|  |  |
| --- | --- |
| Version: | 1.0 |
| Status: | Freigegeben |
| Dokumentenklassifizierung: | intern |

technische Richtlinie "kryptographische Verfahren und deren Algorithmen"

1. Januar 2020

Allgemeine Informationen zum vorliegenden Dokument

|  |  |  |
| --- | --- | --- |
| Bezeichnung | Inhalt | Bearbeitungshinweis |
| Eigentümer |  | [verantwortlich für die Erstellung und Pflege des Dokuments = Abteilungsleitung] |
| Autor |  | [operative Verantwortung für das Dokument] |
| Status | Freigegeben | [Einstufung des aktuellen Dokumentenstatus <Entwurf, Finaler Entwurf, Final/Freigegeben>] |
| Klassifizierung | intern | [Einstufung der Dokumentenvertraulichkeitoffen, intern, vertraulich, streng vertraulich] |
| Dokumen­tenkennung | TRL400001 | [Die Dokumenten-Kennung wird von der Dokumentenlenkung (vergeben] |
| Name des Dokuments | technische Richtlinie "kryptographische Verfahren und deren Algorithmen" | [Bezeichnung des Dokuments wie auf dem Titelblatt beschrieben.] |
| Version  | 1.0 | [zweistellige Versionsnummer] |
| Veröffentlichungsform | digital | [Veröffentlichungsform Papier, digital] |
| Speicherort |  | [Ablageort des Dokumentes] |
| Freigabe am | <TT.MM.YYYY> | [Datum der Freigabe durch den Eigentümer] |
| Freigabe bis | <TT.MM.YYYY> | [Datum der Freigabe bis durch den Eigentümer] |
| Revisionszyklus | Alle zwei Jahre | [Revisionszyklus alle 1, 2 Jahre] |
| Archivierungszeitraum | 10 Jahre | [Archivierungszeitraum nach Ablauf 5, 10 Jahre] |

# Dokumentenhistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Beschreibung | Autor | Datum |
| 0.1 | initiale Erstellung |  |  |
| 0.2 – 0.8 | draft |  |  |
| 0.9 | final draft |  |  |
| 1.0 | final/freigegeben |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Inhaltsverzeichnis

[Allgemeine Informationen zum vorliegenden Dokument 2](#_Toc78233253)

[Dokumentenhistorie 3](#_Toc78233254)

[Inhaltsverzeichnis 4](#_Toc78233255)

[Allgemeine Festlegungen 7](#_Toc78233256)

[Ziel / Zweck 7](#_Toc78233257)

[Geltungsbereich 7](#_Toc78233258)

[Zuständigkeiten 8](#_Toc78233259)

[Genehmigungs- und Änderungsverfahren 8](#_Toc78233260)

[Kryptographische Verfahren und deren Algorithmen 9](#_Toc78233261)

[X.509 Identitäten und Zertifikate 9](#_Toc78233262)

[Qualifizierte elektronische Signatur 11](#_Toc78233263)

[Transport-Layer-Security (TLS) 12](#_Toc78233264)

[TLS Version 1.3 12](#_Toc78233265)

[TLS Version 1.2 15](#_Toc78233266)

[Internet Protocol Security (IPSec) 16](#_Toc78233267)

[Internet Key Exchange in Version 2 (IKEv2) 16](#_Toc78233268)

[Encapsulating Security Payload (ESP) 19](#_Toc78233269)

[Authentication Header (AH) 19](#_Toc78233270)

[SA-Lifetime und Re-Keying 20](#_Toc78233271)

[Kerberos 20](#_Toc78233272)

[Verschlüsselungsalgorithmen 20](#_Toc78233273)

[MAC-Sicherung 20](#_Toc78233274)

[Security Assertion Markup Language (SAML) 21](#_Toc78233275)

[Authentisierung - Aktive Sitzungen 21](#_Toc78233276)

[SAML SOAP Binding 22](#_Toc78233277)

[XML-Verschlüsselungsalgorithmen 22](#_Toc78233278)

[Ableitung des Schlüssels 22](#_Toc78233279)

[Schlüsseltransport 22](#_Toc78233280)

[Key Agreement 23](#_Toc78233281)

[Symmetric Key Wrap 23](#_Toc78233282)

[Message Digest 23](#_Toc78233283)

[Passwörter 24](#_Toc78233284)

[Multi-Faktor-Authentifizierung 25](#_Toc78233285)

[HMAC-basierte Einmal-Passwörter (HOTP) 25](#_Toc78233286)

[Zeit-basierte Einmal-Passwörter (TOTP) 25](#_Toc78233287)

[SMS-basierte Einmal-Passwörter 26](#_Toc78233288)

[Datei-Hashing-Verfahren 26](#_Toc78233289)

[Festplattenverschlüsselung 27](#_Toc78233290)

[Daten(Content)-Verschlüsselung 27](#_Toc78233291)

[Storage Area Network (SAN) 28](#_Toc78233292)

[Daten(Content)-Verschlüsselung 28](#_Toc78233293)

[Transport-Layer-Verschlüsselung 28](#_Toc78233294)

[Voice over IP (VoIP) 28](#_Toc78233295)

[Signalisierung 28](#_Toc78233296)

[Sprachübertragung (The Secure Real-time Transport Protocol (SRTP)) 28](#_Toc78233297)

[Microsoft Windows Remote Desktop 29](#_Toc78233298)

[Verschlüsselungsalgorithmen 29](#_Toc78233299)

[Transport-Layer-Verschlüsselung 29](#_Toc78233300)

[Datenbanken 29](#_Toc78233301)

[PostgreSQL 29](#_Toc78233302)

[Microsoft SQL (MS-SQL) 30](#_Toc78233303)

[mariaDB 32](#_Toc78233304)

[JSON Web Tokens (JWT) 33](#_Toc78233305)

[Authentisierungsverfahren 33](#_Toc78233306)

[Verschlüsselungsalgorithmen 34](#_Toc78233307)

[Hash-Algorithmen 34](#_Toc78233308)

[Network Time Protocol (NTP) 34](#_Toc78233309)

[NTP Absicherung mittels Hash 34](#_Toc78233310)

[NTP Absicherung mittels Zertifikate 35](#_Toc78233311)

[Secure Shell Version 2 (SSHv2) 35](#_Toc78233312)

[Schlüsseleinigung 35](#_Toc78233313)

[Verschlüsselungsalgorithmen 36](#_Toc78233314)

[MAC-Sicherung 36](#_Toc78233315)

[Server-Authentisierung 36](#_Toc78233316)

[Simple Network Management Protocol Version 3 (SNMPv3) 37](#_Toc78233317)

[Message Digest-Algorithmus 37](#_Toc78233318)

[Symmetrischer Verschlüsselungsalgorithmus 37](#_Toc78233319)

[Routing-Protokolle 38](#_Toc78233320)

[OSPF mit Hash-Algorithmen 38](#_Toc78233321)

[BGP (S-BGP) mit IPSec 38](#_Toc78233322)

[Media Access Control Security (MACsec) 39](#_Toc78233323)

[MACsec mit PSK 39](#_Toc78233324)

[MACsec mit EAP 40](#_Toc78233325)

[Control And Provisioning of Wireless Access Points (CAPWAP) 42](#_Toc78233326)

[DTLS Version 1.0 42](#_Toc78233327)

[DTLS Version 1.2 43](#_Toc78233328)

[Mailverschlüsselung 43](#_Toc78233329)

[S/MIME 43](#_Toc78233330)

# Allgemeine Festlegungen

## Ziel / Zweck

Die <Institution> überträgt im Rahmen ihrer internen und externen Kommunikation Informationen mit hohen Anforderungen an die Integrität und Vertraulichkeit. Hieraus ableitend müssen kryptographische Algorithmen und Protokolle benannt und für einen entsprechenden Zeitraum freigegeben werden, welche einen sicheren Betrieb von Einzelkomponenten und des Gesamtsystems sicherstellen. Hierdurch werden zusätzlich die Prinzipien (Security-By-Design, Minimal-Need-To-Know-Prinzip, Defence-In-Depth Prinzip und Redundanz-Prinzip) eines sicheren Systemdesigns unterstützt.

Security-By-Design: Das Gesamtsystem und seine Einzelkomponenten sind von Grund auf im Hinblick auf Sicherheit entwickelt. Vorsätzliche Angriffe und unberechtigte Handlungen werden explizit betrachtet, die Auswirkungen von Sicherheitsvorfällen werden durch das Systemdesign minimiert.

Minimal-Need-To-Know-Prinzip: Jede Komponente und jeder Benutzer erhält nur die Rechte, die für die Ausführung einer Aktion notwendig sind. So werden z. B. Anwendungen und Netzwerk-Dienste nicht mit Administratorprivilegien, sondern nur mit den minimal nötigen Systemrechten betrieben.

Defence-In-Depth Prinzip: Sicherheitsrisiken werden nicht durch einzelne Schutzmaßnahmen angegangen, sondern durch die Implementierung gestaffelter, auf mehreren Ebenen ansetzender und sich ergänzender Sicherheitsmaßnahmen begrenzt.

Redundanz-Prinzip: Das Gesamtsystem ist so ausgelegt, dass der Ausfall einzelner Komponenten die sicherheitsrelevanten Funktionen nicht beeinträchtigt. Das Systemdesign verringert die Wahrscheinlichkeit und die Auswirkungen von Problemen, die durch das uneingeschränkte Anfordern von Systemressourcen oder Netzwerkbandbreite entstehen.

Dieses Dokument benennt freigegebene Software, Produkte sowie Algorithmen für den Einsatz von Kryptographie. Es werden die Einsatzumgebung und Randbedingungen in Form von Varianten und Ausprägungen beschrieben. Der Krypto-Bedarf wird in einer verfahrensspezifischen Schutzbedarfserhebung identifiziert. Er wird in diesem Dokument lediglich aufgegriffen, zusammengefasst und anhand freigegebener kryptographischer Lösungen umgesetzt.

## Geltungsbereich

Die Vorgaben des Dokumentes sind für alle Prozessverantwortlichen der <Institution> verbindlich und entsprechend durch die zuständigen Rollenträger umzusetzen.

Anzuwenden sind die Vorgaben für alle durch die <Institution> verantworteten Geschäftsprozesse, Hard- und Softwarekomponenten sowie ihren Konfigurationen. Die Umsetzung dieser Arbeitsanweisung ist durch die entsprechenden Führungskräfte sicherzustellen.

Die im Folgenden beschriebenen Vorgaben sind hingegen nicht bindend für Prozessverantwortliche von Geschäftsprozessen, die nicht durch die <Institution> wahrgenommen werden. In diesen Fällen besitzen die beschriebenen Vorgaben einen empfehlenden Charakter, auf eine Einhaltung muss durch die <Institution> hingewirkt werden.

Interne Regelungen sind geschlechterneutral zu formulieren. Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung männlicher und weiblicher Sprachformen verzichtet. Sämtliche personenbezogenen Bezeichnungen in männlicher Form werden verallgemeinernd verwendet und beziehen sich stets auf alle Geschlechter.

## Zuständigkeiten

Zuständig für die Einhaltung der in diesem Dokument aufgeführten Pflichten und Anforderungen sind:

* Eigene Mitarbeitende und beauftragte Dienstleister, welche administrative Arbeiten an IT- Systemen und Anwendungen von der <Institution> durchführen,
* Eigene Mitarbeitende und beauftragte Dienstleister, welche Applikationsbetreuung mit administrativem Charakter (z. B. Versionspflege, Benutzerverwaltung) betreiben. Die Kontrolle der korrekten Umsetzung der Vorgaben erfolgt durch den <Bereich ???> bei der <Institution>.

## Genehmigungs- und Änderungsverfahren

Diese technische Richtlinie wird durch den <Informationssicherheitsbeauftragter> verantwortet. Die Pflege dieses Dokuments unterliegt dem <Bereich ???> vertreten durch den <Informationssicherheitsbeauftragter>. Änderungen werden ausschließlich von dieser Person oder seinem Stellvertreter vorgenommen. Eine Genehmigung und Freigabe erfolgt durch den <Informationssicherheitsbeauftragter>.

