisés dans l'établissement de liaison TLS sont explicitement signalés dans le cadre de l'échange des messages `ClientHello` et `ServerHello`.

5.4 Rétrocompatibilité

La TLS assure la rétrocompatibilité en signalant la version TLS dans le message `Client.Hello` , qui est compatible tout au long des messages de version 1.x. Il est à noter que le mécanisme de négociation de version TLSv1.2 a été abandonné en faveur d'une liste de versions dans une extension. Plus précisément, une extension `supported_versions` peut être utilisée dans le message `Client.Hello` pour négocier la version TLS à utiliser, en préférence au champ `legacy_version` du `Client.Hello`. Ceci est destiné à augmenter la compatibilité avec les serveurs existants qui ont incorrectement mis en œuvre la négociation de version.

5.5 Extensions

Les extensions offrent une option pour inclure des informations supplémentaires dans certains des messages d'établissement de liaison TLS. Comme indiqué en 5.2, certaines des caractéristiques disponibles dans TLSv1.2 ne sont pas prises en charge dans TLSv1.3 ou sont gérées différemment. Par conséquent, certaines extensions ne s'appliquent pas ou leur prise en charge n'est pas nécessaire dans TLSv1.3.

Les extensions obligatoirement utilisées dans l'édition 1 du présent document (2007) comprennent:

– `renegotiation_info`: extension de renégociation définie dans la RFC 5746 pour lier la session renégociée à la session précédente;

– `reliable_ca_keys`: permet à un client d'indiquer les certificats CA disponibles localement, qui peuvent être utilisés pour valider des certificats de serveur reçus (définis en RFC 6066);

– `signature_algorithms`: permet au client de signaler des algorithmes de signature (par combinaison d'un algorithme de hachage et de signature) qu'il prend en charge [définie dans la RFC 5426 (TLSv1.2) et la RFC 8446 (TLSv1.3)].

Parmi ces extensions, seule l'extension `signature_algorithm` est prise en charge dans les deux versions TLS. À la place de l'extension `trusted_ca_keys`, TLSv1.3 spécifie l'extension "`certificate_authorities`" qui peut être utilisée par l'un ou l'autre des points d'extrémité pour signaler la CA prise en charge comme guide pour le point d'extrémité destinataire. L'extension `renegotiation_info` n'est pas prise en charge dans TLSv1.3, car la renégociation de session n'est pas prise en charge.

En outre, TLSv1.3 définit la prise en charge obligatoire des extensions qui sont profilées dans le présent document. La liste des extensions à prendre en charge est fournie dans les articles spécifiques relatifs aux différentes versions TLS.

6 Exigences génériques

6.1 Généralités

Le présent article définit les exigences qui s'appliquent à toutes les versions TLS traitées dans le présent document.

Si l'établissement de liaison permettant de configurer une session s'effectue avec succès (selon les exigences décrites à l'Article 6, à l'Article 7 pour TLSv1.2 ou à l'Article 8 pour TLSv1.3, un événement de sécurité doit être déclenché ("avis: établissement de liaison TLS réussi"). Si des erreurs sont apparues pendant l'établissement du canal TLS et si la session TLS ne peut pas être établie, les événements de sécurité définis à l'Article 6, à l'Article 7 pour TLSv1.2 et à l'Article 8 pour TLSv1.3 fournissent des informations détaillées.

Les normes de référence du présent document doivent spécifier un numéro de port, qui est utilisé pour effectuer la communication TLS protégée.

6.2 Signalisation des versions TLS prises en charge

TLSv1.2 telle que définie dans la RFC 5246 est la version par défaut qui doit être prise en charge. TLSv1.3 est la version TLS la plus récente et il est recommandé qu'elle soit prise en charge.

Il est recommandé que le client TLS qui initie une connexion TLS indique la version TLS la plus élevée prise en charge dans le message `ClientHello` de l'établissement de liaison TLS. Le serveur TLS récepteur peut accepter des versions supérieures si elles sont prises en charge du point de vue fonctionnel et autorisées par la politique de sécurité de l'environnement d'exploitation.

TLSv1.2 et TLSv1.3 gèrent différemment la signalisation de version. Dans TLSv1.2, le numéro de version est fourni dans le champ `ProtocolVersion` du message `ClientHello` et signale la version TLS la plus élevée prise en charge par le client.

Le client doit inclure l'identificateur TLSv1.2 `0x0303` dans le champ `ProtocolVersion` pour signaler la prise en charge de TLSv1.2. Si, de plus, le client prend en charge TLSv1.3, il indique ses préférences de version dans l'extension `supported_versions` du message `ClientHello`. Dans ce cas, le champ `ProtocolVersion` dans l'extension `supported_versions` doit être réglé à l'identificateur TLSv1.3 `0x0304` conformément à la RFC 8446. Il est à noter que le serveur prenant en charge TLSv1.3 recevant un message `ClientHello` avec l'extension `supported_versions` n'utilise pas la version TLSv1.2 signalée dans le champ `ProtocolVersion` du message `ClientHello`.

Pour assurer la rétrocompatibilité, les mises en œuvre peuvent éventuellement prendre en charge les versions 1.0 et 1.1 de TLS (parfois appelées SSL v3.1 et v3.2). L'établissement de liaison TLS fournit un mécanisme intégré qui doit être utilisé pour prendre en charge la négociation de la version. L'homologue initiant une connexion TLS doit toujours indiquer la version TLS la plus élevée prise en charge pendant le message d'établissement de liaison TLS. L'application de versions TLS autres que v1.2 relève de la politique de sécurité locale. La proposition de versions antérieures à TLSv1.0 ne doit pas entraîner l'établissement d'une connexion sécurisée (voir aussi la RFC 6176).

NOTE 1 Pour TLSv1.0 et TLSv1.1, certaines vulnérabilités sont connues. De plus, l'IETF a publié un BCP pour obsolescence de TLSv1.0 et TLSv1.1 dans le RFC 8996. La prise en charge facultative est uniquement destinée à assurer la rétrocompatibilité.

La prise en charge uniquement de TLSv1.0 ou uniquement de TLSv1.1 ou uniquement de ces deux versions doit être désactivée par défaut et exige une activation distincte autorisée par une politique de sécurité de l'organisme. Si la prise en charge est désactivée, la proposition d'utiliser uniquement TLSv1.0 ou uniquement TLSv1.1 ou uniquement ces deux versions doit

déclencher un événement de sécurité ("alarme: communication non sécurisée: version TLS proposée obsolète") et l'établissement de session doit être interrompu par le récepteur. Si la prise en charge est activée, la proposition des versions TLSv1.0 ou TLSv1.1 doit déclencher un événement de sécurité ("avertissement: version TLS non sécurisée").

Les versions TLS antérieures à TLSv1.0 ne doivent pas être utilisées. La proposition de versions antérieures à TLSv1.0 ou SSLv3.1 doit déclencher un événement de sécurité ("alarme: communication non sécurisée: version TLS proposée non autorisée") et l'établissement de session doit être interrompu par le récepteur.

NOTE 2 L'IETF a déconseillé SSLv3.0 dans la RFC 7568.

La session TLS doit être interrompue si la version TLS négociée change à partir de l'établissement de liaison TLS initial:

- TLSv1.2 et les versions inférieures: dans une session TLS en cours pendant une renégociation de session TLS ou un établissement de liaison de reprise de session;
- TLSv1.3: une reprise de session basée sur une identité PSK fournie d'un côté ou de l'autre.

L'interruption de la session doit déclencher un événement de sécurité ("alarme: détection d'un changement de version TLS demandé (vers version inférieure possible) dans la communication en cours").

NOTE 3 La RFC 8446 interdit spécifiquement les changements de version au cours d'une renégociation TLSv1.2 pour passer à TLSv1.3.

6.3 Utilisation de suites chiffrées sans chiffrement

Une suite chiffrée qui spécifie NULL pour le chiffrement ne doit pas être utilisée pour la communication à l'extérieur du domaine administratif, si le chiffrement de cette connexion de communication par d'autres moyens ne peut pas être garanti.

NOTE 1 Le présent document n'exclut pas l'utilisation de communications chiffrées à l'aide de tunnels VPN cryptographiques. L'utilisation de tels VPN n'entre pas dans le domaine d'application de la présente norme.

Au sein d'un domaine administratif, l'utilisation de suites chiffrées sans chiffrement peut être autorisée, mais son utilisation relève de la responsabilité de l'opérateur.

NOTE 2 L'application de suites chiffrées sans chiffrement permet le contrôle du trafic tout en continuant à maintenir une authentification entre extrémités et une protection de l'intégrité du trafic.

