Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

Technische Richtlinie BSI TR-03116 Kryptographische Vorgaben für Projekte der Bundesregierung

Teil 3: Intelligente Messsysteme

Stand 2023

Datum: 6. Dezember 2022

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: smartmeter@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die Technische Richtlinie BSI TR-03116 stellt eine Vorgabe für Projekte des Bundes dar. Die Technische Richtlinie ist in sechs Teile gegliedert:

- Teil 1 der Technischen Richtlinie beschreibt die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren im Gesundheitswesen für die elektronische Gesundheitskarte (eGK), den Heilberufeausweis (HBA) und der technischen Komponenten der Telematikinfrastruktur.
- Teil 2 beschreibt die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in hoheitlichen Dokumenten und eID-Dokumenten basierend auf Extended Access Control, zurzeit für den elektronischen Reisepass, den elektronischen Personalausweis, den elektronischen Aufenthaltstitel, die eID-Karte für Unionsbürger, die Smart-eID, Änderungsaufkleber hoheitlicher Dokumente, VISA-Aufkleber und den Ankunftsnachweis.
- Im vorliegenden Teil 3 werden die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in der Infrastruktur intelligenter Messsysteme im Energiesektor beschrieben.
- Teil 4 der Technischen Richtlinie beschreibt die Sicherheitsanforderungen für den Einsatz der Kommunikationsverfahren SSL/TLS, S/MIME, SAML/XML Security und OpenPGP in Anwendungen des Bundes.
- Teil 5 der Technischen Richtlinie beschreibt die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in Anwendungen der Secure Element API (wie Technischen Sicherheitseinrichtungen elektronischer Aufzeichnungssysteme).
- Teil 6 der Technischen Richtlinie beschreibt die Sicherheitsanforderungen für den Einsatz kryptographischer Verfahren in der Infrastruktur kooperativer intelligenter Verkehrssysteme (Cooperative Intelligent Transport Systems / C-ITS).

1.1 Geltungsbereich

Die Anforderungen an die Funktionalität, Interoperabilität und Sicherheit der Komponenten von Smart-Metering-Systemen werden in der Technischen Richtlinie TR-03109 [1] spezifiziert.

Basierend auf den Technischen Richtlinien TR-02102 [2] und TR-03111 [3] werden in diesem Dokument verbindlich die einzusetzenden kryptographischen Verfahren und Primitive sowie zu verwendenden Schlüssellängen für die Absicherung der Infrastruktur von intelligenten Messsystemen und der Marktkommunikation[1] vorgegeben.

Tabelle 1 gibt einen Überblick über die verwendeten Verfahren und ihren Einsatzzweck.

1 Die Bezugnahme auf den Begriff der Marktkommunikation erfolgt in diesem Dokument jeweils vorbehaltlich einer noch zu treffenden Festlegung der Bundesnetzagentur.

Einsatzzweck	*Verfahren*
Sicherstellung der Authentizität von öffentlichen Schlüsseln (vgl. TR-03109-4 [4])	Public Key Infrastruktur (vgl. Abschnitt 3)
Absicherung der Kommunikation zwischen Kommunikationspartnern der Smart-Meter-PKI auf Transportebene im WAN (vgl. TR-03109 [1]) und in der Marktkommunikation[1]	TLS (vgl. Abschnitt 4)
Absicherung der Kommunikation zwischen Smart-Meter-Gateway und Teilnehmern im HAN (vgl. TR-03109 [1])	TLS (vgl. Abschnitt 5)
Absicherung der Kommunikation von Zählern mit dem Smart-Meter-Gateway im LMN (vgl. TR-03109 [1])	TLS (vgl. Abschnitt 6)
Absicherung der Kommunikation von Zählern mit dem Smart-Meter-Gateway im LMN für Daten mit niedrigem Schutzbedarf (vgl. TR-03109 [1])	Symmetrische Kryptographie (vgl Abschnitt 7)
Vertrauliche, authentische Ende-zu-Ende-Übertragung von Daten über das WAN an den Endempfänger auf Inhaltsebene (vgl. TR-03109 [1])	CMS (vgl. Abschnitt 8)
Vertrauliche, authentische Ende-zu-Ende-Übertragung von Daten in der Marktkommunikation auf Inhaltsebene[1]	XML-Security (vgl. Abschnitt 9)
Gegenseitige Authentisierung zwischen Smart-Meter-Gateway und Sicherheitsmodul sowie Aufbau eines sicheren Kanals	PACE (vgl. Abschnitt 10)
Vertrauliche, authentische Kommunikation zwischen Smart-Meter-Gateway und Sicherheitsmodul	Secure Messaging (vgl. Abschnitt 10)

Tabelle 1: Verfahren zur Absicherung der Infrastruktur von Messsystemen

1.2 Begrifflichkeiten

1.2.1 Verwendungszeiträume

Die Vorgaben des vorliegenden Teils 3 der Technischen Richtlinie basieren auf Prognosen über die Sicherheit der verwendeten kryptographischen Verfahren und Schlüssellängen über einen Zeitraum von 7 Jahren, zur Zeit bis einschließlich 2029. Ist eine weitere Verwendung über diesen Zeitraum hinweg nicht ausgeschlossen, so wird dies mit 2029+ gekennzeichnet.

Diese Technische Richtlinie führt auch Vorgaben mit abgelaufenem Verwendungszeitraum auf, sofern entsprechende Anwendungen möglicherweise weiterhin im Feld bzw. in Nutzung sind.

Wird ein Verwendungsende ohne „+“ angegeben, so sollte ein Migrationsplan für die Ausphasung der entsprechenden Anwendung erstellt werden.

Liegt ein Verwendungsanfang in der Zukunft, so sollten rechtzeitig angemessene Tests konzipiert und durchgeführt werden, durch die überprüft werden kann, ob alle notwendigen Voraussetzungen für die Einführung erfüllt sind. Der Testzeitpunkt sollte so gewählt sein, dass mögliche aufgedeckte Mängel noch vor dem Verwendungsanfang behoben werden können.

1.2.2 Schlüsselworte

Anforderungen als Ausdruck normativer Festlegungen werden durch die in Großbuchstaben geschriebenen deutschen Schlüsselworte MUSS/MÜSSEN, DARF NICHT/DÜRFEN NICHT, SOLLTE/SOLLTEN, SOLLTE NICHT/SOLLTEN NICHT, EMPFOHLEN, KANN/KÖNNEN, und OPTIONAL gekennzeichnet. Die verwendeten Schlüsselworte sind auf Basis der Übersetzungen in Tabelle 2 gemäß RFC 2119 [5] zu interpretieren.

Deutsch	***Englisch***
MUSS / MÜSSEN	MUST
DARF NICHT / DÜRFEN NICHT	MUST NOT
VERPFLICHTEND	REQUIRED
SOLLTE / SOLLTEN	SHOULD
SOLLTE NICHT / SOLLTEN NICHT	SHOULD NOT
EMPFOHLEN	RECOMMENDED
KANN / KÖNNEN / DARF / DÜRFEN	MAY
OPTIONAL	OPTIONAL

Tabelle 2: Schlüsselworte

2 Kryptographische Algorithmen

2.1 Kryptographische Basisverfahren

Tabelle 3 gibt eine Übersicht über die kryptographischen Primitive, die in diesem Dokument verwendet werden.

Digitale Signatur	ECDSA [3]
Schlüsseleinigung	ECKA-DH [3]
Schlüsseltransport	ECKA-EG [3]
Blockchiffre	AES [6] • CBC-Mode [7] • GCM-Mode [8]
MAC-Algorithmus	AES-CMAC [9]
Hashfunktionen	SHA-2 Familie [10]

Tabelle 3: Kryptographische Primitive

2.2 Domainparameter für elliptische Kurven

Für kryptographische Algorithmen und Protokolle basierend auf elliptischen Kurven (d.h. TLS, ECDSA und ECKA) werden NIST-Domain-Parameter über Primkörpern nach RFC 5114 [11] bzw. Brainpool-Domain-Parameter nach RFC 5639 [12] in den entsprechenden Bitlängen verwendet.

Um eine einfache Migration auf andere Verfahren zu ermöglichen, MUSS das Sicherheitsmodul eines Smart-Meter-Gateways

- die Schlüsselerzeugung
- PACE, ECKA-DH, ECKA-EG, ECDSA Signaturerzeugung und -verifikation

gemäß den Vorgaben in der TR-03111 [3] für alle Kurvenparameter aus Tabelle 4 unterstützen. Die Vorgaben beziehen sich auf den Zeitpunkt der Herstellung des Smart-Meter-Gateways.

Kurvenparameter	*Verwendung von*	*Verwendung bis*
brainpoolP256r1 [12]	2015	2029+
brainpoolP384r1 [12]	2015	2029+
brainpoolP512r1 [12]	2015	2029+
NIST P-256 (secp256r1) [11]	2015	2029+
NIST P-384 (secp384r1) [11]	2015	2029+

Tabelle 4: Vom Sicherheitsmodul zu unterstützende Kurvenparameter

Als Encoding für die Punkte der elliptischen Kurven MUSS das Uncompressed Encoding gemäß TR-03111 [3] verwendet werden.