# Kryptographische Verfahren und deren Algorithmen

Diese technische Richtlinie richtet sich an Entwickler und Produkt- bzw. Verfahrensverant-wortliche, die die Einführung neuer kryptographischer Systeme planen bzw. an Administra-toren, welche für den Betrieb von Anwendungen, IT-Systemen und Services (IaaS, PaaS, SaaS, FaaS) sind. Zusätzlich wird in diesem Dokument bewusst nicht auf die Angabe krypto-graphischer Verfahren verzichtet, die zwar zum heutigen Zeitpunkt noch als sicher gelten, mittelfristig, aber nicht mehr empfohlen werden können, da sie, wenn auch noch nicht aus-nutzbar, so doch zumindest theoretische Schwächen beinhalten. Die internen Vorgaben be-rücksichtigen die Empfehlungen der TR-02102 und TR-03116-4 vom BSI.

Es wird ausdrücklich angemerkt, dass die nachfolgend aufgezeigten Kryptographischen Ver-fahren als Whitelist zu verstehen sind. Alle in diesem Dokument nicht aufgeführten Algo-rithmen dürfen ohne eine Beantragung einer Ausnahme (siehe Template „Beantragung von Kryptographieausnahmen“) und Freigabe durch den <Informationssicherheitsbeauftragter> nicht eingesetzt werden. Dies bedeutet auch, dass diese nicht als Fallback ausgehandelt werden dürfen.

## X.509 Identitäten und Zertifikate

X.509 ist ein ITU-T-Standard für eine Public-Key-Infrastruktur zum Erstellen digitaler Zertifikate. Der Standard ISO/IEC 9594-8 wurde zuletzt im Oktober 2016 aktualisiert. In diesem Standard sind spezifiziert die folgenden Datentypen: Public-Key-Zertifikat, Attributzertifikat, Certificate Revocation List (CRL) und Attribute Certificate Revocation List (ACRL). In der elektronischen Kommunikation finden X.509-Zertifikate Anwendung bei den TLS-Versionen diverser Übertragungsprotokolle, wie z. B. beim Abruf von Web-Seiten mit dem HTTPS-Protokoll, oder zum Unterschreiben und Verschlüsseln von E-Mails nach dem S/MIME-Standard. X.509 in Version 3 beinhaltet die Flexibilität, mit Profilen erweitert zu werden.

Der Begriff kryptographische Identität (nachfolgend nur noch als Identität benannt) bezeichnet einen Verbund aus Identitätsdaten und einem kryptographischen Objekt, der zur Authentisierung und Authentifizierung verwendet werden kann. Im Allgemeinen handelt es sich um Schlüsselpaare, bestehend aus öffentlichem und privatem Schlüssel, sowie einem Zertifikat, dass die Kombination aus Authentifizierungsmerkmal und öffentlichem Schlüssel durch eine übergeordnete Instanz (CA – Certification Authority) bestätigt. Eine X.509-Identität ist eine kryptographische Identität, bei der als Zertifikat ein X.509- Zertifikat für einer der folgenden Einsatzzwecke verwendet wird:

* Für den Aufbau von TLS-Verbindungen,
* Für den Aufbau von IPSec-Verbindungen,
* Zur Verschlüsselung von Informationen oder
* Zur Erstellung fortgeschrittener Signaturen

Die zugelassenen Algorithmen zur Erzeugung von Schlüsseln und Signaturen, sowie die minimal zu verwendeten Schlüssellängen, sind den nachfolgenden Tabellen zu entnehmen.

1. Die Lebensdauer der Schlüssel und somit die in einem Zertifikat angegebene Gültigkeitsdauer, soll maximal 5 Jahre betragen.
2. Für nicht aufgeführte Algorithmen und Schlüssellängen müssen die Verantwortlichen der technischen Systeme sowie der Anwendungen die folgenden Vorgaben umsetzen:
	1. Algorithmen und Schlüssellängen, die kryptographisch schwächer sind dürfen nur eingesetzt werden, falls dies aus technischen Gründen erforderlich ist und eine Ausnahmegenehmigung durch den Informationssicherheitsbeauftragten erteilt wurde.
	2. Algorithmen und Schlüssellängen, die kryptographisch stärker sind dürfen in Absprache mit dem Informationssicherheitsbeauftragten verwendet werden, wenn deren Verwendung und mögliche Kompatibilitätsprobleme vorher umfassend evaluiert wurden.
3. Die privaten Schlüssel zu den Zertifikaten sollen mit einem Passwort gesichert werden. Ist ein Passwort-Schutz aus technischen Gründen nicht möglich müssen gesonderte technische und organisatorische Maßnahmen zur Sicherstellung der Vertraulichkeit des privaten Schlüssels getroffen und dokumentiert werden.
4. Zertifikate für Personen und extern erreichbare Systeme müssen von einer allgemein als vertrauenswürdig eingestuften Certificate Authority (CA) signiert sein.
5. Zertifikate für nur intern erreichbare Systeme sind aus der PKI abzuleiten.
6. Die Zertifikate und die ausgebende CA müssen über eine Form der Rückrufprüfung verfügen (CRL oder OCSP).

Der Anwendungszweck (X.509 Felder KeyUsage und ExtendedKeyUsage) ist soweit wie möglich einzuschränken. Für verschiedene Anwendungszwecke sind grundsätzlich unabhängige Schlüsselpaare und damit auch eigene Zertifikate zu verwenden.

Folgende Algorithmen können bei der Erstellung von Zertifikaten verwendet werden.

| Verfahren | Schlüssellänge | Gültig bis |
| --- | --- | --- |
| ECDSA | 224 Bit | 2022 |
| ECDSA | ≥ 256 Bit | 2026+ |
| RSASSA-PSS | 2024 Bit | 2022 |
| RSASSA-PSS | ≥ 3072 Bit | 2026+ |
| DSA | 2024 Bit | 2022 |
| DSA | ≥ 3072 Bit | 2026+ |

Tabelle 1: X.509 - Signatur-Algorithmen

Folgende Hash-Algorithmen mit einer minimalen Schlüssellänge können bei der Erstellung von Zertifikaten genutzt werden:

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(2)-384 | 384 Bit | 2026+ |
| SHA(2)-256 | 256 Bit | 2026+ |
| SHA(3)-512 | 512 Bit | 2026+ |
| SHA(3)-384 | 384 Bit | 2026+ |
| SHA(3)-256 | 256 Bit | 2026+ |

Tabelle 2: X.509 --Hash-Verfahren

## Qualifizierte elektronische Signatur

Die Sicherheit einer qualifizierten elektronischen Signatur hängt entscheidend von der Stärke der zugrunde liegenden Algorithmen ab. Im Folgenden werden Algorithmen genannt, die für qualifizierte elektronische Signaturen als geeignet anzusehen sind. Grundlage für die Benennung war der öffentlich einsehbare Algorithmenktatlog der Bundesnetzagentur aus dem Jahr 2016.

Folgende Hash-Algorithmen sind für die Nutzung einer qualifizierten elektronische Signatur freigegeben.

| Hashing-Familie | Länge des Hashwertes | Verwendung bis |
| --- | --- | --- |
| SHA(2)-256 | 256 Bit | 2026+ |
| SHA(2)-384 | 384 Bit | 2026+ |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(3)-256 | 256 Bit | 2026+ |
| SHA(3)-384 | 384 Bit | 2026+ |
| SHA(3)-512 | 512 Bit | 2026+ |

Tabelle 3: Hash-Algorithmen für die qualifizierte elektronische Signatur

In Anwendungen, in denen etwa aufgrund sehr langfristig ausgelegter Sicherheitsziele die Integrität der zu sichernden Daten über mehrere voneinander unabhängige Mechanismen sichergestellt werden soll, werden hierfür strukturell verschiedene Mechanismen verwendt. Daher sollte eine Implementierung zwei voneinander unabhängiger Hashbäume für die gleichen zu schützenden Daten erfolgen. Dies bedeutet einer der beiden Hashbäume basiert auf SHA(2) und der anderer auf SHA(3).

Folgende Algorithmen können bei der Erstellung qualifizierten elektronischen Signatur genutzt werden.

| Algorithmus | Schlüssellänge | Verwendung bis |
| --- | --- | --- |
| ECDSA | 224 Bit | 2022 |
| ECDSA | ≥ 256 Bit | 2026+ |
| RSASSA-PSS | 2048 Bit | 2022 |
| RSASSA-PSS | ≥ 3072 Bit | 2026+ |
| DSA | 2048 Bit | 2022 |
| DSA | ≥ 3072 Bit | 2026+ |

Tabelle 4: Algorithmen für die Erstellung von qualifizierten elektronischen Signatur

## Transport-Layer-Security (TLS)

Transport-Layer-Security (TLS) besser bekannt unter der Vorgängerbezeichnung Secure Sockets Layer (SSL) ist ein hybrides Verschlüsselungsprotokoll zur sicheren Datenübertragung im Internet. Die letzte finale Version des SSL-Protokolls war 3.0 und anschließend unter dem neuen Namen TLS, beginnend mit Version 1.0, weiterentwickelt und standardisiert. Die aktuelle Version TLS lautet 1.3.

Für die Übertragung mittels TLS sind die folgenden Vorgaben an Algorithmen verbindlich:

* Der Server muss sich immer mit einem X.509-Zertifikat ausweisen.
* Clients müssen so konfiguriert werden, dass der Nutzer fehlerhafte Zertifikate und Fehler bei der Aushandlung der Verschlüsselung deutlich erkennen kann. Insbesondere darf die Zertifikatsprüfung nicht deaktiviert werden.
* Als Cipher Suite muss eine Suite aus den nachfolgenden Ausprägungen verwendet werden.
* Die verwendeten Chiffren müssen Perfect Forward Secrecy (PFS) unterstützen.

### TLS Version 1.3

TLS Version 1.3 ist eine grundlegende Überarbeitung der bisherigen Spezifikationen. Manche der bisherigen Dinge funktionieren mit dieser Version ganz anders. Nachfolgend eine unvollständige Zusammenfassung:

* Die bisherigen Cipher Suiten der Version 1.2 können nicht für TLSv1.3-Verbindungen eingesetzt werden.
* Die neuen Cipher Suiten sind unterschiedlich definiert und geben weder den Zertifikatstyp (z. B. RSA, DSA, ECDSA) noch den Schlüsselaustauschmechanismus (z. B. DHE oder ECHDE) an. Dies hat Auswirkungen auf die Konfiguration der Cipher Suite.
* Eine Nachverhandlung ist in einer TLSv1.3-Verbindung nicht möglich.
* DSA-Zertifikate sind in TLSv1.3-Verbindungen nicht mehr erlaubt.

Cipher-Suiten

Die folgenden Cipher Suites sind unter TLSv1.3 für den produktiven Betrieb freigegeben.

| Cipher-Suite | Gültig bis |
| --- | --- |
| TLS\_AES\_128\_GCM\_SHA256 | 2026+ |
| TLS\_AES\_256\_GCM\_SHA384 | 2026+ |
| TLS\_AES\_128\_CCM\_SHA256 | 2026+ |

Tabelle 5: TLSv1.3 - Cipher-Suiten in empfohlener Reihenfolge

Handshake-Modi

Neben der standardmäßigen Diffie-Hellman Schlüsseleinigung über endlichen Körpern (DHE) oder elliptischen Kurven (ECDHE) gibt es in TLS 1.3 weitere Handshake Modi, die Pre-shared Keys (PSK) verwenden. Unter Pre-shared Keys versteht man hierbei Schlüsselmaterial, das entweder vorab verteilt wurde oder das in einer vergangenen Session über den Session-Ticket-Mechanismus ausgetauscht wurde.

Die folgenden PSK-Modi sind für den produktiven Betrieb freigegeben.

| Pre-shared Keys-Modi | Gültig bis |
| --- | --- |
| psk\_ke | 2026+ |
| psk\_dhe\_ke | 2026+ |

Tabelle 6: TLSv1.3 – Handshake PSK-Modi

Diffie-Hellman Gruppen

In TLS 1.3 können die Kommunikationspartner mittels der „supported\_groups“ Erweiterung signalisieren, welche Diffie-Hellman Gruppen für (EC)DHE verwendet werden sollen.

