

Sicherheitsanforderung

Kryptographische Algorithmen und Sicherheitsprotokolle

Deutsche Telekom Gruppe

Version	8.0
Datum	01.12.2025
Status	Freigegeben

Impressum

Herausgeber
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Deutschland

Dateiname	Dokumentennummer 3.50	Dokumententyp Sicherheitsanforderung
Version 8.0	Stand 01.12.2025	Status Freigegeben
Fachlicher Ansprechpartner Telekom Security psa.telekom.de	Gültigkeitsdauer 01.12.2025 - 30.11.2030	Freigegeben von Stefan Pütz, Leiter SEC-T-TST

Zusammenfassung
Kryptographische Algorithmen und Sicherheitsprotokolle

Copyright © 2025 by Deutsche Telekom AG.

Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Inhaltsverzeichnis

1.	Einleitung	4
1.1.	Allgemeine Einleitung	4
1.2.	Anmerkungen	4
1.3.	Nutzerrollen	4
2.	Allgemeine Grundsätze für kryptographische Algorithmen	5
3.	Symmetrische und asymmetrische Verfahren	9
4.	Hash-Funktionen	13
5.	Message Authentication Codes (MAC)	15
6.	Schlüsselverwaltung	18
6.1.	Erzeugung von Schlüsselmaterial	18
6.2.	Schlüsselvereinbarung und -austausch	19
6.3.	Asymmetrische Key Encapsulation Mechanismen (KEM)	20
6.4.	Schlüsselableitung	22
6.5.	Transport und Speicherung von Schlüsseln	24
6.6.	Schlüsselsperrung und Vernichtung	27
7.	Signaturalgorithmen	29
8.	Verwendung von kryptographischen Verfahren in Sicherheitshardware	31
9.	Kryptographische Verfahren zur sicheren Speicherung und Verifikation von Passwörtern	32
10.	Internet Protocol Security (IPSec)	35
10.1.	Allgemeine IPSec Parameter	35
10.2.	Phase 1 (ISAKMP Security Association / IKE-SA)	40
10.3.	Phase 2 (Quick Mode / Child-SA)	42
11.	Transport Layer Security (TLS/SSL) Cipher Suites	44
12.	Secure Shell (SSH) Cipher Suites	52

1. Einleitung

1.1. Allgemeine Einleitung

Dieses Dokument wurde auf Basis der Vorgaben aus der Konzernrichtlinie Sicherheit (Group Security Policy) erstellt. Die Sicherheitsanforderung dient u.a. als Grundlage zur Freigabe im PSA-Prozess.

Sie dient darüber hinaus als Umsetzungsempfehlung für Vorgaben der Konzernrichtlinie Sicherheit (Group Security Policy) in Einheiten, die nicht am PSA-Verfahren teilnehmen. Diese Anforderungen müssen bereits während der Planungs- und Entscheidungsprozesse berücksichtigt werden.

Bei der Umsetzung der Sicherheitsanforderung ist das jeweils vorrangige nationale, internationale und supranationale Recht zu beachten.

Sollte die Einhaltung der beschriebenen Anforderungen in individuellen Fällen nicht oder nur eingeschränkt umsetzbar sein, ist gemeinsam mit einem Sicherheits- und/oder Datenschutzexperten (entsprechend der betroffenen Anforderung) eine Risikobewertung vorzunehmen und mögliche alternative Schutzmaßnahmen abzustimmen.

1.2. Anmerkungen

Dieses Dokument bildet die allgemeine Anforderungsbasis für die Arbeit mit kryptographischen Algorithmen. Das bedeutet, dass nicht immer alle Anforderungen relevant sind, falls system- und/oder anwendungsspezifische kryptographische Anforderungen bereits existieren. Die in diesem Dokument geforderten Algorithmen und Verfahren sind jeweils so sortiert, dass die stärkste (favorisierte) Variante als erstes genannt wird. Die Bezeichnung der Algorithmen in diesem Dokument ist so gewählt, dass sie mit den zitierten Standards übereinstimmt.

Das Dokument enthält Anforderungen an die Entwicklung von kryptographischen Systemen, aber auch Anforderungen an die Entwicklung von Systemen, die fertige kryptographische Komponenten / Module einsetzen, um die Systeme und Systemschnittstellen abzusichern.

Kryptographische Algorithmen für Mobilfunknetze sind nicht Gegenstand der Betrachtung in diesem Dokument. Diese Anforderungen sind bereits in Spezifikationen der 3rd Generation Partnership Project (3GPP) und der European Telecommunication Standardization Institution (ETSI) veröffentlicht.

Auf die referenzierten URLs wurde zuletzt am 30.09.2025 zugegriffen.

Hinweis zu Datenschutz:

Die Anforderungen in dem vorliegenden Dokument richten sich an die Daten- und Informationssicherheit im Sinne Vertraulichkeit und Integrität. Sollen kryptografische Methoden eingesetzt werden, um Anonymisierungs- bzw. Pseudonymisierungsanforderungen des Datenschutzes zu erfüllen, findet man die Bedingungen für den Einsatz der Methoden im Dokument „Datenschutzanforderung: Anonymisierung und Pseudonymisierung“.

1.3. Nutzerrollen

Wir haben Nutzerrollen eingeführt und diese Information zum Langtext einer jeden Anforderung hinzugefügt, um den Adressatenkreis zu bestimmen.

Folgende Definitionen wurden eingeführt:

Entwicklung:

Diese Nutzerrolle beinhaltet alle Tätigkeiten zur Entwicklung von kryptographischen Systemen. Hierzu zählt das Entwickeln von eigenem Programmcode und oder Krypto-Bibliotheken (z.B. eine eigene C-Implementierung von TLS und deren kryptografischen Primitiven erstellen).

Integration:

Diese Nutzerrolle beinhaltet alle Tätigkeiten zur Integration bestehender kryptografischer Module oder auch Bibliotheken und Einbettung in einer spezifischen Anwendung (z.B. Einbindung von OpenSSL Bibliotheken in einen Anwendungsprogrammcode zur Verschlüsselung einer Netzwerk-Schnittstelle).

Betrieb:

Diese Nutzerrolle beinhaltet alle Tätigkeiten für den Betrieb eines kryptographischen Systems. Darunter versteht man das Zertifikatsmanagement, Schlüsselmanagement und Härtung von kryptografischen Anwendungen (z.B. Härtung des SSH-Daemons).

2. Allgemeine Grundsätze für kryptographische Algorithmen

Req 1 Es dürfen nur standardisierte kryptographische Verfahren und Primitive verwendet werden, die von anerkannten Organisationen veröffentlicht werden.

Nutzerrollen: Entwicklung, Integration

Jede Abweichung muss mit der Sicherheitsorganisation der Deutschen Telekom Gruppe abgestimmt und vereinbart werden.

Die in diesem Dokument definierten Anforderungen entsprechen den zur Zeit der Veröffentlichung dieses Dokuments gültigen Sicherheitsstandards folgender anerkannter Organisationen:

- Bundesamt für Sicherheit in der Informationstechnik (BSI) [BSI]
- European Telecommunication Standards Institute (ETSI) [ETSI]
- Senior Officials Group Information Systems Security (SOGIS) [SOGIS]
- National Institute of Standards and Technology (NIST) [NIST]
- International Organization for Standardization (ISO) [ISO]
- Internet Engineering Task Force (IETF) [IETF]

Quellen:

[BSI] https://www.bsi.bund.de/DE/Home/home_node.html

[ETSI] <https://www.etsi.org>

[IETF] <https://www.ietf.org>

[ISO] <https://www.iso.org/home.html>

[NIST] <https://www.nist.gov>

[SOGIS] <https://www.sogis.eu>

Motivation: Nur ausreichend untersuchte Algorithmen bieten einen verlässlichen systematischen Schutz gegen kryptographische Angriffe. Damit wird sichergestellt, dass die Empfehlungen auf einem breiten fachlichen Konsens basieren.

Umsetzungsbeispiel: Standards der ISO/IEC JTC 1/SC 27 „IT Security techniques“, BSI Technische Richtlinie für SSH-Anwendungen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-1/8.0

Req 2 Für die Implementierung von kryptografischen Verfahren müssen etablierte und aktuelle Krypto-Bibliotheken genutzt werden.

Nutzerrollen: Entwicklung, Integration

Eigene Implementierungen von kryptographischen Algorithmen sollen vermieden werden. Stattdessen müssen etablierte und aktuelle kryptographische Bibliotheken genutzt werden, welche die in diesem Dokument freigegebenen Algorithmen implementiert haben. Wenn eigene Implementierungen erforderlich sind, müssen Best Practices beachtet werden und die Sicherheitsorganisation der Deutschen Telekom Gruppe zur Beratung hinzugezogen werden.

Motivation: Nur weit verbreitete kryptographischen Bibliotheken wurden in ausreichendem Maße durch Sicherheitsforscher analysiert und bieten einen hinreichenden Schutz. Diese Bibliotheken werden regelmäßig aktualisiert (u.a. bei Sicherheitsschwachstellen).

Umsetzungsbeispiel: Botan, Bouncy Castle, GnuTLS, OpenSSL, wolfCrypt

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-2/8.0

Req 3 Krypto-Agilität muss gewährleistet werden.

Nutzerrollen: Entwicklung, Integration

Kryptographische Verfahren (einschließlich Algorithmen, Schlüsselgrößen und Protokolle) sind in austauschbaren, klar abgegrenzten Modulen zu implementieren. Die Auswahl und Konfiguration der Verfahren darf nicht fest im Code hinterlegt sein, sondern muss über konfigurierbare, autorisierte Schnittstellen möglich sein.

Es sind Funktionen vorzusehen, um kryptographische Verfahren zu deaktivieren, zu ersetzen und bei persistent gespeicherten Daten eine Migration auf andere Verfahren durchzuführen. Dies schließt insbesondere die Möglichkeit ein, bei Bedarf auf andere kryptographische Verfahren zu wechseln, wenn z.B. das implementierte Verfahren nicht mehr freigegeben ist. Für solche Wechsel sollte die Hardware bzgl. Performance und Speicherkapazitäten Reserven bereithalten oder muss erweiterbar sein.

Die Umsetzung muss nach den folgenden Prinzipien erfolgen [TR-02102-1]:

- Modularisierung der kryptographischen Implementierung
- Trennung von Implementierung und Konfiguration
- Dokumentierte Migrationspfade für den Wechsel auf neue Verfahren
- Vermeidung harter Kodierungen kryptographischer Parameter

Quelle:

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Krypto-Agilität ist von besonderer Bedeutung im Hinblick auf die zu erwartenden Auswirkungen von Quantum Computing, das viele aktuell verwendete asymmetrische Verfahren (z. B. RSA, ECC) in absehbarer Zeit brechen könnte. Da Post-Quantum-Kryptographie (PQC)-Algorithmen derzeit noch nicht final standardisiert und nicht in allen Anwendungsfällen hinreichend analysiert sind, muss das System so ausgelegt sein, dass ein späterer Wechsel auf vom BSI oder internationalen Normierungsorganisationen empfohlene PQC-Verfahren ohne grundlegende Änderungen der Systemarchitektur und mit minimaler Betriebsunterbrechung erfolgen kann.

Umsetzungsbeispiel: Austausch einer Krypto-Bibliothek in einer Webanwendung, Wechsel der Verschlüsselung von AES-128-CBC auf AES-256-GCM oder Ersetzen eines klassischen kryptographischen Verfahrens mit einem hybriden Verfahren aus klassischer Kryptographie und Post-Quanten-Kryptographie (PQC).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-3/8.0

Req 4	Für langfristig schutzbedürftige Daten, deren Vertraulichkeit über das Jahr 2031 hinaus gewahrt werden muss, müssen Hybrid-Implementierungen aus klassischer Kryptographie und Post-Quanten-Kryptographie (PQC) verwendet werden.
-------	---

Nutzerrolle: Entwicklung, Integration

Klassische quantenunsichere Verfahren (z. B. RSA, ECC ohne PQC-Hybrid) dürfen in diesen Fällen nicht allein eingesetzt werden, da mit diesen die Vertraulichkeit über das Jahr 2031, insbesondere aufgrund der Entwicklung bei Quantencomputern, nicht sichergestellt ist [TR-02102-1].

Da die PQC-Verfahren vergleichsweise neu und damit nicht den gleichen Evaluierungsgrad wie die klassischen kryptographischen Verfahren, beispielsweise bezüglich der Resistenz gegen Seitenkanalattacken, der Fehlerinjektion und der Implementierungssicherheit haben, sollten auch diese nicht alleine eingesetzt werden.

Diese Anforderung dient der Absicherung vertraulicher oder anderweitig schutzbedürftiger Daten gegen zukünftige Angriffe durch leistungsfähige Quantencomputer. Die Auswahl und Konfiguration der Verfahren muss den jeweils aktuellen Standards ([NIST-PQC]) und Empfehlungen des BSI ([TR-02102-1], [TR-02102-3], [BSI-PQC]) folgen.

Quellen:

[BSI-PQC] BSI: Migration zu Post-Quanten-Kryptografie, 2020

[FIPS-203] NIST FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard

[FIPS-204] NIST FIPS 204, Module-Lattice-Based Digital Signature Standard

[FIPS-205] NIST FIPS 205, Stateless Hash-Based Digital Signature Standard

[NIST-PQC] NIST PQC Standardization: <https://csrc.nist.gov/projects/post-quantum-cryptography>

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: Quantencomputer werden voraussichtlich in der Lage sein, gängige asymmetrische Verfahren wie RSA und ECC zu brechen. Daten mit einer langen Aufbewahrungsfrist – insbesondere über 2031 hinaus – müssen daher bereits heute so geschützt werden, dass ihre Vertraulichkeit und Integrität auch in einer Post-Quanten-Ära gewährleistet ist. Hybride Verfahren erlauben eine sichere Übergangsphase bis zur breiten Verfügbarkeit und Standardisierung von PQC-Verfahren.