2.3 Zufallszahlen

Für die Erzeugung von Zufallszahlen und kryptographischen Schlüsseln (inkl. Ephemerschlüsseln) MUSS in jedem der verwendeten kryptographischen Protokolle ein Zufallszahlengenerator aus einer der folgenden Klassen (siehe AIS 20/31 [13]) verwendet werden:

- DRG.4,
- PTG.3,
- NTG.1.

Abweichend von obigen Vorgaben durfte in bis Ende 2022 hergestellten Geräten auch ein Zufallszahlengenerator der Klasse DRG.3 verwendet werden.

Bei der Erzeugung von unvorhersagbaren Initialisierungsvektoren für symmetrische Verschlüsselungsverfahren im CBC-Mode MÜSSEN die Anforderungen aus der TR-02102 [2] beachtet werden, sofern nicht explizit Abweichendes genannt wird.

2.4 Umgang mit Ephemerschlüsseln

Ephemerschlüssel und Sitzungsschlüssel MÜSSEN nach ihrer Verwendung unwiderruflich gelöscht werden. Ephemer- bzw. Sitzungsschlüssel DÜRFEN NICHT für mehr als *eine* Sitzung benutzt werden oder persistent abgespeichert werden. Dies gilt insbesondere für Master-Secret und Pre-Master-Secret bei TLS (vgl. Kapitel 4) sowie Content bzw. Key Encryption Keys bei CMS (vgl. Kapitel 8) und XML Security (vgl. Kapitel 9).

3 Public Key Infrastruktur

Die Authentizität der öffentlichen Schlüssel von Kommunikationspartnern im WAN und der Marktkommunikation, welche zur gegenseitigen Authentisierung und zum Aufbau eines verschlüsselten, integritätsgesicherten TLS-Kanals bzw. zur Verschlüsselung oder Signatur von Daten auf Inhaltsebene eingesetzt werden, wird durch die Smart Metering Public Key Infrastruktur (SM-PKI) sichergestellt. Die SM-PKI wird in der TR-03109-4 [4] und in der Certificate Policy der SM-PKI [14] spezifiziert.

Die SM-PKI besteht demnach aus einer *Root-CA* als nationale Wurzelinstanz, *Sub-CAs* für die Ausstellung der Endnutzerzertifikate sowie den *Endnutzerzertifikaten*. Zu den Endnutzern gehören insbesondere die Marktteilnehmer, die Gateway-Administratoren und die Smart-Meter-Gateways (vgl. [14]).

Als Signaturverfahren, mit dem die X.509-Zertifikate und -Sperrlisten signiert werden, MUSS das Verfahren ECDSA gemäß TR-03111 [3], Kapitel 5.2.2, verwendet werden.

Tabelle 5 enthält die Hashfunktionen und Kurvenparameter, die von CAs für die Ausstellung von Zertifikaten verwendet werden MÜSSEN. Die Verwendungszeiträume beziehen sich auf die Erstellung der Zertifikate und somit auf den Zeitpunkt, an dem eine CA das jeweilige Zertifikat eines Antragstellers innerhalb der SM-PKI signiert. Für die Root-CA betrifft diese Vorgabe also sowohl die Selbstsignatur bei der Ausstellung von Root-Zertifikaten als auch die Signatur bei der Ausstellung von Sub-CA-Zertifikaten.

Verfahren/Parameter	*Vorgaben*	*Verwendung von*	*Verwendung bis*
Root-CA			
Signaturalgorithmus	`ecdsa-with-SHA384` [3] `ecdsa-with-SHA512` [3]	2015 2025[2]	2025[2] 2029+
Kurvenparameter	brainpoolP384r1 [12] brainpoolP512r1 [12]	2015 2025[2]	2025[2] 2029+
Sub-CAs			
Signaturalgorithmus	`ecdsa-with-SHA256` [3] `ecdsa-with-SHA384` [3]	2015 2025[3]	2025[3] 2029+
Kurvenparameter	brainpoolP256r1 [12] brainpoolP384r1 [12]	2015 2025[3]	2025[3] 2029+

Tabelle 5: Verfahren und Parameter für die Signatur von Zertifikaten

Die Laufzeiten der Zertifikate werden in der TR-03109-4 [4] verbindlich vorgegeben. Zur Durchführung der Zertifikatsverifikation gemäß Certificate Policy der SM-PKI [14] MÜSSEN sämtliche Teilnehmer der SM-PKI insbesondere alle in der Zertifikatskette eingesetzten Verfahren und Parameter unterstützen.

2 Die Umstellung erfolgt mit dem planmäßigen Zertifikatswechsel der Root-CA.
3 Die Umstellung erfolgt mit dem Wechsel der Schlüssellänge im übergeordneten Root-Zertifikat.

4 TLS-Kommunikation im WAN und in der Marktkommunikation

Zwischen Kommunikationspartnern der SM-PKI im WAN und in der Marktkommunikation wird zum Aufbau eines verschlüsselten/integritätsgesicherten und gegenseitig authentisierten Transportkanals das TLS-Protokoll verwendet. Dieses Kapitel legt die hierbei einzusetzenden kryptographischen Parameter verbindlich fest. Es ist ratsam, die TLS-Konfiguration durch Tests zu überprüfen. Die Testspezifikation TR-03116-TS [15] beschreibt entsprechende Testfälle.

4.1 Allgemeine Vorgaben

4.1.1 Authentifizierung und TLS-Zertifikate

Zur gegenseitigen Authentifizierung MUSS jede Partei ein TLS-Zertifikat aus der SM-PKI für ein Schlüsselpaar verwenden, das zur Erzeugung von Signaturen mit ECDSA (gemäß TR-03111 [3]) geeignet ist.

Hierbei MÜSSEN die Kurvenparameter aus Tabelle 6 verwendet werden. Der Verwendungszeitraum bezieht sich auf die Erstellung der Zertifikate.

Kurvenparameter	*Verwendung von*	*Verwendung bis*
brainpoolP256r1 [12]	2015	2029+

Tabelle 6: Kurvenparameter in TLS-Zertifikaten

Für die Zertifikatsverifikation in der SM-PKI gelten die Anforderungen aus Kapitel 3.

4.1.2 TLS-Version und Sessions

Das TLS-Protokoll MUSS mindestens nach Version 1.2 nach RFC 5246 [16] implementiert werden. Ein Fallback auf eine ältere TLS-Version als TLS 1.2 DARF NICHT möglich sein. Die Unterstützung der TLS-Version 1.3 nach RFC 8446 [17] wird EMPFOHLEN. Das Smart-Meter-Gateway SOLLTE beim TLS-Handshake, im Client-Hello, anstelle der eigenen Zeit, eine Zufallszahl als `gmt_unix_time` verwenden.

Kommunikationspartner im WAN MÜSSEN eine TLS-Session (inklusive eventueller Session Resumptions) auf einen Wert begrenzen, der 48 Stunden nicht überschreitet. Beim Smart-Meter-Gateway SOLLTE dieser Wert durch den Gateway Administrator konfigurierbar sein. Insbesondere MUSS das Smart-Meter-Gateway bestehende TLS-Verbindungen nach Ablauf dieser Zeit beenden und für eine neue Verbindung einen neuen TLS-Handshake durchführen.

Innerhalb der erlaubten Session-Lebensdauer KANN Session Resumption verwendet werden. Hierbei KANN eine Stateless Resumption nach RFC 5077 [18] unterstützt und genutzt werden. In diesem Falle MÜSSEN die Anforderungen aus RFC 5077 [18], insbesondere Kapitel 4 und 5, beachtet werden. Die Server-Schlüssel für die Verschlüsselung und Sicherung der Authentizität von Tickets für eine Session Resumption MÜSSEN vom Server sicher gespeichert, verarbeitet und regelmäßig gewechselt werden. Nach Ablauf der Nutzungszeit MÜSSEN die Schlüssel unverzüglich vernichtet werden.

Session Renegotiation DARF NICHT möglich sein.

4.2 Vorgaben für TLS 1.2

4.2.1 Cipher Suites und Kurvenparameter

Die TLS-Implementierung MUSS gemäß RFC 5289 [19] mit ephemerem ECDH erfolgen. Dabei stehen grundsätzlich folgende Cipher Suites zur Verfügung:

- `TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256`
- `TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384`
- `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`
- `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`

Die Wahl der Cipher Suite legt bei TLS 1.2 folgende Bereiche des TLS-Protokolls fest:

- Schlüsselaustausch
- Authentifizierung
- Hashfunktion
- Verschlüsselung und Message Authentication Code (MAC).

Tabelle 7 enthält die Cipher Suites und elliptischen Kurven nach RFC 5114 [11] und RFC 7027 [20], die für die TLS-Kommunikation von Kommunikationspartnern im WAN und in der Marktkommunikation mindestens unterstützt werden MÜSSEN.

Verfahren/Parameter	*Verwendung von*	*Verwendung bis*
Cipher Suites		
`TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256`	2015	2029+
`TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256`	2024	2029+
Kurvenparameter		
NIST P-256 (secp256r1) [11] (IANA-Nr. 23)	2015	2029+
brainpoolP256r1 [20] (IANA-Nr. 26)	2015	2029+
brainpoolP384r1 [20] (IANA-Nr. 27)	2022	2029+

Tabelle 7: Mindestens zu unterstützende Verfahren und Parameter für TLS 1.2

Um eine langfristige Nutzung zu ermöglichen, SOLLTEN für TLS zusätzlich auch die in Tabelle 8 genannten Cipher Suites und elliptischen Kurven unterstützt werden.