Die folgenden Diffie-Hellman Gruppen sind für den Einsatz in TLSv1.3 freigegeben.

| Diffie-Hellman Gruppe | Gültig bis |
| --- | --- |
| secp256r1  | 2026+ |
| secp384r1  | 2026+ |
| brainpoolP256r1tls13 | 2026+ |
| brainpoolP384r1tls13 | 2026+ |
| brainpoolP512r1tls13 | 2026+ |
| ffdhe2048  | 2026+ |
| ffdhe3072  | 2026+ |
| ffdhe4096  | 2026+ |

Tabelle 7: TLSv1.3 - freigegebene DH-Gruppen

Signaturverfahren

In TLS 1.3 können die Kommunikationspartner mittels der Erweiterungen „signature\_algorithms“ und „signature\_algorithms\_cert“ signalisieren, welche Signaturverfahren zur zertifikatsbasierten Authentisierung verwenden werden sollen. Die „signature\_algorithms“ Erweiterung bezieht sich dabei auf Signaturen, die der Client oder Server für eine CertificateVerify-Nachricht erstellt, und die „signature\_algorithms\_cert“ Erweiterung auf Zertifikatssignaturen.

Die folgenden Signaturverfahren werden für die „signature\_algorithms“ Erweiterung freigegeben.

| Signaturverfahren – Erweiterung „signature\_algorithms“ | Gültig bis |
| --- | --- |
| rsa\_pss\_rsae\_sha256 | 2026+ |
| rsa\_pss\_rsae\_sha384  | 2026+ |
| rsa\_pss\_rsae\_sha512 | 2026+ |
| rsa\_pss\_pss\_sha256 | 2026+ |
| rsa\_pss\_pss\_sha384 | 2026+ |
| rsa\_pss\_pss\_sha512 | 2026+ |
| ecdsa\_secp384r1\_sha384 | 2026+ |
| ecdsa\_brainpoolP256r1tls13\_sha256 | 2026+ |
| ecdsa\_brainpoolP384r1tls13\_sha384 | 2026+ |
| ecdsa\_brainpoolP512r1tls13\_sha512 | 2026+ |

Tabelle 8: TLSv1.3 - Signaturverfahren für Erweiterung „signature\_algorithms“

Die folgenden Signaturverfahren werden für die „signature\_algorithms\_cert“ Erweiterung freigegeben.

| Signaturverfahren – Erweiterung „signature\_algorithms\_cert“ | Gültig bis |
| --- | --- |
| rsa\_pkcs1\_sha256 | 2025 |
| rsa\_pkcs1\_sha384 | 2025 |
| rsa\_pkcs1\_sha512 | 2025 |
| rsa\_pss\_rsae\_sha256 | 2026+ |
| rsa\_pss\_rsae\_sha384 | 2026+ |
| rsa\_pss\_rsae\_sha512 | 2026+ |
| rsa\_pss\_pss\_sha256 | 2026+ |
| rsa\_pss\_pss\_sha384 | 2026+ |
| rsa\_pss\_pss\_sha512 | 2026+ |
| ecdsa\_secp256r1\_sha256 | 2026+ |
| ecdsa\_secp384r1\_sha384 | 2026+ |
| ecdsa\_brainpoolP256r1tls13\_sha256 | 2026+ |
| ecdsa\_brainpoolP384r1tls13\_sha384 | 2026+ |
| ecdsa\_brainpoolP512r1tls13\_sha512 | 2026+ |

Tabelle 9: TLSv1.3 - Signaturverfahren für Erweiterung „signature\_algorithms\_cert“

### TLS Version 1.2

Für den Einsatz von TLSv1.2 wurden die folgenden Cipher-Suiten mit Perfect Forward Secrecy freigegeben:

| Cipher-Suite | Gültig bis |
| --- | --- |
| TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | 2026+ |
| TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384 | 2026+ |
| TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 | 2026+ |
| TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 | 2026+ |
| TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM | 2026+ |
| TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM | 2026+ |
| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 | 2026+ |
| TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384 | 2026+ |
| TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 | 2026+ |
| TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 | 2026+ |
| TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA256 | 2026+ |
| TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA256 | 2026+ |
| TLS\_DHE\_DSS\_WITH\_AES\_128\_GCM\_SHA256 | 2026+ |
| TLS\_DHE\_DSS\_WITH\_AES\_256\_GCM\_SHA384 | 2026+ |
| TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 | 2026+ |
| TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 | 2026+ |
| TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 | 2026+ |
| TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 | 2026+ |
| TLS\_DHE\_RSA\_WITH\_AES\_128\_CCM | 2026+ |
| TLS\_DHE\_RSA\_WITH\_AES\_256\_CCM | 2026+ |

Tabelle 10: TLSv1.2 - Cipher-Suiten mit PFS

## Internet Protocol Security (IPSec)

Internet Protocol Security (IPSec) ist eine Protokoll-Suite, die eine kryptographisch abgesicherte Kommunikation über nicht vertrauenswürdige IP-Netze ermöglicht und direkt auf der Vermittlungsschicht (Network Layer ISO-OSI-Modell) arbeitet. IPSec bietet durch die verbindungslose Integrität sowie die Zugangskontrolle und Authentisierung der Daten die gewünschte Möglichkeit der Absicherung an. Zusätzlich wird durch IPSec die Vertraulichkeit sowie Authentizität der Paketreihenfolge durch die Verschlüsselung sichergestellt. Dazu werden verschiedene Mechanismen eingesetzt, etwa Verschlüsselung einzelner IP-Pakete und Einfügen eines zusätzlichen Paket-Headers mit einem Message Authentication Code. IPSec kann zum Aufbau virtueller privater Netzwerke (VPN) verwendet werden oder zum Schutz vor Replay-Angriffen eingesetzt werden.

### Internet Key Exchange in Version 2 (IKEv2)

Für den Aufbau einer VPN-Verbindung sind das Verfahren IPSec mit IKEv2 zugelassen. Bei der Aushandlung der IKE-SA werden die Zertifikate zwischen VPN-Client und VPN-Server über UDP übertragen. Der VPN-Client und der VPN-Server müssen den Aufbau von Security Associations (SA) zwischen ihnen, entsprechend dem Internet Key Exchange Protocol Version 2 (IKEv2) gemäß RFC5996, durchführen. Aufgrund der genutzten Zertifikatsprofile und Schlüssellängen können die ISAKMP-Pakete größer als die MTU des Transportnetzes werden, so dass diese fragmentiert werden müssen. Hierbei kann es potenziell zu Problemen mit auf der Übertragungsstrecke liegenden Netzwerkkomponenten kommen, die fragmentierte UDP-Pakete nicht weiterleiten.

Verschlüsselungsverfahren für den Schutz der IKE-Nachrichten

Die Freigaben betreffen die Verschlüsselung der im IKE\_AUTH-, CREATE\_CHILD\_SA- sowie INFORMATIONAL-Exchange ausgetauschten Nachrichten.

| Verfahren | AES-Schlüssellänge | Gültig bis |
| --- | --- | --- |
| ENCR\_AES\_CBC | 128 / 256 | 2026+ |
| ENCR\_AES\_CTR | 128 / 256 | 2026+ |
| ENCR\_AES\_GCM\_16 | 128 / 256 | 2026+ |
| ENCR\_AES\_GCM\_12 | 128 / 256 | 2026+ |
| ENCR\_AES\_CCM\_16 | 128 / 256 | 2026+ |
| ENCR\_AES\_CCM\_12 | 128 / 256 | 2026+ |

Tabelle 11: IPSec - Verschlüsselungsverfahren für den Schutz der IKE-Nachrichten

Pseudo-Zufallsfunktionen zur Schlüsselerzeugung

Zur Erzeugung von Schlüsselmaterial wird eine Pseudo-Random-Function (PRF) eingesetzt. Die folgenden PRFs sind für den produktiven Einsatz freigegeben.

| Verfahren | Gültig bis |
| --- | --- |
| PRF\_AES128\_XCBC | 2026+ |
| PRF\_AES128\_CMAC | 2026+ |
| PRF\_HMAC\_SHA2\_256 | 2026+ |
| PRF\_HMAC\_SHA2\_384 | 2026+ |
| PRF\_HMAC\_SHA2\_512 | 2026+ |

Tabelle 12: IPSec - Pseudo-Zufallsfunktionen zur Schlüsselerzeugung

Integritätsschutz der IKE-Nachrichten

Die folgenden Verfahren sind für die Integritätssicherung der im IKE\_AUTH-, CREATE\_CHILD\_SA- sowie INFORMATIONAL-Exchange ausgetauschten Nachrichten vom freigegeben.

| Verfahren | Gültig bis |
| --- | --- |
| AUTH\_AES\_XCBC\_96 | 2026+ |
| AUTH\_HMAC\_SHA2\_256\_128 | 2026+ |
| AUTH\_HMAC\_SHA2\_384\_192 | 2026+ |
| AUTH\_HMAC\_SHA2\_512\_256 | 2026+ |

Tabelle 13: IPSec – Integritätsschutz der IKE-Nachrichten

Diffie-Hellman-Schlüsselaustausch

Die folgenden Gruppen für den Schlüsselaustausch mittels des Diffie-Hellman-Verfahrens sind für die Nutzung in VPNs freigegeben. Zur Realisierung der Eigenschaft Perfect Forward Secrecy (PFS) kann im CREATE\_CHILD\_SA-Austausch ein erneuter Diffie-Hellman-Schlüsselaustausch durchgeführt werden. Die dabei empfohlenen elliptischen Kurven und Gruppen sollten dabei die gleichen sein.

| DH-Gruppe | Gültig bis |
| --- | --- |
| 2048-bit MODP Group | 2022 |
| 3072-bit MODP Group | 2026+ |
| 4096-bit MODP Group | 2026+ |
| 256-bit random ECP group | 2026+ |
| 384-bit random ECP group | 2026+ |
| 521-bit random ECP group | 2026+ |
| 2048-bit MODP Group with 256-bit Prime Order Subgroup | 2022 |
| brainpoolP256r1 | 2026+ |
| brainpoolP384r1 | 2026+ |
| brainpoolP512r1 | 2026+ |

Tabelle 14: IPSec – DH-Gruppen für Schlüsselaustausch

Authentisierungs-Verfahren für IKEv2

Im Januar 2015 wurde der RFC7427 veröffentlicht. In diesem Dokument wird eine breite Unterstützung von Signaturverfahren und Hashfunktionen spezifiziert. Durch diesen RFC ist es nun möglich, bei IKEv2 weitere elliptischen Kurven, neue Hashfunktionen sowie weitere Signaturverfahren einzusetzen.

| Verfahren | Schlüssellänge | Hash-Funktion | Gültig bis |
| --- | --- | --- | --- |
| ECDSA-256 mit Kurve secp256r1 | 256 Bit | SHA256 | 2026+ |
| ECDSA-384 mit Kurve secp384r1 | 384 Bit | SHA384 | 2026+ |
| ECDSA-512 mit Kurve secp521r1 | 512 Bit | SHA512 | 2026+ |
| ECDSA-256 mit brainpoolP256r1 | 256 Bit | SHA256 | 2026+ |
| ECDSA-384 mit brainpoolP384r1 | 384 Bit | SHA384 | 2026+ |
| ECDSA-512 mit brainpoolP512r1 | 512 Bit | SHA512 | 2026+ |
| RSASSA-PSS | 2048 Bit | SHA256 | 2023 |
| RSASSA-PSS | 4096 Bit | SHA384 | 2026+ |
| ECGDSA-256 mit brainpoolP256r1 | 256 Bit | SHA256 | 2026+ |
| ECGDSA-384 mit brainpoolP384r1 | 384 Bit | SHA384 | 2026+ |
| ECGDSA-512 mit brainpoolP512r1 | 512 Bit | SHA512 | 2026+ |

Tabelle 15: IPSec - Authentisierungs-Verfahren für IKEv2

### Encapsulating Security Payload (ESP)

Verschlüsselungsverfahren für ESP

Die aufgeführten Verfahren sind unabhängig davon, ob der Tunnel- oder Transportmodus von ESP verwendet wird freigegeben. Für weiterführende Details über die zu verschlüsselnden Bereiche wird an dieser Stelle auf RFC4303 verwiesen.

| Verfahren | AES-Schlüssellänge | Gültig bis |
| --- | --- | --- |
| ENCR\_AES\_CBC | 128/256 | 2026+ |
| ENCR\_AES\_CTR | 128/256 | 2026+ |
| AES-GCM with a 16 octet ICV | 128/256 | 2026+ |
| AES-GCM with a 12 octet ICV | 128/256 | 2026+ |
| AES-CCM with a 16 octet ICV | 128/256 | 2026+ |
| AES-CCM with a 12 octet ICV | 128/256 | 2026+ |