Si la connexion de communication est destinée à être uniquement sécurisée en ce qui concerne l'intégrité des informations transmises, les suites chiffrées suivantes peuvent être utilisées avec TLSv1.2 et les versions inférieures:

– TLS_RSA_WITH_NULL_SHA256 (définie dans la RFC 5246)

NOTE 3 L'IEC 62351-3:2014+AMD1:2018+AMD2:2020 prenait en charge la suite chiffrée TLS_RSA_WITH_NULL_SHA. Cette prise en charge facultative a été supprimée, car SHA-1 est déconseillé.

Si la connexion de communication est destinée à être sécurisée uniquement en ce qui concerne l'intégrité des informations transmises, les suites chiffrées suivantes peuvent être utilisées avec TLSv1.3:

– TLS_SHA256_SHA256 (RFC 9150)
– TLS_SHA384_SHA384 (RFC 9150)

Les mises en œuvre permettant des suites chiffrées TLS avec un chiffrement NULL revendiquant la conformité à la présente partie doivent fournir un mécanisme pour permettre explicitement ces suites chiffrées TLS. Par défaut, les suites chiffrées TLS sans chiffrement doivent être désactivées.

6.4 Prise en charge des certificats

6.4.1 Prise en charge d'ancres de confiance multiples

Une ancre de confiance associée à la CA fournit la base pour la validation du certificat, en particulier la validation du chemin de certificat. Comme les entités sont susceptibles d'avoir des interactions avec les entités appartenant à un domaine différent, il est nécessaire de prendre en charge différentes ancres de confiance. Le nombre réel dépend du cas d'utilisation cible.

Une mise en œuvre revendiquant la conformité au présent document doit prendre en charge au moins 5 ancres de confiance (généralement appelés CA racine) en tant que nombre minimal.

Les critères et la sélection d'une CA ne relèvent pas du domaine d'application du présent document.

Dans les cas où plusieurs certificats X.509 (et la clé privée correspondante) sont disponibles sur un IED, il peut être souhaitable de permettre au demandeur de choisir un certificat du côté serveur qui correspond aux certificats de confiance (CA racine) disponibles du côté du demandeur.

6.4.2 Taille de certificat

Une mise en œuvre revendiquant la conformité à la présente norme doit pouvoir gérer des certificats dont la taille maximale est supérieure ou égale à 8 192 octets. La taille réelle des certificats pris en charge doit être documentée.

NOTE 1 Le certificat peut également comporter des informations sur les rôles, conformément à l'IEC 62351-8, qui influencent sa taille finale.

NOTE 2 La taille du certificat peut être influencée par la sélection attentive des noms des émetteurs, des domaines et des extensions prises en charge, etc.

Le dépassement de la taille maximale supportée d'un certificat au cours d'un établissement de liaison TLS doit déclencher un événement de sécurité ("alarme: taille du certificat TLS dépassée") et l'interruption de l'établissement de session TLS.

Il est à noter que la limitation de la taille du certificat se rapporte aux exigences potentielles des protocoles des couches supérieures tels que MMS. Pour le MMS, afin d'assurer l'interopérabilité des certificats de clé publique, il est nécessaire de fixer une taille maximale autorisée pour les certificats de clé publique échangés par l'ACSE. Cette restriction de taille est décrite dans l'IEC 62351-4. Si le même certificat est utilisé dans le contexte du MMS et de la TLS, il est nécessaire de respecter cette condition limite.

Le fabricant peut prendre en charge des tailles de certificat plus importantes pour la TLS si les certificats ne sous-tendent pas des restrictions de taille de certificat dans les couches supérieures.

6.4.3 Échange de certificats

L'échange et la validation de certificats doivent être bidirectionnels pour obtenir l'authentification réciproque. Si une entité ne fournit pas son certificat, la connexion doit prendre fin.

NOTE Le certificat du serveur est transmis dans le message `ServerHello`. Le certificat du client est transmis dans le message `Certificate`.

L'interruption de la connexion due à l'absence de certificat homologue pendant l'établissement de liaison TLS doit déclencher un événement de sécurité ("alarme: certificat homologue non disponible").

L'impossibilité d'accéder au certificat du point d'extrémité local et à la clé privée correspondante doit déclencher un événement de sécurité ("alarme: certificat de point d'extrémité indisponible").

6.4.4 Validation des certificats de clé publique

6.4.4.1 Généralités

Les certificats doivent être vérifiés à la fois par les côtés appelant et appelé. En outre, chaque côté doit valider son propre certificat, comme décrit dans le présent paragraphe, et plus spécifiquement valider également les extensions de certificat incluses, comme défini dans l'IEC 62351-9, avant l'utilisation du certificat.

Il existe deux mécanismes qui doivent être configurables pour la validation des certificats:

– l'acceptation d'un certificat provenant d'une CA autorisée;

– l'acceptation de certificats individuels provenant d'une CA autorisée (domaine d'application limite de la CA émettrice).

6.4.4.2 Vérification basée sur des CA émettrices sélectionnées

Une mise en œuvre revendiquant la conformité à la présente norme doit pouvoir être configurée pour accepter les certificats d'une ou plusieurs autorités de certification, sans configuration des certificats individuels.

L'échec de la recherche d'un certificat de CA correspondant pendant l'établissement de liaison TLS doit déclencher un événement de sécurité ("alarme: validation de certificat: certificat de CA non disponible") et l'interruption de l'établissement de session TLS.

6.4.4.3 Vérification basée sur des certificats individuels provenant de CA émettrices sélectionnées

Il convient qu'une mise en œuvre revendiquant la conformité à la présente norme puisse être configurée pour accepter des certificats individuels (liste de certificats de confiance) provenant d'une ou plusieurs autorités de certification autorisées. La configuration des certificats individuels peut être fondée sur le nom de sujet ou le numéro de série du certificat ainsi que sur les informations de la CA émettrice.

NOTE 1 La vérification basée sur des certificats individuels peut être influencée par des procédures automatisées de mise à jour des certificats, qui changent généralement le numéro de série des certificats.

NOTE 2 La vérification basée sur des certificats individuels peut être prise en charge par des listes d'autorisation et de validation de certificats, comme indiqué dans l'IEC 62351-9.

L'échec de la recherche d'un certificat individuel correspondant pendant l'établissement de liaison TLS doit déclencher un événement de sécurité ("alarme: validation de certificat: certificat individuel de confiance provenant d'une CA autorisée non disponible") et l'interruption de l'établissement de session TLS.

6.4.4.4 Vérification de la révocation des certificats

6.4.4.4.1 Généralités

La révocation d'un certificat doit suivre les paramètres obligatoires et les procédures spécifiés dans l'ISO/IEC 9594-8. La gestion d'une CRL dépend de la mise en œuvre locale. L'IEC TS 62351-1 traite des questions de gestion des CRL. L'IEC 62351-9 présente l'application des CRL.

En variante des CRL locales, le protocole OCSP peut être utilisé pour vérifier l'état de révocation des certificats appliqués. L'application d'interactions OCSP à l'aide d'un répondeur OCSP est décrite dans l'IEC 62351-9.

Si le protocole OCSP est appliqué, la TLS assure la prise en charge du transport des messages de réponse OCSP par un "agrafage OCSP". Ceci est possible pour le certificat de serveur dans TLSv1.2 et est décrit en 7.5.5. TLSv1.3 prend en charge l'agrafage des réponses OCSP pour les certificats de clients et les certificats de serveurs, comme décrit en 8.8.10.

Une mise en œuvre revendiquant la conformité à la présente norme doit pouvoir vérifier l'état de révocation des certificats reçus selon un intervalle de temps configurable.

La vérification de la révocation d'un certificat peut être appelée sur la base des caractéristiques du protocole TLS telles que la renégociation de session pour TLSv1.2, ou par l'application qui surveille les certificats utilisés.

En cas de surveillance d'application des certificats utilisés, il est exigé que l'application garde la trace des certificats homologues utilisés pendant les établissements de sessions TLS et appelle la vérification de la révocation des certificats en fonction d'un temps configurable. Pour permettre des actions opérationnelles en cas de certificats révoqués (par exemple, la fourniture d'un justificatif d'identité valide ou l'exécution d'un contrôle de sécurité de l'IED associé), l'application peut accorder un délai de grâce (en heures) pendant lequel ces actions peuvent être réalisées sans interruption immédiate de la session en cours. Si aucun délai de grâce n'est prévu, il est exigé de l'application qu'elle mette fin à la connexion associée. La valeur par défaut pour le délai de grâce est de 0 h (c'est-à-dire aucun délai de grâce). Il est à noter que la configuration du délai de grâce pour l'application dépend de la mise en œuvre locale et peut faire l'objet d'une approbation conformément à la politique de sécurité de l'organisme.

Les certificats révoqués ne doivent pas être utilisés pour établir une session TLS (TLSv1.2 et TLSv1.3). Une entité recevant un certificat révoqué pendant l'établissement de la session doit refuser la connexion. Une entité recevant un certificat révoqué pendant une renégociation de session TLSv1.2 doit mettre fin à la connexion.