Verfahren/Parameter	*Verwendung von*	*Verwendung bis*
Cipher Suites		
`TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384`	2015	2029+

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	2015	2024[4]
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	2015	2029+
Kurvenparameter		
brainpoolP384r1 [20] (IANA-Nr. 27)	2015	2021[5]
brainpoolP512r1 [20] (IANA-Nr. 28)	2015	2029+
NIST P-384 (secp384r1) [11] (IANA-Nr. 24)	2015	2029+

Tabelle 8: Zusätzlich Verfahren und Kurvenparameter für TLS 1.2, deren Unterstützung empfohlen wird

Die unterstützten Verfahren und Kurven MÜSSEN vom Client hierbei in den entsprechenden Datenfeldern[6] als Named Curves angezeigt werden. Zudem MÜSSEN Clients und Server die `ec_point_format`-Extension gemäß RFC 8422 [21] verwenden.

Andere Cipher Suites oder elliptische Kurven, als die in Tabelle 7 oder Tabelle 8 genannten, DÜRFEN NICHT für die Kommunikation im WAN oder in der Marktkommunikation unterstützt werden.

4.2.2 Weitere Vorgaben und Empfehlungen

4.2.2.1 Signaturalgorithmen

Digitale Signaturen während des TLS Handshakes MÜSSEN mit ECDSA erstellt werden. Hierbei MÜSSEN die Hashfunktionen aus Tabelle 9 unterstützt werden.

Verfahren/Parameter	*Verwendung von*	*Verwendung bis*
SHA-256	2015	2029+
SHA-384	2015	2029+
SHA-512	2022	2029+

Tabelle 9: Verpflichtend zu unterstützende Hashfunktionen für Signaturen während des TLS-Handshakes

Zusätzlich SOLLTEN die Hashfunktionen aus Tabelle 10 unterstützt werden.

Verfahren/Parameter	*Verwendung von*	*Verwendung bis*
SHA-512	2015	2021[7]

Tabelle 10: Optional zu unterstützende Hashfunktionen für Signaturen während des TLS-Handshakes

Die unterstützten Verfahren MÜSSEN dem Kommunikationspartner dabei in den entsprechenden Datenfeldern[8] angezeigt werden. Andere als die in Tabelle 9 und Tabelle 10 angegebenen Hashfunktionen DÜRFEN NICHT verwendet werden.

4 Ab 2024 ist die Unterstützung verpflichtend, vgl. Tabelle 7.
5 Seit 2022 ist die Unterstützung verpflichtend, vgl. Tabelle 7.
6 Vgl. RFC 5246 [16], Kapitel 7.4.1.2 (`ClientHello` Message), und RFC 8422 [21], Kapitel 5.1 (Supported Elliptic Curves Extension `elliptic_curves`), vgl. RFC 8446 [17], Kapitel 4.2.7 (`supported_groups` Extension).
7 Seit 2022 ist die Unterstützung verpflichtend, vgl. Tabelle 9.

4.2.2.2 Extensions

Für Extensions gelten ergänzend zu TR-03109-1 [22] folgende Regeln:

- Eine Verkürzung der Ausgabe des HMAC DARF NICHT verwendet bzw. akzeptiert werden[9].
- Im Allgemeinen werden bei TLS gemäß RFC 5246 [16] Klartextdaten zunächst integritätsgesichert (MAC) und anschließend werden Klartext und MAC verschlüsselt (MAC-then-Encrypt). Grundsätzlich ist aber die Verwendung von Encrypt-then-MAC oder Authenticated Encryption vorzuziehen (vgl. Bellare und Nampempre [24]). Daher wird die Verwendung von Encrypt-then-MAC gemäß RFC 7366 [25] EMPFOHLEN, d.h.
  - TLS-Clients SOLLTEN die Encrypt-then-MAC-Extension im Client-Hello anbieten und
  - TLS-Server SOLLTEN entweder eine GCM Cipher Suite auswählen oder die Encrypt-then-MAC-Extension im Server-Hello verwenden.

  Spätestens ab 2025 MUSS die Encrypt-then-MAC-Extension von TLS-Clients unterstützt und im Client-Hello angeboten werden. Spätestens ab 2025 MÜSSEN TLS-Server, wenn die Extension vom TLS-Client angeboten wird, entweder eine GCM Cipher Suite auswählen oder die Encrypt-then-MAC-Extension verwenden.

- Im Allgemeinen erfolgt bei TLS gemäß RFC 5246 [16] die Berechnung des Master Secrets so, dass nicht alle kryptographischen Parameter aus dem TLS-Handshake in die Berechnung einbezogen werden. Je nach verwendeten kryptographischen Parametern kann die fehlende Einbeziehung dieser Daten zu Angriffen auf eine TLS-Session führen (vgl. etwa Triple-Handshake-Angriff nach Bhargavan, Delignat-Lavaud, Fournet, Pironti und Strub [26]). Auch grundsätzlich ist es empfohlen, kontextspezifische Daten in die Berechnung von Session-Schlüsseln einzubeziehen. Daher SOLLTEN Kommunikationspartner im WAN die Extended Master Secret Extension gemäß RFC 7627 [27] verwenden. Hierbei fließen die kryptographischen Parameter in Form eines *Session Hash* (Hashwert über alle Nachrichten des TLS-Handshakes) in die Berechnung des Master Secrets ein.

4.3 Vorgaben für TLS 1.3

Bei der Verwendung von TLS 1.3 gemäß RFC 8446 [17] MUSS folgender Handshake-Modus unterstützt werden:

- ECDHE (ephemerer ECDH)

Zur Unterstützung von Session Resumption KANN zudem folgender Handshake-Modus unterstützt werden:

- PSK mit ECDHE

Das Senden oder Annehmen von 0-RTT Daten DARF NICHT erfolgen. Andere Handshake-Modi DÜRFEN NICHT unterstützt werden.

4.3.1 Cipher Suites und Kurvenparameter

Bei TLS 1.3 definiert eine Cipher Suite die zu verwendenden Algorithmen für

8 Vgl. RFC 5246 [16], Kapitel 7.4 (`ServerKeyExchange`, `CertificateVerify`, `CertificateRequest` Messages und `supported_signature_Algorithms` Extension).
9 Vgl. RFC 6066 [23], Kapitel 7 (`truncated_hmac`-Extension).

- die authentisierte Verschlüsselung der Datenpakete (Blockchiffre inkl. Betriebsmodus) und
- die Hashfunktion für die Schlüsselableitung.

Tabelle 11 enthält die Cipher Suites, die bei der Verwendung von TLS 1.3 für die Kommunikation zwischen Kommunikationspartnern im WAN und in der Marktkommunikation mindestens unterstützt werden MÜSSEN.

Cipher Suites	*Verwendung von*	*Verwendung bis*
TLS_AES_128_GCM_SHA256	2020	2029+

Tabelle 11: Mindestens zu unterstützende Cipher Suites für TLS 1.3

Um eine langfristige Nutzung zu ermöglichen, SOLLTEN für TLS 1.3 zusätzlich auch die in Tabelle 12 genannten Cipher Suites unterstützt werden.

Cipher Suites	*Verwendung von*	*Verwendung bis*
TLS_AES_256_GCM_SHA384	2020	2029+
TLS_AES_128_CCM_SHA256	2020	2029+

Tabelle 12: Zusätzliche Cipher Suites für TLS 1.3, deren Unterstützung empfohlen wird

Tabelle 13 enthält die elliptischen Kurven nach RFC 8446 [17] und RFC 8734 [28], die bei der Verwendung von TLS 1.3 mindestens unterstützt werden MÜSSEN.

Kurvenparameter	*Verwendung von*	*Verwendung bis*
brainpoolP256r1tls13 (IANA-Nr. 31) [28]	2020	2029+
brainpoolP384r1tls13 (IANA-Nr. 32) [28]	2020	2029+
secp256r1 (IANA-Nr. 23) [17]	2020	2029+

Tabelle 13: Mindestens zu unterstützende Kurvenparameter für TLS 1.3

Zudem SOLLTEN bei der Verwendung von TLS 1.3 zusätzlich auch die in Tabelle 14 genannten elliptischen Kurven unterstützt werden.

Kurvenparameter	*Verwendung von*	*Verwendung bis*
brainpoolP512r1tls13 (IANA-Nr. 33) [28]	2020	2029+
secp384r1 (IANA-Nr. 24) [17]	2020	2029+

Tabelle 14: Zusätzliche Kurvenparameter, deren Unterstützung für TLS 1.3 empfohlen wird

Andere Cipher Suites oder elliptische Kurven, als die in den Tabellen 11-14 genannten, DÜRFEN NICHT für die Kommunikation im WAN oder in der Marktkommunikation unterstützt werden.

4.3.2 Weitere Vorgaben und Empfehlungen

4.3.2.1 Signaturalgorithmen

Bei der Verwendung von TLS 1.3 MÜSSEN Kommunikationspartner im WAN die Signature-Algorithm-Extension verwenden, um die unterstützten Signaturalgorithmen für die Verifikation von Serversignaturen im Rahmen des Handshakes anzuzeigen.