Tabelle 16: IPSec - Verschlüsselung der ESP-Pakete

Integritätsschutz der ESP-Pakete

Die aufgeführten Verfahren für den Integritätsschutz der ESP-Pakete sind unabhängig davon, ob der Tunnel- oder Transportmodus von ESP verwendet wird freigegeben. Für tiefergehende Details über die zu sichernden Bereiche innerhalb des ESP-Pakets wird auf RFC4303 verwiesen.

| Verfahren | Gültig bis |
| --- | --- |
| AUTH\_AES\_XCBC\_96 | 2026+ |
| AUTH\_HMAC\_SHA2\_256\_128 | 2026+ |
| AUTH\_HMAC\_SHA2\_384\_192 | 2026+ |
| AUTH\_HMAC\_SHA2\_512\_256 | 2026+ |

Tabelle 17: IPSec – Integritätsschutz der ESP-Nachrichten

### Authentication Header (AH)

Die aufgeführten Verfahren für den Integritätsschutz der AH-Pakete sind unabhängig davon, ob der Tunnel- oder Transportmodus von AH verwendet wird freigegeben. Für tiefergehende Details über die zu sichernden Bereiche innerhalb des AH-Pakets wird auf RFC4302 verwiesen.

| Verfahren | Gültig bis |
| --- | --- |
| AUTH\_AES\_XCBC\_96 | 2026+ |
| AUTH\_HMAC\_SHA2\_256\_128 | 2026+ |
| AUTH\_HMAC\_SHA2\_384\_192 | 2026+ |
| AUTH\_HMAC\_SHA2\_512\_256 | 2026+ |

Tabelle 18: IPSec – Integritätsschutz der AH-Nachrichten

### SA-Lifetime und Re-Keying

Zur Sicherstellung der Sicherheitsanforderungen der Anwendung ist die Lebensdauer einer SA (SA-Lifetime) festzulegen; dies gilt sowohl für IKE-SAs als auch für IPsec-SAs. Dabei sollte die IKE-SA- Lifetime den maximalen Wert von 24 h und die IPSsec-SA-Lifetime von 4 h nicht überschreiten.

## Kerberos

Kerberos ist ein verteilter Authentifizierungsdienst (Netzwerkprotokoll) für offene und unsichere Netze, welches basierend auf dem Needham-Schroeder-Protokoll zur Authentifizierung (1978) weiterentwickelt wurde. Die zurzeit aktuelle Version ist Kerberos 5. Sie ist in RFC 4120 definiert und nutzt ASN.1 zur Codierung. Um Man-in-the-Middle-Angriffe zu unterbinden können, authentifiziert der Kerberos-Dienst sowohl den Server gegenüber dem Client, als auch den Client gegenüber dem Server. Zur Minimierung der Nutzung oder Ausnutzung von schwachen und veralteten Verschlüsselungs- und Authentisierungsverfahren, sind für die produktive Nutzung nur die in den folgenden Tabellen aufgeführten freigegeben.

### Verschlüsselungsalgorithmen

Die an dieser Stelle definierten kryptographische Algorithmen müssen von standardkonformen Anwendungen bzw. Betriebssystemen unterstützt werden.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES\_256 | CTS | 2026+ |
| AES-128 | CTS | 2026+ |

Tabelle 19: Kerberos - Verschlüsselungsalgorithmen

### MAC-Sicherung

Für die MAC-Sicherung sind die folgenden Verfahren freigegeben. Von Windows Betriebssystemen können derzeit keine Verfahren der sha2-Familie angewandt werden.

| Verfahren | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| hmac-sha384-192 | 192 Bit | 2026+ |
| hmac-sha256-128 | 128 Bit | 2026+ |
| hmac-sha1-96 | 96 Bit | 2022 |

Tabelle 20: Kerberos – Verschlüsselungsalgorithmen

## Security Assertion Markup Language (SAML)

Die Security Assertion Markup Language (SAML) 2.0 ist ein XML-basierter Rahmen, der es ermöglicht, Identitäts- und Sicherheitsinformationen über Sicherheitsdomänen hinweg gemeinsam zu nutzen. Die SAML-Spezifikation zielt zwar in erster Linie auf die Bereitstellung von domänenübergreifender Web-Browser-Single-Sign-On (SSO) ab, wurde aber auch so konzipiert, dass sie modular und erweiterbar ist, um die Verwendung in anderen Kontexten zu erleichtern. Vom OASIS-Konsortium wurde SAML ab 2001 entwickelt und berücksichtigt daher die folgenden Anwendungsfälle:

* Single Sign-on - ein Benutzer ist nach der Anmeldung an einer Webanwendung automatisch auch zur Benutzung weiterer Anwendungen authentifiziert.
* Verteilte Transaktionen - mehrere Benutzer arbeiten gemeinsam an einer Transaktion und teilen sich die Sicherheitsinformationen.
* Autorisierungsdienste - die Kommunikation mit einem Dienst läuft über eine Zwischenstation, die die Berechtigung überprüft.

In vielen Fällen hängt die Sicherheit einer SAML-Konversation von dem zugrunde liegenden Vertrauensmodell ab, das in der Regel auf einer Schlüsselverwaltungsinfrastruktur (z.B. PKI oder geheimer Schlüssel) basiert. So sind z.B. SOAP-Nachrichten, die mittels XML-Signatur gesichert sind, nur insoweit gesichert, als die im Austausch verwendeten Schlüssel vertrauenswürdig sind. Unerkannte kompromittierte Schlüssel oder widerrufene Zertifikate könnten beispielsweise eine Sicherheitsverletzung ermöglichen. Sogar der Verzicht auf die Anforderung eines Zertifikats öffnet die Tür für Imitationsangriffe. Die Einrichtung einer PKI ist nicht trivial und muss korrekt implementiert werden, damit darauf aufbauende Schichten sicher sind.

Die nachfolgend benannten Protokolle, Algorithmen und entsprechenden Werte sollen eine geeignete Implementierung von Sicherheitsprotokollen ermöglichen, um die Sicherheit eines Systems aufrechtzuerhalten, einschließlich sicherer Zufalls- oder Pseudozufallszahlengenerierung und sicherer Schlüsselspeicherung.

### Authentisierung - Aktive Sitzungen

Die nicht-persistente Authentisierung wird durch den Kommunikationskanal bereitgestellt, der für den Transport einer SAML-Nachricht verwendet wird. Die Authentisierung kann unilateral - vom Sitzungsinitiator zum Empfänger - oder bilateral erfolgen.

Die Absicherung des Kommunikationskanals muss auf Basis eines sicheren Netzwerkprotokolls, wie TLS ab Version 1.2 (siehe Kapitel "Transport-Layer-Security (TLS)") oder IPsec (siehe Kapitel "Internet Protocol Security (IPSec)") erfolgen. Die Identität des Kommunikationspartners ist auf Basis von X.509 v3 (siehe Kapitel "X.509 Identitäten und Zertifikate") zu verifizieren.

### SAML SOAP Binding

Da die SAML SOAP-Bindung keine Authentisierung erfordert und auch keine Anforderungen an die Vertraulichkeit oder Nachrichtenintegrität In-Transit stellt, ist diese offen für eine Vielzahl gängiger Angriffe. Daher werden in diesem Kapitel übergreifende Anforderungen zur kryptographischen Absicherung der SOAP-Nachrichten benannt. Durch Hinzufügen von In-Transit-Vertraulichkeit auf SOAP-Ebene ist die SOAP-Nachricht so zu konstruieren, dass unabhängig vom SOAP-Transport niemand außer der vorgesehenen Kommunikationspartner auf die Nachricht zugreifen kann. Dies sollte primär die Implementierung bzw. Nutzung XML-Verschlüsselung sichergestellt werden. Alternativ kann die Absicherung auf der SOAP-Transportschicht mittels TLS ab Version 1.2 oder eine darunter liegende Schicht mittels IPSec zurückgegriffen werden.

### XML-Verschlüsselungsalgorithmen

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-256 | GCM | 2026+ |
| AES-192 | GCM | 2026+ |
| AES-128 | GCM | 2022 |
| AES-256 | CBC | 2026+ |
| AES-192 | CBC | 2026+ |
| AES-128 | CBC | 2022 |

Tabelle 21: SAMLv2 – XML-Verschlüsselungsalgorithmen

### Ableitung des Schlüssels

| Verfahren | Beschreibung | Gültig bis |
| --- | --- | --- |
| PBKDF2 | Es wird empfohlen folgenden Parameter zu verwenden: 10000 Durchläufe. | 2026+ |
| ConcatKDF | Die Ableitung von Schlüsseln basiert nur auf Basis von Hashes. Grundsätzlich muss SHA-1 zusätzlich zu SHA-256 wegen der geforderten Rückwärtskompatibilität genutzt werden. Sofern möglich sollte daher PBKDF2 bevorzugt werden. | 2026+ |

Tabelle 22: SAMLv2 – Ableitung des Schlüssels

### Schlüsseltransport

| Verfahren | Schlüssellänge | Hash | MGF | Gültig bis |
| --- | --- | --- | --- | --- |
| RSA-OAEP | AES-256 | SHA512 | MGF1 | 2026+ |
| RSA-OAEP | AES-192 | SHA512 | MGF1 | 2026+ |
| RSA-OAEP | AES-128 | SHA512 | MGF1 | 2022 |
| RSA-OAEP | AES-256 | SHA384 | MGF1 | 2026+ |
| RSA-OAEP | AES-192 | SHA384 | MGF1 | 2026+ |
| RSA-OAEP | AES-128 | SHA384 | MGF1 | 2022 |
| RSA-OAEP | AES-256 | SHA256 | MGF1 | 2026+ |
| RSA-OAEP | AES-192 | SHA256 | MGF1 | 2026+ |
| RSA-OAEP | AES-128 | SHA256 | MGF1 | 2022 |

Tabelle 23: SAMLv2 – Schlüsseltransport

### Key Agreement

| Verfahren | Schlüssellänge | Hash | Gültig bis |
| --- | --- | --- | --- |
| ECDSA - P-521 | 512 | SHA512 | 2026+ |
| ECDSA - P-384 | 384 | SHA384 | 2026+ |
| ECDSA - P-256 | 256 | SHA256 | 2026+ |

Tabelle 24: SAMLv2 – Key-Agreement

### Symmetric Key Wrap

| Verfahren | Key Encryption Key (KEK) | Gültig bis |
| --- | --- | --- |
| AES-256 | 256 | 2026+ |
| AES-192 | 256 | 2026+ |
| AES-192 | 192 | 2026+ |
| AES-128 | 256 | 2026+ |
| AES-128 | 192 | 2026+ |
| AES-128 | 128 | 2026+ |

Tabelle 25: SAMLv2 – Symmetric Key Wrap

### Message Digest

| Hash-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2) | 512 Bit | 2026+ |
| SHA(2) | 384 Bit | 2026+ |
| SHA(2) | 256 Bit | 2026+ |

Tabelle 26: SAMLv2 – Message Digest

## Passwörter

Diese Variante definiert die Anforderungen für die internen Entwickler von Anwendungen hinsichtlich des Umgangs mit dem Speichern von Passwörtern und dient ebenfalls als Grundlage für die Beauftragung bei externer Entwicklung von Anwendungen. Grundsätzlich ist das Abspeichern von Passwörtern zu vermeiden. Interne Anwendungen sollten, statt einer eigenen Benutzerverwaltung, die Anmeldung über vorhandene Verzeichnisdienste umsetzen. Anwendungen, die interne und externe Nutzer haben, sollten für die internen Nutzer ebenfalls vorhandene Verzeichnisdienste verwenden. Gibt es einen Verzeichnisdienst, in dem die externen Nutzer bereits enthalten sind, sollte eine Anbindung geprüft werden.

Passwörter müssen immer gehasht und gesalzen (salted) gespeichert werden. Es dürfen nur Verfahren eingesetzt werden, die explizit für das Speichern von Passwörtern gedacht sind und deren Berechnungsaufwand (z. B. über die Rundenanzahl) eingestellt werden kann. Zusätzlich kann ein Pfeffer (pepper) eingesetzt werden.