Umsetzungsbeispiel: Schlüsseltausch / Key Encapsulation

- ML-KEM – NIST standardisierter Algorithmus (finalisiert), auch bekannt als FIPS 203, Weiterentwicklung von CRYSTALS-Kyber, verfügbar in OpenSSL 3.5.0+, liboqs 0.11.0+, WolfSSL 5.7.4+, Golang v1.24+ (letzteres nur als Hybrid mit X25519, siehe unten)
- HQC (Hamming Quasi-Cyclic) – vom NIST für die Standardisierung als Alternative zu ML-KEM ausgewählt, noch nicht finalisiert
- FrodoKEM – ausgewählt für die Standardisierung durch ISO, aber nicht durch NIST, verfügbar in liboqs, geeignet für spezielle Anwendungsfälle, noch nicht abgeschlossen
- Klassisches McEliece – NIST erwägt eine Standardisierung, nachdem die Standardisierung nach ISO abgeschlossen ist, nur für sehr spezielle Anwendungsfälle nützlich, da es einen großen Overhead für den Austausch öffentlicher Schlüssel bietet, aber eine schnelle Kapselung und Entkapselung. Die Standardisierung ist noch nicht abgeschlossen.
- Tatsächliche Hybrid-Verschlüsselung "X25519MLKEM768" (Hybrid aus EC mit curve25519 und PQC unter

Verwendung von ML-KEM mit 768 Bits) – Unterstützt von Openssl 3.5.0+ und von der crpto-Bibliothek von Go v1.24, und daher von der meisten Software, die mit dieser Go-Version programmiert wurde (z.B. Traefik v3.5)

Digitale Signaturen

- ML-DSA – NIST-Standardisierter Algorithmus (finalisiert), auch bekannt als FIPS 204, Weiterentwicklung von CRYSTALS-Dilithium, verfügbar in Openssl 3.5.0+, liboqs 0.12.0+, WolfSSL 5.8.0+
- SLH-DSA – NIST-standardisierter Algorithmus (finalisiert), auch bekannt als FIPS 205 als Alternative zu ML-DSA. Es basiert auf SPHINCS+, verfügbar in OpenSSL v3.5.0+, liboqs
- FN-DSA – basierend auf Falcon, NIST-Standardisierung in Arbeit, wird als FIPS 206 veröffentlicht, kleine Signatursgröße, verfügbar in liboqs, noch nicht finalisiert

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-4/8.0

3. Symmetrische und asymmetrische Verfahren

Req 5 Als symmetrische Blockverschlüsselung müssen AES-256, AES-192 oder AES-128 verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Der „Advanced Encryption Standard“ (AES) ist eine im Jahre 2001 standardisierte und seitdem intensiv untersuchte Variante des „Rijndael“-Algorithmus. Der Algorithmus ist in [ISO-18033-3] und in [FIPS-197] standardisiert. AES ist eine Blockverschlüsselung, die 128 Bit lange Blöcke verarbeitet und eine Schlüssellänge von 128, 192 bzw. 256 Bit unterstützt. AES-128, AES-192 und AES-256 bezeichnen die entsprechenden Schlüssellängen.

Zum Schutz vor zurzeit lediglich theoretischen Angriffen mit Quantencomputern (Grover-Algorithmus) wird empfohlen bereits heute eine Schlüssellänge von 256 Bit zu wählen [TR-02102-1].

Quellen:

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

[ISO-18033-3] ISO/IEC, Encryption algorithms Part 3: Block ciphers

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: AES ist intensiv untersucht, hat langanhaltender Kryptoanalyse standgehalten und bietet mit 256 Bit Schlüssellänge ausreichend Sicherheit gegenüber zukünftigen Angriffen mit Quantencomputern.

Umsetzungsbeispiel: Ein Webserver nutzt eine Cipher Suite mit AES um Client Verbindung mit TLS abzusichern: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-5/8.0

Req 6 Bei symmetrischer Blockverschlüsselung müssen folgende Authenticated Encryption with Associated Data (AEAD) Betriebsarten verwendet werden: Galois-Counter-Mode (GCM) oder Counter with CBC-MAC (CCM).

Nutzerrollen: Betrieb, Entwicklung, Integration

Wenn die Länge der zu verschlüsselnden Daten größer als die AES-Blocklänge (128 Bit) ist, muss eine Betriebsart verwendet werden, um alle Daten zu verschlüsseln.

GCM und CCM sind Authenticated Encryption with Associated Data (AEAD) Betriebsarten und bieten Vertraulichkeit und Integrität für den Klartext, sowie optional Integrität für zusätzliche Daten (associated data), wie beispielsweise einem Protokoll-Header. GCM und CCM sind in [ISO-19772], [NIST-SP-800-38C] und [NIST-SP-800-38D] standardisiert. Solange die Anwendung AEAD unterstützt, ist Authenticated Encryption (GCM, CCM) den klassischen Betriebsarten (CBC, CTR, OFB) vorzuziehen, da CBC, CTR und OFB nur Vertraulichkeit bieten.

Anforderungen an AEAD-Betriebsarten [NIST-SP-800-38D]:

- Anforderungen an den Galois-Counter-Mode (GCM):
 - Der Authentication Tag muss mindestens 96 Bit lang sein.
 - Der Initialisierungsvektor (Nonce) muss zufällig generiert werden.
 - Der Initialisierungsvektor (Nonce) darf sich bei gleichem Schlüssel nicht wiederholen und soll 96 Bit lang sein. Kürzere Initialisierungsvektoren sind nicht erlaubt.
 - Der gleiche Schlüssel darf für maximal 2^{32} Iterationen verwendet werden.
- Anforderungen an den Counter Mode mit CBC-MAC (CCM):

- Der Authentication Tag muss mindestens 96 Bit lang sein.
- Der Initialisierungsvektor (Nonce) muss zufällig generiert werden.
- Der gleiche Schlüssel darf für maximal 2^{32} Iterationen verwendet werden.

Quellen:

[ISO-19772] ISO/IEC, Authenticated encryption

[NIST-SP-800-38C] NIST, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality

[NIST-SP-800-38D] NIST, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC

Motivation: Werden Betriebsarten falsch implementiert, kann es zu einer Schwächung des kryptographischen Systems kommen.

Umsetzungsbeispiel: Nutzung von AES-128-GCM als AEAD-Chiffre im TLS v1.3 Kontext.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-6/8.0

Req 7	Falls bei symmetrischer Verschlüsselung keine AEAD-Betriebsart zur Verfügung steht, muss eine der folgenden Vertraulichkeits-Betriebsarten verwendet werden: Counter Mode (CTR), Cipher-Block Chaining (CBC) und Output Feedback (OFB).
-------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

Wenn die Länge der zu verschlüsselnden Daten größer als die AES-Blocklänge (128 Bit) ist, muss eine Betriebsart verwendet werden, um alle Daten zu verschlüsseln.

AEAD-Betriebsarten sollen verwendet werden, solange es die Anwendung erlaubt.

Die klassischen Betriebsarten CBC, CTR und OFB bieten lediglich Vertraulichkeit für den Klartext, weshalb sie generell durch separate Mechanismen zum Integritätsschutz (z.B. Message Authentication Codes (MAC)) ergänzt werden müssen. Die Vertraulichkeits-Betriebsarten sind in [ISO-10116] und [NIST-SP-800-38A] standardisiert.

Anforderungen an Betriebsarten die lediglich Vertraulichkeit bieten (vergleiche Annex B aus ISO/IEC 10116):

Anforderungen an den Counter Mode (CTR):

- Der Zählerstand (=Counter) und damit auch der initiale Zählerstand (Initialisierungsvektor) sind 128 Bit lang.
- Der Zähler darf sich bei gleichem Schlüssel nicht wiederholen.
- Der Initialisierungsvektor ist so zu wählen, dass sich der Zählerstand bei gleichem Schlüssel nicht wiederholt.

Anforderungen an den Output Feedback Modus (OFB):

- Der Initialisierungsvektor (IV) ist 128 Bit lang und muss zufällig gewählt werden.

Anforderungen an den Cipher Block Chaining Mode (CBC):

- Der Initialisierungsvektor (IV) ist 128 Bit lang und muss zufällig gewählt werden.
- Der Initialisierungsvektor darf nicht beeinflussbar oder ableitbar sein.
- Es muss ein Padding-Verfahren gemäß ISO, CMS, ESP, oder Ciphertext Stealing verwendet werden:

Verfahren	Beschreibung	Referenzspezifikation
ISO (Method 2)	ISO/IEC 9797-1 Padding Method 2	[ISO-9797-1]
CMS	Cryptographic Message Syntax (CMS) Padding	[RFC-5652]
ESP	Encapsulating Security Payload (ESP) Padding	[RFC-4303]
Ciphertext Stealing	Ciphertext Stealing (CTS)	[NIST-SP-800-38A-Add]

Hinweis zu CBC:

Symmetrische Blockchiffren besitzen eine feste Blocklänge (z. B. AES 128 Bit) . Bei einer Verwendung der Betriebsart CBC, muss die Länge der verschlüsselnden Daten ein Vielfaches der Blocklänge ($N \times 128$ Bit) betragen. Padding-Verfahren übernehmen diese Aufgabe und füllen den letzten Datenblock auf ein Vielfaches der festen Blocklänge ($N \times 128$ Bit) auf.

Hinweis zu Ciphertext Stealing:

Mithilfe von der abgewandelten Betriebsart Cipher Block Chaining (CBC) mit "Ciphertext Stealing" CBC-CS kann die Erweiterung des Schlüsseltextes verhindert werden. Fehlende Bits werden mit Chiffretext aufgefüllt.

Quellen:

[ISO-9797-1] ISO/IEC, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher

[ISO-10116] ISO/IEC, Modes of operation for an n-bit block cipher

[NIST-SP-800-38A] NIST, Recommendation for Block Cipher Modes of Operation: Methods and Techniques

[NIST-SP-800-38A-Add] NIST, Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode

[RFC-4303] IP Encapsulating Security Payload (ESP)

[RFC-5652] Cryptographic Message Syntax (CMS)

Motivation: Geringere Anzahl von Implementierungsfehlern.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.50-7/8.0

Req 8	Bei asymmetrischen kryptographischen Verfahren basierend auf dem Faktorisierungsproblem oder dem Diskreten-Logarithmus-Problem muss eine Mindestschlüssellänge von 3000 Bit sichergestellt sein, bei Verfahren basierend auf elliptischen Kurven (ECC) 250 Bit.
-------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

Bei der Verwendung von (klassischen) asymmetrischen Algorithmen und Verfahren müssen mindestens die folgenden Schlüssellängen eingehalten werden ([TR-02102-1]):

- Verfahren basierend auf dem Faktorisierungsproblem wie RSA: 3000 Bit
- Verfahren basierend auf dem Diskreten-Logarithmus-Problem wie Diffie-Hellman (DH) oder Digital Signature Algorithm (DSA): 3000 Bit
- Verfahren basierend auf elliptischen Kurven (ECC) wie Elliptic Curve Diffie-Hellman (ECDH), Elliptic Curve Digital Signature Algorithm (ECDSA) oder Edwards-Curve Digital Signature Algorithm (EdDSA): 250 Bit

Hinweis zu zu PQC:

Wenn eine hybride Implementierung von klassischer und Post-Quanten-Kryptographie verwendet werden, dann muss der klassische Teil dieser Anforderung entsprechen.

Quelle:

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Die geforderten Schlüssellängen gelten bis mindestens 2030 als sicher.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-8/8.0

4. Hash-Funktionen

Req 9	Es müssen kryptographische Hash-Funktionen der SHA-3 und SHA-2 Familie mit einer Ausgabelänge von mindestens 256 Bit verwendet werden.
-------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Die Sicherheit einer Hash-Funktion wird maßgeblich durch ihre Ausgabelänge bestimmt, welche als Suffix angegeben wird (z.B. hat SHA-3-512 eine Ausgabelänge von 512 Bit).

- Standardisierte Varianten der SHA-2 Familie sind SHA-512, SHA-384 und SHA256 ([ISO-10118-3] und [FIPS-180-4]).
- Standardisierte Varianten der SHA-3 Familie sind SHA-3-512, SHA-3-384 und SHA-3-256 ([ISO-10118-3] und [FIPS-202]).

Zum Schutz vor zurzeit lediglich theoretischen Angriffen mit Quantencomputern (Grover-Algorithmus) wird empfohlen bereits heute eine Ausgabelänge von größer gleich 384 Bit zu wählen [TR-02102-1].

Die zulässigen Anwendungsgebiete für diese Hash-Funktionen sind:

- Digitale Signaturen
- Message Authentication Codes (MAC)
- Mask generation functions (MGF)
- Schlüsselableitungsfunktionen (Key Derivation Function = KDF) und passwortbasierte Schlüsselableitung (PBKDF)
- Pseudozufallszahlengeneratoren (PRNG)
- Integritätsschutz

Hinweis zu Authentizität und Vertraulichkeit:

Hash-Funktionen bieten keine Vertraulichkeit (im Sinne von Verschlüsselung) und keinen Schutz vor aktiven Angriffen (z.B. gegenüber Manipulationen). Hash-Funktionen sind öffentlich bekannte und schlüssellose Funktionen. Deshalb kann bei Angriffen, bei denen eine Nachricht mit Hashwert manipuliert wird, ebenfalls einfach der Hashwert der neuen Nachricht berechnen werden. Wenn die Authentizität einer Nachricht mit symmetrischen Verfahren garantiert werden soll, müssen Message Authentication Codes (wie z.B. HMAC) verwendet werden.

Hinweis zu SHA-224/SHA-3-224:

SHA-224/SHA-3-224 darf nur noch für Message Authentication Codes (MAC) und zur Schlüsselableitung (KDF) in Bestandssystemen genutzt werden und sollte gegen Hash Funktionen längerer Outputlänge (≥ 256 Bit) ausgetauscht werden.

Quellen:

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

[FIPS-202] NIST, FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

[ISO-10118-3] ISO/IEC, Hash-functions Part 3: Dedicated hash-functions

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Kryptographische Hash-Funktionen werden in vielen Anwendungsgebieten eingesetzt und bilden die Grundlage vieler kryptographischer Verfahren, wie digitalen Signaturen.

Umsetzungsbeispiel: Verwendung einer Hash-Funktion für die digitale Signatur eines Zertifikats.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

5. Message Authentication Codes (MAC)

Req 10 Für Message Authentication Codes (MACs) müssen die MAC-Verfahren HMAC, CMAC, GMAC oder KMAC mit den freigegebenen Parametern verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Message Authentication Codes (MAC) dienen dem Integritätsschutz von Nachrichten in Sicherheitsprotokollen wie beispielsweise in IPSec oder TLS/SSL. Zudem werden Message Authentication Codes in anderen kryptographischen Verfahren, wie Key Derivation Functions eingesetzt.