Tabelle 15 enthält die Signaturverfahren, die für die Verifikation von Serversignaturen bei der Verwendung von TLS 1.3 unterstützt werden MÜSSEN. Andere Signaturalgorithmen für die Verifikation von Serversignaturen, als die in Tabelle 15 genannten, DÜRFEN NICHT für TLS 1.3 unterstützt werden.

Verfahren/Parameter	*Verwendung von*	*Verwendung bis*
`ecdsa_brainpoolP256r1tls13_sha256`	2020	2029+
`ecdsa_brainpoolP384r1tls13_sha384`	2020	2029+
`ecdsa_brainpoolP512r1tls13_sha512`	2020	2029+
`ecdsa_secp256r1_sha256`	2020	2029+
`ecdsa_secp384r1_sha384`	2020	2029+

Tabelle 15: Zu unterstützende Signaturalgorithmen für die Verifikation von Serversignaturen bei TLS 1.3

Bei der Verwendung von TLS 1.3 MÜSSEN die Kommunikationspartner im WAN die `Signature_Algorithm_Cert`-Extension verwenden, um die für die Zertifikatsverifikation unterstützten Signaturalgorithmen anzuzeigen.

Tabelle 16 enthält die Signaturverfahren, die bei der Verwendung von TLS 1.3 für die Zertifikatsverifikation entsprechend Kapitel 4.1.1 unterstützt werden MÜSSEN. Andere Signaturalgorithmen für die Zertifikatsverifikation, als die in Tabelle 16 genannten, DÜRFEN NICHT für TLS 1.3 unterstützt werden.

Verfahren/Parameter	*Verwendung von*	*Verwendung bis*
`ecdsa_brainpoolP256r1tls13_sha256`	2020	2029+
`ecdsa_brainpoolP384r1tls13_sha384`	2020	2029+
`ecdsa_brainpoolP512r1tls13_sha512`	2020	2029+

Tabelle 16: Zu unterstützende Signaturalgorithmen für die Zertifikatsverifikation bei TLS 1.3

5 TLS-Kommunikation im HAN

Das TLS-Protokoll dient im HAN zum Aufbau eines authentisierten sicheren Kanals zwischen Smart-Meter-Gateway und Komponenten im HAN (wie etwa die Anzeigeeinheit oder CLS-Systeme) (vgl. TR-03109-1 [22]). Es ist ratsam, die TLS-Konfiguration durch Tests zu überprüfen. Die Testspezifikation TR-03116-TS [15] beschreibt entsprechende Testfälle.

5.1 Allgemeine Vorgaben

5.1.1 Authentisierung und TLS-Zertifikate

Für die gegenseitige Authentisierung in der TLS-Kommunikation im HAN werden in der TR-03109-1 [22] zwei mögliche Umsetzungsszenarien (PKI-basierte vs. selbstsignierte Zertifikate) beschrieben.

Für Zertifikate im HAN MÜSSEN Kurvenparameter aus Tabelle 4 dieser Technischen Richtlinie verwendet werden. Hierbei ist grundsätzlich die Verwendung von Brainpool-Kurven ratsam.

Bei einer Umsetzung mittels TLS-Zertifikaten der SM-PKI gelten darüber hinaus die weiteren Anforderungen und Hinweise aus Kapitel 4.1.1.

Die maximalen Zertifikatslaufzeiten hängen vom konkreten Umsetzungsszenario (PKI-basierte vs. selbstsignierte Zertifikate) ab und werden von der TR-03109-1 [22] festgelegt.

5.1.2 TLS-Version und Sessions

Das TLS-Protokoll MUSS mindestens nach Version 1.2 nach RFC 5246 [16] implementiert werden. Ein Fallback auf eine ältere TLS-Version DARF NICHT möglich sein. Die Unterstützung der TLS-Version 1.3 nach RFC 8446 [17] wird EMPFOHLEN.

Das Smart-Meter-Gateway MUSS eine TLS-Session (inklusive eventueller Session Resumptions) auf einen Wert begrenzen, der 48 Stunden nicht überschreitet. Dieser Wert SOLLTE durch den Gateway Administrator konfigurierbar sein. Insbesondere MUSS das Smart-Meter-Gateway bestehende TLS-Verbindungen mit Ablauf dieser Zeit beenden und für eine neue Verbindung einen neuen TLS-Handshake durchführen.

Innerhalb der erlaubten Session-Lebensdauer KANN Session Resumption verwendet werden. Im Falle einer Stateless Resumption MÜSSEN die Anforderungen aus RFC 5077 [18], insbesondere Kapitel 5, beachtet werden.

Session Renegotiation DARF NICHT möglich sein.

5.2 Cipher Suites und Kurvenparameter

Smart-Meter-Gateway und HAN-Komponenten MÜSSEN die Anforderungen an Cipher Suites und Kurvenparameter aus Kapitel 4.2 und 4.3 auch für die TLS-Kommunikation im HAN einhalten.

Zusätzlich MÜSSEN Smart-Meter-Gateway und HAN-Komponenten für die TLS-Kommunikation im HAN die in Kapitel 4.2 und 4.3 empfohlenen Cipher Suites und Kurvenparameter unterstützen.

5.3 Weitere Vorgaben und Empfehlungen

5.3.1 Signaturalgorithmen

Smart-Meter-Gateways und Komponenten im HAN MÜSSEN für die TLS-Kommunikation im HAN bzgl. Unterstützung und Verwendung von Signaturalgorithmen während des TLS-Handshakes die Anforderungen aus Kapitel 4.2.2.1 bei der Verwendung von TLS 1.2 und Kapitel 4.3.2.1 bei der Verwendung von TLS 1.3 einhalten.

5.3.2 Extensions

Smart-Meter-Gateways und HAN Komponenten MÜSSEN bei der Verwendung von TLS 1.2 bzgl. der Unterstützung und Verwendung von Extensions die Anforderungen aus Kapitel 4.2.2.2 auch für die Kommunikation im HAN einhalten.

Für Extensions gelten ergänzend zu TR-03109-1 [22] folgende Regeln:

- Ebenso wie in Kapitel 4.2.2.2 gilt: Eine Verkürzung der Ausgabe des HMAC DARF NICHT verwendet bzw. akzeptiert werden[10].
- Ebenso wie in Kapitel 4.2.2.2 gilt: Im Allgemeinen werden bei TLS 1.2 gemäß RFC 5246 [16] Klartextdaten zunächst integritätsgesichert (MAC) und anschließend werden Klartext und MAC verschlüsselt (MAC-then-Encrypt). Grundsätzlich ist aber die Verwendung von Encrypt-then-MAC oder Authenticated Encryption vorzuziehen (vgl. Bellare und Nampempre [24]). Daher wird die Verwendung von Encrypt-then-MAC gemäß RFC 7366 [25] EMPFOHLEN, d.h.
  - TLS-Clients SOLLTEN die `encrypt_then_MAC`-Extension im Client-Hello anbieten und
  - TLS-Server SOLLTEN entweder eine GCM Cipher Suite auswählen oder die `encrypt_then_MAC`-Extension im Server-Hello verwenden.

  Spätestens ab 2025 MUSS die `encrypt_then_MAC`-Extension von TLS-Clients unterstützt und im Client-Hello angeboten werden. Spätestens ab 2025 MÜSSEN TLS-Server, wenn die Extension vom TLS-Client angeboten wird, entweder eine GCM Cipher Suite auswählen oder die `encrypt_then_MAC`-Extension verwenden.
- Im Allgemeinen erfolgt bei TLS 1.2 gemäß RFC 5246 [16] die Berechnung des Master Secret ohne die direkte Einbeziehung kryptographischer Parameter aus dem TLS-Handshake. Je nach verwendeten kryptographischen Parametern kann dies zu Angriffen auf eine TLS-Session führen. Die `extended_master_secret`-Extension dient der Abwehr solcher Angriffe, indem ein Session-Hash in die Berechnung des Master Secrets einfließt. Daher SOLLTE spätestens ab 2024 die `extended_master_secret`-Extension gemäß RFC 7627 [27] unterstützt und im Client-Hello sowie im Server-Hello angeboten werden.

5.3.3 Migration kryptographischer Verfahren und Schlüssel

Es wird EMPFOHLEN, Komponenten im HAN mit der Möglichkeit auszustatten, in Zukunft neue Schlüssel einzuspielen bzw. zu erzeugen und ggf. per Firmware-Update neue kryptographische Verfahren einzuspielen, um so eine weitere Verwendbarkeit der Komponenten auch nach einer erforderlichen Migration kryptographischer Verfahren zu ermöglichen.

10 Vgl. RFC 6066 [23], Kapitel 7 (`truncated_hmac`-Extension).

6 TLS-Kommunikation im LMN

Sofern nicht die Voraussetzungen von Kapitel 7 erfüllt sind, MUSS die Kommunikation zwischen Zählern und Smart-Meter-Gateways per TLS erfolgen. Insbesondere MUSS für die bidirektionale Kommunikation zwischen Smart-Meter-Gateway und Zähler mindestens in folgenden Einsatzszenarien TLS unterstützt werden (vgl. TR-03109-1 [22]):

- Wechsel des gemeinsamen, zählerindividuellen Schlüssels gemäß Kapitel 7.1.1,
- Auslesen von Zählerdaten,
- Auswahl der auszulesenden Daten.

Es ist ratsam, die TLS-Konfiguration durch Tests zu überprüfen. Die Testspezifikation TR-03116-TS [15] beschreibt entsprechende Testfälle.