Das genutzte Salz (salt) muss zufällig beim Setzen des Passworts bestimmt werden und muss mit dem Passwort gespeichert werden. Die übrigen Parameter, wie beispielsweise Funktionsauswahl und Rundenanzahl, sollten mit jedem Passwort gespeichert werden (z. B. im crypt-Format), damit diese später im laufenden Betrieb für neue Passwörter geändert werden können, ohne alle Passwörter zurücksetzen zu müssen.

Wird ein Pfeffer (pepper) eingesetzt, muss es bei der Installation der Anwendung zufällig generiert werden und muss außerhalb der Datenbank mit den Passwort-Hashes gespeichert werden. Das Pfeffer wird beim Abspeichern und vor jeder Prüfung an das Klartextpassword angehangen.

Der Berechnungsaufwand muss experimentell bestimmt werden und ist so groß wie möglich zu wählen, ohne die Nutzung der Anwendung zu beeinträchtigen.

Folgende Verfahren (in Klammern jeweils Richtwerte für den Berechnungsaufwand) sind für den produktiven Betrieb freigegeben.

| Verfahren | Richtwerte für den Berechnungsaufwand | Gütig bis |
| --- | --- | --- |
| Argon2 | Argon2 ist in zwei Hauptversionen verfügbar. Argon2i ist am sichersten gegen Seitenkanalangriffe, während Argon2d den höchsten Widerstand gegen GPU-Cracking-Angriffe bietet.Es wird empfohlen 3 Durchläufe ab Argon2i v1.3 durchzuführen. | 2026+ |
| script | Es wird empfohlen folgende Parameter zu verwenden: CPU- und Speicherfaktor: 1048576, Speicherfaktor: 8, Parallelfaktor: 1. | 2026+ |
| bcrypt | Es wird empfohlen folgenden Parameter zu verwenden: Kostenfaktor 12. | 2026+ |
| PBKDF2 | Es wird empfohlen folgenden Parameter zu verwenden: 10000 Durchläufe. | 2026+ |

Tabelle 27: freigegebene Verfahren zum Speichern von Passwörtern

Reine Hashverfahren wie MD5, SHA1, SHA2 oder SHA3 und veraltete Passworthashverfahren wie DES-crypt und Windows LM-Hash sind nicht freigegeben.

Passwörter und Teile aus Passwörtern dürfen nicht in anderen Werten als den Passworthash (z. B. Cookies für „Eingeloggt bleiben“-Funktion, Authentication Token) oder mit zwei verschiedenen Hashverfahren gespeichert werden.

## Multi-Faktor-Authentifizierung

Multi-Faktor-Authentifizierung (MFA) nutzt die Kombination von zwei oder mehr Berechtigungsnachweisen für die Prüfung der Identität. Die Sicherheit von Anmeldeverfahren lässt sich dank MFA deutlich erhöhen. Der Identitätsdiebstahl wird erschwert. Mit der Zunahme von Cyber-Sicherheitsbedrohungen wird es immer notwendiger die Sicherheitsstandards von Anwendungen zu verbessern. Heutzutage können viele Anwendungen durch eine zusätzliche Sicherheitsebene (Aktivierung von 2-Faktor-Authentifizierung) besser abgesichert werden.

### HMAC-basierte Einmal-Passwörter (HOTP)

Das Verfahren wurde von der Initiative For Open Authentication (OATH) entwickelt und im Rahmen der Internet Engineering Task Force (IETF) im Dezember 2005 als RFC 4226 veröffentlicht. Ein ähnliches und jüngeres Verfahren, welches aber statt mit einem Zähler mit der Uhrzeit arbeitet, stellt der Time-based One-time Password Algorithmus(TOTP) dar.

Für die Absicherung der Benutzerkonten durch HMAC-Based One-Time Passwords sind die folgenden Werte freigegeben.

| Verfahren | Zeichenlänge | Gültig bis |
| --- | --- | --- |
| SHA(2) – 512 Bit | 6 Zeichen8 Zeichen10 Zeichen | 2026+ |
| SHA(2) – 256 Bit | 6 Zeichen8 Zeichen10 Zeichen | 2026+ |

Tabelle 28: HMAC-basierte Einmal-Passwörter (HOTP)

Der Benutzer sollte spätestens nach 8 h idealerweise nach 1h einen Hinweis für einen neuen Request seines Zählers erhalten.

### Zeit-basierte Einmal-Passwörter (TOTP)

Das TOTP Verfahren wurde 2011 im Rahmen der Internet Engineering Task Force (IETF) als RFC 6238 veröffentlicht. Beim TOTP Algorithmus handelt es sich um eine Hashfunktion, in welcher ein geheimes Passwort zusammen mit der aktuellen Uhrzeit gehasht wird.

Für die Absicherung der Benutzerkonten durch HMAC-Based One-Time Passwords dürfen die folgenden Werte verwendet werden.

| Verfahren | Zeichenlänge | Gültiges Zeitinterval | Gültig bis |
| --- | --- | --- | --- |
| SHA(2) – 512 Bit | 6 Zeichen8 Zeichen10 Zeichen | 30 sec60 sec | 2026+ |
| SHA(2) – 256 Bit | 6 Zeichen8 Zeichen10 Zeichen | 30 sec60 sec | 2026+ |

Tabelle 29: Zeit-basierte Einmal-Passwörter (TOTP)

Die Schlüssel sind in einem manipulationssicheren Endgerät aufzubewahren und sind vor unbefugtem Zugriff und unbefugter Benutzung zu schützen. Die gesamte Kommunikation sollte über kryptographisch abgesicherte Verbindungen mittels TLS Version 1.2 oder IPSec erfolgen. Sofern die Übermittlung des gemeinsamen Preshared Keys (PSK) mittels QR-Code erfolgt, ist sicherzustellen, dass ebenfalls die Übertragung des QR-Codes kryptographisch abgesichert erfolgt und das der Empfänger, den QR-Code sicher gegenüber unberechtigten Dritten aufbewahrt.

Bei IT-Systemen oder Anwendungen mit sehr hohen Schutzbedarf hinsichtlich der Vertraulichkeit und Integrität dürfen keine QR-Codes zum Austausch des gemeinsamen Preshared Keys (PSK) für die HMAC-Based One-Time Password Authentisierung verwendet werden.

### SMS-basierte Einmal-Passwörter

Für die Absicherung der Benutzerkonten durch SMS-Based One-Time Passwords dürfen die folgenden Werte verwendet werden.

| Verfahren | Zeichenlänge | Gültiges Zeitinterval | Gültig bis |
| --- | --- | --- | --- |
| SHA(1) – 100 Bit | 6 Zeichen | 30 sec60 sec | 2021 |

Tabelle 30: SMS-basierte Einmal-Passwörter

Wenn das Passwort für die 2FA an das Mobilegerät gesendet wird, sollte das Gerät das Authentisierungsgeheimnis nicht anzeigen, während das Mobilegerät vom Eigentümer gesperrt ist (d. h. die Eingabe einer PIN, eines Passworts oder einer biometrischen Datenerfassung ist erforderlich). Authentifikatoren sollten jedoch den Empfang eines Authentifizierungsgeheimnisses auf einem gesperrten Gerät anzeigen.

Bei IT-Systemen oder Anwendungen mit hohen oder sehr hohen Schutzbedarf hinsichtlich der Vertraulichkeit und Integrität darf aufgrund der Sicherheitsmängel im SS7-Routingprotokoll keine SMS-Based One-Time Password Authentisierung verwendet werden.

## Datei-Hashing-Verfahren

Eine gute Hashfunktion liefert für die erwarteten Eingabedaten Werte, sodass zwei unterschiedliche Eingaben auch zu unterschiedlichen Ausgabewerten führen. Ein Hashwert wird aus diesem Grunde auch als Fingerprint einer Datei bezeichnet, da er eine nahezu eindeutige Kennzeichnung einer größeren Datenmenge darstellt, so wie ein Fingerabdruck einen Menschen nahezu eindeutig identifiziert. Die Länge des Hash-Outputs ist als Sicherheitsparameter von zentraler Bedeutung, weil er den zu erbringenden Aufwand von generischen Angriffen bestimmt. Zur Begegnung des Geburtstagsparadoxons für eine Hashfunktion muss die Länge des Hashwertes für die SHA(2)-Familie da diese auf das Merkle-Damgård-Konstruktion aufbaut mindestens ≥ 240 Bit sein.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(2)-512 | 256 Bit | 2026+ |
| SHA(2)-384 | 384 Bit | 2026+ |
| SHA(2)-256 | 256 Bit | 2026+ |
| SHA(3)-512 | 512 Bit | 2026+ |
| SHA(3)-384 | 384 Bit | 2026+ |
| SHA(3)-256 | 256 Bit | 2026+ |
| SHA(3)-224 | 224 Bit | 2026+ |

Tabelle 31: Datei-Hash-Verfahren

## Festplattenverschlüsselung

Eine Festplattenverschlüsselung kann für die gesamte Festplatte oder einzelne Partitionen implementiert sein. Die zum Booten benötigten Daten müssen jedoch unverschlüsselt auf der Boot-Festplatte vorhanden sein oder durch einen speziellen Bootmanager entschlüsselt werden. Die an dieser Stelle definierten kryptographische Algorithmen müssen von standardkonformen Anwendungen für die Festplattenverschlüsselung unterstützt werden.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-256 | XTS | 2026+ |
| AES-128 | XTS | 2026+ |

Tabelle 32: Festplattenverschlüsselungsverfahren

## Daten(Content)-Verschlüsselung

Für die Verschlüsselung von einzelnen Dateien oder/und Ordnern ist der an dieser Stelle definierte kryptographische Algorithmus anzuwenden.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-128 | GCM | 2026+ |

Tabelle 33: Daten(Content)-Verschlüsselung

## Storage Area Network (SAN)

Als Storage-Area-Network (SAN) bzw. Speichernetzwerk bezeichnet man im Bereich der Datenverarbeitung ein Netz zur Anbindung von Festplattensubsystemen (Disk-Array) und Tape-Libraries an Server-Systeme. Storage Area Networks sind für serielle kontinuierliche Hochgeschwindigkeitsübertragungen großer Datenmengen konzipiert worden. Sie basieren heute für hochverfügbare, hoch performante Installationen auf der Implementierung des Fibre-Channel-Standards oder aus Kostenüberlegungen auch auf IP.

### Daten(Content)-Verschlüsselung

Der an dieser Stelle definierte kryptographische Algorithmus ist für die Verschlüsselung der Dateien (Content) zu verwenden.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-128 | GCM | 2026+ |

Tabelle 34: SAN-Content-Verschlüsselung

### Transport-Layer-Verschlüsselung

Eine zusätzliche Absicherung des Transportes sollte mittels der freigegebenen Verfahren TLS in Version 1.2 oder 1.3 etabliert werden. Die dediziert freigegebenen Algorithmen der Werte sowie bis wann die Verfahren gültig sind, wird im Kapitel Transport-Layer-Security dieses Dokumentes beschrieben.

## Voice over IP (VoIP)

IP-Telefonie ist eine Technologie, die es ermöglicht, den Telefondienst auf der IP-Infrastruktur zu realisieren, so dass diese die herkömmliche Technologie samt ISDN und allen Komponenten ersetzen.

### Signalisierung

Die Signalisierung eines Telefonats sollte bevorzugt mit mindestens TLSv1.2 verschlüsselt werden. Welche Algorithmen, Schlüssellängen und deren Gültigkeit von Seitens der <Institution> freigegeben wurden, ist im Kapitel TLS Version 1.2 bzw. TLS Version 1.3 beschrieben. Alternativ kann zur Absicherung der Signalisierung auch auf die Verwendung von IPSec (siehe Kapitel Internet Protocol Security) zurückgegriffen werden

### Sprachübertragung (The Secure Real-time Transport Protocol (SRTP))

Die an dieser Stelle definierten kryptographische Algorithmen dürfen für den Zeitraum der Gültigkeit bevorzugt in der Reihenfolge der Aufzählung verwendet werden.

| Verfahren | Chiffrierungsmodus | Hash | Gültig bis |
| --- | --- | --- | --- |
| AES-256 | CTR | HMAC-SHA1-80 | 2026+ |
| AES-192 | CTR | HMAC-SHA1-80 | 2026+ |
| AES-128 | CTR | HMAC-SHA1-80 | 2024 |

Tabelle 35: VoIP – Verschlüsselung der Sprachübertragung

## Microsoft Windows Remote Desktop

Das Microsoft Remote Desktop Protocol (RDP) bietet Remote-Anzeige- und Eingabemöglichkeiten über Netzwerkverbindungen für Windows-basierte Anwendungen, die auf einem Server laufen. RDP selbst wird in TCP gekapselt und kann hier durch mittels etablierter Transportverschlüsselungsverfahren zusätzlich abgesichert werden.