Folgende Tabelle beschreibt die zulässigen Algorithmen und Parameter:

MAC-Verfahren	Zulässige Algorithmen	Besondere Betriebsbedingungen / Bemerkungen	Referenzspezifikationen
HMAC	SHA-512 SHA-384 SHA-256 SHA-3-512 SHA-3-384 SHA-3-256	Die Schlüssellänge muss mindestens 128 Bit betragen. Die Länge des MAC muss mindestens 128 Bit betragen.	[ISO-9797], [FIPS-198]
CMAC	AES-256 AES-192 AES-128	Die Schlüssellänge wird durch die AES-Variante vorgegeben (128/192/ 256 Bit). Die Länge des MAC muss mindestens 96 Bit betragen.	[ISO-9797], [NIST-SP-800-38B]
GMAC	AES-256 AES-192 AES-128	Die Schlüssellänge wird durch die AES-Variante vorgegeben (128/192/ 256 Bit). Der Initialisierungsvektor muss mindestens 96 Bit betragen. Die Länge des MAC muss mindestens 96 Bit betragen.	[ISO-9797], [NIST-SP-800-38D]
KMAC128, KMAC256		Die Länge des MAC muss mindestens 96 Bit betragen.	[NIST-SP-800-185]

Zum Schutz vor derzeit nur theoretischen Angriffen mit Quantencomputern (Grover-Algorithmus) wird empfohlen, für AES eine Schlüssellänge von 256 Bit und für Hash-Funktionen eine Ausgabelänge größer oder gleich 384 Bit zu wählen [TR-02102-1].

Hinweis zur Länge des MAC

Kürzere MAC-Längen als 96 Bit bei CMAC, GMAC und KMAC sowie 128 Bit bei HMAC sind nur für bestehende Systeme zulässig und sollten zeitnah gegen längere MAC-Längen ausgetauscht werden.

Hinweis zu CMAC:

Einfache Implementierungen des CBC-MAC sind nicht sicher gegenüber Erweiterungsangriffen. Eine in [ISO-9797] oder [NIST-SP-800-38B] standardisierte und sichere Variante des CBC-MAC ist CMAC (MAC Algorithm 5 in [ISO-9797]).

Einschränkungen zu SHA-224/SHA-3-224/SHA-1:

Die Nutzung von SHA-224/SHA-1 für Message Authentication Codes ist nur noch für bestehende Systeme zulässig, wenn kein stärkeres Verfahren verfügbar ist und sollte zeitnah gegen stärkere Varianten ausgetauscht werden (mindestens SHA-256/SHA-3-256).

Quellen:

[FIPS-198] NIST, FIPS198-1, The Keyed-Hash Message Authentication Code (HMAC)

[ISO-9797] ISO/IEC, Message Authentication Codes (MACs), Part 1: Mechanisms using a block cipher, Part 2: Mechanisms using a dedicated hash-function

[NIST-SP-800-38B] NIST, Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication

[NIST-SP-800-38D] NIST, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC

[NIST-SP-800-185] NIST, SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Message Authentication Codes (MAC) dienen dem Integritätsschutz von Nachrichten in Sicherheitsprotokollen wie beispielsweise in IPsec oder TLS/SSL. Zudem werden Message Authentication Codes in anderen kryptographischen Verfahren, wie Key Derivation Functions eingesetzt.

Umsetzungsbeispiel: Integritätsprüfung von empfangenen Nutzungsdatensätzen zur Rechnungserstellung mittels HMAC-SHA-256.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.50-10/8.0

Req 11	Falls bei symmetrischer Verschlüsselung lediglich Vertraulichkeits-Betriebsarten (CTR, CBC, OFB) verwendet werden, müssen zum Schutz der Integrität Message Authentication Codes über den Ciphertext berechnet werden („Encrypt-then-MAC“).
--------	---

Nutzerrolle: Entwicklung

Wenn keine Authenticated-Encryption-Betriebsart verwendet wird, muss ein Message Authentication Code verwendet werden, um die Authentizität der Daten herzustellen. Die einzige generische Konstruktion aus Verschlüsselung und MAC die der stärksten Sicherheitsanforderung genügt ist „Encrypt-then-MAC“. Dabei wird die Nachricht zunächst verschlüsselt und anschließend ein MAC (z.B. HMAC) über die Nachricht berechnet. Andere Kombinationen aus Verschlüsselung und MAC können zwar unter bestimmten Rahmenbedingungen sicher implementiert werden, allerdings wird von der generellen Verwendung anderer generischer Konstruktionen als „Encrypt-then-MAC“ abgeraten.

Hinweis zur Verwendung in standardisierten Protokollen:

Standardisierte Sicherheitsprotokolle wie SSH, SSL/TLS oder IPsec verwenden verschiedene generische Kombinationen aus Verschlüsselung und MAC. Dies hat in der Vergangenheit zu verschiedenen Angriffen geführt woraufhin die Protokolle überarbeitet wurden. Die Anforderungen an diese Sicherheitsprotokolle sind in separaten Abschnitten dieses Dokuments aufgeführt. Es wird empfohlen dedizierte Authenticated Encryption Konstruktionen zu verwenden

Motivation: Vermeidung von konstruktiven Schwachstellen in der Verschlüsselung.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

6. Schlüsselverwaltung

6.1. Erzeugung von Schlüsselmaterial

Req 12	Bei der Erzeugung von Schlüsselmaterial für kryptographische Algorithmen müssen entweder physische Zufallszahlengeneratoren oder folgende kryptographisch starke Pseudozufallszahlengeneratoren (deterministische Quellen) mit ausreichender Entropie verwendet werden: HMAC-DRBG, Hash-DRBG oder CTR-DRBG.
--------	---

Nutzerrollen: Entwicklung, Integration

Wird für die Erzeugung kryptographischer Schlüssel ein Pseudozufallszahlengenerator (deterministic random bit generator = DRBG) verwendet, muss in die Berechnung des Startwertes (Seed) ein durch einen Menschen unbeeinflussbarer (nichtdeterministischer) Wert einfließen. Dieser Wert muss einen hinreichend großen Informationsgehalt (Entropie) aufweisen.

Pseudozufallszahlengenerator	basierend auf	Referenz-Spezifikationen
HMAC-DRBG	SHA-256, SHA-512	[NIST-SP-800-90A], [ISO-18031]
Hash-DRBG	SHA-256, SHA-512	[NIST-SP-800-90A], [ISO-18031]
CTR-DRBG	AES-128, AES-256	[NIST-SP-800-90A], [ISO-18031]

Quellen:

[ISO-18031] ISO/IEC, Random bit generation

[NIST-SP-800-90A] NIST, Recommendation for Random Number Generation Using Deterministic Random Bit Generators

Motivation: Werden Ausgangswerte mit unterschiedlicher Wahrscheinlichkeit erzeugt kann dies ausgenutzt werden, um einen Schlüssel durch statistische Analysen abzuleiten bzw. herzuleiten.

Umsetzungsbeispiel: Verwendung der Random-Funktion unter Linux (/dev/urandom) zur Seed-Generierung für einen Pseudozufallsgenerator.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-12/8.0

Req 13	Für die Erzeugung von Schlüsselmaterial für Anwendungen mit hohen Sicherheitsanforderungen muss ein Hardware Security Module (HSM) oder ein Secure Element mit physischem Zufallszahlengenerator verwendet werden.
--------	--

Nutzerrolle: Entwicklung

Das erzeugte Schlüsselmaterial darf die Sicherheitshardware nicht ungeschützt verlassen.

Motivation: Manipulationssichere ("tamper-proof") HSMs oder Smart Cards dienen als sicherer Speicher für erzeugtes Schlüsselmaterial und sie verfügen über starke kryptographische Primitive. Zudem bieten sie zusätzlichen Schutz vor Kompromittierung geheimer Schlüssel.

Umsetzungsbeispiel: - Erzeugung von zufälligen Subscriber Keys im Rahmen der SIM-Herstellungsprozesses in einem HSM durch einen physischen Zufallszahlengenerator.
- Utimaco Enterprise Secure Key Manager

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-13/8.0

6.2. Schlüsselvereinbarung und -austausch

Req 14 Verfahren zur Schlüsselvereinbarung müssen Perfect Forward Secrecy (PFS) bieten und durch Authentisierung gegen Man-in-the-Middle Angriffe geschützt werden.

Nutzerrolle: Entwicklung

Asymmetrische Schlüsselvereinbarung wird in der Regel auf Basis des (Elliptic Curve) Diffie-Hellman Schlüsselaustausches durchgeführt (z.B. Ephemeral EC-DH oder auch MQV). Der Vorteil dieses Schlüsselaustausches ist, dass der komplette Schlüssel nie über den Kommunikationskanal transportiert wird. Zudem bietet der vergängliche (ephemeral) DH-Schlüsselaustausch, wenn die verwendeten Secrets nach Gebrauch gelöscht werden, Perfect Forward Secrecy bedeutet, dass selbst bei Angriffen, bei denen Zugriff auf ein Langzeitgeheimnis erlangt wird (z.B. privaten RSA-Schlüssel), die Vertraulichkeit der vorherigen Kommunikation nicht gebrochen werden kann. Verschiedene asymmetrische Schlüsselaustauschprotokolle sind in [ISO-11770-3] standardisiert. Die Eigenschaften müssen bei Implementierungen und Auswahl von Verfahren berücksichtigt werden (z.B. Vergänglichkeit und Authentifizierung). Um Implementierungsfehler zu vermeiden, müssen bei Eigenimplementierungen zudem die Empfehlungen der [BSI TR 02102-1] eingehalten werden.

Hinweis zu standardisierten DH-Gruppen:

Da die Erzeugung der Systemparameter – sogenannte Diffie-Hellman Gruppen – komplex, fehleranfällig und aufwendig ist, werden häufig standardisierte Systemparameter verwendet [RFC5114].

Quellen:

[ISO-11770-3] ISO/IEC, Key management Part 3: Mechanisms using asymmetric techniques

[RFC-5114] Additional Diffie-Hellman Groups for Use with IETF Standards

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Die Schlüsselvereinbarung löst das Schlüsselaustauschproblem der symmetrischen Verschlüsselung. Mit Perfect Forward Secrecy kann bei einem Angriff, bei dem Langzeitschlüsseln (z.B. privater RSA-Schlüssel) kompromittiert werden, nicht die Vertraulichkeit bereits übertragener Ciphertexte gebrochen werden. Authentisierung ist notwendig, um Man-in-the-Middle Angriffe abzuwehren.

Umsetzungsbeispiel: Ein Beispiel für eine sichere Schlüsselvereinbarung ist ein Diffie-Hellman Schlüsselaustausch im TLS-Protokoll. Der Server authentisiert sich mit einem TLS-Zertifikat, das von einer vertrauenswürdigen CA ausgestellt wurde. Die Client Authentisierung kann ebenfalls durch ein Zertifikat stattfinden oder durch eine nachgelagerte Passworteingabe über den gesicherten Kanal. Ab TLS 1.3 werden nur noch Schlüsselaustauschverfahren mit Perfect Forward Secrecy angeboten.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten

- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-14/8.0

6.3. Asymmetrische Key Encapsulation Mechanismen (KEM)

Req 15 Falls kein Schlüsselaustauschverfahren mit Perfect Forward Secrecy (PFS) verwendet werden kann, muss für die asymmetrische Verschlüsselung von symmetrischen Schlüsseln einer der folgenden Key Encapsulation Mechanismen mit den freigegebenen Parametern verwendet werden: ECIES-KEM, PSEC-KEM, RSAES-OAEP, DLIES-KEM oder RSA-KEM.

Nutzerrollen: Entwicklung, Integration

Key Encapsulation Mechanismen (KEM) werden verwendet, um einen symmetrischen Schlüssel über einen unsicheren Kanal mittels asymmetrischer Kryptographie zu übertragen. Sie bieten bei Wiederverwendung von Langzeitschlüsseln keine Perfect Forward Secrecy. Die tatsächliche Verschlüsselung der Nutzdaten erfolgt anschließend über einen symmetrischen Verschlüsselungsalgorithmus (beispielsweise AES-GCM). Es wird nicht empfohlen, große Mengen an Benutzerdaten mit asymmetrischen Algorithmen zu verschlüsseln.

ECIES (Elliptic Curve Integrated Encryption Scheme) und DLIES (Discrete Logarithm Integrated Encryption Scheme) sind hybride Verschlüsselungsverfahren, die eine asymmetrische KEM, z.B. ECIES-KEM, mit einem symmetrischen Verschlüsselungsverfahren bzw. Authenticated Encryption, z.B. AES-GCM, kombinieren.

Folgende Tabelle enthält die zulässigen asymmetrischen KEMs sowie die Mindestlängen der entsprechenden Sicherheitsparameter und Empfehlungen zu weiteren Systemparametern für hybride Verschlüsselung:

Asymmetrischer Verschlüsselungsalgorithmus	Mindestlänge der Sicherheitsparameter	Symmetrische Verschlüsselung	Weitere benötigte Funktionen	Referenzspezifikationen
ECIES-KEM (auf elliptischen Kurven)	250 Bit (die Ordnung des Basispunktes der elliptischen Kurve muss mindestens 250 betragen)	AES-GCM	KDF: HMAC based KDF	[ISO-18033-2]
PSEC-KEM (auf elliptischen Kurven)	$\text{ord}(g) \geq 250$ Bit (die Ordnung des Basispunktes der elliptischen Kurve muss mindestens 250 betragen)	AES-GCM	KDF: HMAC based KDF	[ISO-18033-2]
RSAES-OAEP	3000 Bit RSA Modulus (n) mit einem öffentlichen Schlüssel $e > 2^{16}$	AES-GCM	Hash: SHA3, SHA2 KDF: HMAC based KDF	[RFC-8017]
DLIES-KEM (auf $\text{GF}(p)$)	3000 Bit Primzahl p (definiert den endlichen Körper), 250 Bit Primzahl q	AES-GCM	KDF: HMAC based KDF	[ISO-18033-2]

RSA-KEM	3000 Bit RSA Modulus (n) mit einem öffentlichen Schlüssel $e > 2^{16}$	AES-GCM	KDF: HMAC based KDF	[ISO-18033-2]
---------	--	---------	---------------------	---------------

Für weitere Informationen zu den Sicherheitsparametern asymmetrischer Verschlüsselungsverfahren wird auf [TR-02102-1] verwiesen. Die Key Encapsulation Mechanismen (KEM) ECIES-KEM, RSAES-OAEP, RSA-KEM, PSEC-KEM, sind in [ISO-18033-2] standardisiert.

Hinweis zum verwendeten symmetrischen Verschlüsselungsverfahren:

Es wird empfohlen Authenticated Encryption zu verwenden (z.B. AES-GCM, AES-CCM). Falls keine AEAD-Betriebsarten verwendet werden können, muss das Verschlüsselungsverfahren durch einen Message Authentication Code (MAC) in der „Encrypt-then-MAC“-Konstruktion ergänzt werden.

Hinweis zu RSA-PKCS#1 v1.5 Verschlüsselung:

RSA nach PKCS#1 v1.5 gilt nicht mehr als sicher und darf zur Verschlüsselung nicht verwendet werden.