6.1 Allgemeine Vorgaben

Das TLS-Protokoll MUSS mindestens nach Version 1.2 nach RFC 5246 [16] implementiert werden. Ein Fallback auf eine ältere TLS-Version DARF NICHT möglich sein. Die Unterstützung der TLS-Version 1.3 nach RFC 8446 [17] wird EMPFOHLEN.

Das Smart-Meter-Gateway MUSS die Länge einer TLS-Session (inklusive eventueller Session Resumptions) auf einen Wert begrenzen, der 31 Tage nicht überschreitet. Dabei DÜRFEN NICHT mehr als 5 MB (5.000.000 Bytes) an Daten innerhalb einer Session ausgetauscht werden[11]. Insbesondere MUSS das Smart-Meter-Gateway bestehende TLS-Verbindungen mit Ablauf dieser Zeit beenden und für eine neue Verbindung einen neuen TLS-Handshake erzwingen.

Innerhalb der erlaubten Session-Lebensdauer KANN Session Resumption verwendet werden. Im Falle einer Stateless Resumption MÜSSEN die Anforderungen aus RFC 5077 [18], insbesondere Kapitel 5, beachtet werden.

Session Renegotiation DARF NICHT möglich sein.

6.2 Cipher Suites und Kurvenparameter

Smart-Meter-Gateways und TLS-Zähler MÜSSEN die Anforderungen an Cipher Suites und Kurvenparameter aus Kapitel 4.2 und 4.3 auch für die Kommunikation im LMN einhalten. Für Zähler beziehen sich die Vorgaben zeitlich auf den Einbau des jeweiligen Zählers.

6.3 Authentisierung und TLS-Zertifikate

Die Zertifikate für die Kommunikation zwischen Smart-Meter-Gateway und TLS-Zähler sind selbstsigniert (vgl. TR-03109-1 [22]). Insbesondere MUSS das Smart-Meter-Gateway also für die TLS-Kommunikation im WAN und im LMN separate Zertifikate verwenden.

Für Zertifikate im LMN MÜSSEN Kurvenparameter aus Tabelle 4 dieser Technischen Richtlinie verwendet werden. Hierbei wird die Verwendung von Brainpool-Kurven empfohlen.

11 Die Datenmenge bezieht sich auf das Gesamtvolumen der ausgetauschten Nachrichten (ohne die Nachrichten des TLS-Handshakes).

Die maximalen Zertifikatslaufzeiten MÜSSEN den Anforderungen aus Tabelle 17 entsprechen.

Zertifikat	*Gültigkeitszeit*	*Private Key Usage*
Zählerzertifikat	Maximal 7 Jahre	Maximal 7 Jahre
SMGW-Zertifikat	Maximal 7 Jahre	Maximal 7 Jahre

Tabelle 17: Laufzeiten der LMN-Zertifikate

6.3.1 Initialer Austausch und Update der LMN-Zertifikate

Für Zähler, die TLS unterstützen, MUSS das initiale Schlüsselpaar vom Hersteller oder vom Smart-Meter-Gateway erzeugt und authentisch in den Zähler eingebracht werden.

Wird das initiale Schlüsselpaar vom Hersteller erzeugt und in den Zähler eingebracht, so MUSS bei erstmaligem Anschluss an ein neues Smart-Meter-Gateway ein authentischer Austausch der TLS-Zertifikate mit dem Smart-Meter-Gateway erfolgen.

Wird das initiale Schlüsselpaar vom Smart-Meter-Gateway erzeugt, so MUSS dieses zusammen mit den Zertifikaten des Smart-Meter-Gateways an den Zähler verschlüsselt und authentisch an den Zähler übertragen werden.

Die initiale Übertragung der Zertifikate bzw. die Einbringung des initialen Schlüsselmaterials vom Smart-Meter-Gateway auf dem Zähler MUSS mit dem in Kapitel 7 beschriebenen symmetrischen Verfahren erfolgen.[12]

Unmittelbar nach Austausch/Einbringung der TLS-Zertifikate MUSS ein TLS-Kanal aufgebaut und der zählerindividuelle Schlüssel für die Kommunikation auf Basis symmetrischer Kryptographie gemäß den Vorgaben von Kapitel 7.1.1 gewechselt werden.

Für das Update eines Smart-Meter-Gateway-Zertifikats MUSS ein neues Schlüsselpaar erzeugt werden und anschließend MUSS das neue selbst-signierte Zertifikat über den aufgebauten TLS-Kanal an den Zähler gesendet werden.

Für das Update eines Zähler-Zertifikats MUSS das Smart-Meter-Gateway ein neues Schlüsselpaar erzeugen und anschließend MUSS das neue selbst-signierte Zertifikat mit dem zugehörigen privaten Schlüssel über den aufgebauten TLS-Kanal an den Zähler gesendet werden.

6.4 Weitere Vorgaben und Empfehlungen

6.4.1 Signaturalgorithmen

Smart-Meter-Gateways und TLS-Zähler MÜSSEN bzgl. Unterstützung und Verwendung von Signaturalgorithmen während des TLS-Handshakes die Anforderungen aus Kapitel 4.2.2.1 und 4.3.2.1 auch für die Kommunikation im LMN einhalten.

12 Es ist geplant, den Austausch der Zertifikate auch durch Aufbau eines verschlüsselten Kanals via Passworteingabe und PACE (siehe auch TR-03109-2 [29], TR-03110 [30]) zu ermöglichen.

6.4.2 Extensions

Für Extensions gelten ergänzend zur TR-03109-1 [22] folgende Regeln:

- Im Allgemeinen werden bei TLS 1.2 gemäß RFC 5246 [16] Klartextdaten zunächst integritätsgesichert (MAC) und anschließend werden Klartext und MAC verschlüsselt (MAC-then-Encrypt). Grundsätzlich ist aber die Verwendung von Encrypt-then-MAC vorzuziehen (vgl. Bellare und Nampempre [24]). Daher wird die Verwendung von Encrypt-then-MAC gemäß RFC 7366 [25] EMPFOHLEN. Ab 2025 MÜSSEN Smart-Meter-Gateways und neu verbaute Zähler die Encrypt-then-MAC-Extension unterstützen. Wenn das Smart-Meter-Gateway im LMN als TLS-Client agiert, MUSS es spätestens ab 2025 im Client-Hello die Encrypt-then-MAC-Extension anbieten. Wenn das Smart-Meter-Gateway im LMN als TLS-Server mit einem entsprechenden Zähler als TLS-Client kommuniziert, MUSS es spätestens ab 2025 entweder eine GCM Cipher Suite auswählen oder die Encrypt-then-MAC-Extension verwenden, sofern eine CBC Cipher Suite ausgewählt wird.
- Im Allgemeinen erfolgt bei TLS 1.2 gemäß RFC 5246 [16] die Berechnung des Master Secret ohne die direkte Einbeziehung kryptographischer Parameter aus dem TLS-Handshake. Je nach verwendeten kryptographischen Parametern kann dies zu Angriffen auf eine TLS-Session führen. Daher ist es ratsam, die Extended Master Secret Extension gemäß RFC 7627 [27] zu unterstützen und zu verwenden. Hierbei fließen die kryptographischen Parameter in Form eines Session-Hashs in die Berechnung des Master Secrets ein.

6.4.3 Migration kryptographischer Verfahren und Schlüssel

Die gewünschte Verwendungszeit von TLS-Zählern kann deutlich über den Prognose-Zeitraum hinausgehen. Daher wird EMPFOHLEN, Zähler mit der Möglichkeit auszustatten, neue Schlüssel einzuspielen bzw. zu erzeugen und ggf. per Firmware-Update neue kryptographische Verfahren einzuspielen, um so eine weitere Verwendbarkeit des TLS-Zählers auch in Zukunft zu ermöglichen.

7 Kommunikation im LMN auf Basis symmetrischer Kryptographie

Für Zähler, die nur unidirektional kommunizieren können, ist die Möglichkeit von TLS nicht gegeben. Da außerdem Bandbreite und Verfügbarkeit des Kommunikationskanals im LMN auch für bidirektional kommunizierende Zähler unter Umständen starken Einschränkungen unterliegt, sind Zähler nicht immer imstande gemäß den zeitlichen Anforderungen einen TLS-Kanal mit einem Smart-Meter-Gateway aufzubauen.

Daher MUSS das Smart-Meter-Gateway diesen Zählern eine alternative Möglichkeit zum Senden von Mess- und Zähldaten bereitstellen. Diese Möglichkeit wird im Folgenden beschrieben. Diese Art der Kommunikation KANN für die Auswahl, den Abruf oder die Übertragung von Daten verwendet werden, falls eine Verbindung per TLS nicht möglich ist. Ansonsten DARF diese Art der Kommunikation NICHT verwendet werden.

7.1 Voraussetzungen

Zähler und Smart-Meter-Gateway verfügen über einen gemeinsamen geeigneten, symmetrischen Schlüssel *MK*. Dieser Schlüssel *MK* MUSS eine Länge von 128 Bit besitzen und für jeden Zähler individuell zufällig (gemäß den generellen Vorgaben von Kapitel 2.3 an Zufallszahlen) erzeugt werden. Dieser Schlüssel wird im Folgenden meist kurz zählerindividueller Schlüssel *MK* genannt. Die Erzeugung des zählerindividuellen Schlüssels *MK* MUSS gemäß den Anforderungen aus Kapitel 2.3

- durch den Hersteller vorgenommen werden, der den zählerindividuellen Schlüssel *MK* in den Zähler einbringt, oder
- durch den Zähler erfolgen, der den zählerindividuellen Schlüssel *MK* an den Hersteller ausgibt.