### Verschlüsselungsalgorithmen

Der an dieser Stelle definierte kryptographische Algorithmus ist mittels Policy permanent zu konfigurieren.

| Verfahren | Schlüssellänge | Gültig bis |
| --- | --- | --- |
| RC4 (High Level) | 128 Bit | 2026+ |

Tabelle 36: Microsoft Windows-Remote-Desktop-Verschlüsselung

### Transport-Layer-Verschlüsselung

Begründet der Tatsache, dass RDP für die Verschlüsselung auf den Algorithmus RC4 zurückgreift, ist zur Vermeidung eines Angriffes bzw. zur Wahrung der Vertraulichkeit der Daten eine zusätzliche Absicherung mittels TLS sicherzustellen, dieses steht für die aktuell von Microsoft unterstützten Betriebssysteme ab der TLS Version 1.2 zur Verfügung. Welche Algorithmen, Schlüssellängen für welchen Zeitraum freigegeben sind, wird im Kapitel „TLS Version 1.2“ benannt.

## Datenbanken

### PostgreSQL

**Passwort-Verschlüsselung**

Datenbank-Benutzerpasswörter werden als Hashes gespeichert, welches Verfahren angewandt wird, bestimmt der Einstellungsparameter von „password\_encryption“. Das in der nachfolgenden Tabelle aufgeführte Verfahren ist bei eine Benutzerauthentisierung mittels Passwortes seitens der <Institution> freigegeben.

| Verfahren | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| scram-sha-256 | 256 Bit | 2026+ |

Tabelle 37: PostSQL – Passwort-Verschlüsselung

**Hash-Funktionen**

Eine gute Hashfunktion liefert für die erwarteten Eingabedaten Werte, sodass zwei unterschiedliche Eingaben auch zu unterschiedlichen Ausgabewerten führen. Ein Hashwert wird aus diesem Grunde auch als Fingerprint einer Datei bezeichnet, da er eine nahezu eindeutige Kennzeichnung einer größeren Datenmenge darstellt, so wie ein Fingerabdruck einen Menschen nahezu eindeutig identifiziert. Die Länge des Hash-Outputs ist als Sicherheitsparameter von zentraler Bedeutung, weil er den zu erbringenden Aufwand von generischen Angriffen bestimmt. Zur Begegnung des Geburtstagsparadoxons für eine Hashfunktion muss die Länge des Hashwertes für die SHA(2)-Familie da diese auf das Merkle-Damgård-Konstruktion aufbaut mindestens ≥ 240 Bit sein.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(2)-384 | 384 Bit | 2026+ |
| SHA(2)-256 | 256 Bit | 2026+ |
| SHA(3)-512 | 512 Bit | 2026+ |
| SHA(3)-384 | 384 Bit | 2026+ |
| SHA(3)-256 | 256 Bit | 2026+ |
| SHA(3)-224 | 224 Bit | 2026+ |

Tabelle 38: PostSQL - -Hash-Verfahren

### Microsoft SQL (MS-SQL)

Transparente Datenverschlüsselung (TDE) führt die Echtzeit-I/O-Verschlüsselung und -Entschlüsselung der Daten und Protokolldateien durch. Die Verschlüsselung selbst erfolgt mit einem Datenbank-Verschlüsselungsschlüssel (DEK), der im Datenbank-Boot-Record für die Verfügbarkeit während der Wiederherstellung gespeichert wird. Der DEK ist ein symmetrischer Schlüssel, der mit einem in der Master-Datenbank des Servers gespeicherten Zertifikat oder einem asymmetrischen Schlüssel, der durch ein EKM-Modul geschützt ist, gesichert ist.

Datenbank-Verschlüsselungsschlüssel (DEK)

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-256 | CTR | 2026+ |
| AES-192 | CTR | 2026+ |
| AES-128 | CTR | 2024 |

Tabelle 39: MS-SQL –Datenbank-Verschlüsselungsschlüssel (DEK)

**X.509 Identitäten**

Bei Verwendung von Zertifikaten sind die Anforderungen an Protokolle und Algorithmen der Variante X.509 Identitäten und Zertifikate zu beachten. Sofern auf einen symmetrischen Schlüssel zurückgegriffen wird sind folgende Algorithmen anzuwenden.

| Verfahren | Schlüssellänge | Gültig bis |
| --- | --- | --- |
| RSA | 4096 Bit | 2026+ |
| RSA | 3072 Bit | 2026+ |
| RSA | 2048 Bit | 2022 |

Tabelle 40: MS-SQL –X.509 Identitäten

Deterministic oder Randomized Encryption

Es wird empfohlen Randomized Encryption primär einzusetzen. Denn die deterministische Verschlüsselung erzeugt immer den gleichen verschlüsselten Wert für einen beliebigen Klartextwert. Die Verwendung deterministischer Verschlüsselung ermöglicht es Point Lookups, Equality Joins, Gruppierung und Indexierung auf verschlüsselten Spalten. Hierdurch besteht die Gefahr, dass es auch nicht autorisierten Benutzern möglich ist, Informationen über verschlüsselte Werte zu erraten, indem die nicht autorisierten Benutzer Muster in der verschlüsselten Spalte untersuchen, insbesondere, wenn in den verschlüsselten Spalten nur ein kleiner Satz möglicher verschlüsselter Werte hinterlegt ist.

Die deterministische Verschlüsselung muss eine Spaltenzusammenstellung mit einer binären-Sortierreihenfolge für Zeichenspalten verwenden.

Always Encrypted

Always Encrypted ist eine Funktion zum Schutz vertraulicher Daten innerhalb der Azure SQL Datenbank oder SQL Server Datenbanken. Die Funktion Always Encrypted ermöglicht es den Clients selbst, vertrauliche Daten innerhalb von Client-Anwendungen zu verschlüsseln und die Verschlüsselungsschlüssel niemals an die Datenbank-Engine (SQL-Datenbank oder SQL-Server) weiterzugeben. Daher bietet Always Encrypted eine Trennung zwischen denen, die die Daten besitzen und denen, die die Daten verwalten (aber zur Wahrung der Vertraulichkeit keinen Zugriff besitzen dürfen).

Hashing Algorithmen

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(2)-384 | 384 Bit | 2026+ |
| SHA(2)-256 | 256 Bit | 2026+ |

Tabelle 41: MS-SQL - -Hash-Verfahren

**Verschlüsselungsalgorithmen**

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-256 | CBC | 2026+ |
| AES-192 | CBC | 2026+ |
| AES-128 | CBC | 2022 |

Tabelle 42: MS-SQL –Verschlüsselungsalgorithmen

### mariaDB

MariaDB ist ein relationales Open-Source-Datenbankmanagementsystem, welches durch einen Fork aus MySQL entstanden ist. Dieser Kapitel benennt die freigegeben kryptographischen Parameter.

TLS-Verschlüsselung von Daten zwischen Server und Alien (in transit)

Standardmäßig überträgt MariaDB Daten zwischen Server und Clients, ohne sie zu verschlüsseln. Dies ist in einer verteilten Umgebung nicht akzeptabel. Um diesem Problem entgegenzuwirken, bietet MariaDB die Möglichkeit, Daten während der Übertragung zwischen Server und Clients mit dem Transport-Layer-Security (TLS)-Protokoll zu verschlüsseln. Welche Ciphersuiten bis zu welchem Jahr genutzt werden sollten ist im Kapitel Transport-Layer-Security (TLS) beschrieben. Alle nicht in diesem Kapitel benannten TLS/SSL Verfahren bzw. deren Ciphersuiten sind nicht für den produktiven Einsatz freigegeben.

Native Verschlüsselung von Daten im Ruhezustand (data-at-rest)

Die Data-at-Rest-Verschlüsselung von MariaDB erfordert die Verwendung eines Plugins zur Schlüsselverwaltung und Verschlüsselung. Diese Plugins sind sowohl für die Verwaltung der Verschlüsselungscodes als auch für die eigentliche Ver- und Entschlüsselung der Daten zuständig.

Verschlüsselungsalgorithmen

Es wird eine AES-basierte Verschlüsselung mit einer Schlüssellänge von 128 Bit verwendet, diese kann durch modifizieren der Quelle auf bis zu 256 Bit erweitern können. Von den Entwicklern von MariaDB wurde 128 Bit gewählt, da es schneller und für die meisten Zwecke sicher genug ist.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-256 | CBC | 2026+ |
| AES-192 | CBC | 2026+ |
| AES-128 | CBC | 2022 |

Tabelle 43: mariaDB –Verschlüsselungsalgorithmen

Hashing Algorithmen

Bevor auf die Funktionen von SHA2 zurückgegriffen wird, ist die Nutzung von TLS zu konfigurieren.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(2)-384 | 384 Bit | 2026+ |
| SHA(2)-256 | 256 Bit | 2026+ |

Tabelle 44: mariaDB - -Hash-Verfahren

Verwaltung des Verschlüsselungsschlüssel

Um die Wirksamkeit der Verschlüsselung zu erhöhen, ist der Verschlüsselungsschlüssel auf einem separaten System von den Daten abzulegen. Mit MariaDB Enterprise können drei Plugin-Optionen zur Verwaltung von Verschlüsselungscodes genutzt werden:

* MariaDB Verschlüsselungs-Plugin (file\_key\_management) verfügbar für alle MariaDB-Server
* Amazon Key Management System Plugin
* Eperi Key Management System Plugin

## JSON Web Tokens (JWT)

Ein JSON Web Token (JWT) ist ein auf JSON basiertes und nach RFC 7519 genormtes Access-Token. Das JWT ermöglicht den Austausch von verifizierbaren Claims. Es wird typischerweise verwendet, um in einem System mit einem Drittanbieter die Identität eines Benutzers zwischen einem Identity-Provider und einem Service-Provider auszutauschen. Des Weiteren eignet sich JWT zur Implementierung einer Stateless Session, denn da alle für die Authentifikation benötigten Informationen in dem Token übertragen werden, muss die Sitzung nicht auf dem Server gespeichert werden. Die Spezifikation (RFC 7518) registriert kryptografische Algorithmen und Bezeichner, die mit den Spezifikationen JSON Web Signature (JWS), JSON Web Encryption (JWE) und JSON Web Key (JWK) verwendet werden sollen. Sie definiert mehrere IANA-Register für diese Identifikatoren.

### Authentisierungsverfahren

Folgende Algorithmen für die Authentisierung bzw. Signatur können für den sicheren Einsatz von JWA genutzt werden. Die aufgezeigten Algorithmen sollten priorisiert entsprechend der hinterlegten Reihenfolge genutzt werden.

| ID | Verfahren | Bit-Länge | Hash | MGF | Gültig bis |
| --- | --- | --- | --- | --- | --- |
| ES512 | ECDSA-P-521 | 512 | SHA(2)-512 | - | 2026+ |
| ES384 | ECDSA-P-384 | 384 | SHA(2)-384 | - | 2026+ |
| ES256 | ECDSA-P-256 | 256 | SHA(2)-256 | - | 2026+ |
| PS512 | RSASSA-PSS | ≥ 3072 Bit | SHA(2)-512 | MGF1 | 2026+ |
| PS512 | RSASSA-PSS | 2024 | SHA(2)-512 | MGF1 | 2022 |
| PS384 | RSASSA-PSS | ≥ 3072 Bit | SHA(2)-384 | MGF1 | 2026+ |
| PS384 | RSASSA-PSS | 2024 | SHA(2)-384 | MGF1 | 2022 |
| PS256 | RSASSA-PSS | ≥ 3072 Bit | SHA(2)-256 | MGF1 | 2026+ |
| PS256 | RSASSA-PSS | 2024 | SHA(2)-256 | MGF1 | 2022 |

Tabelle 45: JWT – Authentisierungsverfahren

### Verschlüsselungsalgorithmen

Dieser Kapitel benennt die Freigegebenen Algorithmen zur Verschlüsselung eines JWE Content Encryption Key (CEK) mittels Advanced Encryption Standard (AES).