La détection d'un certificat révoqué lors d'un établissement de session ou une renégociation de session doit déclencher un événement de sécurité ("alarme: validation de certificat: certificat révoqué").

La fin d'une connexion TLS due à un certificat révoqué doit déclencher un événement de sécurité ("alarme: session terminée pour cause de certificat révoqué").

Il est à noter que les certificats révoqués peuvent conduire à la fin d'une session TLS. Il convient donc que les administrateurs de systèmes développent des procédures de gestion des certificats afin de remédier à cette situation (voir aussi l'IEC 62351-9). De même, il est prévu qu'un processus de gestion de la sécurité soit en place afin d'évaluer la CRL ou la réponse OCSP avant la diffusion, et ce afin d'éviter une interruption non désirée d'une connexion de communication qui peut influer sur la fiabilité du système. Les mises en œuvre peuvent également prendre en charge plusieurs certificats par dispositif pour passer à un autre certificat (valide) en cours de route. Il est à noter que cette dernière option dépend de la mise en œuvre locale.

6.4.4.4.2 Vérification de la révocation des certificats en utilisant des CRL

Le processus de vérification de la CRL ne doit pas entraîner la fin d'une session établie.

Pour TLSv1.2, l'interruption d'une session TLS due à un certificat révoqué devient effective à la prochaine renégociation de session, comme indiqué en 7.4.5. Ceci peut laisser un délai de grâce (entre la révocation du certificat et la détection du certificat révoqué utilisé) pour qu'un opérateur traite un certificat révoqué. Une alternative consiste à traiter la vérification de la révocation de certificat par l'application décrite en 6.4.4.4.1.

NOTE 1 Comme TLSv1.2 permet la renégociation de session, elle est généralement exécutée par la pile TLS et peut invoquer l'application afin d'effectuer la validation du certificat, et plus particulièrement la vérification de la révocation.

Comme TLSv1.3 ne prend pas en charge la renégociation de session, l'application est tenue d'appeler la validation de certificat et d'agir sur les certificats révoqués détectés, en particulier dans les sessions de longue durée. Cela exige que l'application garde la trace des certificats utilisés pendant l'établissement de la connexion. Cela est décrit en 6.4.4.4.1.

La période recommandée pour actualiser une CRL locale est de 24 h. La CRL elle-même contient également une information concernant la prochaine mise à jour de la CRL, dans le champ `nextUpdate`. Il convient que l'actualisation de la CRL soit alignée sur l'échéance qui survient en premier. Pour obtenir une nouvelle CRL, il convient de contacter le point de diffusion de CRL de la CA émettrice. Une impossibilité d'accéder au point de diffusion de CRL ne doit pas entraîner la fin d'une session. L'indisponibilité d'une CRL locale doit déclencher un événement de sécurité ("avertissement: CRL locale non accessible"). Une impossibilité d'accéder à une CRL locale ne doit pas entraîner la fin d'une session.

NOTE 2 La CRL peut être diffusée de différentes manières (manuelle sous forme de fichier, mise à disposition au niveau des points de diffusion de CRL, etc.).

NOTE 3 Si aucun certificat n'est révoqué, la CRL est une liste vide, mais qui est toujours disponible.

L'expiration d'une CRL locale doit déclencher un événement de sécurité ("avertissement: CRL expirée"), mais ne doit pas entraîner la fin de la session.

6.4.4.4.3 Vérification de la révocation des certificats en utilisant le protocole OCSP

Il convient qu'une mise en œuvre revendiquant la conformité au présent document puisse interagir avec un répondeur OCSP pour obtenir des informations sur la révocation.

Une mise en œuvre revendiquant la conformité au présent document doit pouvoir traiter les messages de réponse OCSP, lorsqu'ils sont fournis dans le cadre de l'établissement de liaison TLS sous forme de réponse OCSP agrafée.

NOTE L'interaction avec un répondeur OSCP fournit également un message de réponse OCSP.

L'indisponibilité d'une réponse OCSP doit déclencher un événement de sécurité ("avertissement: réponse OCSP expirée").

Les réponses OCSP ont un délai d'expiration indiqué par la valeur `nextUpdate`. Cela permet de mettre une réponse OCSP en mémoire cache. Si le protocole OCSP est appliqué, la mise en mémoire cache des réponses OCSP doit être prise en charge dans un délai maximal de 24 h.

Si un OCSP est utilisé pour les vérifications de révocation de certificat, l'inaccessibilité au répondeur OCSP doit entraîner un événement de sécurité: ("avertissement: répondeur OCSP non accessible").

Pour TLSv1.2, la RFC 6066 définit une demande d'état de certificat (`status_request`) et une extension de réponse permettant à TLSv1.2 de fournir une réponse OSCP pour le certificat de serveur avec son certificat. Comme cette extension permet de ne fournir qu'une seule réponse OCSP, une extension supplémentaire est définie dans la RFC 6961 permettant de demander (`status_requestv2`) et d'agrafer des réponses OCSP également pour les certificats CA intermédiaires contenus dans la liste de certificats du message de certificat de serveur. 7.5.5 fournit des détails complémentaires.

TLSv1.3 fournit une assistance pour la demande et l'agrafage des réponses OCSP comme décrit dans la RFC 6066, pour tous les certificats de la liste de certificats fournie par le côté client ou serveur. 8.8.10 fournit des détails complémentaires.

6.4.4.5 Certificats expirés

Les certificats expirés ne doivent pas être utilisés pendant l'établissement (TLSv1.2 et TLSv1.3) ou la renégociation (TLSv1.2 uniquement) d'une session TLS. Une entité recevant un certificat expiré au cours de l'établissement de la session doit refuser la connexion (TLSv1.2 et TLSv1.3). Une entité recevant un certificat expiré pendant la renégociation de session dans le cas de TLSv1.2 doit mettre fin à la connexion.

Il est à noter qu'il est prévu de mettre en place un processus de gestion de la sécurité afin d'initier une procédure de renouvellement de certificat dans un délai opportun (voir aussi l'IEC 62351-9). Un exemple de délai peut être un mois. Il peut également être possible de prendre en charge de multiples certificats par dispositif pour pouvoir passer à un autre certificat (valide).

Le refus d'un établissement de session initial ou d'une renégociation de session dû à un certificat expiré doit déclencher un événement de sécurité ("alarme: validation de certificat: certificat expiré").

6.4.4.6 Algorithmes de signature

La prise en charge des algorithmes cryptographiques pour les signatures pendant l'échange de clés est définie pour TLSv1.2 en 7.4 et pour TLSv1.3 en 8.3.

L'échec de la recherche d'un algorithme de signature correspondant aux composants du certificat pendant l'établissement d'une session TLS doit déclencher un événement de sécurité ("alarme: validation de certificat: algorithmes de signature non pris en charge").

L'échec de la validation de la signature des certificats reçus au cours d'un établissement de liaison TLS doit déclencher un événement de sécurité ("alarme: validation de certificat: impossible de vérifier la signature du certificat"). Il est à noter que l'indication concernant les algorithmes de signature pris en charge doit être donnée en utilisant les extensions appropriées définies en 7.5.4 pour TLSv1.2 et en 8.8.4 (algorithme de signature) et 8.8.4.2 (certificat d'algorithmes de signature) pour TLSv1.3.

6.5 Coexistence avec un trafic de protocole non sécurisé

Des normes de référence exigeant une prise en charge d'un trafic de protocole non sécurisé peuvent spécifier la façon dont la coexistence entre les communications sécurisées et non sécurisées est gérée. Cela peut être réalisé en prévoyant un accès TCP/IP séparé pour le trafic TLS sécurisé.

7 Exigences spécifiques à TLSv1.2

7.1 Généralités

Dans les versions antérieures de la présente norme, une norme de référence était exigée pour fournir des définitions supplémentaires concernant la gestion de la TLS ou la prise en charge des caractéristiques du protocole TLS.

À partir de cette édition, ces définitions ont été intégrées dans le présent document. De plus, la description de TLSv1.2 a été alignée sur la spécification des paramètres associés à la TLS et fournis dans les normes de référence à l'IEC 62351-3:2014+AMD1:2018+AMD2:2020, qui sont l'IEC 62351-4:2018+AMD1:2018 et l'IEC 62351-8:2020.

7.2 Suites chiffrées prises en charge

Pour TLSv1.2 (et les versions inférieures), le présent document spécifie les prises en charge obligatoires (o) et facultatives (f) des suites chiffrées dans le Tableau 1.