Vor dem Anschluss des Zählers an ein Smart-Meter-Gateway MUSS der Eigentümer des Zählers den initialen zählerindividuellen Schlüssel *MK* vertraulich und authentisch an den Administrator des Gateways übertragen. Der Gateway-Administrator MUSS den initialen zählerindividuellen Schlüssel dann wie in der TR-03109-1 [22] beschrieben gesichert in das Gateway einbringen[13].

Handelt es sich bei den Zähler um einen bidirektional kommunizierenden Zähler, so MÜSSEN Zähler und Smart-Meter-Gateway unmittelbar nach dem Anschluss des Zählers an das Smart-Meter-Gateway TLS-Zertifikate austauschen und den zählerindividuellen Schlüssel *MK* wechseln (vgl. hierzu Kapitel 6.3.1 und 7.1.1).

Jeder Zähler MUSS über einen Transmission Counter *C* mit einer Länge von 32 Bit verfügen. Erfolgt die Kommunikation zwischen Zähler und Smart-Meter-Gateway bidirektional, d.h. auf den Eingang eines Datensatzes erfolgt die Versendung einer Antwortnachricht, so MUSS auch das Smart-Meter-Gateway über einen zählerindividuellen Transmission Counter *C'* mit einer Länge von 32 Bit verfügen. Transmission Counter DÜRFEN NICHT überlaufen oder zurückgesetzt werden. Das Zurücksetzen der Transmission Counter ist ausnahmsweise nur direkt nach der Erzeugung eines neuen zählerindividuellen Schlüssels *MK* und vor dessen erster Verwendung erlaubt.

Vor der Versendung einer Nachricht MUSS der Zähler den Counterwert *C* gegenüber dem Counterwert der zuletzt authentisch empfangenen bzw. gesendeten Nachricht erhöhen (vgl. Kapitel 7.2).

13 Hierbei eingesetzte kryptographische Verfahren MÜSSEN den allgemeinen Empfehlungen der TR-02102 [2] entsprechen.

Erfolgt die Kommunikation bidirektional, so MUSS auch das Smart-Meter-Gateway vor der Versendung einer Nachricht den Counterwert *C'* gegenüber dem Counterwert der zuletzt authentisch empfangenen bzw. gesendeten Nachricht erhöhen (vgl. Kapitel 7.2).

7.1.1 Wechsel des gemeinsamen, zählerindividuellen Schlüssels *MK* für bidirektionale Zähler

Alle Zähler, die bidirektional kommunizieren können, MÜSSEN mindestens einmal innerhalb von 2 Jahren erfolgreich einen TLS-Kanal aufbauen, um den gemeinsamen, für jeden Zähler individuell zufällig erzeugten Schlüssel *MK* für das symmetrische Verfahren zu wechseln.

Zur Berechnung des neuen zählerindividuellen Schlüssels *MK'* MUSS das Smart-Meter-Gateway eine Zufallszahl z_1 von 128 Bit erzeugen und diese innerhalb des TLS-Kanals an den Zähler senden. Der Schlüssel ist dann $MK'=MAC(MK, z_1)$. Optional KANN zusätzlich der Zähler eine Zufallszahl z_2 von 128 Bit erzeugen und diese an das Smart-Meter-Gateway übertragen. Der neue gemeinsame zählerindividuelle Schlüssel wird dann als $MK'=MAC(MK, z_1||z_2)$ gesetzt. Hierbei MUSS der MAC-Algorithmus aus Tabelle 18 verwendet werden.

7.2 Schlüsselableitung

Vor jeder Übertragung eines neuen Datensatzes MÜSSEN aus dem Schlüssel *MK* die Schlüssel K_{Enc} (für die Verschlüsselung) und K_{MAC} (für die MAC-Berechnung) abgeleitet werden.

Die Berechnung der Schlüssel K_{Enc} bzw. K_{MAC} MUSS jeweils durch MAC-Bildung des aktuellen Counterwertes mit dem im Folgenden beschriebenen Verfahren unter Verwendung der Primitive aus Tabelle 18 geschehen. Hierbei MUSS stets sichergestellt werden, dass der in die Schlüsselableitung für einen zu sendenden Datensatz eingehende Counterwert größer ist als der zugehörige Counterstand der zuletzt authentisch empfangenen bzw. gesendeten Nachricht.

Verfahren	*Mode*	*Länge*	*Verwendung von*	*Verwendung bis*
Berechnung von MK' bzw. K_{Enc}, K_{MAC}, L_{Enc} oder L_{MAC}				
AES	CMAC gemäß RFC 4493 [9]	128	2015	2029+

Tabelle 18: Berechnung der abgeleiteten Schlüssel

Die Berechnung der Schlüssel K_{Enc} bzw. K_{MAC} für einen Datensatz erfolgt durch

- $K_{Enc} = MAC(MK,0x00||C||Zähler\text{-}ID)$ bzw.
- $K_{MAC} = MAC(MK,0x01||C||Zähler\text{-}ID)$,

wobei *0x00* bzw. *0x01* jeweils von der Länge 1 Byte sind. Der Input des MAC MUSS dabei vor Eingang in den MAC stets wie folgt auf Blocklänge aufgefüllt werden, wobei *l* jeweils die Bytelänge des Inputs und *I2OS()* die Konvertierungsfunktion von Integers nach Oktetts gemäß TR-03111 [3], Kapitel 3.1.2, bezeichnet:

- Anzahl der aufzufüllenden Oktetts: *16-(l mod 16)* Oktetts;
- Wert je aufzufüllender Oktett: *I2OS(16-(l mod 16))*.

Erfolgt die Kommunikation bidirektional, so MUSS auch das Smart-Meter-Gateway für die Versendung jedes Datensatzes neue Schlüssel L_{Enc} und L_{MAC} via

- L_{Enc} *= MAC(MK,0x10||C'||Zähler-ID) bzw.*
- L_{MAC} *= MAC(MK,0x11||C'||Zähler-ID)*

und dem gleichen Padding ableiten. Hierbei MUSS das Smart-Meter-Gateway einen Transmission Counter *C'* verwenden, der größer ist als Counterstand des zuletzt authentisch empfangenen bzw. gesendeten Datensatzes.

7.3 Übertragung von Zählerdaten

Die Übertragung der Daten MUSS stets verschlüsselt und MAC-gesichert erfolgen. Die übertragenen Daten MÜSSEN hierbei zuerst (mit dem abgeleiteten Schlüssel K_{Enc} *bzw.* L_{Enc}) verschlüsselt und danach werden die verschlüsselten Daten (mit K_{MAC} *bzw.* L_{MAC}) MAC-gesichert werden.

Hierbei MÜSSEN die Verfahren aus Tabelle 19 verwendet werden. Der Verwendungszeitraum bezieht sich auf die Herstellung des Zählers.

Verfahren	*Mode*	*Länge*	*Verwendung von*	*Verwendung bis*
Verschlüsselung (mit K_{Enc} bzw. L_{Enc})				
AES	CBC gemäß RFC 4493 [7] (IV=0)[14]	128	2015	2029+
Authentizität und Integritätssicherung (mit K_{MAC} bzw. L_{MAC})				
AES	CMAC gemäß RFC 4493 [9]. Der MAC-Wert kann optional auf die ersten 64 Bit gekürzt werden.	128/64	2015	2029+

Tabelle 19: Symmetrische Absicherung der Datenübertragung

Anmerkung:

1. Der Stand des Transmission Counters, der für die Ableitung der Schlüssel K_{Enc} und K_{MAC} bzw. L_{Enc} und L_{MAC} benötigt wird, MUSS unverschlüsselt aber MAC-gesichert übertragen werden. Zudem MUSS ein Zähler die Zähler-ID unverschlüsselt an das Smart-Meter-Gateway übertragen.
2. Die Wahl von IV=0 in der obigen Tabelle 19 ist möglich, da bei jeder Datenübertragung aus dem jeweiligen Counter insbesondere ein neuer Schlüssel K_{Enc} bzw. L_{Enc} abgeleitet wird, sodass K_{Enc} bzw. L_{Enc} jeweils nur einmalig mit IV=0 verwendet wird.
3. Die Verschlüsselung und MAC-Sicherung MUSS stets über einen vollständigen Datensatz von zu schützenden Daten erfolgen. Der Transport des Datensatzes KANN in mehreren Paketen erfolgen.
4. Zur Detektion von Replay-Attacken MUSS der Empfänger einer Nachricht bei jedem empfangenen Datensatz prüfen, dass der Transmission Counter des empfangenen Datensatzes größer als der des letzten empfangenen Datensatzes ist.
5. Um Replay-Attacken auch im Falle eines Stromausfalls zu vermeiden, MUSS stets sichergestellt werden, dass auch nach einem Stromausfall des Smart-Meter-Gateways ein Transmission Counter *C* vorliegt, der einen Wert aufweist, der nicht kleiner als 12 Stunden vor Beginn des Stromausfalls ist.