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES-256 | GCM | 2026+ |
| AES-192 | GCM | 2026+ |
| AES-128 | GCM | 2026+ |
| AES-256 | CBC | 2026+ |
| AES-192 | CBC | 2026+ |
| AES-128 | CBC | 2022 |

Tabelle 46: JWT –Verschlüsselungsalgorithmen

### Hash-Algorithmen

Folgende Hash-Algorithmen mit einer minimalen Schlüssellänge können für den Einsatz von JSON Web Algorithmen (JWA) genutzt werden. Die aufgezeigten Algorithmen sollten priorisiert entsprechend der hinterlegten Reihenfolge genutzt werden. Hierbei ist zu beachten, dass reine Hashing-Verfahren aus Gründen der generellen Angriffsflächen von JWT zu vermeiden sind.

| ID | Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- | --- |
| HS512 | HMAC-SHA(2) | 512 Bit | 2026+ |
| HS384 | HMAC-SHA(2) | 384 Bit | 2026+ |
| HS256 | HMAC-SHA(2) | 256 Bit | 2026+ |

Tabelle 47: JWT - Hash-Algorithmen

## Network Time Protocol (NTP)

### NTP Absicherung mittels Hash

Die Synchronisation der Zeit ist in einem Netzwerk mit mannigfaltigen Sicherheitsrisiken behaftet und das Network Time Protocol (NTP) ist ein häufiges Ziel für die Angriffe. NTP besitzt jedoch eine Möglichkeit die Zeitsynchronisation gegen Störungen mittels eines Hash abzusichern. Das Network Time Protocol verwendet meistens MD5-codierte Schlüssel, um Zeitstempel zu überprüfen, welche von einem NTP Servern an einen NTP-Client oder -Server geliefert werden. Für die Absicherung der Nachbarschaftsbeziehungen sind im Rahmen ihrer Gültigkeit die folgenden Hash-Algorithmen freigegeben.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-256 | 256 Bit | 2026+ |
| SHA(1) | 160 Bit | 2022 |

Tabelle 48: NTP - -Hash-Verfahren

### NTP Absicherung mittels Zertifikate

Die Synchronisation der Zeit ist in einem Netzwerk mit mannigfaltigen Sicherheitsrisiken behaftet und das Network Time Protocol (NTP) ist ein häufiges Ziel für die Angriffe. NTP besitzt jedoch eine Möglichkeit die Zeitsynchronisation gegen Störungen mittels Zertifikate abzusichern. In den meisten Implementierungen wird das Network Time Protocol mittels MD5-codierte Schlüssel abgesichert. Eine Absicherung mittels Zertifikats ist aus Gründen der besseren Absicherung zu bevorzugen. Die freigegebenen Parameter für die Verwendung von Zertifikaten sind im Kapitel "X.509 Identitäten und Zertifikate" aufgeführt.

## Secure Shell Version 2 (SSHv2)

Secure Shell oder SSH bezeichnet sowohl ein Netzwerkprotokoll als auch entsprechende Programme, mit deren Hilfe man auf eine sichere Art und Weise eine verschlüsselte Netzwerkverbindung mit einem entfernten Gerät herstellen kann. Die an dieser Stelle definierten kryptographische Algorithmen müssen von standardkonformen SSH Anwendungen unterstützt werden.

### Schlüsseleinigung

Im Rahmen des SSH-Verbindungsaufbaus wird ein Schlüsselaustausch (Key Exchange) durchgeführt, um gemeinsame Sitzungsschlüssel für die Authentisierung und die Verschlüsselung zu erzeugen und auszutauschen. Die nachfolgend benannten Key Exchange Methods sind für die produktive Verwendung freigegeben.

| Schlüsselaustauschverfahren | Gültig bis |
| --- | --- |
| diffie-hellman-group-exchange-sha256 | 2026+ |
| diffie-hellman-group14-sha256 | 2022 |
| diffie-hellman-group15-sha512 | 2026+ |
| diffie-hellman-group16-sha512 | 2026+ |
| rsa2048-sha256 | 2023 |
| ecdh-sha2-nistp256 | 2026+ |
| ecdh-sha2-nistp384 | 2026+ |
| ecdh-sha2-nistp521 | 2026+ |

Tabelle 49: SSH - Verfahren für die Schlüsseleinigung

### Verschlüsselungsalgorithmen

Während des Key Exchange einigen sich der SSH-Client und der SSH-Server auf einen gemeinsamen Verschlüsselungsalgorithmus sowie einen gemeinsamen Verschlüsselungsschlüssel. Die folgenden Verschlüsselungsverfahren sind für den produktiven betrieb freigegeben.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AEAD AES256 | GCM | 2026+ |
| AEAD AES128 | GCM | 2026+ |
| AES256 | CBC | 2024 |
| AES192 | CBC | 2024 |
| AES128 | CBC | 2024 |
| AES256 | CTR | 2026+ |
| AES192 | CTR | 2026+ |
| AES128 | CTR | 2026+ |

Tabelle 50: SSH - Verschlüsselungsalgorithmen

### MAC-Sicherung

Für die MAC-Sicherung sind die folgenden Verfahren freigegeben.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 Bit | 2026+ |
| SHA(2)-256 | 256Bit | 2026+ |

Tabelle 51: SSH - -Hash-Verfahren

### Server-Authentisierung

Der SSH-Server authentisiert sich gegenüber dem SSH-Client. Die Authentisierung erfolgt im Rahmen des Transport Layer Protocol. Die Key Exchange-Nachrichten enthalten eine digitale Signatur des SSH-Servers oder einen anderen Nachweis, um dessen Authentizität zu beweisen. Folgende Verfahren für die IT-Systeme und Anwendungen freigegeben.

| Verfahren | Schlüssellänge | Gültig bis |
| --- | --- | --- |
| pgp-sign-dss | 2000 Bit | 2022 |
| pgp-sign-dss | 250 Bit | 2022 |
| ecdsa-sha2-nistp256 | 250 Bit | 2026+ |
| ecdsa-sha2-nistp384 | 250 Bit | 2026+ |
| ecdsa-sha2-nistp521 | 250 Bit | 2026+ |
| x509v3-ecdsa-sha2-nistp256 | 250 Bit | 2026+ |
| x509v3-ecdsa-sha2-nistp384 | 250 Bit | 2026+ |
| x509v3-ecdsa-sha2-nistp521 | 250 Bit | 2026+ |

Tabelle 52: SSH - -Server-Authentisierungsverfahren

## Simple Network Management Protocol Version 3 (SNMPv3)

Der RFC7860 spezifiziert mehrere Authentifizierungsprotokolle auf Basis der SHA-2-Hash-Funktionen für das User-based Security Model (USM) für SNMPv3.

### Message Digest-Algorithmus

Die nachfolgenden Verfahren können für die Integritätssicherung ausgetauschten Nachrichten verwendet werden.

| Hashing-Familie | Anmerkungen | Gültig bis |
| --- | --- | --- |
| SHA(1) | usmHMACSHA1AuthProtocol: verwendet SHA1 und kürzt die Ausgabe auf 96 Bit. | 2021 |
| SHA(2)-224 | usmHMAC128SHA224AuthProtocol: verwendet SHA-224 und kürzt die Ausgabe auf 128 Bit. | 2026+ |
| SHA(2)-256 | usmHMAC192SHA256AuthProtocol: verwendet SHA-256 und kürzt die Ausgabe auf 192 Bit. | 2026+ |
| SHA(2)-384 | usmHMAC256SHA384AuthProtocol: verwendet SHA-384 und kürzt die Ausgabe auf 256 Bit. | 2026+ |
| SHA(2)-512 | usmHMAC384SHA512AuthProtocol: verwendet SHA-512 und kürzt die Ausgabe auf 384 Bit. | 2026+ |

Tabelle 53: SNMPv3 - -Message Digest Algorithmus

### Symmetrischer Verschlüsselungsalgorithmus

Das nachfolgend aufgeführte Verfahren betrifft die Verschlüsselung der mittels SNMPv3 ausgetauschten Nachrichten.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES256 | CBC | 2026+ |
| AES192 | CBC | 2026 |
| AES128 | CBC | 2024 |

Tabelle 54: SNMPv3 - Verschlüsselungsalgorithmen

## Routing-Protokolle

Routing-Protokolle sorgen für den Austausch von Routing-Informationen zwischen den Netzen und erlauben es den Routern, ihre Routing-Tabellen dynamisch aufzubauen. Traditionelles IP-Routing bleibt einfach, da Next-Hop-Routing benutzt wird: Der Router sendet das Paket an denjenigen Nachbar-Router, von dem er glaubt, dass er am günstigsten zum Zielnetz liegt. Um den weiteren Weg des Pakets braucht sich der Router nicht zu kümmern. Selbst wenn er falsch lag und das Paket nicht an den „optimalen“ Nachbarn gesendet hat, sollte das Paket trotzdem früher oder später am Ziel ankommen.

### OSPF mit Hash-Algorithmen

Kernstück von OSPF ist die Nachbarschafts-Datenbank / LSD (Link State Database), die eine Liste aller benachbarten Router enthält, zu denen eine bidirektionale Verbindung besteht. Sie spiegelt die Topologie des Netzes wider. Damit diese Datenbank aufgebaut oder bei Topologie-Änderungen aktualisiert wird, ist der Austausch von Routing-Informationen notwendig. Diese werden mittels Flooding übermittelt.

Für die Absicherung der Nachbarschaftsbeziehungen dürfen die folgenden Hash-Algorithmen verwendet werden.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| HMAC-SHA(2)-512 | 512 Bit | 2026+ |
| HMAC-SHA(2)-384 | 384Bit | 2026+ |
| HMAC-SHA(2)-256 | 256Bit | 2026+ |
| HMAC-SHA-1 | 160 Bit | 2024 |

Tabelle 55: OSPF - -Hash-Verfahren

### BGP (S-BGP) mit IPSec

Im BGP selbst sind keine echten Authentifizierungsmechanismen eingebaut, vielmehr vertrauen sich BGP-Router gegenseitig. Es gibt grundsätzlich zwei Möglichkeiten, wie ein Angreifer eine BGP-Sitzung stören bzw. manipulieren kann. Die erste ist, sich als ein Peer-Router auszugeben und die IP-Adresse dieses Peers zu übernehmen. Sofern auf den Routern keine strengen Filter hinterlegt sind, kann der Angreifer dann schlechte Informationen (Metriken) in die Routing-Tabellen einspeisen oder umgekehrt ein Angreifer kann umfangreiche Informationen über das Netzwerk sammeln. Die andere Form des Angriffs besteht darin, einen Reset der BGP-Sitzung zu erzwingen, was mehr als ärgerlich ist. Denn BGP unterliegt den gleichen Arten von Angriffen wie IP-Spoofing, Session Stealing, Denial of Service und dergleichen. Zur Absicherung von BGP sollte daher das Secure-Border-Gateway-Protokoll oder S-BGP verwendet werden. S-BGP nutzt IPSec-Verschlüsselung, um die Übertragungen ab zu sichern.

Die Seitens der <Institution> freigegebenen IPSec Algorithmen und Werte wurden bereits im Kapitel 0 definiert.

| Hashing-Familie | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| HMAC-SHA(2)-512 | 512 Bit | 2026+ |
| HMAC-SHA(2)-384 | 384Bit | 2026+ |
| HMAC-SHA(2)-256 | 256Bit | 2026+ |
| HMAC-SHA(2)-224 | 224Bit | 2026+ |
| HMAC-SHA-1 | 160 Bit | 2024 |
| HMAC-MD5 | 96Bit | 2022 |

Tabelle 56: IS-IS - -Hash-Verfahren

## Media Access Control Security (MACsec)

MACsec wird im Standard IEEE 802.1AE beschrieben und ermöglicht eine sichere Kommunikation für den gesamten Datenverkehr über die Punkt-zu-Punkt Ethernet-Verbindungen zwischen direkt verbundenen Knoten.

Zur Absicherung einer reinen Switch-to-Switch-Verbindungen empfiehlt sich die etwas unsichere Verwendung des statischen CAK-Sicherheitsmodus. Sollte der statische CAK-Sicherheitsmodus zum Einsatz kommen, ist alle 30 Tage auf einen neuen zufälligen Sicherheitsschlüssel zu aktualisieren. Ebenfalls ist sicherzustellen, dass bei einem Sicherheitsvorfall mit Bezug auf unberechtigte Zugriffe auf die Konfiguration oder Backupsystemen bzw. nach einem Personalwechsel (verantwortliche Administratoren oder Dienstleister) der Sicherheitsschlüssel auf allen relevanten Systemen erneuert wird.