Tableau 1 – Prise en charge des suites chiffrées pour TLSv1.2

Échange de clés		Chiffrement	Hachage	Valeur IANA	Source	Soutien
Algorithme	Signature					
TLS_RSA_		WITH_NULL	SHA256	0x00,0x3B	RFC 5246	c1
TLS_RSA		WITH_AES_128_CBC_	SHA256	0x00,0x3C	RFC 5246	o
TLS_DHE_	RSA_	WITH_AES_128_GCM_	SHA256	0xC0,0x9E	RFC 5288	o
TLS_DHE_	RSA_	WITH_AES_256_GCM_	SHA384	0x00,0xA1	RFC 5288	f
TLS_ECDHE_	RSA_	WITH_AES_128_GCM_	SHA256	0xC0,0x2F	RFC 5289	o
TLS_ECDHE_	RSA_	WITH_AES_256_GCM_	SHA384	0xC0,0x30	RFC 5289	f
TLS_ECDHE_	ECDSA_	WITH_AES_128_GCM_	SHA256	0xC0,0x2B	RFC 5289	o
TLS_ECDHE_	ECDSA_	WITH_AES_256_GCM_	SHA384	0xC0,0x2C	RFC 5289	f
c1: prise en charge possible si seule la protection de l'intégrité est souhaitée. Ces suites chiffrées doivent être désactivées par défaut et exiger une activation distincte autorisée par la politique de sécurité de l'organisme.						

Des suites chiffrées supplémentaires peuvent être prises en charge. Pour la sélection de ces suites chiffrées, il convient de suivre les recommandations de l'IETF. L'interopérabilité des suites chiffrées supplémentaires dépend de la prise en charge de ces suites chiffrées par les homologues.

Le choix final des suites chiffrées utilisées dans l'environnement opérationnel dépend de la politique de sécurité de l'organisme. Comme le présent document prend en charge plusieurs suites chiffrées obligatoires, un opérateur peut opter pour ne pas utiliser de suites chiffrées ne fournissant pas de PFS, si aucune rétrocompatibilité n'est souhaitée.

Le traitement des suites chiffrées prenant en charge NULL pour le chiffrement est décrit en 6.3.

7.3 Suites chiffrées non autorisées

La liste des suites chiffrées non autorisées comprend entre autres les suites suivantes:

– TLS_NULL_WITH_NULL_NULL;
– suites chiffrées contenant TLS_*_*_MD5 (qui n'autorisent pas MD5 pour l'authentification de toute combinaison); et
– suites chiffrées contenant TLS_*_DES_* (qui n'autorisent pas DES pour le chiffrement de toute combinaison).

La signalisation d'une suite chiffrée non autorisée pendant l'établissement de session TLS est gérée différemment côté client et côté serveur:

Si seules des suites chiffrées non autorisées sont proposées dans le message `ClientHello`, un événement de sécurité doit être déclenché ("alarme: suite chiffrée TLSv1.2 non autorisée proposée") sur le serveur TLS.

Si une suite chiffrée non autorisée est signalée dans le message `ServerHello`, le client TLS doit déclencher un événement de sécurité ("alarme: suite chiffrée TLSv1.2 non autorisée proposée").

Si, dans un ensemble de suites chiffrées, une suite chiffrée non autorisée est proposée dans le message `ClientHello`, le serveur TLS doit ignorer la proposition non autorisée.

L'établissement de session doit être interrompu si seules des suites chiffrées non autorisées sont signalées d'un côté ou de l'autre.

Les clients/serveurs TLS sont susceptibles d'utiliser les suites chiffrées TLS énumérées dans le Tableau 1. Il est également autorisé à utiliser d'autres suites chiffrées, mais sans aucune garantie d'interopérabilité, à l'exception de celles non autorisées. Il convient d'utiliser d'autres suites chiffrées sur la base de l'évaluation des risques de l'opérateur.

Si la liste des suites chiffrées exclut les suites chiffrées indiquées en 7.2 et est écrasée par des politiques de sécurité locales, cela entraîne alors une non-conformité à la Partie 3 (cas d'essai négatif en 100-3 PIXIT).

7.4 Échange de clés

7.4.1 Généralités

Les mécanismes d'échange de clés sont déterminés par les suites chiffrées prises en charge, comme indiqué en 7.2.

7.4.2 Mécanismes d'échange de clés

Les mécanismes d'échange de clés à prendre en charge sont alignés sur les suites chiffrées prises en charge énumérées en 7.2 et comprennent:

– le chiffrement à clé publique en utilisant RSA;
– l'accord d'échange de clés Diffie-Hellman éphémères;
– l'accord d'échange de clés Diffie-Hellman éphémères basé sur des courbes elliptiques.

NOTE Certaines suites chiffrées définies pour TLSv1.2 prennent également en charge un mélange de signatures basées sur RSA et des échanges Diffie-Hellman basés sur des courbes elliptiques.

7.4.3 Algorithmes cryptographiques

La signature au moyen d'algorithmes RSA et ECDSA doit être prise en charge.

Pour les signatures basées sur RSA, la longueur de clé suivante doit être prise en charge:

– obligatoire: opération de signature: RSA avec une longueur de clé minimale de 2 048 bits.

L'algorithme RSA avec une longueur de clé de 2 048 bits doit être pris en charge. La longueur de clé de 2 048 bits est la longueur de clé minimale à prendre en charge pour les signatures RSA. Sur la base des recommandations NIST SP 800-57 ou BSI TR 01202-1, il est fortement recommandé de prendre également en charge la longueur de clé RSA de 3 072 bits et plus, afin de faire face aux progrès de la cryptographie. Le choix de l'algorithme de signature dépend de la politique de sécurité de l'organisme.

La prise en charge d'un RSA avec une longueur de clé de 1 024 bits est déconseillée. Son utilisation est limitée à la rétrocompatibilité. La prise en charge des clés RSA de 1 024 bits doit être désactivée par défaut et exige une activation distincte, autorisée par une politique de sécurité de l'organisme. Si l'utilisation de clés RSA de 1 024 bits est désactivée, la détection des clés RSA avec un RSA inférieur à 2 048 bits doit déclencher un événement de sécurité ("alarme: longueur de clé RSA insuffisante (< 2 048 bits)") et l'établissement de session doit être interrompu par le récepteur. Si l'utilisation de clés RSA de 1 024 bits est activée, la détection d'une clé RSA dans l'intervalle 1 024 ≤ longueur de clé RSA < 2 048 doit déclencher un événement de sécurité ("avertissement: longueur de clé minimale").

Si les clés de longueur supérieure à 1 024 bits ne peuvent pas être utilisées, il est également recommandé de prendre des mesures de sécurité supplémentaires.

Les clés RSA avec une longueur inférieure à 1 024 bits ne doivent pas être utilisées. La détection des clés RSA de moins de 1 024 bits doit déclencher un événement de sécurité ("alarme: longueur de clé insuffisante") et l'établissement de session doit être interrompu par le récepteur.

La longueur de clé suivante doit être prise en charge pour l'algorithme de clé publique ECDSA:

– obligatoire: opération de signature: longueur de clé ECDSA d'au moins 256 bits.

NOTE 1 Les recommandations concernant la longueur de clé exigée pour les algorithmes de signature sont revues constamment et peuvent être consultées dans les documents NIST SP 800-57, BSI TR 02102-1 ou NSA Suite B.

Pour les signatures basées sur des courbes elliptiques définies sur des champs de nombres premiers finis avec l'algorithme de signature ECDSA, la longueur de clé minimale est de 256 bits (en combinaison avec SHA-256). L'OID pour `ecdsa-with-SHA256` à utiliser est: iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) ecdsa-with-SHA256(2).

Les mises en œuvre revendiquant la conformité à la présente norme doivent prendre en charge l'ECDSA avec la courbe: {OID}:

`secp256r1`: {iso(1) member-body(2) us(840) ansi-X9-62(10045) curves(3) prime(1) prime256v1(7)}.

Les courbes {OID} suivantes peuvent facultativement être prises en charge pour l'ECDSA:

– `secp384r1`: {iso(1) identified-organization(3) certicom(132) curve(0) ansip384r1(34)}
– `secp521r1`: {iso(1) identified-organization(3) certicom(132) curve(0) ansip521r1(35)}
– `brainpoolP256r1`: {iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP256r1(7)}
– `brainpoolP384r1`: {iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP384r1(11)}
– `brainpoolP512r1`: {iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP512r1(13)}

SHA-256 doit être pris en charge et représente l'algorithme de hachage préférentiel à utiliser.

La prise en charge de SHA-1 est déconseillée. Son utilisation est limitée à la rétrocompatibilité. SHA-1 n'est plus reconnu comme sécurisé concernant la résistance à la collision et il est donc fortement recommandé d'effectuer une évaluation des risques avant d'utiliser cet algorithme. Si SHA-256 ne peut pas être utilisé, il est également recommandé de prendre des mesures de sécurité supplémentaires.

NOTE 2 L'IETF a publié la RFC 9155 qui déconseille les hachages de signatures MD5 et SHA-1 dans TLSv1.2.