14 Das Padding MUSS so gewählt werden, dass hierdurch keine Angriffe auf die Verschlüsselung möglich sind. Eine Möglichkeit ist das Padding aus Kapitel 7.2. Das Padding DARF NICHT verarbeitet werden, bevor der MAC über das Chiffrat erfolgreich verifiziert wurde.

8 Inhaltsdatensicherung im WAN

Im Weitverkehrsnetz (WAN) der Smart-Meter-Gateway-Kommunikation erfolgt die Übermittlung von Daten nicht immer über einen direkten Transportkanal zwischen Sender und Endempfänger, sondern teilweise über dritte Parteien (z.B. den Gateway-Administrator). Daher geschieht der Austausch von Daten zwischen Kommunikationspartnern im WAN innerhalb eines TLS-Kanals auf der Basis von für den Endempfänger verschlüsselten und signierten Nachrichten im Cryptographic-Message-Syntax-Format gemäß RFC 5652 [31].

Hierbei muss das folgende Schema implementiert werden.

8.1 Authenticated-Enveloped-data Content Type

Für die Inhaltsdatenverschlüsselung MUSS der Authenticated-Enveloped-Data Content Type (vgl. RFC 5083 [32]) unter Verwendung eines ephemer-statischen Diffie-Hellman nach den Vorgaben der TR-03109-1 [22] verwendet werden.

8.1.1 Content-Authenticated-Encryption

Die Inhaltsdaten werden symmetrisch verschlüsselt und die verschlüsselten Daten werden MAC-gesichert (Content-Authenticated-Encryption). Hierbei MÜSSEN Verfahren aus Tabelle 20 für die Verschlüsselung und die MAC-Sicherung der Inhaltsdaten verwendet werden. Alle Verfahren der Tabelle 20 MÜSSEN unterstützt werden.

Verfahren/Parameter		*Länge*	*Verwendung von*	*Verwendung bis*
AES-GCM				
Verschlüsselung und Authentizität	AES-GCM gemäß RFC 5084 [33][15]	128	2015	2029+
AES-CBC-CMAC				
Verschlüsselung	AES-CBC (IV=0) gemäß RFC 4493 [7] mit Padding gemäß RFC 5652 [31], Abschnitt 6.3	128	2015	2029+
Authentizität	AES-CMAC gemäß RFC 4493 [9]	128	2015	2029+

Tabelle 20: Inhaltsdatenverschlüsselung

Die symmetrischen Schlüssel für die Verschlüsselung und MAC-Sicherung der Inhaltsdaten MÜSSEN (unmittelbar vor ihrer Verwendung) zufällig erzeugt werden. Ein Schlüssel DARF NICHT für die Versendung mehrerer Nachrichten verwendet werden.

Bemerkung: Die Wahl von IV=0 in der obigen Tabelle ist möglich, da bei jeder erneuten Inhaltsdatenverschlüsselung, d.h. für jedes neue Authenticated-Enveloped-Data-Paket, die symmetrischen Schlüssel neu generiert werden.

15 Die Bitlänge des Initialisierungsvektors MUSS 96-Bit und die des Authentication Tags MUSS 128 Bit sein.

8.1.2 Schlüsselableitung und Key Encryption

Die zufällig erzeugten Schlüssel für die Verschlüsselung und MAC-Sicherung der Inhaltsdaten sind verschlüsselt im CMS-Container enthalten (Key Encryption).

Der Schlüssel *K* für die Key Encryption MUSS per ECKA-EG gemäß TR-03111 [3] berechnet werden. Die Ableitung von *K* MUSS mittels der X9.63 Key Derivation Function erfolgen (vgl. TR-03111 [3], Kapitel 4.3.3). ECKA-EG MUSS dazu gemäß TR-03111 [3], Kapitel 4.3.2.2 bzw. 5.3.1 (OIDs mit X9.63-KDF) implementiert werden.

Tabelle 21 enthält die Hashfunktionen und Kurvenparameter, die für ECKA-EG verwendet werden MÜSSEN. Die Verwendungszeiträume beziehen sich auf die Erstellung des zugrundeliegenden Verschlüsselungszertifikats.

Verfahren/Parameter		*Verwendung von*	*Verwendung bis*
Hash	SHA-256	2015	2029+
Kurvenparameter	brainpoolP256r1	2015	2029+

Tabelle 21: Schlüsseltransport für die Inhaltsdatenverschlüsselung

Die Key Encryption MUSS mit dem abgeleiteten Schlüssel *K* auf Basis symmetrischer Kryptographie erfolgen. Hierbei MÜSSEN die Verfahren aus Tabelle 22 verwendet werden.

Verfahren/Parameter		*Verwendung von*	*Verwendung bis*
Verschlüsselung	id-aes128-wrap [34]	2015	2029+

Tabelle 22: Algorithmen für die CMS Key Encryption

8.2 Signed-Data Content Type

Die verschlüsselten und MAC-gesicherten Inhaltsdaten (Authenticated-Enveloped-Data Content Type, siehe Kapitel 8.1) müssen anschließend signiert werden. Hierzu MUSS ECDSA, implementiert nach TR-03111 [3], verwendet werden.

Tabelle 23 enthält die Hashfunktionen und Kurvenparameter, die für die Signatur verwendet werden MÜSSEN. Die Verwendungszeiträume beziehen sich auf die Erstellung des zugrundeliegenden Signaturzertifikats.

Verfahren/Parameter		*Verwendung von*	*Verwendung bis*
Hash	SHA-256	2015	2029+
Kurvenparameter	brainpoolP256r1 [12]	2015	2029+

Tabelle 23: Signatur der verschlüsselten Inhaltsdaten

9 Inhaltsdatensicherung in der Marktkommunikation

In der Marktkommunikation MÜSSEN die zu übermittelnden Daten neben der Absicherung auf Transportebene via TLS (vgl. Kapitel 4) innerhalb des TLS-Kanals auch auf Inhaltsebene unter Verwendung von XML Security entsprechend den W3C-Recommendations [35] und [36] und der SM-PKI entsprechend der TR-03109-4 [4] und der Certificate Policy der SM-PKI [14] abgesichert werden. Dieses Kapitel legt die hierbei einzusetzenden kryptographischen Parameter verbindlich fest.

9.1 Inhaltsdatensignatur

Die zu übermittelnden Daten MÜSSEN zuerst mit dem zum SM-PKI-Signaturzertifikat gehörenden privaten Schlüssel des Absenders signiert werden. Tabelle 24 enthält den Signaturalgorithmus und die Kurvenparameter, die für die Signatur verwendet werden MÜSSEN. Die Verwendungszeiträume beziehen sich auf die Erstellung des zugrundeliegenden Signaturzertifikats.

Verfahren/Parameter		*Verwendung von*	*Verwendung bis*
Signaturverfahren	http://www.w3.org/2001/04/ xmldsig-more#ecdsa-sha256 [36]	2020	2029+
Kurvenparameter	brainpoolP256r1 [12]	2020	2029+

Tabelle 24: Signatur der verschlüsselten Inhaltsdaten

9.2 Inhaltsdatenverschlüsselung

Für die Verschlüsselung der signierten Inhaltsdaten für den Endempfänger MUSS das im Folgenden spezifizierte hybride kryptographische XML-Security-Profil verwendet werden.

Hierbei MUSS analog zu Kapitel 8 der öffentliche Schlüssel des SM-PKI-Verschlüsselungszertifikats des Endempfängers dazu genutzt werden, die Session Keys zu verschlüsseln (*Key Encryption*). Die Verschlüsselung der eigentlichen Datenpakete (*Content Encryption*) MUSS via symmetrischer Verschlüsselungsverfahren erfolgen.

9.2.1 Content Encryption

Die eigentlichen Datenpakete MÜSSEN symmetrisch verschlüsselt und MAC-gesichert werden. Hierbei MÜSSEN Verfahren aus Tabelle 25 verwendet werden.

Verfahren/Parameter	*Länge*	*Verwendung von*	*Verwendung bis*
http://www.w3.org/2009/xmlenc11#aes128-gcm[16] [37]	128	2020	2029+

Tabelle 25: Inhaltsdatenverschlüsselung

Die symmetrischen Schlüssel für die Verschlüsselung der Inhaltsdaten MÜSSEN (unmittelbar vor ihrer Verwendung) zufällig erzeugt werden. Ein Schlüssel DARF NICHT für die Versendung mehrerer Nachrichten verwendet werden.

16 Die Bitlänge des Initialisierungsvektors MUSS 96 Bit und die des Authentication Tags MUSS 128 Bit sein.

9.2.2 Key Agreement und Key Encryption

Die Key Encryption MUSS als Schlüsselableitung (Key Agreement) umgesetzt werden.

Der Schlüssel *K* für die Key Encryption MUSS http://www.w3.org/2009/xmlenc11#ECDH-ES [37] gemäß TR-03111 [3], Kapitel 4.3.2.2 (hier als ECKA-EG bezeichnet), berechnet werden. Die Ableitung von *K* MUSS mittels http://www.w3.org/2009/xmlenc11#ConcatKDF [37] erfolgen.[17]

Tabelle 26 enthält die Hashfunktionen und Kurvenparameter, die für ECDH-ES verwendet werden MÜSSEN. Die Verwendungszeiträume beziehen sich auf die Erstellung des zugrundeliegenden Verschlüsselungszertifikats.