Folgende Kurven dürfen eingesetzt werden und erreichen nach heutigem Stand der Forschung das in der obigen Tabelle geforderte Sicherheitsniveau:

- brainpoolP512r1
- NIST P-521
- brainpoolP384r1
- NIST P-384
- brainpoolP320r1
- brainpoolP256r1
- NIST P-256

Die Brainpool-Kurven sind in [RFC-5639] und die NIST-Kurven in [FIPS-186-5] spezifiziert.

Abkürzungen:

ECIES = Elliptic Curve Integrated Encryption Scheme

KEM = Key Encapsulation Mechanism

OAEP = Optimal Asymmetric Encryption Padding

PSEC = Provable Secure Elliptic Curve (encryption)

RSA = Rivest-Shamir-Adleman

DLIES = Discrete Logarithm Integrated Encryption Scheme

Quellen:

[FIPS-186-5] NIST FIPS 186-5 Digital Signature Standard (DSS)

[ISO-18033-2] ISO/IEC, Encryption algorithms Part 2: Asymmetric ciphers

[RFC-5639] Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation

[RFC-8017] PKCS #1: RSA Cryptography Specifications Version 2.2

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Diese Verfahren bieten einen angemessenen Schutz und sind von anerkannten Institutionen empfohlen. Hybride Verschlüsselung löst das Schlüsselaustauschproblem der symmetrischen Kryptographie und wird auch bei der Emailverschlüsselung eingesetzt.

Umsetzungsbeispiel: Verschlüsselung des AES-Schlüssels mittels RSA Public Key des Empfängers: RSA-KEM_PubKey_B (AES_Key).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-15/8.0

6.4. Schlüsselableitung

Req 16	Zur Ableitung von Sitzungsschlüsseln aus einem Geheimnis müssen standardisierte Schlüsselableitungsfunktionen (Key Derivation Functions) verwendet werden, die auf folgenden Primitiven basieren: HMAC, CMAC, AES, SHA-3, SHA-2.
--------	--

Nutzerrolle: Entwicklung

Bei der Nutzung eines gemeinsamen Geheimnisses (z.B. nach einem DH-Schlüsselaustausch) werden symmetrische "Unter-Schlüssel" benötigt (z.B. für Sitzungsschlüssel, Nachrichten Authentisierung oder Verschlüsselung). Schlüsselableitungsfunktionen (Key Derivation Functions = KDF) dienen dazu, aus einem einzigen Geheimnis (z. B. Master Key) mehrere abgeleitete Schlüssel zu erzeugen, die für dedizierte Anwendungsfälle nutzbar sind.

Generell wird zwischen Extraction-then-Expansion Schlüsselableitungsfunktionen (two-step KDFs nach [ISO-11770-6] bzw. [NIST-SP-800-56C]) basierend auf HMAC oder CMAC und Schlüsselableitungsfunktionen auf Basis von pseudo-zufälligen Funktionen (one-step KDFs nach [NIST-SP-800-56C], [NIST-SP-800-108] bzw. [ISO-11770-6]) unterschieden. Der Unterschied liegt dabei im Eingangs-Schlüsselmateri al. one-step KDFs benötigen ein zufälliges Geheimnis als Eingabe wohingegen two-step KDFs im ersten Schritt die Entropie des Eingangsmaterial mit einem Salt anreichern. Grundsätzlich wird empfohlen Extraction-then-Expansion KDFs zu verwenden.

Anwendungsbereich von Schlüsselableitungsfunktionen:

- Bindung von Protokolldaten an einen Schlüssel (zum Beispiel Sendername, Empfängername) durch das sogenannte „Label“
- Ableitung von symmetrischen Sitzungsschlüsseln oder von symmetrischen Schlüsseln für dedizierte Aufgaben (MAC-Berechnung, Verschlüsselung)
- Anreicherung von Geheimnissen mit Entropie zur Verwendung als symmetrischer Schlüssel (Extraction-then-Expansion)

Quellen:

[ISO-11770-6] ISO/IEC, Key management Part 6: Key derivation

[NIST-SP-800-56C] NIST, Recommendation for Key-Derivation Methods in Key-Establishment Schemes

[NIST-SP-800-108] NIST, Recommendation for Key Derivation Using Pseudorandom Functions

Motivation: Untersuchte und von vertrauenswürdigen Institutionen empfohlene Schlüsselableitungsfunktionen bieten einen ausreichenden Schutz und sichern eine vergleichbare kryptographische Stärke der abgeleiteten Schlüssel.

Umsetzungsbeispiel: Erzeugung eines HMAC-Schlüssels und eines AES-Schlüssels nach einem Diffie-Hellman Schlüsselaustausch mittels HMAC-SHA-256 basierter Extraction-then-Expansion KDF.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-16/8.0

Req 17	Falls ein kryptographischer Schlüssel aus einem Passwort abgeleitet wird, dürfen nur Argon2id oder scrypt verwendet werden. Wenn Argon2id und scrypt nicht verfügbar sind, darf PBKDF2 verwendet werden.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Folgende Parameter sind für die Schlüsselableitung zu verwenden:

Verfahren	Salt-Länge	Sicherheitsparameter / Work Faktor	Referenz-Spezifikationen
Argon2id	128 Bit	Mindestspeichergröße (m), Mindestanzahl an Iterationen (t), Parallelisierungsfaktor (p) m=47.104 (46 MiB), t=1, p=1 m=19.456 (19 MiB), t=2, p=1 m=12.288 (12 MiB), t=3, p=1 m=9.216 (9 MiB), t=4, p=1 m=7.168 (7 MiB), t=5, p=1	[RFC-9106]
scrypt	128 Bit	CPU/Memory Kostenfaktor (N); Blockgröße (r); Parallelisierungsfaktor (p) N=2 ¹⁷ ; r=8; p=1 N=2 ¹⁶ ; r=8; p=2 N=2 ¹⁵ ; r=8; p=3 N=2 ¹⁴ ; r=8; p=5 N=2 ¹³ ; r=8; p=10	[RFC7914]
PBKDF2-HMAC-SHA256/512	128 Bit	PBKDF2-HMAC-SHA256: 600,000 Iterationen PBKDF2-HMAC-SHA512: 210,000 Iterationen	[RFC8018]

PBKDF2 ist als kryptographisch schwächer einzustufen.

Hinweis zu Sicherheitsparameter Argon2id und scrypt

Die Parameter entsprechen der OWASP-Empfehlung [OWASP]. Da die Parameterwahl von Faktoren wie der Leistungsfähigkeit der Hardware abhängt, kann die Anpassung des Parameters bei bestimmten Anwendungsfällen notwendig sein.

Hinweis zu Sicherheitsparameter PBKDF2

Der Parameter $n \geq 600.000/210.000$ entspricht der OWASP-Empfehlung [OWASP]. Da die Parameterwahl von Faktoren wie der Leistungsfähigkeit der Hardware abhängt, kann die Anpassung des Parameters bei bestimmten Anwendungsfällen notwendig sein.

Hinweis zur Passwort-Bildung:

Die aktuellen Anforderungen an die Passwort-Bildung findet man im Dokument [TBS].

Quellen:

[OWASP] https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

[RFC-7914] The scrypt Password-Based Key Derivation Function

[RFC-8018] PKCS #5: Password-Based Cryptography Specification

[RFC-9106] Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications

[TBS] Deutsche Telekom AG: Technischer Basisschutz von IT-/NT-Systemen

Motivation: Spezialisierte Passwortbasierte-KDF sind dafür entworfen, Schlüssel mit einer ausreichenden Entropie aus

Passwörtern sicher abzuleiten.

Umsetzungsbeispiel: Ableitung eines Sitzungsschlüssels und MAC-Schlüssels aus einem Passwort durch den Argon2 Algorithmus.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-17/8.0

6.5. Transport und Speicherung von Schlüsseln

Req 18	Anwendungen mit hohen Sicherheitsanforderungen müssen ein Hardware Security Module (HSM) oder ein Secure Element zur Speicherung von Schlüsseln verwenden. Geheime Schlüssel und vertrauliche Parameter dürfen die Hardware nicht ungeschützt verlassen.
--------	--

Nutzerrolle: Entwicklung

Die Verwendung von manipulationssicheren ("tamper-proof") HSMs garantiert, dass das erzeugte Schlüsselmaterial den höchstmöglichen Schutz erhält. Zusätzlich schützt ein Hardware Security Modul bzw. Secure Element vor Seitenkanalangriffen.

Motivation: Kryptographische Schlüssel müssen vor unbefugtem Zugriff geschützt werden. Entsprechende Sicherheitshardware bietet sehr hohen Schutz gegenüber Angriffen und Seitenkanalanalysen.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-18/8.0

Req 19	Symmetrische Schlüssel müssen gegen unberechtigten Zugriff geschützt werden und dürfen außerhalb der Systemgrenzen nicht in ungeschützter Form übertragen, sowie gespeichert werden.
--------	--

Nutzerrolle: Entwicklung, Betrieb

Der Datenzugriff auf Schlüsselmaterial muss auf die notwendigen Dienste und Nutzergruppen innerhalb und außerhalb der Systemgrenzen eingeschränkt werden.

Für den Transport und Speicherung von symmetrischem Schlüsselmaterial (z.B. Pre-Shared Keys) außerhalb der Systemgrenzen müssen AES-Synthetic Initialization Vector (AES-SIV) oder AES-Key-Wrap verwendet werden. Diese Anforderung ist relevant, wenn kryptographische Schlüssel über einen unsicheren Kanal übertragen werden müssen oder auf einem unsicheren Datenträger gespeichert werden. Diese werden allgemein als "Key-Wrap" Verfahren bezeichnet und sind in [RFC-5297] bzw. [NIST-SP-800-38F] standardisiert. Die Anforderungen an Sicherheitsparameter dieser Verfahren entsprechen den Anforderungen an AES in diesem Dokument.

Hinweis zu sicherer Hardware:

Schlüssel können auch auf sicherer Hardware wie Hardware-Sicherheits-Modulen (HSMs) oder Secure Elements (wie z.B. Smart Cards) gespeichert und transportiert werden. Schlüssel werden dann von der sicheren Hardware und ggf. weiteren kryptographischen Mechanismen (z.B. Key Encryption Keys und Keywrap) geschützt.

Quellen:

[NIST-SP-800-38F] NIST, Recommendation for Block Cipher Modes of Operation: Methods for Key Wrapping
[RFC-5297] Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)

Motivation: Die Sicherheit von kryptographischen Verfahren folgt Kerckhoffs' Prinzip, was bedeutet, dass lediglich der Schlüssel geheim gehalten werden muss und das Verfahren öffentlich bekannt sein soll. Deshalb müssen kryptographische Schlüssel bei der Speicherung und beim Transport gesondert geschützt werden, indem sie mit anderen Schlüsseln verschlüsselt werden.

Umsetzungsbeispiel: Key-Wrap eines AES-128 Schlüssels (Schlüssellänge = 128 Bit) mittels AES-SIV.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-19/8.0

Req 20	Private Schlüssel von asymmetrischen Verfahren müssen gegen unberechtigten Zugriff geschützt werden und dürfen außerhalb der Systemgrenzen nicht in ungeschützter Form übertragen oder gespeichert werden.
--------	--

Nutzerrollen: Betrieb, Integration

Der Datenzugriff auf Schlüsselmaterial muss auf die notwendigen Dienste und Nutzergruppen innerhalb und außerhalb der Systemgrenzen eingeschränkt werden.

Private Schlüssel dienen der Signatur von Nachrichten bzw. zur Entschlüsselung von verschlüsselten Nachrichten. Eine digitale Signatur (persönliche Unterschrift) ist nur mittels eines privaten Schlüssels möglich und gewährleistet die Authentizität. Private Schlüssel dürfen nicht auf zentralen Speichersystemen gespeichert werden oder verteilt werden. Zusätzlich müssen private Schlüssel von natürlichen Personen durch ein persönliches Geheimnis (PIN, Passwort) gegen unbefugte Nutzung geschützt werden.

Hinweis zu Fernsignaturen:

Fernsignaturen dürfen nur durch zertifizierte Vertrauensdienstleister (Trust Center wie www.telesec.de) erstellt werden. Das Zertifikat muss die Erfüllung der eIDAS Standards (electronic IDentification, Authentication and trust Services) ausweisen und muss regelmäßig überprüft, verlängert bzw. erneuert werden.

Motivation: Wenn bei einem Angriff der private Schlüssel kompromittiert wird, kann die entsprechende digitale Identität übernommen werden und sich mit dieser Identität authentifiziert bzw. Nachrichten entschlüsselt werden.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-20/8.0

Req 21	Kryptographische Langzeitschlüssel müssen eine der Anwendung angemessene begrenzte Gültigkeit besitzen.
--------	---

Nutzerrolle: Betrieb, Entwicklung, Integration

Generell wird zwischen Langzeitschlüsseln und kurzlebigem Schlüsselmaterial wie Session Keys unterschieden. Langzeitschlüssel werden über einen längeren Zeitraum für kryptographische Operationen verwendet (z.B. TLS-Serverzertifikate oder SSH-Schlüssel), wohingegen Session Keys nur für eine Protokollsitzung verwendet und danach gelöscht werden. Der Begriff Langzeitschlüssel wird nicht über eine konkrete Zeitspanne definiert, sondern über die mehrfache Verwendung desselben Schlüsselmaterials. Die Begrenzung der Gültigkeit kryptographischer Langzeitschlüssel beschränkt die Möglichkeiten der Kryptoanalyse, die Auswirkungen im Fall einer Kompromittierung des privaten Schlüssels sowie das Risiko, dass der zugrundeliegende Algorithmus oder die Schlüssellänge während des Gültigkeitszeitraums des Schlüssels als unsicher klassifiziert werden.

Die Gültigkeit von Langzeitschlüsseln muss organisatorisch festgelegt werden. Bei asymmetrischen Langzeitschlüsseln kann ggf. der Gültigkeitszeitraum über das zugehörige digitale Zertifikat definiert werden. Dabei ist zu beachten, dass bei einer Zertifikatserneuerung nicht zwingend der Schlüssel erneuert werden muss (hierbei ist die Policy der ausstellenden Certificate Authority zu beachten). Grundsätzlich sollte die gesamte Gültigkeit von Schlüsseln (ggf. verteilt auf mehrere Zertifikate), die für Endanwendungen und damit häufig verwendet werden, zwischen einem und drei Jahren liegen.

In einzelnen, kontrollierbaren Fällen mit geringer Verwendung wie beispielsweise dem Schlüssel einer Certification Authority dürfen längere Gültigkeiten gewählt werden. Im Notfall (z.B. einer Kompromittierung) ist der entsprechende Schlüssel vorzeitig zu sperren.