Für die Absicherung der Verbindung zu Endpunktgeräten (Server, Telefon oder einen PC) ist auf den Einsatz mittels MACsec EAP zurückzugreiffen.

Die Verwendung eines Prey-Shared-Key (PSK) anstelle von EAP ist nicht erlaubt zur Absicherung kompletter Netzinfrastrukturen, sondern ist als Alternative zur LWL-Verschlüsselung für die Absicherung der Kommunikation zwischen den Border-Switches im Rechenzentrumsverbund vorzusehen. Im Rahmen der Erstellung des Netzinfrastruktur Betriebskonzeptes ist sicherzustellen, dass durch den Passwortwechsel beim Einsatz von MacSec mit PSK keine Betriebsausfälle für die eigentliche Verfahrenskommunikation der <Institution> verursacht werden.

### MACsec mit PSK

**Geräte Identifikation**

Sollte ein Pre-Shared Key (PSK) anstelle von EAP für die Geräte Identifikation eingesetzt werden, muss der Connectivity Association Key (CAK) und Connectivity Association Key Name (CKN) in der Konfiguration des Netz-Devices direkt hinterlegt werden. Beim Einsatz von MACsec mit einem PSK ist zu beachten, dass die Vertraulichkeit der zu schützenden Informationen direkt abhängig ist vom MACsec-Key (MAK), welcher geschützt wird durch den CAK.

| Keyname | Passwortlänge | Verwendung bis |
| --- | --- | --- |
| PSK für den Connectivity Association Key Name (CKN) | 60 Zeichen (mindestens 20 Kleinbuchstaben, mindestens 10 Großbuchstaben, mindestens 10 Ziffern und keine Sonderzeichen) | 2026+ |
| PSK für den Connectivity Association Key (CAK) | 30 Zeichen (mindestens 10 Kleinbuchstaben, mindestens 10 Großbuchstaben und 3 Sonderzeichen) | 2026+ |

Tabelle 57: MACsec – Geräte Identifikation via PSK

### MACsec mit EAP

**Geräte Identifikation**

Für die Geräte Identifikation mittels EAP sind die folgenden Algorithmen zu verwenden.

| Algorithmus | Schlüssellänge | Verwendung bis |
| --- | --- | --- |
| ECDSA | 224 Bit | 2022 |
| ECDSA | ≥ 256 Bit | 2026+ |
| RSA | 2048 Bit | 2022 |
| RSA | ≥ 3072 Bit | 2026+ |

Tabelle 58: MACsec – Geräte Identifikation via X.509 Identität

**Authentisierung und Schlüsselaustausch**

Für die Authentisierung und dem Schlüsselaustausch (IEEE 802.1X (RFC 5126, RFC 4851)) sind die beiden folgenden Verfahren zu verwenden.

| Schlüsseleinigung und Authentisierung | Verschlüsselung | Chiffrierungsmodus | Hash | Verwendung bis |
| --- | --- | --- | --- | --- |
| ECDSA | AES256 | GCM | SHA(2)-512 | 2026+ |
| CBC | 2024 |
| RSA | AES256 | GCM | SHA(2)-512 | 2026+ |
| CBC | 2024 |
| ECDSA | AES256 | GCM | SHA(2)-384 | 2026+ |
| CBC | 2024 |
| RSA | AES256 | GCM | SHA(2)-384 | 2026+ |
| CBC | 2024 |
| ECDSA | AES128 | GCM | SHA(2)-256 | 2026+ |
| CBC | 2024 |
| RSA | AES128 | GCM | SHA(2)-256 | 2026+ |
| CBC | 2024 |

Tabelle 59: MACsec – Authentisierung und Schlüsselaustausch

**Control Key Management**

Für das Management der Control Keys (IEEE 802.1X-2010) sind die beiden folgenden Verfahren zu verwenden.

| Algorithmus | Betriebsmodus | Verwendungszeitraum |
| --- | --- | --- |
| AES-256 | CMAC | 2026+ |
| AES-128 | CMAC | 2024 |

Tabelle 60: MACsec - Algorithmen für Control Key Management

**Authorization and Key Distribution**

Für die Authorization und Key Distribution (RFC 6218) sind die folgenden Verfahren zu verwenden.

| Schlüsseleinigung und Authentisierung | Verschlüsselung | Chiffrierungsmodus | Hash | Verwendung bis |
| --- | --- | --- | --- | --- |
| ECDSA | AES256 | GCM | SHA(2)-512 | 2026+ |
| CBC | 2024 |
| RSA | AES256 | GCM | SHA(2)-512 | 2026+ |
| CBC | 2024 |
| ECDSA | AES256 | GCM | SHA(2)-384 | 2026+ |
| CBC | 2024 |
| RSA | AES256 | GCM | SHA(2)-384 | 2026+ |
| CBC | 2024 |
| ECDSA | AES128 | GCM | SHA(2)-256 | 2026+ |
| CBC | 2024 |
| RSA | AES128 | GCM | SHA(2)-256 | 2026+ |
| CBC | 2024 |

Tabelle 61: MACsec - Algorithmen für Authorization und Key Distribution

**Bulk Data Encryption**

Für die Bulk Data Encryption (IEEE 802.1 AE) sind die beiden folgenden Verfahren zu verwenden.

| Algorithmus | Betriebsmodus | Verwendungszeitraum |
| --- | --- | --- |
| AES-256 | GCM | 2026+ |
| AES-128 | GCM | 2024 |

Tabelle 62: MACsec - Algorithmen für Bulk Data Encryption

## Control And Provisioning of Wireless Access Points (CAPWAP)

CAPWAP ist ein Software-Protokoll zur Steuerung und Einrichtung von Access Points für WLANs. Zweck ist die zentrale Verwaltung von Access Points. Das CAPWAP-Protokoll ist beschrieben im RFC 5415. Verwendet werden die UDP-Ports 5246 als Steuerkanal und 5247 als Datenkanal.

Um keine Implikation bezüglich der Sicherheit in DTLS herbeizuführen, wurden nur an den Stellen Änderungen vorgenommen, an denen dies bei Verwendung eines nicht zuverlässigen Transportprotokolls notwendig ist. Diese Änderungen sind:

* Wiederherstellen der Zuverlässigkeit des Handshakes zu Beginn der Kommunikation, da in diesem Teil die Ankunft aller Pakete garantiert werden muss, um eine Authentifizierung und den Schlüsseltausch ermöglichen zu können. Dies geschieht dadurch, dass die Pakete nach einer bestimmten Zeit erneut gesendet werden.
* Explizite Nummerierung der einzelnen Pakete während der Übertragung. Dies geschieht bei TLS nur implizit, wodurch bei einem Paketverlust kein korrekter HMAC mehr berechnet werden kann, was eine Integritätsverletzung darstellt und wiederum zu einem Verbindungsabbruch führt.
* Eine optionale Replay-Detection für einzelne Pakete.

DTLS 1.0 basiert auf TLS 1.1 und DTLS 1.2 auf TLS 1.2. Die Versionsnummer 1.1 wurde übersprungen, um die Versionsnummern mit TLS zu harmonisieren.

### DTLS Version 1.0

Die folgenden Cipher Suites sind bei der Verwendung DTLSv1.0 in der empfohlenen Reihenfolge freigegeben.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verfahren | Verschlüsselung | Chiffrierungsmodus | Hash | Gütlig bis |
| RSA | AES256 | CBC | SHA(2)-256 | 2026+ |
| RSA | AES256 | CBC | SHA(1) | 2021 |
| RSA | AES128 | CBC | SHA(1) | 2021 |

Tabelle 63: DTLSv1.0 Cipher Suites

### DTLS Version 1.2

Die folgenden Cipher Suites sind bei der Verwendung DTLSv1.2 in der empfohlenen Reihenfolge freigegeben.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verfahren | Verschlüsselung | Chiffrierungsmodus | Hash | Gütlig bis |
| ECDHE\_ECDSA | AES256 | GCM | SHA(2)-384 | 2026+ |
| ECDHE\_ECDSA | AES128 | GCM | SHA(2)-256 | 2026+ |

Tabelle 64: DTLSv1.2 Cipher Suites

## Mailverschlüsselung

### S/MIME

Emails können mit dem S/MIME-Standard so verschickt werden, dass der Inhalt vertraulich bleibt und der Empfänger die Unverfälschtheit der Nachricht und der Absenderinformationen prüfen kann. Beide Kommunikationspartner müssen dazu ein Zertifikat einsetzen. Für interne Nutzer gelten dabei die Vorgaben der Variante 1. Den Zertifikaten externer Nutzer ist nur dann zu vertrauen, wenn diese ebenfalls die Anforderungen aus Variante 1 erfüllen.

S/MIME Versionen

Seitens der <Institution> wird die Verwendung von S/MIME 4.0 empfohlen. Insofern bei der Verwendung S/MIME 3.2 oder 3.1 sichergestellt wird, dass die nachfolgend aufgeführten Algorithemen und deren Parameter eingehalten werden, können in bestehenden Anwendungen diese weiter eingesetzt werden.

| Version | Gültig bis |
| --- | --- |
| S/MIME 4.0 | 2026+ |
| S/MIME 3.2 | 2024 |
| S/MIME 3.1 | 2021 |
| S/MIME 3.0 | 2021 |

Tabelle 65: S/MIME Versionen

Hash-Algorithmen

S/MIME verwendet Hashfunktionen insbesondere bei der Signierung von Nachrichten sowie bei der Ableitung von Schlüsseln. Eines der in der Tabelle aufgeführten Verfahren muss verwendet werden.

| Algorithmus | Länge des Hashwertes | Gültig bis |
| --- | --- | --- |
| SHA(2)-512 | 512 | 2027+ |
| SHA(2)-256 | 256 | 2027+ |
| SHA(2)-224 | 224 | 2022 |

Tabelle 66: S/MIME - Hash-Algorithmen

Signatur-Algorithmen

Für die Erstellung von Signaturen mit S/MIME muss eines der benannten Signaturverfahren verwendet werden.

| Algorithmus | Schlüssellänge | Gültig bis |
| --- | --- | --- |
| ECDSA | 224 Bit | 2022 |
| ECDSA | ≥ 256 Bit | 2027+ |
| RSASSA-PSS | 2048 Bit | 2022 |
| RSASSA-PSS | ≥ 3072 Bit | 2027+ |

Tabelle 67: S/MIME - Signatur-Algorithmen

Inhalte-Verschlüsselung (Content Encryption)

Für die Content Encryption sind die beiden folgenden Verfahren zu verwenden.

| Verfahren | Chiffrierungsmodus | Gültig bis |
| --- | --- | --- |
| AES128 | CBC | 2025 |
| AES256 | GCM | 2027+ |
| AES128 | GCM | 2027+ |

Tabelle 68: S/MIME – Inhalte-Verschlüsselung

Schlüsselverschlüsselung (Key Enryption)

Abhängig von der verwendeten Kryptographie wird der Content Encryption Key direkt mit dem öffentlichen Schlüssel des Empfängers asymmetrisch verschlüsselt (Key Transport) oder der Sender erzeugt ein ephemeres Schlüsselpaar und leitet aus diesem und dem öffentlichen Schlüssel des Empfängers einen symmetrischen Schlüssel ab (Key Agreement), mit dem der Content Encryption Key dann symmetrisch verschlüsselt wird.

Für die asymmetrische Key Encryption sind die nachfolgend aufgeführten Verfahren anzuwenden.

| Algorithmus | Schlüssellänge | Verwendung bis |
| --- | --- | --- |
| RSAES-OAEP | 2048 Bit | 2022 |
| RSAES-OAEP | 3072 Bit | 2027+ |

Tabelle 69: Asymmetrische Key Encryption bei S/MIME

Für die Key Encryption via Schlüsseleinigung muss ein Verfahren aus der nachfolgenden Tabelle verwendet werden.

|  |  |  |
| --- | --- | --- |
| Algorithmus | Schlüssellänge | Verwendung bis |
| Schlüsselaushandlung |
| ECDH | 224 Bit | 2022 |
| ECDH | 256 Bit | 2027+ |
| **Key-Wrap-Algorithmus** |
| AES-Wrap | 128 Bit | 2027+ |

Tabelle 70: Key Encryption via Schlüsseleinigung bei S/MIME