La détection de l'utilisation de SHA-1 doit déclencher un événement de sécurité ("avertissement: utilisation d'un algorithme de hachage obsolète").

Les politiques de sécurité des entités doivent pouvoir interdire ou éviter la prise en compte de tout algorithme de hachage facultatif.

NOTE 3 Les recommandations concernant les algorithmes de hachage de signature sont revues constamment et peuvent être consultées dans les documents NIST SP 800-57, BSI TR 02102-1 ou NSA Suite B.

Pour signaler les algorithmes de signature pris en charge, l'extension "`signature_algorithm`" décrite en 7.5.4 doit être utilisée.

7.4.4 Reprise de session

La reprise de session dans TLSv1.2 permet la reprise ou le renouvellement des clés d'une session sur la base de l'ID de session connecté avec un secret principal (existant) préalablement dédié, ce qui génère une nouvelle clé de session. Cela limite l'impact sur les performances des établissements de liaison asymétriques et cette reprise peut être effectuée pendant une session en cours ou après la fin d'une session sur une période de temps définie.

TLS suggère une durée non supérieure à 24 h dans la RFC 5246 pour la durée de vie de `sessionID`. Le présent document reprend cette suggestion. Une reprise de session doit être effectuée à intervalles réguliers pour les sessions actives. Pour les sessions finies, la session peut être reprise au plus tard 24 h après (en fonction de la durée de vie de `sessionID`). Il convient de définir les paramètres actuels sur la base d'une évaluation des risques par l'opérateur.

Si la reprise de session est effectuée sur la base de l'identificateur de session (comme spécifié dans la RFC 5246) et si `sessionID` a expiré, le client et le serveur TLS procèdent à un établissement de liaison complet conformément à la section 7.3 de la RFC 5246. Dans ce cas, un événement de sécurité doit être déclenché ("avis: la session n'a pas pu être reprise, car la durée de vie de `sessionID` est expirée. Réalisation d'un établissement de liaison TLS complet à la place").

La reprise de sessions actives pour mettre à jour la clé de session est susceptible d'être plus fréquente que la renégociation de session. De ce fait, l'intervalle de reprise de session est inférieur à l'intervalle de renégociation de session. La formule suivante fournit une recommandation pour définir les valeurs respectives: 0 < intervalle de reprise de session < intervalle de renégociation de session ≤ 24 h.

Les mises en œuvre revendiquant la conformité au présent document doivent prendre en charge la configuration des intervalles de reprise de session pour les sessions actives. Il convient qu'une valeur par défaut pour l'intervalle de reprise de session soit de 6 h.

NOTE 1 Pour le raisonnement concernant les valeurs par défaut, voir aussi l'exemple informatif de 7.4.5.

Les clients doivent initier une reprise de session en utilisant le message `ClientHello`.

NOTE 2 Les précédentes éditions du présent document autorisaient le serveur à envoyer un message `HelloRequest` pour déclencher une reprise de session initiée par le serveur dans TLSv1.2. Ceci n'est pas possible selon la RFC 5246, car le message `HelloRequest` n'est destiné qu'à déclencher une renégociation de session et a donc été supprimé.

La reprise de session peut être initiée par le client, tant que les politiques de sécurité du client et du serveur le permettent. En cas d'échec de reprise d'une session, la procédure de gestion des défaillances décrite dans TLSv1.2 doit être suivie. La reprise de session peut être effectuée sur la base de l'identificateur de session (TLS native selon la RFC 5246). Autrement, la reprise de session peut être effectuée sur la base de tickets de session (RFC 5077). Cette dernière option permet d'éviter les états côté serveur des sessions, qui peuvent être reprises. Cette

option peut s'appliquer aux dispositifs sous contrainte afin d'éviter une plus grande mémoire cache de session.

Dans l'approche de reprise de session TLS basée sur les tickets, le serveur TLS génère un ticket de session. Ce ticket des informations chiffrées sur le contexte de la session, permettant au serveur TLS de recréer la session précédemment fermée ou d'établir une nouvelle session basée sur le matériel de clé de la session existante. Le contexte de session est chiffré sous forme de clé de ticket uniquement connue du serveur TLS. Il convient de renouveler régulièrement cette clé de ticket sur la base de la politique de sécurité de l'organisme. Il s'agit d'une recommandation liée à l'exploitation qui n'a pas d'influence sur l'interopérabilité. Pour s'aligner sur le temps de reprise de session admis, la durée de vie maximale des tickets émis doit être de 24 h.

NOTE 3 L'application de tickets de session pour éviter le stockage spécifique à une session du côté serveur présente des avantages dans les environnements qui coupent une connexion et se reconnectent après une durée spécifique. Si la reprise de session est utilisée pour mettre à jour la clé de session d'une session en cours, il peut n'y avoir aucun avantage.

Si la reprise de session TLS est utilisée pour la même connexion TLS, l'extension de renégociation de session TLS doit être utilisée comme décrit en 7.5.2.

Il est à noter que si la reprise de session est effectuée pour rétablir une session TLS précédemment fermée, le certificat utilisé pendant l'établissement de la session d'origine n'est pas validé au cours de la reprise de session. Ceci est géré selon une approche similaire à la validation normale des certificats après la période définie, comme indiqué en 6.4.4. Il est à noter qu'une mise en œuvre peut nécessiter de stocker les certificats homologues pour pouvoir effectuer la validation de certificat.

7.4.5 Renégociation de session

La renégociation de session dans TLSv1.2 initie un établissement de session TLS complet où toutes les opérations asymétriques pendant l'échange de clés, y compris la validation du certificat, doivent être exécutées. Comme indiqué en 6.4.4.4 et 6.4.4.5, les certificats révoqués ou expirés ne doivent pas être utilisés dans la renégociation de session.

NOTE 1 Si une application vérifie le certificat pendant une session en cours en entraînant la détection d'un certificat expiré ou révoqué, le délai jusqu'à la renégociation de session TLS suivante peut constituer un délai de grâce pour mener des actions (par exemple, la fourniture d'un justificatif d'identité valide ou la réalisation d'un contrôle de sécurité de l'IED associé). Cela est généralement géré par la politique de sécurité de l'organisme.

La renégociation de session génère une nouvelle session basée sur une clé maîtresse récemment négociée et de nouvelles clés de session. Comme cela implique spécifiquement la validation du certificat, il convient de choisir le délai de renégociation de session conformément à l'actualisation des informations concernant l'état de révocation (CRL), comme décrit en 6.4.4.4.

Les intervalles de renégociation de session doivent être configurables tant qu'ils sont dans les limites de la période maximale spécifiée et doivent être alignés sur la période de mise à jour de la CRL. Si le protocole OCSP est utilisé pour des vérifications de révocation de certificat, la renégociation de session doit être alignée sur la durée de mise en mémoire cache de la réponse OCSP. Dans tous les cas, la renégociation des connexions de longue durée doit être effectuée au moins toutes les 24 h afin d'instaurer une vérification de la validité du certificat. Des intervalles plus courts peuvent être définis par une norme de référence.

NOTE 2 Un exemple d'alignement est la moitié de la durée d'actualisation de la CRL ou la moitié de la durée de mise en mémoire cache de la réponse OCSP, afin de limiter la possibilité de certificats révoqués non détectés.

Les mises en œuvre revendiquant la conformité à la présente norme doivent prendre en charge la configuration des intervalles de renégociation de session. La session TLS doit être renégociée dans un délai maximal de 24 h. Il convient qu'une valeur par défaut pour la renégociation de session soit de 12 h.

La limite inférieure pour la renégociation de session prise en charge par une mise en œuvre doit être de 10 min.

NOTE 3 Cette exigence protège les performances du client et du serveur en empêchant un trop grand nombre d'établissements de liaison TLS complets et de validations de certificats de clé publique associées.

La renégociation de session peut être initiée par l'un ou l'autre côté tant que la politique de sécurité du client TLS et du serveur TLS les autorise à utiliser cette caractéristique. En cas d'échec de renégociation d'une session, la procédure de gestion des défaillances décrite dans TLSv1.2 doit être suivie.

L'entité appelante (client TLS) et l'entité appelée (serveur TLS) sont chargées de vérifier que la renégociation de session TLS a lieu aux intervalles prévus. Si l'entité appelante ne reçoit pas de demande de renégociation de session TLS (`HelloRequest`) de la part de l'entité appelée à l'intervalle prévu, l'entité appelante doit alors initier elle-même la renégociation de session TLS à l'aide d'un message `ClientHello`. Si l'entité appelée ne reçoit pas un message de renégociation de session TLS (`ClientHello`) en réponse à un message `HelloRequest`, l'entité appelée doit mettre fin à la connexion. Il convient que l'interruption d'une connexion due à une renégociation de session TLS manquée déclenche un événement de sécurité ("alarme: intervalle de renégociation de session expiré").