Die folgenden Gültigkeitszeiträume werden empfohlen:

- Endentitäten (z.B. Anwender, TLS-Server): 1-3 Jahre
- Zwischenzertifizierungsstellen (Subordinate/ Intermediate CA): 8-15 Jahre
- Wurzelzertifizierungsstellen (Root CA): 20-30 Jahre

Es kann erforderlich sein, dass Schlüssel auch über ihre Gültigkeit hinaus verwendet werden müssen. Dies kann für öffentliche Schlüssel zur Prüfung von Signaturen in Langzeitanwendungen oder für private Schlüssel zur Entschlüsselung von Langzeitdaten wie beispielsweise E-Mails der Fall sein.

Motivation: Durch beschränkte Gültigkeitszeiträume von asymmetrischen Langzeitschlüsseln kann die Sicherheit der verwendeten Verfahren/Schlüssellängen in regelmäßigen Abständen angepasst werden.

Umsetzungsbeispiel: TLS-Webserverzertifikat mit einer Laufzeit von einem bis drei Jahren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-21/8.0

Req 22	Die Sicherheit von kryptographischen Langzeitschlüsseln muss in regelmäßigen Abständen überprüft und ggf. angepasst werden.
--------	---

Nutzerrolle: Betrieb, Entwicklung, Integration

Auf Grund von steigender Rechnerleistung und neuen Angriffen müssen die Schlüssellängen der eingesetzten kryptographischen Verfahren von Zeit zu Zeit angepasst werden. Im Falle besonderer Entwicklungen der Kryptoanalyse (z.B. durch universelle Quantencomputer) kann sogar eine Anpassung des kryptographischen Verfahrens notwendig sein. Dies betrifft besonders kryptographische Langzeitschlüssel wie Schlüsselpaare, für die ein digitales Zertifikat ausgestellt wurde oder SSH-Schlüssel. Spätestens bei der Erneuerung der Zertifikate muss überprüft werden, ob das eingesetzte Verfahren und die Schlüssellänge noch als sicher gelten.

Quellen:

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Fortschritte im Computing machen es notwendig Schlüssellängen anzupassen.

Umsetzungsbeispiel: Überprüfung, ob die Schlüssellänge von Langzeitschlüsseln weiterhin ausreichend ist, wenn ein Zertifikat erneuert wird.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-22/8.0

Req 23 Kryptographisches Schlüsselmaterial (symmetrisch und asymmetrisch) muss erneuerbar sein.

Nutzerrolle: Betrieb, Entwicklung

Aktive Angriffe und Fortschritte in der Computertechnologie machen es notwendig kryptographisches Schlüsselmaterial auszutauschen, z.B. wenn zum Schutz vor zurzeit theoretischen Angriffen mit Quantencomputern von AES-128 auf AES-256 gewechselt wird.

Zudem hängt die Sicherheit eines kryptographischen Verfahrens von der Geheimhaltung des Schlüssels ab, weshalb feste Schlüssel bei Kompromittierung ein großes Risiko darstellen.

Die Schlüsselerneuerung (Rekeying) ist eine notwendige Maßnahme, die in modernen Kryptoverfahren umgesetzt wird. Weitere Informationen können in [ISO-JTC-1] gefunden werden.

Erneuerte Schlüssel dürfen nicht von aktiv verwendetem Schlüsselmaterial (z.B. Verschlüsselungsschlüssel) abgeleitet werden.

Quelle:

[ISO-JTC-1] ISO/IEC JTC 1/SC 27 Information security, cybersecurity and privacy protection

Motivation: Die Sicherheit von kryptographischen Verfahren hängt von der Geheimhaltung und Güte des Schlüsselmaterials ab.

Umsetzungsbeispiel: Definition der Lebensdauer eines IPSec Session Keys in der IPSec Konfiguration eines IP-Sec-Gateways.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-23/8.0

6.6. Schlüsselsperrung und Vernichtung

Req 24 Kompromittierte Schlüssel oder Zertifikate müssen deaktivierbar bzw. sperrbar sein.

Nutzerrollen: Betrieb, Entwicklung, Integration

Unmittelbar nach Erkennen einer möglichen Kompromittierung von privaten Schlüsseln ist das betroffene Zertifikat zu sperren und es darf nicht mehr möglich sein, sich mit dem privaten Schlüssel zu authentisieren. Sperrprozesse für digitale Zertifikate müssen in andere Regelprozesse integriert werden (z.B. bei Verlust eines Endgerätes). Wenn wie bei SSH keine Zertifikate verwendet werden, muss dem betroffenen Schlüsselmateriale das Vertrauen entzogen werden.

Motivation: Sollte bei einem Angriff der private Schlüssel kompromittiert werden (z. B. "privater RSA-Schlüssel"), kann die entsprechende digitale Identität übernommen und damit beispielsweise Anwendungen oder Funktionen unberechtigt genutzt werden.

Umsetzungsbeispiel: Automatisierter Prozess zum Sperren/widerrufen eines kompromittierten Zertifikats mittels Certificate Revocation List (CRL) als Bestandteil des Certificate Management Protokolls (CMP).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.50-24/8.0

Req 25	Nicht mehr benötigtes Schlüsselmateriale muss mit sicheren Verfahren unmittelbar und unwiderruflich vernichtet oder unbrauchbar gemacht werden.
--------	---

Nutzerrolle: Betrieb

Diese Anforderung gilt für Schlüsselmateriale, welches auf persistenten Speichermedien gespeichert ist, auf welche die Anwendung direkt zugreifen kann.

Folgende elektronische Lösungsverfahren sind empfohlen:

- DoD-5220.22-M (ECE)
- Bruce-Schneier-Algorithmus
- Peter-Gutmann-Algorithmus
- ATA-"Secure-Erase" (bei SSD oder SSHD-Festplatten) - alle Bits müssen im Zustand 1 sein

Hinweis zu virtuellen Umgebungen:

In virtuellen bzw. Cloud Umgebungen ist das Überschreiben von Schlüsselmateriale durch die Anwendung schwer umsetzbar, wobei durch die dynamische Speicherverwaltung das Risiko auf Kompromittierung auch gering ist.

Motivation: Sollten bei einem Angriff nicht mehr verwendete Geheimnisse kompromittiert werden (z. B. "alter AES-Key"), können aufgezeichnete Nachrichten mit dem (damals gültigen) "alten Schlüssel" entschlüsselt werden.

Umsetzungsbeispiel: - Überschreiben des Key Storage mittels DoD-5220.22-M (ECE) Algorithmus- Zurückziehen eines Webserver Zertifikats

- Automatisierter Prozess zum Sperren/widerrufen ungenutzter Zertifikate mittels Certificate Revocation List (CRL) als Bestandteil des Certificate Management Protokolls (CMP).

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-25/8.0

7. Signaturalgorithmen

Req 26 Folgende Signatur-Algorithmen müssen mit den freigegebenen Parametern verwendet werden: ECDSA, EdDSA, RSA-PSS, DSA/DSS.

Nutzerrollen: Entwicklung, Integration

Signaturalgorithmen werden verwendet, damit der Empfänger (einer signierten Nachricht) den Ersteller und Nachrichtintegrität eindeutig verifizieren kann.

Die Verifikation des signierten Pakets erfolgt mithilfe des öffentlichen Schlüssels. Signaturalgorithmen mit Appendix (bei denen die Signatur der Nachricht angehängt wird) benötigen immer eine kryptographische Hash-Funktion.

Folgende Tabelle stellt die zulässigen Parameter für digitale Signaturen mit Appendix dar:

Signatur-Algorithmus	Mindestlänge der Sicherheitsparameter	Zulässige Hash-Funktionen	Referenz-Spezifikationen
ECDSA	250 Bit (die Ordnung des Basispunktes der elliptischen Kurve muss mindestens 250 betragen)	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit	[ISO-14888-3] [FIPS-186-5]
EdDSA	250 Bit	SHAKE256 für Ed448, SHA-512 für Ed25519	[RFC-8032] [FIPS-186-5]
RSA-PSS	3000 Bit RSA Modulus (n) mit einem öffentlichen Schlüssel $e > 2^{16}$	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit	[ISO-14888-2] [FIPS-186-5]
DSA / DSS	3000 Bit Primzahl p (definiert den endlichen Körper); 250 Bit Primzahl q	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit	[ISO-14888-3] [FIPS-186-5]

Für weitere Informationen zu den Sicherheitsparametern asymmetrischer Signaturverfahren wird auf [TR-02102-1] verwiesen.

Hinweis zu EdDSA:

EdDSA wird nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass EdDSA als sicher einzustufen ist.

Hinweis zu DSA / DSS:

[FIPS-186-5] erlaubt diesen Algorithmus nur noch zur Prüfung von alten Signaturen.

Einschränkungen zu SHA-224/SHA-3-224:

SHA-224/SHA-3-224 darf nur noch in Bestandssystemen verwendet werden und muss bei nächster Gelegenheit gegen einen Hashalgorithmus mit einer Output-Länge von mindestens 256 Bit ausgetauscht werden.

Einschränkungen zu RSA PKCS#1 v1.5:

RSA PKCS#1 v1.5 ist nur noch bei Bestandssystemen zulässig und sollte bei Systemänderungen gegen stärkere digitale Signaturverfahren ersetzt werden.

Freigegebene elliptische Kurven:

- brainpoolP512r1
- NIST P-521
- brainpoolP384r1

- NIST P-384
- brainpoolP320r1
- brainpoolP256r1
- NIST P-256
- Curve25519/Ed25519
- Curve448/Ed448

Abkürzungen:

DSA = Digital Signature Algorithm

DSS = Digital Signature Standard

EC = Elliptic Curve

PKCS = Public Key Cryptography Standards

PSS = Probabilistic Signature Scheme

Quellen:

[FIPS-186-5] NIST FIPS 186-5 Digital Signature Standard (DSS)

[ISO-14888-2] ISO/IEC 14888-2:2008 Information technology — Security techniques — Digital signatures with appendix - Part 2: Integer factorization based mechanisms

[ISO-14888-3] ISO/IEC 14888-3:2018 IT Security techniques — Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms

[RFC-8032] Edwards-Curve Digital Signature Algorithm (EdDSA)

[TR-02102-1] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen, Version 2025-01

Motivation: Eine sichere digitale Signatur stellt die Grundlage für eine eindeutige (digitale) Identität dar.

Umsetzungsbeispiel: x509 Zertifikat mit Signaturalgorithmus sha - 256 with RSA

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.50-26/8.0

8. Verwendung von kryptographischen Verfahren in Sicherheitshardware

Req 27 Anwendungen mit hohen Sicherheitsanforderungen müssen ein Hardware Security Module (HSM) oder ein Secure Element für kryptographische Verfahren verwenden.

Nutzerrolle: Entwicklung

Sicherheitshardware ist speziell für kryptographische Operationen konstruiert und enthält neben physischen Schutzvorrichtungen auch zusätzlichen Schutz gegen Seitenkanalangriffe. Neben der Schlüsselerzeugung und Verwaltung müssen Anwendungen mit besonders hohem Schutzbedarf HSMs oder Secure Elements für die Ausführung von kryptographischen Operationen verwenden.

Motivation: Sicherheitshardware (HSM) ermöglicht die performante Berechnung von kryptographischen Verfahren und stellt sicher, dass die kryptographischen Schlüssel nicht kompromittiert werden können. Außerdem sind die Implementierungen gegen Seitenkanalangriffe geschützt.

Umsetzungsbeispiel: Digitale Signatur von Zertifikaten durch eine Zertifizierungsstelle (CA) deren Schlüsselmaterial auf einem HSM gespeichert ist.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-27/8.0

9. Kryptographische Verfahren zur sicheren Speicherung und Verifikation von Passwörtern

Req 28 Nur folgende kryptographischen Verfahren zur Erzeugung von Passwort-Hashes und Verifikation von Passwörtern dürfen mit den freigegebenen Parametern verwendet werden: scrypt, bcrypt oder Argon2id. Wenn scrypt, bcrypt und Argon2id nicht verfügbar sind, dürfen auch SHA-512-crypt, SHA-256-crypt, oder PBKDF2 verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Passwörter zur Authentifizierung dürfen nur als Passwort-Hash gespeichert werden.

Sichere Verfahren zur kryptographischen Passwort-Hash-Erzeugung und Passwort-Verifikation:

Verfahren	Salt-Länge	Sicherheitsparameter / Work-Faktor	Zusätzliche Parameter	Referenzspezifikationen
scrypt	128 Bit	CPU/Memory Kostenfaktor (N); Blockgröße (r); Parallelisierungsfaktor (p) $N=2^{17}; r=8; p=1$ $N=2^{16}; r=8; p=2$ $N=2^{15}; r=8; p=3$ $N=2^{14}; r=8; p=5$ $N=2^{13}; r=8; p=10$	keine	[RFC-7914]
bcrypt (nur zur Passwortspeicherung)	128 Bit	Iterationen (input cost (n)): 2^n mit $n \geq 13$	keine	[Bcrypt]
Argon2id	128 Bit	Mindestspeichergröße (m), Mindestanzahl an Iterationen (t), Parallelisierungsfaktor (p) $m=47.104$ (46 MiB), $t=1, p=1$ $m=19.456$ (19 MiB), $t=2, p=1$ $m=12.288$ (12 MiB), $t=3, p=1$ $m=9.216$ (9 MiB), $t=4, p=1$ $m=7.168$ (7 MiB), $t=5, p=1$	keine	[RFC-9106]
SHA-512-crypt	96 Bit (16 Zeichen)	Iterationen (n)=640.000	id=6	[Crypt3]

SHA-256-crypt	96 Bit (16 Zeichen)	Iterationen (n)=640.000	id=5	[Crypt3]
PBKDF2-HMAC-SHA256/512	128 Bit	PBKDF2-HMAC-SHA256: 600,000 Iterationen PBKDF2-HMAC-SHA512: 210,000 Iterationen	keine	[RFC-8018]

SHA-512-crypt, SHA-256-crypt und PBKDF2 sind als kryptographisch schwächer einzustufen.

Hinweis zu den Parametern:

Die Parameter-Empfehlungen sind so gewählt, sodass ein Offline-Angriff einer Passwort-Hash Datei unattraktiv ist, indem die Passwort-Hash-Erzeugung verlangsamt wird.

Hinweis zu Sicherheitsparameter PBKDF2:

Der Parameter $n \geq 600.000/210.000$ entspricht der OWASP-Empfehlung [OWASP]. Da die Parameterwahl von Faktoren wie der Leistungsfähigkeit der Hardware abhängt, kann die Anpassung des Parameters bei bestimmten Anwendungsfällen notwendig sein.