Verfahren/Parameter		***Verwendung von***	***Verwendung bis***
Hash	http://www.w3.org/2001/04/xmlenc#sha256 [35]	2020	2029+
Kurvenparameter	brainpoolP256r1 [12]	2020	2029+

Tabelle 26: Schlüsseltransport für die Inhaltsdatenverschlüsselung

Die Key Encryption MUSS mit dem abgeleiteten Schlüssel *K* auf Basis symmetrischer Kryptographie erfolgen. Hierbei MÜSSEN die Verfahren aus Tabelle 27 verwendet werden.

Verfahren/Parameter		***Verwendung von***	***Verwendung bis***
Verschlüsselung	http://www.w3.org/2001/04/xmlenc#kw-aes128 [35]	2020	2029+

Tabelle 27: Algorithmen für die XML Key Encryption

17 Die ConcatKDF wird in NIST SP800-56A [38] spezifiziert. Die Vorgaben für Kürzung auf die jeweilige Schlüssellänge sind hierbei in NIST SP800-56C [39] dargelegt.

10 PACE und Secure Messaging

Für den Zugriff des Smart-Meter-Gateways auf das Sicherheitsmodul erfolgt eine gegenseitige Authentisierung beider Komponenten gemäß TR-03109-2 [29] mittels des PACE-Protokolls. PACE (Password Authenticated Connection Establishment) (vgl. TR-03109-2 [29] bzw. TR-03110 [30]) ist ein passwort-basiertes Authentisierungs- und Schlüsseleinigungsverfahren, bei dem aus einer gemeinsamen PIN[18] Sitzungsschlüssel hoher Entropie für das anschließende Secure Messaging abgeleitet werden. Secure Messaging liefert einen verschlüsselten, authentisierten Kanal zwischen dem Smart-Meter-Gateway und dem Sicherheitsmodul (vgl. TR-03109-2 [29]).

Tabelle 28 enthält die kryptographischen Verfahren sowie die Anzahl der dezimalen Zeichen der PIN, die für PACE verwendet werden MÜSSEN. Die angegebenen Verwendungszeiträume beziehen sich auf die Herstellung des Sicherheitsmoduls.

Verfahren/Parameter		*Verwendung von*	*Verwendung bis*
Algorithmus	`id-PACE-ECDH-GM-AES-CBC-CMAC-128` vgl. [29] bzw. [30]	2015	2029+
Kurvenparameter	brainpoolP256r1	2015	2029+
PACE-PIN	Mindestens 10 Dezimalziffern	2015	2029+

Tabelle 28: PACE und Secure Messaging

Da die gewünschte Verwendungszeit der Komponenten eines Smart-Metering-Systems deutlich über den Verwendungszeitraum hinausgeht, wird (insbesondere für Komponenten ohne Update-Möglichkeit) EMPFOHLEN, zusätzlich auch die folgenden weiteren PACE-Algorithmen mit den elliptischen Kurven der entsprechenden Bitlängen zu unterstützen:

- `id-PACE-ECDH-GM-AES-CBC-CMAC-192` vgl. [29] bzw. [30]
- `id-PACE-ECDH-GM-AES-CBC-CMAC-256` vgl. [29] bzw. [30]

Das Smart-Meter-Gateway MUSS eine Secure Messaging Session auf maximal 48 Stunden begrenzen.

18 Die PIN, die bei PACE zur Authentisierung des Smart-Meter-Gateways gegenüber dem Sicherheitsmodul verwendet wird, MUSS im Smart-Meter-Gateway geeignet geschützt werden, vgl. TR-03109-1 [22].

11 Zertifizierung

Die Smart-Meter-Gateways und deren Sicherheitsmodule MÜSSEN nach den Common Criteria zertifiziert sein.

11.1 Smart-Meter-Gateway

Im Rahmen der erforderlichen Zertifizierung MUSS die Konformität des Smart-Meter-Gateways zum Schutzprofil BSI-CC-PP-0073 [40] nachgewiesen werden.

Das Common Criteria Zertifikat MUSS einen Hinweis enthalten, dass die Anforderungen dieser Technischen Richtlinie an das Smart-Meter-Gateway (siehe auch die entsprechenden Application Notes im PP zertifiziert unter BSI-CC-PP-0073 [40]) berücksichtigt wurden.

11.2 Sicherheitsmodul

Im Rahmen der erforderlichen Zertifizierung MUSS die Konformität des Sicherheitsmoduls zum Schutzprofil BSI-CC-PP-0077-V2 [41] nachgewiesen werden.

Das Common Criteria Zertifikat MUSS einen Hinweis enthalten, dass die Anforderungen dieser Technischen Richtlinie an das Sicherheitsmodul (siehe auch die entsprechenden Application Notes im PP zertifiziert unter BSI-CC-PP-0077-V2 [41]) berücksichtigt wurden.

Literaturverzeichnis

[1] BSI TR-03109, Technische Richtlinie BSI TR-03109, 2021

[2] BSI TR-02102, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Version 2022-01, 2022

[3] BSI TR-03111, Elliptic Curve Cryptography (ECC), Version 2.10, 2018

[4] BSI TR-03109-4, Smart Metering PKI - Public Key Infrastruktur für Smart Meter Gateways, 2017

[5] IETF RFC 2119, S. Bradner, Key words for use in RFCs to indicate requirement levels, 1997

[6] NIST FIPS 197, Advanced Encryption Standard (AES), 2001

[7] ISO/IEC 10116:2006, Information technology -- Security techniques -- Modes of operation for an n-bit block cipher, 2006

[8] NIST SP800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, 2007

[9] IETF RFC 4493, JH. Song, R. Poovendran, J. Lee, T. Iwata, The AES-CMAC Algorithm, 2006

[10] NIST FIPS 180-4, Secure Hash Standard (SHS), 2015

[11] IETF RFC 5114, M. Lepinski, S. Kent, Additional Diffie-Hellman Groups for Use with IETF Standards, 2008

[12] IETF RFC 5639, M. Lochter, J. Merkle, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, 2010

[13] BSI AIS 20/31, A proposal for: Functionality classes for random number generators, Version 2.0, 2011

[14] BSI, Certificate Policy der Smart Metering PKI, 2017

[15] BSI TR-03116-TS, TLS Test-Specification, 2020

[16] IETF RFC 5246, T. Dierks, E. Rescorla, Transport Layer Security (TLS) Version 1.2, 2008

[17] IETF RFC 8446, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, 2018

[18] IETF RFC 5077, J. Salowey, H. Zhou, P. Eronen, H. Tschofenig, Transport Layer Security (TLS) Session Resumption without Server-Side State, 2008

[19] IETF RFC 5289, E. Rescorla, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), RFC 5289, 2008

[20] IETF RFC 7027, J. Merkle, M. Lochter, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS), 2013

[21] IETF RFC 8422, Y. Nir, S. Josefsson, M. Pegourie-Gonnard, Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Version 1.2 and Earlier, 2018

[22] BSI TR-03109-1, Anforderungen an die Interoperabilität der Kommunikationseinheit eines intelligenten Messsystems, 2021

[23] IETF RFC 6066, D. Eastlake 3rd, Transport Layer Security (TLS) Extensions: Extension Definitions, 2011

[24] M. Bellare, C. Namprempre, Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm; in Advances in Cryptology - Asiacrypt 2000 Proceedings, Lecture Notes in Computer Science Vol. 1976, T. Okamoto ed, Springer-Verlag, 2000

[25] IETF RFC 7366, P. Gutman, Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), 2014

[26] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, P.-Y. Strub, Triple Handshake and Cookie Cutters: Breaking and Fixing Authentication over TLS, IEEE Symposium on Security and Privacy, 2014

[27] IETF RFC 7627, K. Bhargavan, Ed., A. Delignat-Lavaud, A. Pironti, A. Langley, M. Ray, Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension, 2015

[28] IETF RFC 8734, L. Bruckert, J. Merkle, M. Lochter, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS) Version 1.3, 2020
[29] BSI TR-03109-2, Smart Meter Gateway – Anforderungen an die Funktionalität und Interoperabilität des Sicherheitsmoduls, 2014
[30] BSI TR-03110, Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token, Version 2.21, 2016
[31] IETF RFC 5652, R. Housley, Cryptographic Message Syntax (CMS), 2009
[32] IETF RFC 5083, R. Housley, Cryptographic Message Syntax (CMS) Authenticated-Enveloped-Data Content Type, 2007
[33] IETF RFC 5084, R. Housley, Using AES-CCM amd AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS), 2007
[34] IETF RFC 3565, J. Schaad, Use of AES Encryption Algorithm in CMS, 2003
[35] W3C Recommendation, XML Signature Syntax and Processing Version 1.1, 2013
[36] IETF RFC 6931, D. Eastlake, Additional XML Security Uniform Resource Identifiers (URIs), 2013
[37] W3C Recommendation, XML Encryption Syntax and Processing Version 1.1, 2013
[38] NIST SP800-56A, Recommendations for Pair-Wise Key-Establishment Using Discrete Logarithm Cryptography, Revision 3, 2018
[39] NIST SP800-56C, Recommendations for Key-Derivation Methods in Key-Establishment Schemes, Revision 2, 2020
[40] BSI CC-PP-0073, Protection Profile for the Gateway of a Smart Metering System, 2014
[41] BSI CC-PP-0077-V2, Protection Profile for the Security Module of a Smart Metering System, 2015