Les mises en œuvre revendiquant la conformité au présent document doivent prendre en charge le traitement de `HelloRequest`.

NOTE 4 Selon la RFC 5246, `HelloRequest` est un message facultatif que le serveur peut envoyer à un client.

NOTE 5 Il est prévu que le client et le serveur soient configurés avec la même politique de sécurité TLS.

Pour éviter des faiblesses dans la renégociation de session, l'extension de la renégociation de session définie dans la RFC 5746 doit être utilisée comme décrit en 7.5.1.

Il est à noter que l'exemple informatif suivant est fourni pour configurer la renégociation de session et la reprise de session:

– la durée d'actualisation estimée de la CRL (ou la validité de la réponse OCSP) est de 24 h;
– la renégociation de session implique la validation des certificats homologues, y compris la vérification de révocation. La renégociation de session impliquant les certificats homologues pour authentification peut être réalisée au moins toutes les 12 h;
– pour permettre une mise à jour des clés de session pendant l'intervalle de 12 h entre deux renégociations de session, une reprise de session est réalisée toutes les 6 h pendant la session. Le temps maximal accordé pour reprendre une session précédemment clôturée est de 24 h.

7.5 Prise en charge des extensions

7.5.1 Généralités

Le présent paragraphe définit les extensions de l'établissement de liaison TLS dont la prise en charge par les mises en œuvre conformes est obligatoire ou facultative. Chaque paragraphe fait référence à la spécification relative à l'extension considérée.

7.5.2 Extension de la renégociation de session TLS

La définition de TLSv1.2 et des versions inférieures n'a pas envisagé de créer une liaison cryptographique entre une session initiale et une session renégociée. Cette liaison manquante peut permettre à un attaquant d'injecter un contenu lors d'un établissement de connexion TLS (également connu sous le nom d'attaque à triple établissement de liaison). Pour faire face à cette attaque, la RFC 5746 définit une extension de renégociation qui rattache la session TLS renégociée à la session TLS, dans laquelle la renégociation est effectuée. Les mises en œuvre conformes doivent prendre en charge l'extension de renégociation de session.

Selon la RFC 5746, un client TLS inclut l'extension "renegotiation_info" dans le message ClientHello initial. Pendant l'établissement de liaison initial, cette extension est vide et signale la prise en charge du client en vue d'une renégociation sécurisée. De même, le serveur TLS inclut une extension "renegotiation_info" vide dans le message ServerHello s'il a détecté les informations provenant du client et s'il prend en charge la renégociation de session. Dans le contexte de la présente norme, la prise en charge de cette extension est obligatoire. Si l'un ou l'autre des côtés détecte l'absence de l'extension renegotiation_info dans l'établissement de liaison initial, il doit déclencher un événement de sécurité ("avertissement: renégociation de session sécurisée non prise en charge (établissement de liaison initial)"). Une renégociation de session ne doit pas être effectuée sans utiliser cette extension.

Selon la RFC 5746, un client TLS inclut les données de vérification du client "client_verify_data" envoyées dans le message Finish du précédent établissement de liaison dans l'extension renegotiation_info contenue dans le message ClientHello renégocié. Le serveur TLS inclut les données "client_verify_data", ainsi que "server_verify_data" dans l'extension, lors de l'envoi du message ServerHello correspondant. Si l'un ou l'autre des côtés détecte l'absence de l'extension renegotiation_info ou l'un de ses composants dans l'établissement de liaison renégocié, pour lequel la prise en charge a été signalée dans l'établissement de liaison initial, il doit déclencher un événement de sécurité ["alarme: renégociation de session sécurisée non prise en charge (établissement de liaison renégocié)"]. La session doit être abandonnée. Les mises en œuvre revendiquant la conformité à la présente spécification doivent prendre en charge cette extension conformément à la RFC 5746.

Conformément à l'extension de renégociation de session de la RFC 5746, section 3.1, s'applique aux établissements de liaison complets et aux reprises d'établissement de liaison dans la même connexion TLS. Ainsi, le paramètre client_verify_data et le server_verify_data sont spécifiques à la connexion et ne font pas partie de la mémoire cache de session TLS. Si la reprise de session TLS est effectuée pour une connexion TLS différente, l'extension de renégociation de session TLS ne peut pas être utilisée. Pour la reprise de session TLS de la même connexion TLS, elle doit être utilisée.

NOTE L'utilisation d'extensions est typiquement effectuée par la pile TLS.

7.5.3 Signalisation des certificats de CA pris en charge par le client par l'intermédiaire d'une CA de confiance

L'extension d'indication de la CA de confiance spécifiée dans la RFC 6066 (section 6) permet à un client TLS de donner des informations relatives aux certificats de CA pris en charge localement, car la CA racine des centres de services publics peut ne pas être publique. L'extension permet au demandeur d'influer sur la sélection du certificat X.509 du côté IED pour l'authentification du côté serveur afin de permettre la vérification du certificat X.509 utilisé du côté demandeur.

L'indication de la CA de confiance est contenue dans le message ClientHello. Le champ "extension_data" de cette extension doit être vide. Un serveur TLS recevant une indication de CA de confiance peut utiliser cette information pour guider sa sélection d'une chaîne de certificats appropriée à renvoyer au client. Selon la RFC 6066, le serveur doit, dans ce cas, inclure une extension de type "trusted_ca_keys" dans le message ServerHello (étendu).

Il convient que les mises en œuvre revendiquant la conformité à la présente norme prennent en charge cette extension.

La prise en charge de cette extension peut s'appliquer aux cas où différents domaines administratifs ont accès aux IED, par exemple deux centres de services publics avec leur propre infrastructure de clé publique. Si différents domaines administratifs sont à prendre en charge, il convient d'utiliser l'extension d'indication de la CA de confiance de la TLS.

Les mises en œuvre revendiquant la conformité à la présente norme utilisant cette extension peuvent spécifier la sélection des certificats émis par la CA demandés du côté serveur TLS.

L'échec de la sélection d'un certificat correspondant émis par la CA doit déclencher un événement de sécurité ("alarme: certificat de CA correspondant pris en charge côté client TLSv1.2 non trouvé").

7.5.4 Signalisation des algorithmes de signature pris en charge

La RFC 5246 définit l'extension d'algorithme de signature pour permettre à un client d'indiquer les combinaisons de hachage et de signature qui peuvent être utilisées dans les signatures numériques pour les certificats et les opérations d'établissement de liaison TLS.

Les clients utilisant TLSv1.2 doivent inclure l'extension `signature_algorithm` avec au moins la combinaison `{SHA-1, RSA; SHA-256, RSA; SHA-256, ECDSA}` dans le message `ClientHello`. La combinaison SHA-1 et RSA est uniquement destinée à la rétrocompatibilité (voir également ci-dessous) et dépend de la politique de sécurité de l'organisme (voir également 7.4.3). D'autres combinaisons peuvent être prises en charge.

NOTE Si l'extension `signature_algorithm` n'est pas incluse, le serveur TLSv1.2 revient à la valeur par défaut, c'est-à-dire qu'il applique toujours SHA-1 pour le hachage.

La détection d'une extension `signature_algorithm` manquante doit déclencher un événement de sécurité ("avertissement: extension d'algorithme de signature manquante").

SHA-256 doit être pris en charge et représente l'algorithme de hachage préférentiel à utiliser.

La prise en charge de SHA-1 est déconseillée. Son utilisation est limitée à la rétrocompatibilité. SHA-1 n'est plus reconnu comme sécurisé concernant la résistance à la collision et il est donc fortement recommandé d'effectuer une évaluation des risques avant d'utiliser cet algorithme. Si SHA-256 ne peut pas être utilisé, il est également recommandé de prendre des mesures de sécurité supplémentaires. L'utilisation de SHA-1 n'est pas autorisée dans la prochaine édition de la présente norme.

La détection d'une combinaison {SHA-1, RSA} dans l'extension `signature_algorithm` doit déclencher un événement de sécurité ("avertissement: algorithme de signature obsolète signalé").

Les politiques de sécurité des entités doivent pouvoir interdire ou éviter la prise en compte de tout algorithme de hachage facultatif.

NOTE Les recommandations concernant les algorithmes de hachage de signature sont revues constamment et peuvent être consultées dans les documents NIST SP 800-57, BSI TR 02102-1 ou NSA Suite B.

La mise en œuvre revendiquant la conformité à la présente spécification doit prendre en charge cette extension conformément à la RFC 6066.

7.5.5 Agrafage des messages de réponse OCSP

7.5.5.1 Généralités

L'agrafage de la réponse OCSP permet à un serveur de fournir à un client les informations de révocation des certificats côté serveur dans le cadre de l'établissement de liaison TLS. L'utilisation de ce mécanisme peut s'avérer utile si le client n'est pas en mesure de récupérer les informations de révocation par l'intermédiaire de la CRL ou du protocole OCSP.