Hinweis zu SHA-512/256-crypt:

Wichtig zu erwähnen ist, dass es nicht zulässig ist native Hashfunktionen für die Erzeugung von Passworthashes zu verwenden, da diese mit speziellen Angriffsmethoden (GPU, ASIC/FPGA) effizient gebrochen werden können. Daher ist SHA-512-crypt eine spezialisierte Implementierung basierend auf dem SHA-512 Algorithmus, um Passwort-Hashwerte zu erzeugen und zu verifizieren.

Hinweis für Schlüsselableitungen:

Empfehlungen für die Passwort-basierte Ableitung von Schlüsselmaterial findet man im Kapitel „Schlüsselverwaltung“.

Hinweis zu Passwort-Management-Lösungen

Freigegebene Passwort-Management-Lösungen wie T-PAM erfüllen diese Anforderung.

Hinweis zur Passwort-Bildung:

Die aktuellen Anforderungen an die Passwort-Bildung findet man im Dokument [TBS].

Quellen:

[Bcrypt] bcrypt - A Future-Adaptable Password Scheme, Provos, Mazieres, Usenix 1999

[Crypt3] Crypt(3) man page

[OWASP] https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html

[RFC-7914] The crypt Password-Based Key Derivation Function

[RFC-8018] PKCS #5: Password-Based Cryptography Specification

[RFC-9106] Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications

[TBS] Deutsche Telekom AG: Technischer Basisschutz von IT-/NT-Systemen

Motivation: Nur spezielle Passwort-Hash-Algorithmen sind resistent gegenüber Online- und Offline-Angriffen auf Passwort-Hash-Tabellen.

Umsetzungsbeispiel: Python:

```
>>> from passlib.hash import sha256_crypt
```

```
>>> # generate new salt, hash password
```

```
>>> hash = sha256_crypt.hash("password")
```

```
>>> hash
```

```
'$5$rounds=80000$wnsT7Yr92oJoP28r$cKhJlmk5mfuSKV9b3mumNzlbstFUpIKtQXXMo4G6Ep5'
```

```
>>> # same, but with explicit number of rounds
```

```
>>> sha256_crypt.using(rounds=640000).hash("password")
```

```
'$5$rounds=640000$q3hvJE5mn5jKRsw.$BbbYTFialmz9rTy03GGi.Jf9YY5bmxN0LU3p3ul1iUB'
```

```
>>> # verify password
>>> sha256_crypt.verify("password", hash)
True
>>> sha256_crypt.verify("letmein", hash)
False
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unbemerkt durchführbare Angriffe

ID: 3.50-28/8.0

10. Internet Protokoll Security (IPSec)

Werden Anforderungen nur für IP Security (IPSec) benötigt, sind nur die Anforderungen in Kapitel 10 relevant.

10.1. Allgemeine IPSec Parameter

Req 29	IPSec muss mit Internet Key Exchange (IKE) in der Version 2 genutzt werden.
--------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

IPSec (Internet Protocol Security) ist ein Protokoll zum Schutz der Vertraulichkeit, Integrität und Authentizität von Informationen, die über das IP-Protokoll übertragen werden. Es wird zwischen den folgenden IPSec-Protokollen unterschieden:

- Authentication Header (AH): Schutz der Integrität und Authentizität
- Encapsulated Security Payload (ESP): zusätzlicher Schutz der Vertraulichkeit

Ein essenzielles Konzept von IPSec sind die Security Associations (SA). Eine SA stellt eine gesicherte Verbindung zwischen zwei Kommunikationspartnern dar und umfasst die verwendeten kryptographischen Algorithmen und Parameter. Security Associations werden mithilfe des Protokoll IKEv2 (Internet Key Exchange) ausgehandelt. IPSec und IKEv2 sind in verschiedenen RFCs standardisiert, dazu gehören [RFC-4301], [RFC-4302], [RFC-4303], [RFC-7296], und [RFC-8247].

Quellen:

[RFC-4301] Security Architecture for the Internet Protocol

[RFC-4302] IP Authentication Header

[RFC-4303] IP Encapsulating Security Payload (ESP)

[RFC-7296] Internet Key Exchange Protocol Version 2 (IKEv2)

[RFC-8247] Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)

[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: IKEv2 besitzt stärkere Schlüsselableitungsfunktionen und besitzt Performancevorteile beim IPSec Tunnelaufbau gegenüber IKEv1. Darüber hinaus ist IKEv1 anfällig gegen Denial-of-Service (DoS) Angriffe, da der Status der Session Cookies gespeichert wird [TR-02102-3].

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-29/8.0

Req 30	Für IKEv2 und IPSec müssen die folgenden Verschlüsselungsverfahren verwendet werden: ENCR_AES_GCM_16, ENCR_AES_GCM_12, ENCR_CHACHA20_POLY1305, ENCR_AES_CCM_16, ENCR_AES_CCM_12, ENCR_AES_CTR oder ENCR_AES_CBC.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

In IKEv2 wird das Verschlüsselungsverfahren zur Verschlüsselung der IKE-Nachrichten zur Aushandlung der Eigenschaften sowie kryptographischen Algorithmen und Parameter für das IPSec-Protokoll verwendet. Im IPSec-Protokoll ESP wird das Verschlüsselungsverfahren zum Schutz der ESP-Pakete eingesetzt.

Die nachfolgende Tabelle listet die zulässigen Verschlüsselungsverfahren und die Referenzspezifikationen für die Verwendung in IKEv2 und im IPSec-Protokoll ESP auf:

Verschlüsselungsverfahren	Referenzspezifikationen für IKEv2	Referenzspezifikationen für IPSec (ESP)
ENCR_AES_GCM_16	[RFC-5282], [RFC-8247]	[RFC-4106], [RFC-8247]
ENCR_AES_GCM_12	[RFC-5282], [RFC-8247]	[RFC-4106], [RFC-8247]
ENCR_CHACHA20_POLY1305	[RFC-7634]	[RFC-7634]
ENCR_AES_CCM_16	[RFC-5282]	[RFC-4309]
ENCR_AES_CCM_12	[RFC-5282]	[RFC-4309]
ENCR_AES_CTR	[RFC-5930]	[RFC-3686]
ENCR_AES_CBC	[RFC-7296]	[RFC-3602]

Hinweise:

- Der Verschlüsselungsalgorithmus AES kann grundsätzlich mit drei Schlüssellängen (256, 192 oder 128 Bit) verwendet werden. Die Schlüssellänge kann über das Key-Length-Attribut ausgehandelt werden. Dabei ist zu berücksichtigen, welche Schlüssellängen die entsprechende Implementierung anbietet, wobei alle drei Schlüssellängen sicher sind und genutzt werden können.
- Die Verfahren ENCR_AES_CTR und ENCR_AES_CBC müssen mit einem Verfahren zum Schutz der Integrität kombiniert werden.
- Die Suffixe _16 und _12 bei den Verfahren ENCR_AES_GCM und ENCR_AES_CCM geben die Länge des Authentifizierungs-Tags (MAC) in Byte an.
- ENCR_CHACHA20_POLY1305 wird nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Quellen:

[RFC-3602] The AES-CBC Cipher Algorithm and Its Use with IPsec
[RFC-3686] Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)
[RFC-4106] The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
[RFC-4309] Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)
[RFC-5282] Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol
[RFC-5930] Using Advanced Encryption Standard Counter Mode (AES-CTR) with the Internet Key Exchange version 02 (IKEv2) Protocol
[RFC-7296] Internet Key Exchange Protocol Version 2 (IKEv2)
[RFC-7634] ChaCha20, Poly1305, and Their Use in the Internet Key Exchange Protocol (IKE) and IPsec
[RFC-8247] Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)
[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: Eine starke Verschlüsselung schützt die Datenübertragung vor unberechtigter Einsichtnahme.

Umsetzungsbeispiel: Cisco Konfigurationsbeispiel zur Einstellung von AES für ESP:

```
CE1_R1_4(config)#crypto ipsec transform-set test esp-aes
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-30/8.0

Req 31 Folgende Hash-Funktionen müssen bei IPSec verwendet werden: SHA-3-512, SHA-512, SHA-3-384, SHA-384, SHA-3-256, oder SHA-256.

Nutzerrollen: Betrieb, Entwicklung, Integration

Hash-Funktionen sind die Basis für Message Authentication Codes (MAC) zum Schutz der Integrität von IPSec-Nachrichten sowie für Pseudozufallsfunktionen zur Schlüsselableitung. Darüber hinaus werden sie für die Erstellung digitaler Signaturen im Rahmen der Authentifizierung verwendet. Im Allgemeinen wird der Hash-Algorithmus implizit durch die Konfiguration des MAC, des PRF oder des Authentifizierungsmechanismus angegeben. Da jedoch einige IPSec-Systeme die Konfiguration des Hashalgorithmus erfordern, müssen in diesen Fällen die in dieser Anforderung angegebenen Hashalgorithmen verwendet werden.

Hinweis zu SHA-1:

SHA-1 darf nicht für digitale Signaturen verwendet werden. Für Message Authentication Codes oder Pseudozufallsfunktionen dürfen sie nur in Legacy-Systemen verwendet werden und müssen bei nächster Gelegenheit durch einen Hash-Algorithmus mit einer 256-Bit-Ausgabe ersetzt werden.

Hinweis zu SHA-3-224/SHA-224:

SHA-3-224/SHA-224 darf nur in Legacy-Systemen verwendet werden und muss bei nächster Gelegenheit durch einen Hash-Algorithmus mit 256-Bit-Ausgabe ersetzt werden.

Quelle:

[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IPSec and IKE, Version 2025-01

Motivation: Hash-Funktionen werden für verschiedene kryptographische Mechanismen innerhalb von IPSec verwendet. Daher müssen sie ein hohes Sicherheitsniveau bieten.

Umsetzungsbeispiel: Cisco Konfiguration von SHA-256 als Hash-Funktion für Phase 1:

```
CE1_R1_4(config-isakmp)#hash sha256
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen

ID: 3.50-31/8.0

Req 32 Für IKEv2 und IPSec müssen die folgenden Message Authentication Codes (MAC) verwendet werden: AUTH_HMAC_SHA2_512_256, AUTH_HMAC_SHA2_384_192, AUTH_HMAC_SHA2_256_128 oder AUTH_AES_CMAC_96.

Nutzerrollen: Betrieb, Entwicklung, Integration

In IKEv2 werden Message Authentication Codes zum Integritätsschutz der IKE-Nachrichten eingesetzt. In IPSec werden sie zum Schutz der Integrität der ESP- oder AH-Pakete verwendet.

Die nachfolgende Tabelle listet die zulässigen MAC-Verfahren und die Referenzspezifikationen für die Verwendung in IKEv2 und in den IPSec-Protokollen AH und ESP auf. Ein „-“ in einem Tabellenfeld bedeutet, dass das jeweilige Verfahren nicht für das entsprechende Protokoll vorgesehen ist.

Zulässige MAC-Verfahren für IKEv2 und IPSec:

MAC-Verfahren	Referenzspezifikationen für IKEv2	Referenzspezifikationen für IPSec (AH, ESP)
AUTH HMAC SHA2 512 256	[RFC-4868]	[RFC-4868]
AUTH HMAC SHA2 384 192	[RFC-4868]	[RFC-4868]

AUTH HMAC SHA2 256 128	[RFC-4868]	[RFC-4868]
AUTH AES CMAC 96	-	[RFC-4494]

Hinweise:

- Die Suffixe der HMAC-Verfahren haben folgende Bedeutung:
 - Das erste Suffix (512, 384, 256) beschreibt die Output-Länge der Hashfunktion und damit des HMACs in Bit. Dies entspricht auch der Schlüssellänge.
 - Das zweite Suffix (256, 192, 128) beschreibt die Bit-Länge, auf welche die Ausgabe des HMAC, d. h. der Authentication Tag, gekürzt werden kann.

Zukünftig spezifizierte MAC-Verfahren, die den in diesem Dokument festgelegten Anforderungen an kryptographische Verfahren entsprechen, können ebenfalls verwendet werden. Dies gilt beispielsweise für MAC-Verfahren, die Hashfunktionen der SHA 3-Familie einsetzen.

Quellen:

[RFC-4494] The AES-CMAC-96 Algorithm and Its Use with IPsec

[RFC-4868] Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec

[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: Der Integritätsschutz ermöglicht es, Datenmanipulation zu erkennen.

Umsetzungsbeispiel: Einstellung von HMAC_SHA2_512_256 für AH und ESP.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-32/8.0

Req 33 Für IKEv2 müssen die folgenden Pseudozufallsfunktionen (PRFs) verwendet werden:
PRF_HMAC_SHA2_512, PRF_HMAC_SHA2_384, PRF_HMAC_SHA2_256 or
PRF_AES128_CMAC.

Nutzerrollen: Betrieb, Entwicklung, Integration

In IKEv2 werden Pseudozufällige Funktionen (PRFs) zur Schlüsselableitung verwendet.

Die nachfolgende Tabelle listet die zulässigen PRFs und die Referenzspezifikationen für die Verwendung in IKEv2 auf:

PRF	IANA Nr.	Referenzspezifikationen
PRF HMAC SHA2 512	7	[RFC-4868]
PRF HMAC SHA2 384	6	[RFC-4868]
PRF HMAC SHA2 256	5	[RFC-4868]
PRF AES128 CMAC	8	[RFC-4615]

Hinweise:

- Das Suffix (512, 384, 256) beschreibt die Output-Länge der Hashfunktion und damit des HMACs in Bit.
- PRF_AES128_CMAC verlangt zur Schlüsselableitung einen 16-Byte-Schlüssel.

Zukünftig spezifizierte PRFs, die den in diesem Dokument festgelegten Anforderungen an kryptographische Verfahren entsprechen, können ebenfalls verwendet werden. Dies gilt beispielsweise für PRFs, die Hashfunktionen der SHA

3-Familie einsetzen.

Quellen:

[RFC-4615] The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)
[RFC-4868] Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec
[RFC-7296] Internet Key Exchange Protocol Version 2 (IKEv2)
[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: Eine PRF ermöglicht die sichere Ableitung von Schlüsseln.

Umsetzungsbeispiel: Einstellung von PRF_HMAC_SHA2_384 für IKEv2.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-33/8.0

Req 34 Für IPSec müssen die folgenden Diffie-Hellman Gruppen verwendet werden: IANA-Nr. 15, 16, 19, 20, 21, 28, 29, 30, 31 oder 32.

Nutzerrollen: Betrieb, Entwicklung, Integration

Die Diffie-Hellman Gruppe wird zum Schlüsselaustausch mit Perfect Forward Secrecy (PFS) verwendet. Generell kann zwischen Elliptic Curve Gruppen und Finite Field Gruppen (mod p) unterschieden werden.

Die folgende Tabelle beinhaltet die zulässigen Diffie-Hellman Gruppen:

IANA-Nr.	Diffie-Hellman Gruppe	Referenzspezifikationen
30	brainpoolP512r1	[RFC-6954]
21	521-bit random ECP Group	[RFC-5903]
32	x448	[RFC-8031]
29	brainpoolP384r1	[RFC-6954]
20	384-bit random ECP Group	[RFC-5903]
28	brainpoolP256r1	[RFC-6954]
19	256-bit random ECP Group	[RFC-5903]
31	x25519	[RFC-8031]
16	4096-bit MODP Group	[RFC-3526]
15	3072-bit MODP Group	[RFC-3526]

Hinweis zu DH Gruppen x448 und x25519:

Die DH Gruppen x448 und x25519 werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Hinweis zu DH Gruppe 14:

Die Diffie-Hellman Gruppe 14 (IANA-Nr. 14) besitzt eine Schlüssellänge von 2048 Bit und darf nicht mehr eingesetzt werden.

Quellen:

[RFC-3526] More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)

[RFC-5903] Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2

[RFC-6954] Using the Elliptic Curve Cryptography (ECC) Brainpool Curves for the Internet Key Exchange Protocol Version 2 (IKEv2)

[RFC-8031] Curve25519 and Curve448 for the Internet Key Exchange Protocol Version 2 (IKEv2) Key Agreement

[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: Standardisierte Diffie-Hellman Gruppen verwenden sichere Parameter und beschleunigen den Schlüsselaustausch.

Umsetzungsbeispiel: Cisco Konfigurationsbeispiel zur Einstellung einer Diffie-Hellman Gruppe für den Schlüsselaustausch in Phase 1:

```
CE1_R1_4(config-isakmp)#group 20
```

Cisco Konfigurationsbeispiel zur Einstellung einer Diffie-Hellman Gruppe für den Schlüsselaustausch in Phase 2 (Perfect Forward Secrecy):

```
CE1_R1_4(ipsec-profile)#set pfs group20
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-34/8.0

10.2. Phase 1 (ISAKMP Security Association / IKE-SA)

Req 35 Bei IPSec muss eine zertifikatsbasierte Authentisierung mit den folgenden Verfahren und den freigegebenen Parametern verwendet werden: ECDSA-512 mit brainpoolP512r1, ECDSA-512 mit Kurvesecp521r1, Ed448, ECDSA-384 mit brainpoolP384r1, ECDSA-384 mit KurveSecp384r1, ECDSA-256 mit brainpoolP256r1, ECDSA-256 mit KurveSecp256r1, Ed25519 oder RSASSA-PSS.

Nutzerrollen: Betrieb, Entwicklung, Integration

Zertifikatsbasierte Authentisierung bietet bei Verwendung eines empfohlenen Authentisierungsmechanismus ein höheres Sicherheitsniveau als Pre-Shared-Key Authentisierung.

Die folgende Tabelle enthält die erlaubten Authentisierungsverfahren und Schlüssellängen:

Authentisierungsverfahren	Hash-Funktion	Schlüssellänge	Referenzspezifikation
ECDSA-512 mit brainpoolP512r1	SHA-512	512	[RFC-7427]
ECDSA-512 mit Kurvesecp521r1	SHA-512	512	[RFC-4753], [RFC-4754]
Ed448	SHAKE256 (fix)	448	[RFC-8420]
ECDSA-384 mit brainpoolP384r1	SHA-384	384	[RFC-7427]
ECDSA-384 mit KurveSecp384r1	SHA-384	384	[RFC-4753], [RFC-4754]

ECDSA-256 mit brainpoolP256r1	SHA-256	256	[RFC-7427]
ECDSA-256 mit KurveSecp256r1	SHA-256	256	[RFC-4753], [RFC-4754]
Ed25519	SHA512 (fix)	256	[RFC-8420]
RSASSA-PSS	Mind. SHA-256	Mind. 3000 Bit	[RFC-7427], [RFC-4055]

Hinweis zu Ed448 und Ed25519:

Ed448 und Ed25519 werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Hinweis zu Pre-shared Keys (PSK) Authentisierung:

Eine PSK-basierte Authentisierung darf nur nach Genehmigung durch die Sicherheitsorganisation der Deutschen Telekom Gruppe erfolgen, falls eine zertifikatsbasierte Authentisierung nicht möglich ist.

Wenn eine Pre-Shared Key (PSK) basierte Authentisierung notwendig ist, müssen folgende Bedingungen erfüllt werden:

- Ein PSK muss mindestens 128 Bit (oder 16 ASCII Zeichen) lang sein und zufällig gewählt (=hohe Entropie).
- Jeder Tunnelendpunkt muss eine Hälfte des PSK definieren (bspw. die 64 höchstwertigsten Bits / 8 linken ASCII Zeichen von Endpunkt A und die 64 niedrigstwertigen Bits / 8 rechten ASCII Zeichen von Endpunkt B).
- Ein PSK muss verschlüsselt ausgetauscht werden.
- Für jede Kommunikationsbeziehung muss ein separater Schlüssel (Pre-Shared Key) erzeugt werden.
- Ein PSK muss regelmäßig geändert werden (mindestens alle 24 Monate).

Quellen:

[RFC-4055] Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

[RFC-4753] ECP Groups for IKE and IKEv2

[RFC-4754] IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)

[RFC-7427] Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)

[RFC-8420] Using the Edwards-Curve Digital Signature Algorithm (EdDSA) in the Internet Key Exchange Protocol Version 2 (IKEv2)

[TR-02102-3] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 3 IP-Sec and IKE, Version 2025-01

Motivation: Eine zertifikatsbasierte Authentisierung unterstützt ein flexibles Management der Zugänge und die Nutzung moderner Public Key Infrastructures (PKI) inkl. Certificate Revocation Lists (CRL) bzw. Online Certificate Status Protocol (OCSP).

Umsetzungsbeispiel: Nutzung eines x509v3 Zertifikats für die IKE-SA Authentisierung.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.50-35/8.0

Req 36 Der IKE Security Association (IKE-SA) Lifetime-Parameter muss zeitbasiert sein und einen Wert von maximal 86.400 Sekunden (24 Stunden) haben.

Nutzerrollen: Betrieb, Entwicklung, Integration

Mithilfe der IKE-SA steuern die Security Gateways die Authentifizierung und Autorisierung. Daher muss ein adäquater Wert gesetzt werden.

Motivation: Unveränderte Schlüssel können Ursache für Vertraulichkeitslücken sein. Dies gilt auch bei Verwendung von starken Verschlüsselungsverfahren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.50-36/8.0

10.3. Phase 2 (Quick Mode / Child-SA)

Req 37	Encapsulating Security Payload (ESP) muss im Tunnel Mode verwendet werden.
--------	--

Nutzerrollen: Betrieb, Entwicklung, Integration

Hinweis zum Authentication Header (AH):

Der Authentication Header kann in Verbindung mit ESP genutzt werden. Da jedoch der ESP bereits sämtliche Eigenschaften zum Integritätsschutz und Anti-Replay-Schutz bereitstellt, ist die Nutzung eines Authentication Headers nicht notwendig.

Motivation: ESP stellt bereits sämtliche Eigenschaften zum Integritätsschutz und Anti-Replay bereit.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Abstreiten von durchgeführten Aktionen
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-37/8.0

Req 38	Der Child Security Association (Child-SA) Lifetime-Parameter muss zeitbasiert sein und einen Wert von maximal 7.200 Sekunden (2 Stunden) haben.
--------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

Der Session Key-Hierarchie wird nach Ablauf der Lifetime erneuert.

Motivation: Unveränderte Schlüssel können Ursache für Vertraulichkeitslücken sein. Dies gilt auch bei Verwendung von starken Verschlüsselungsverfahren.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unbemerkt durchführbare Angriffe

ID: 3.50-38/8.0

Req 39	Die Anti Replay Window Size muss kleiner oder gleich 128 Pakete sein.
--------	---

Nutzerrollen: Betrieb, Entwicklung, Integration

Das Anti Replay Protokoll schützt IPSec Verbindungen gegenüber Replay Angriffen.

Motivation: Schutz gegenüber Replay-Attacken.

Umsetzungsbeispiel: Router (crypto-map)# set security-association replay window-size 128

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Modifikation von Daten
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

ID: 3.50-39/8.0

11. Transport Layer Security (TLS/SSL) Cipher Suites

Req 40 Es muss TLS in der Version 1.2 oder 1.3 verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

TLS (Transport Layer Security) ist ein Protokoll zur sicheren Übertragung von Daten über TCP/IP-basierte Verbindungen und der Nachfolger von SSL (Secure Sockets Layer). Es schützt die Vertraulichkeit, Integrität und Authentizität der Daten bzw. der Kommunikationspartner.

TLS in der Version 1.2 [RFC-5246] und Version 1.3 [RFC-8446] bietet Cipher Suites mit Authenticated Encryption Associated Data (AEAD). AEAD bietet neben dem Schutz der Vertraulichkeit auch Integrität sowie Authentizität der übertragenen Daten.

Quellen:

[RFC-5246] The Transport Layer Security (TLS) Protocol Version 1.2

[RFC-8446] The Transport Layer Security (TLS) Protocol Version 1.3

Motivation: Die aktuelle Version von TLS schließt bisherige Sicherheitsschwachstellen und Angriffsflächen auf den TLS-Protokoll-Handshake.

Umsetzungsbeispiel: OpenSSL> protocol = tlsv1_3

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-40/8.0

Req 41 Es müssen die folgenden Cipher Suites mit Perfect Forward Secrecy (PFS) verwendet werden: für TLS v1.3: IANA-Nr. "0x13,0x01", "0x13,0x02", "0x13,0x03" or "0x13,0x04", sowie für TLS v1.2: IANA-Nr. "0xC0,0x2B", "0xC0,0x2C", "0xC0,0xAC", "0xC0,0xAD", "0xC0,0x2F", "0xC0,0x30", "0xCC,0xA9" oder "0xCC,0xA8".

Nutzerrollen: Betrieb, Entwicklung, Integration

Cipher Suites spezifizieren die kryptographischen Verfahren einer TLS-Verbindung.

Perfect Forward Secrecy (kurz PFS, auch Forward Secrecy) bedeutet, dass Verbindungsinhalte auch bei Kenntnis der Langzeit-Schlüssel der Kommunikationspartner nicht nachträglich entschlüsselt werden können.

In TLS v1.2 werden Cipher Suites wie folgt definiert: *TLS_AKE_WITH_Enc_Hash*.

Nachfolgend wird die Bedeutung der einzelnen Bestandteile erläutert:

- *AKE (Authenticated Key Exchange)*: Authentisiertes Schlüsseleinigungsverfahren für das Handshake-Protokoll
- *Enc (Encryption)*: Verschlüsselungsverfahren mit Betriebsart für das Record-Protokoll
- *Hash*: Hashfunktion für einen HMAC zur Schlüsselableitung. Sofern *Enc* kein AEAD-Verschlüsselungsverfahren ist, wird der HMAC außerdem zur Integritätssicherung verwendet.

Die nachfolgende Tabelle listet die zulässigen Cipher Suites mit PFS in TLS v1.2 sowie die Referenzspezifikationen auf. Dabei wurde der Designphilosophie von TLS v1.3 gefolgt, weshalb in dieser Tabelle lediglich AEAD-

Konstruktionen aufgelistet sind.

Cipher Suite	IANA-Nr.	Referenzspezifikation
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xC0,0x2B	[RFC-5289]
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	0xC0,0x2C	[RFC-5289]
TLS_ECDHE_ECDSA_WITH_AES_128_CCM	0xC0,0xAC	[RFC-7251]
TLS_ECDHE_ECDSA_WITH_AES_256_CCM	0xC0,0xAD	[RFC-7251]
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	0xC0,0x2F	[RFC-5289]
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	0xC0,0x30	[RFC-5289]
TLS_ECDHE_ECDSA_WITH_CHACHA20POLY1305_SHA256	0xCC,0xA9	[RFC-7905]
TLS_ECDHE_RSA_WITH_CHACHA20POLY1305_SHA256	0xCC,0xA8	[RFC-7905]

Die folgenden Cipher Suites mit PFS in TLS v1.2 mit AES-CBC (CBC-Modus) und / oder DHE Cipher Suites sind nur in Bestandsystemen erlaubt:

Cipher Suite	IANA-Nr.	Referenzspezifikation	Hinweis
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	0xC0,0x23	[RFC-5289]	CBC
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	0xC0,0x24	[RFC-5289]	CBC
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	0xC0,0x27	[RFC-5289]	CBC
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	0xC0,0x28	[RFC-5289]	CBC
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	0x00,0xA2	[RFC-5288]	DHE
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	0x00,0xA3	[RFC-5288]	DHE
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	0x00,0x9E	[RFC-5288]	DHE
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	0x00,0x9F	[RFC-5288]	DHE

TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	0xC0,0x9E	[RFC-6655]	DHE
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	0xC0,0x9F	[RFC-6655]	DHE
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256	0xCC,0xAA	[RFC-7905]	DHE
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	0x00,0x40	[RFC-5246]	CBC / DHE
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256	0x00,0x6A	[RFC-5246]	CBC / DHE
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	0x00,0x67	[RFC-5246]	CBC / DHE
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	0x00,0x6B	[RFC-5246]	CBC / DHE

Hinweis zu DHE Cipher Suites:

Für das authentifizierte Schlüsseleinigungsverfahren (AKE) werden Verfahren basierend auf elliptischen Kurven (ECDHE_ECDSA) bevorzugt. DHE (diskreter Logarithmus) Schlüsselvereinbarungen sind anfälliger gegen DoS (DHeater) Angriffe als ECDHE (elliptische Kurven). Sie werden voraussichtlich vom IETF abgekündigt und vom BSI [TR-02102-2] nur bis zum Jahr 2029 empfohlen.

Hinweis zu Cipher Suites mit AES-CBC:

Cipher Suites mit AES-CBC sind nur mit der TLS-Erweiterung "Encrypt-then-MAC" vom BSI empfohlen ([TR-02102-2]). Die Erweiterung ist aber in den meisten Implementierungen nicht aktivierbar bzw. eine Überprüfung, ob sie aktiv ist, ist nicht möglich.

In TLS v1.3 werden Cipher Suites wie folgt definiert: *TLS_AEAD_Hash*.