Pour TLSv1.2, il existe deux extensions différentes qui peuvent être facultativement prises en charge et sont décrites en 7.5.5.2 et 7.5.5.3. Pour la spécification des mécanismes, les documents correspondants sont cités en référence.

7.5.5.2 Agrafage OCSP

La RFC 6066 (section 8) définit l'agrafage de la réponse OCSP pour le certificat côté serveur.

Si un client TLS souhaite recevoir une réponse OCSP, il peut inclure l'extension "`status_request`" dans le champ `ExtensionType` du message `ClientHello`. Les données de l'extension sont "`CertificateStatusRequest`", contenant le type de statut dans "`OCSPStatusRequest`" et une liste des répondeurs OCSP auxquels le client fait confiance. Si cette liste est vide, le répondeur OCSP est implicitement connu du serveur. Il existe également d'autres options pour le client TLS, comme l'inclusion d'un nonce dans l'extension, qui est utilisé par le client OCSP pour interroger l'état de révocation.

Un serveur TLS recevant un message `ClientHello` avec une extension "`status_request`" peut renvoyer la réponse OCSP sous la forme "`OCSPResponse`" dans le message `CertificateStatus`, qui est un nouveau message d'établissement de liaison défini par la RFC 6066. Ce message suit immédiatement le message "`Certificat`" envoyé par le serveur TLS.

Une mise en œuvre revendiquant la conformité au présent document doit prendre en charge l'extension status_request_v2 conformément à la RFC 6066.

Il est à noter que cela n'exige pas nécessairement la prise en charge d'interactions OCSP à l'aide d'un répondeur OCSP.

7.5.5.3 Multi-agrafage OCSP

Alors que l'agrafage OCSP défini dans la RFC 6066 offre une option permettant de fournir une réponse OCSP pour le certificat du serveur, la RFC 6961 définit le multi-agrafage des réponses OCSP. Cette caractéristique prend également en charge la fourniture des réponses OCSP pour les CA intermédiaires dans la chaîne de certificats. Ceci est réalisé en fournissant des structures d'informations supplémentaires, ainsi que des améliorations aux champs `ExtensionType` et `CertificateStatusType` définis à l'origine dans la RFC 6066.

Si un client TLS souhaite recevoir plusieurs réponses OCSP, il peut inclure l'extension "`status_request_v2`" dans le champ `ExtensionType` du message `ClientHello`. Les données de l'extension sont "`CertificateStatusRequestItemV2`", contenant le type de statut dans "`OCSPStatusRequest`", et une liste des répondeurs OCSP auxquels le client fait confiance. Si cette liste est vide, le répondeur OCSP est implicitement connu du serveur. Il existe également d'autres options pour le client, comme l'inclusion d'un nonce dans l'extension, qui est utilisé par le client OCSP pour interroger l'état de révocation.

Un serveur TLS recevant un message `ClientHello` avec une extension "`status_request_v2`" peut renvoyer les réponses OCSP dans "`OCSPResponseList`" dans le message `CertificateStatus`, qui a été initialement défini par la RFC 6066. Ce message suit directement le message "`Certificat`" envoyé par le serveur TLS.

Il convient qu'une mise en œuvre prenne en charge cette extension conformément à la RFC 6961.

Il est à noter que cela n'exige pas nécessairement la prise en charge d'interactions OCSP à l'aide d'un répondeur OCSP.

7.5.6 Signalisation du serveur TLS cible prévu par l'indication du nom de serveur

La RFC 6066 (section 3) définit l'extension "`server_name`", qui peut être utilisée par un client TLS dans un message `ClientHello`. Cette extension peut être utilisée par le client TLS pour faciliter les situations où plusieurs serveurs TLS (virtuels) sont hébergés à une seule adresse de réseau. La signalisation du nom de serveur cible prévu permet au serveur de choisir le certificat approprié à renvoyer au client ou pour traiter d'autres aspects concernés par une politique de sécurité.

Il convient qu'une mise en œuvre revendiquant la conformité au présent document prenne en charge cette extension conformément à la RFC 6066.

7.5.7 Prise en charge du chiffrement avant authentification

Dans TLSv1.2, l'approche par défaut pour la sécurité de la couche d'enregistrement est le mécanisme "MAC-then-encrypt". Comme les attaques dirigées contre les suites chiffrées CBC sont connues (par exemple LUCKY13, POODLE), lesquelles tirent parti de l'ordre des opérations, une extension a été définie pour inverser l'ordre jusqu'au premier chiffrement, puis procéder à l'authentification.

La RFC 7366 définit l'extension "`encrypt_then_mac`", qui peut être utilisée par un client TLS dans un message `ClientHello` pour signaler la prise en charge. De même, le serveur TLS inclut également cette extension dans le message `ServerHello`.

Étant donné que le présent document spécifie également la prise en charge des suites chiffrées basées sur CBC au paragraphe 7.2, les mises en œuvre doivent prendre en charge cette extension comme spécifié dans la RFC 7366.

Si l'extension "`encrypt_then_mac`" n'est pas incluse dans les messages `ClientHello` ou `ServerHello`, alors qu'une suite chiffrée TLS (`CipherSuite`) contenant un algorithme de chiffrement en mode CBC est signalée dans la `CipherSuite` (`ClientHello`) prise en charge ou dans la suite chiffrée sélectionnée (`ServerHello`), un événement de sécurité doit être déclenché ("alarme: session TLS avec mécanisme MAC-then-encrypt détectée") et la session doit être interrompue.

8 Exigences spécifiques à TLSv1.3

8.1 Généralités

Le présent article décrit l'obligation de prendre en charge les suites chiffrées, les approches en matière d'échange de clés et les réglages de profil pour TLSv1.3 afin de sécuriser la communication pour l'automatisation des systèmes de puissance.

8.2 Suites chiffrées prises en charge

Pour TLSv1.3, le présent document spécifie les prises en charge obligatoires et facultatives des suites chiffrées dans le Tableau 2.

Tableau 2 – Prise en charge des suites chiffrées pour TLSv1.3

Suite chiffrée	Valeur IANA	Source	Prise en charge (client/serveur)
TLS_AES_128_GCM_SHA256	0x13, 0x01	RFC 8446	o
TLS_AES_256_GCM_SHA384	0x13, 0x02	RFC 8446	o
TLS_CHACHA20_POLY1305_SHA256	0x13, 0x03	RFC 8446	f
TLS_AES_128_CCM_SHA256	0x13, 0x04	RFC 8446	o
TLS_AES_128_CCM_8_SHA256	0x13, 0x05	RFC 8446	f
TLS_SHA256_SHA256	0xC0, 0xB4	RFC 9150	c1
TLS_SHA384_SHA384	0xC0, 0xB5	RFC 9150	c1
c1: prise en charge possible si seule la protection de l'intégrité est souhaitée. Ces suites chiffrées doivent être désactivées par défaut et exiger une activation distincte autorisée par la politique de sécurité de l'organisme.			

Une suite chiffrée qui ne spécifie pas un chiffrement AEAD pour le chiffrement ne doit pas être utilisée pour la communication à l'extérieur du domaine administratif, si le chiffrement de cette connexion de communication par d'autres moyens ne peut pas être garanti.

NOTE 1 Le présent document n'exclut pas l'utilisation de communications chiffrées à l'aide de tunnels VPN cryptographiques. L'utilisation de tels VPN n'entre pas dans le domaine d'application de la présente norme.

Au sein d'un domaine administratif, l'utilisation de suites chiffrées sans chiffrement peut être autorisée, mais son utilisation relève de la responsabilité de l'opérateur.

NOTE 2 L'application de suites chiffrées sans chiffrement permet le contrôle du trafic tout en continuant à maintenir une authentification entre extrémités et une protection de l'intégrité du trafic.

Les mises en œuvre permettant des suites chiffrées TLS sans chiffrement revendiquant la conformité au présent document doivent fournir un mécanisme pour permettre explicitement ces suites chiffrées TLS. Par défaut, les suites chiffrées TLS sans chiffrement ne doivent pas être autorisées.

8.3 Échange de clés

8.3.1 Généralités

Les exigences de la RFC 8446 concernant la prise en charge obligatoire des algorithmes de signature pour les messages d'établissement de liaison et les certificats ont été reproduites ici. Ceci concerne spécifiquement:

– la prise en charge obligatoire des signatures numériques avec rsa_pkcs1_sha256 (pour les certificats), rsa_pss_rsae_sha256 (pour le message `CertificateVerify` et les certificats), et ecdsa_secp256r1_sha256;

– la prise en charge obligatoire de l'échange de clés basé sur la courbe elliptique secp256r1.

Pour renouveler les clés de session, TLSv1.3 prend en charge différents mécanismes nouveaux par rapport à TLSv1.2:

– un nouveau message `KeyUpdate` a été défini comme message de post-établissement de liaison pour la mise à jour du matériel de clé cryptographique de l'expéditeur (clés et vecteurs d'initialisation). Cette fonctionnalité est utilisée pour mettre à jour la clé de session de chaque côté pendant la session. Elle a été réalisée dans TLSv1.2 en utilisant la reprise de session (pour mettre uniquement à jour les clés de session) et avec la renégociation de session (pour mettre à jour les clés de session et pour valider également les certificats des deux homologues);

– de nouveaux modes d'établissement de liaison de clés prépartagées permettent l'utilisation de PSK établies au cours de la dernière session TLS ou fournies par des moyens externes pour négocier de nouvelles clés de session sans impliquer de certificats dans le contexte de la configuration de session. Ceci est comparable à la reprise de session sans état côté serveur (basé sur les tickets) dans TLSv1.2;
– la renégociation de session est interdite conformément à la RFC 8446. Pour TLSv1.2, la renégociation de session est utilisée dans l'IEC 62351-3 pour appliquer une authentification par certificat des deux côtés après un certain temps, qui est à aligner sur la mise à jour des informations de révocation de certificat. Contrairement à l'utilisation de TLSv1.2, l'invocation de la validation du certificat est à gérer par l'application spécifiquement pour les sessions de longue durée (plus longues que la durée d'actualisation de la CRL) comme indiqué en 6.4.4.4.

8.3.2 Modes d'établissement de liaison

TLSv1.3 prend en charge différents modes d'établissement de liaison. La prise en charge des échanges de clés Diffie-Hellman (DHE, ECDHE) est obligatoire et ces échanges sont utilisés par défaut.

TLSv1.3 prend également en charge les modes PSK énumérés dans le Tableau 3 pour permettre une configuration de session plus rapide, en prenant en charge la reprise de session et/ou l'établissement de connexions TLS parallèles.

Indépendamment du mode d'établissement de liaison utilisé, la fourniture de données d'application chiffrées pendant l'établissement de liaison TLS (jusqu'à ce que le message fini du client TLS soit reçu par le serveur TLS) ne doit pas être utilisée. Cela concorde également avec la section 2 de la RFC 8446.

Il est à noter que pour TLSv1.2, l'IEC 62351 utilise la reprise de session pour appliquer une mise à jour de clé de session ou pour établir une seconde connexion TLS, qui est rattachée à l'établissement de liaison complet de la connexion initiale. Dans TLSv1.3, il existe un mécanisme distinct pour la mise à jour de la clé de session pendant une session TLS en cours, qui peut être initié d'un côté ou de l'autre. Par conséquent, la reprise de session avec TLSv1.3 est uniquement destinée à établir une autre connexion sécurisée, qui est rattachée à la session TLS initiale entre les mêmes homologues.

Tableau 3 – Prise en charge des modes d'établissement de liaison basés sur PSK pour TLSv1.3

Modes d'établissement de liaison	Référence	Valeur IANA	Prise en charge (client/serveur)
psk_ke	RFC 8446	0	x
psk_dhe_ke	RFC 8446	1	o
NOTE Lors de l'utilisation de psk_ke, aucune confidentialité de transmission parfaite n'est assurée.			

Sur la base de la PSK, TLSv1.3 active un établissement de liaison 0-RTT, qui permet le transport de données chiffrées (indiquées par l'extension `early_data`) avec le premier message (`ClientHello`). Comme la PSK est fournie hors bande ou est le résultat d'une session précédente, la confidentialité de transmission parfaite ne peut pas être assurée. Dans le contexte de la présente spécification, 0-RTT ne doit pas être utilisé.

Pour assurer une confidentialité de transmission parfaite pour la clé de session à configurer, `psk_dhe_ke` doit être pris en charge par des mises en œuvre revendiquant la conformité au présent document.

L'option d'utiliser uniquement `psk_ke` n'exige pas que le serveur effectue un partage de clé et n'entraîne pas de propriétés de confidentialité de transmission parfaite pour la clé de session

convenue. Dans le contexte de la présente spécification, le mode d'établissement de liaison `psk-ke` ne doit pas être utilisé.

Si un client TLS signale la prise en charge de `psk_ke` uniquement, un événement de sécurité doit être généré ("alarme: prise en charge du mode PSK non éphémère uniquement"). La session doit être interrompue.

8.3.3 Groupes Diffie-Hellman

TLSv1.3 permet aux homologues de communication de spécifier les groupes pris en charge pour (EC)DHE. Dans le contexte du présent document, les groupes Diffie-Hellman marqués comme obligatoires dans le Tableau 4 doivent être pris en charge. Les groupes facultatifs peuvent être pris en charge.

Tableau 4 – Prise en charge des groupes Diffie-Hellman pour TLSv1.3

Groupe Diffie-Hellman	Référence	Valeur IANA	Prise en charge (client/serveur)
secp256r1	RFC 8422	23	o
secp384r1	RFC 8422	24	f
brainpoolP256r1	RFC 8734	31	f
brainpoolP384r1	RFC 8734	32	f
brainpoolP512r1	RFC 8734	33	f
ffdhe2048	RFC 7919	256	o
ffdhe3072	RFC 7919	257	f
ffdhe4096	RFC 7919	258	f

8.3.4 Algorithmes de signature

8.3.4.1 Algorithmes de signature pour les messages d'établissement de liaison

Par rapport à TLSv1.2, TLSv1.3 permet d'indiquer les algorithmes de signature souhaités de façon plus spécifique que TLSv1.2. Ici, deux extensions sont proposées (voir aussi 8.3.4.2). L'extension `signature_algorithm` spécifie les algorithmes pris en charge et à utiliser dans le message `CertificateVerify`. Les algorithmes indiqués dans le Tableau 5 sont à prendre en charge de manière obligatoire et facultative pour une utilisation avec l'extension `signature_algorithm` dans le contexte du présent document.

Tableau 5 – Algorithmes de signature pris en charge pour l'établissement de liaison dans TLSv1.3

Algorithmes de signature	Référence	Valeur IANA	Prise en charge (client/serveur)
rsa_pss_rsae_sha256	RFC 8446	0x0804	o
rsa_pss_rsae_sha384	RFC 8446	0x0805	f
rsa_pss_rsae_sha512	RFC 8446	0x0806	f
rsa_pss_pss_sha256	RFC 8446	0x0809	o
rsa_pss_pss_sha384	RFC 8446	0x080A	f
rsa_pss_pss_sha512	RFC 8446	0x080B	f
ecdsa_secp256r1_sha256	RFC 8446	0x0403	o
ecdsa_secp384r1_sha384	RFC 8446	0x0503	f
ecdsa_brainpoolP256r1_sha256	RFC 8734	0x081A	f
ecdsa_brainpoolP384r1_sha384	RFC 8734	0x081B	f
ecdsa_brainpoolP512r1_sha512	RFC 8734	0x081C	f

8.3.4.2 Algorithmes de signature pour les certificats

TLS1.3 permet d'indiquer les algorithmes de signature souhaités à utiliser pour les signatures des certificats dans l'extension `signature_algorithms_cert`. Selon la RFC 8446, si aucune extension "`signature_algorithms_cert`" n'est incluse, l'extension "`signature_algorithms`" (voir 8.3.4.1) s'applique également aux signatures des certificats. Les algorithmes indiqués dans le Tableau 6 doivent être pris en charge de manière obligatoire et facultative pour une utilisation avec `signature_algorithms_cert`:

Tableau 6 – Algorithmes de signature pris en charge pour les certificats dans TLSv1.3

Algorithmes de signature	Référence	Valeur IANA	Prise en charge (client/serveur)
rsa_pkcs1_sha256	RFC 8446	0x0401	o
rsa_pkcs1_sha384	RFC 8446	0x0501	f
rsa_pkcs1_sha512	RFC 8446	0x0601	f
rsa_pss_rsae_sha256	RFC 8446	0x0804	o
rsa_pss_rsae_sha384	RFC 8446	0x0805	f
rsa_pss_rsae_sha512	RFC 8446	0x0806	f
rsa_pss_pss_sha256	RFC 8446	0x0809	o
rsa_pss_pss_sha384	RFC 8446	0x080A	f
rsa_pss_pss_sha512	RFC 8446	0x080B	f
ecdsa_secp256r1_sha256	RFC 8446	0x0403	o
ecdsa_secp384r1_sha384	RFC 8446	0x0503	f
ecdsa_brainpoolP256r1_sha256	RFC 8734	0x081A	f
ecdsa_brainpoolP384r1_sha384	RFC 8734	0x081B	f
ecdsa_brainpoolP512r1_sha512	RFC 8734	0x081C	f
NOTE Le BSI allemand recommande de n'utiliser RSA (valeurs IANA 0x0401, 0x0501, 0x0601) que jusqu'en 2025, car les problèmes de bourrage dans le PKCS #1 utilisé sont connus.			