Nachfolgend wird die Bedeutung der einzelnen Bestandteile erläutert:

- *AEAD*: Authentisiertes Verschlüsselungsverfahren für das Record-Protokoll
- *Hash*: Hashfunktion für HMAC und HKDF im Handshake-Protokoll

Die nachfolgende Tabelle listet die empfohlenen Cipher Suites mit PFS in TLS v1.3 sowie die Referenzspezifikationen auf.

Cipher Suite	IANA-Nr.	Referenzspezifikation
TLS_AES_128_GCM_SHA256	0x13,0x01	[RFC-8446]
TLS_AES_256_GCM_SHA384	0x13,0x02	[RFC-8446]
TLS_CHACHA20_POLY1305_SHA256	0x13,0x03	[RFC-8446]
TLS_AES_128_CCM_SHA256	0x13,0x04	[RFC-8446]

Hinweis zu CHACHA20 Cipher Suites:

CHACHA20 Cipher Suites: werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Hinweis zu PQC:

Wenn eine hybride Implementierung von klassischer und Post-Quanten-Kryptographie verwendet werden (erfordert TLS 1.3), dann muss der klassische Teil dieser Anforderung entsprechen.

Alle zukünftig spezifizierten Cipher Suites, die den in diesem Dokument definierten Anforderungen entsprechen, können ebenfalls verwendet werden. Dies gilt beispielsweise für Cipher Suites, die eine Hash-Funktion aus der SHA-3-Familie verwenden.

Quellen:

[RFC-5246] The Transport Layer Security (TLS) Protocol Version 1.2

[RFC-5288] AES Galois Counter Mode (GCM) Cipher Suites for TLS

[RFC-5289] TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)

[RFC-6655] AES-CCM Cipher Suites for Transport Layer Security (TLS)

[RFC-7251] AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS

[RFC-7905] ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS)

[RFC-8446] The Transport Layer Security (TLS) Protocol Version 1.3

[TR-02102-2] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 2 TLS, Version 2025-01

Motivation: Der Einsatz von modernen Cipher Suites mit Perfect Forward Secrecy schützt die Transport-Sicherheit in TLS.

Umsetzungsbeispiel: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-41/8.0

Req 42 Für TLS müssen die Diffie-Hellman Gruppen mit IANA-Nr. 23-33 verwendet werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Die Diffie-Hellman Gruppe wird zum Schlüsselaustausch mit Perfect Forward Secrecy (PFS) verwendet.

Die folgende Tabelle beinhaltet die zulässigen Diffie-Hellman Gruppen:

Diffie-Hellman Gruppe	IANA-Nr.	Referenzspezifikation	Hinweis
secp256r1	23	[RFC 8422]	
secp384r1	24	[RFC 8422]	
secp521r1	25	[RFC 8422]	
brainpoolP256r1	26	[RFC 7027]	TLS 1.2
brainpoolP384r1	27	[RFC 7027]	TLS 1.2
brainpoolP512r1	28	[RFC 7027]	TLS 1.2
x25519	29	[RFC-8422]	
x448	30	[RFC-8422]	
brainpoolP256r1tls13	31	[RFC 8734]	TLS 1.3
brainpoolP384r1tls13	32	[RFC 8734]	TLS 1.3
brainpoolP512r1tls13	33	[RFC 8734]	TLS 1.3

ffdhe3072	257	[RFC 7919]	
ffdhe4096	258	[RFC 7919]	

Hinweis zu x448 und x25519:

x448 und x25519 werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Hinweis zu Brainpool-Kurven:

Für die Brainpool-Kurven in TLS v1.2 [RFC-7027] wurden die IANA-Nummern 26, 27 und 28 reserviert, in TLS v1.3 [RFC-8446] die IANA-Nummern 31, 32 und 33.

Hinweis zu DH Gruppe 256:

Die Diffie-Hellman Gruppe 256 (IANA-Nr. 256) besitzt eine Schlüssellänge von 2048 Bit und darf nicht mehr eingesetzt werden.

Hinweis zu DH Gruppen 257 und 258:

Diese Diffie-Hellman Gruppen sind anfälliger gegenüber dem DHeater (DoS) Angriff als elliptische Kurven Diffie-Hellman. Daher wird empfohlen Brainpool oder NIST (secp) Kurven für die Diffie-Hellman Schlüsselvereinbarung zu nutzen.

Hinweis zu PQC:

Wenn eine hybride Implementierung von klassischer und Post-Quanten-Kryptographie verwendet werden (erfordert TLS 1.3), dann muss der klassische Teil dieser Anforderung entsprechen.

Quellen:

[RFC-7027] Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS)

[RFC-7919] Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)

[RFC-8422] Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier

[RFC-8734] Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS) Version 1.3

[TR-02102-2] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 2 TLS, Version 2025-01

Motivation: Standardisierte Diffie-Hellman Gruppen verwenden sichere Parameter und beschleunigen den Schlüsselaustausch.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-42/8.0

Req 43 Für TLS müssen digitale Zertifikate von einer Zertifizierungsstelle mit ausreichender Schlüssellänge und begrenzter Gültigkeit verwenden sowie korrekt validiert werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Digitale Zertifikate werden in TLS zur Authentisierung während des TLS-Handshakes verwendet. TLS-Server müssen ein TLS-Zertifikat verwenden, Clients benötigen ein Zertifikat falls gegenseitige Authentisierung gefordert ist (mutual authentication).

TLS-Zertifikate für Webserver, die aus dem Internet erreichbar sind, müssen von einer öffentlichen Zertifizierungsstelle ausgestellt werden, die von Browsern und Betriebssystemen als Vertrauenswürdig eingestuft ist. Diese können z.B.

über den Dienst „TeleSec ServerPass“ beauftragt werden (siehe <https://www.telesec.de/de/produkte/serverpass/ueberblick>).

Bei Schlüssellängen, Gültigkeit und weiteren Konfigurationsmöglichkeiten muss die Certificate Policy der Zertifizierungsstelle beachtet werden.

Bei Webservern, die ausschließlich für interne Anwendungen verwendet werden und die nicht über das Internet erreichbar sind können digitale Zertifikate von einer privaten (internen) Public Key Infrastruktur verwendet werden.

Die Mindestanforderungen aus der folgenden Tabelle müssen für jede Form von TLS-Zertifikaten, d.h. auch für Client-zertifikate, beachtet werden:

Algorithmenfamilie	Schlüssellänge	Hashfunktion
Elliptic Curve	250 Bit	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit
Digital Signature Algorithm (DSA)	3000 Bit	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit
RSA	3000 Bit	SHA-3, SHA-2 mit einer Ausgabelänge 256 Bit

Einschränkungen zu SHA-224/SHA-3-224:

SHA-224/SHA-3-224 darf nur noch in Bestandssystemen verwendet werden und muss bei nächster Gelegenheit gegen einen Hashalgorithmus mit einer Output-Länge von mindestens 256 Bit ausgetauscht werden.

Die Zertifikatslaufzeit von öffentlichen TLS-Serverzertifikaten (ausgestellt von einer Zertifizierungsstelle, die Zertifikate entsprechend der Vorgaben des [CA/Browser Forum] ausstellt) darf maximal 397 Tage betragen. Für andere, interne TLS-Zertifikate sollte eine Laufzeit von maximal 3 Jahren nicht überschritten werden.

Die Zertifikatslaufzeit für automatisch ausgestellte und automatisch erneuerbare Zertifikate, z.B. bei ACME (Automatic Certificate Management Environment), sollte möglichst kurz gewählt werden. Zu empfehlen ist eine Laufzeit von maximal 90 Tagen.

Zertifikats-Validierung (vgl. [RFC-5280]):

- Die Zertifikate müssen gegen eine wohldefinierte Trust-Anchor-Liste validiert werden, die auf die nötigen Trust-Anchors reduziert sind. Dabei können Intermediate Zertifikate als Trust-Anchor verwendet werden.
- Bei der Validierung von Serverzertifikaten muss eine Hostname Validation durchgeführt werden.
- Die Zertifikate müssen gültig sein.
- Wenn in der Zertifikatskette „Name Constraints“ definiert sind, müssen alle nachfolgenden Zertifikate in der Kette diese „Name Constraints“ einhalten.

Quellen:

[CA/Browser Forum] <https://cabforum.org/working-groups/server/baseline-requirements/documents/>
[RFC5280] Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, chapter 6

Motivation: Digitale Zertifikate bilden die Basis der Authentisierung und schaffen Vertrauen. Ohne ausreichend starke Authentisierung sind Man-in-the-Middle Angriffe möglich.

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe

- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-43/8.0

Req 44 TLS-Serverzertifikate dürfen keine Wildcard-Zertifikate sein.

Nutzerrollen: Betrieb, Entwicklung, Integration

Ein Wildcard-Zertifikat gilt nicht für einzelne Server, sondern für alle Server, die alle zur gleichen (Sub-)Domain gehören, z. B. *.abc.telekom.de.

Als Alternative zu Wildcard-Zertifikaten bieten sich Zertifikate an, die automatisiert über das ACME-Protokoll beauftragt und bereitgestellt werden können.

Falls für ein System keine offizielle CA (Certificate Authority) zum Ausstellen von Zertifikaten per ACME-Protokoll erreicht werden kann, dann ist die Nutzung von Wildcard-Zertifikaten akzeptabel – vorausgesetzt das Zertifikat gilt nicht für Second-Level-Domains (also z.B. *.telekom.de) und es wird nicht für Produktion und Nicht-Produktion verwendet.

Quelle:

[RFC-6125] Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)

Motivation: Wildcard-Zertifikate lassen sich für eine beliebige Anzahl von Domainnamen nutzen, für die der Wildcard-Ausdruck gültig ist. Hierin besteht ein erhöhtes Missbrauchsrisiko. Außerdem ist das Risiko für einen unberechtigten Zugriff auf den privaten Schlüssel des Zertifikates umso größer, je mehr Server dieses Zertifikat einsetzen. Diese Risiken sollen verhindert werden. Auch der [RFC-6125] besagt, dass aus Sicherheitsbetrachtungen auf Wildcard-Zertifikate verzichtet werden sollte.

Umsetzungsbeispiel: Nutzung von Zertifikaten für eindeutige Domainnamen, die über das ACME-Protokoll automatisch aktualisiert werden können, z.B. von TeleSec, <https://telesec.de/de/produkte/serverpass/ueberblick/> .

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-44/8.0

Req 45 Die Sicherheit der TLS-Implementierung und -Konfiguration von Webservern muss regelmäßig überprüft werden.

Nutzerrollen: Betrieb, Entwicklung, Integration

Sichere Cipher Suites enthalten die wesentlichen Bestandteile eines Sicherheitsprotokolls:

- Authentication-Algorithmus
- Key Exchange-Algorithmus
- Encryption-Algorithmus
- MAC-Algorithmus

Die Sicherheit ist jedoch bestimmt von der jeweilig verwendeten Implementierung / Software. Des Weiteren ist sie abhängig von der eingesetzten Protokollversion. Daher müssen die eingesetzten Cipher Suites auf Basis von Security Best Practices ausgewählt werden.

Empfohlene Informationsquellen und Testwerkzeuge:

- Cipher Suite Info (<https://ciphersuite.info/>)
- SSL Labs (<https://www.ssllabs.com/ssltest/>)
- Testssl (<https://testssl.sh/>)
- DTSP Internal Scan Platform (<https://dtsp.telekom-dienste.de/>)

Hinweis zur Einstufung der Cipher Suites:

Die Informationsquellen und Testwerkzeuge stufen u.U. die in diesem Dokument als sicher angegebenen Cipher Suites als unsicher ein. Dies kann daran liegen, dass

- die in diesem Dokument genannten Hinweise nicht eingehalten bzw. berücksichtigt sind,
- die Cipher Suites in einer anderen Form unsicher konfiguriert sind oder
- sich seit Erscheinen dieses Dokumentes die Bewertung verändert wurde.

Motivation: Nur sichere Cipher Suites erlauben eine minimale Angriffsfläche und schützen vor Downgrade-Angriffen.

Umsetzungsbeispiel: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Abstreiten von durchgeführten Aktionen
- Unbemerkt durchführbare Angriffe
- Angriffe motiviert und erleichtert durch Informationspreisgabe oder erkennbare Sicherheitsschwachstellen

ID: 3.50-45/8.0

12. Secure Shell (SSH) Cipher Suites

Req 46 Die Version 2 vom SSH-Protokoll muss verwendet werden. (Ab OpenSSH 7.4+ automatisch erfüllt).

Nutzerrollen: Betrieb, Entwicklung, Integration

SSHv1 muss permanent in der Konfiguration des SSH-Servers deaktiviert werden. Mit OpenSSH 7.4 wurde die Unterstützung von SSHv1 komplett entfernt, so dass keine extra Konfiguration mehr notwendig ist.

Quelle:

[TR-02102-4] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 4 SSH, Version 2025-01

Motivation: Die Version 1 des SSH-Protokolls besitzt kryptographische Schwächen und ist obsolet. Bei einer Nutzung von SSHv1 ist die Vertraulichkeit und Integrität der übertragenen Daten nicht sichergestellt.

Umsetzungsbeispiel: OpenSSH Parameter: Protocol 2

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Störung der Verfügbarkeit
- Unbemerkt durchführbare Angriffe

ID: 3.50-46/8.0

Req 47 SSH Moduli kleiner als 3072 Bit dürfen nicht genutzt werden.

Die Datei /etc/ssh/moduli beinhaltet vorberechnete Parameter – sogenannte Moduli – für den Diffie-Hellmann Algorithmus. Es sind auch Moduli enthalten, die nicht lang genug sind, um gegen bekannte Angriffe resistent zu sein.

Um die Nutzung von Moduli-Werten kleiner als 3072 Bit zu verhindern, müssen diese aus der Datei /etc/ssh/moduli gelöscht werden. Die Spalte "size" in /etc/ssh/moduli listet den Index des höchstwertigen Bits (beginnend bei 0) auf, was bedeutet, dass ein 3072-Bit Primzahl, deren höchstes gesetztes Bit an Position 3071 liegt, als 3071 aufgeführt wird.

Quelle:

[TR 02102-4] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 4 SSH, Version 2025-01

Motivation: Wenn der genutzte DH-Modulus zu kurz ist, ist der Austausch von Schlüsseln nicht ausreichend geschützt.

Umsetzungsbeispiel:

```
$ awk '$5 >= 3071' /etc/ssh/moduli > /etc/ssh/moduli.safe  
$ mv /etc/ssh/moduli.safe /etc/ssh/moduli
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen

ID: 3.50-47/8.0

Req 48 Es dürfen nur die folgenden Algorithmen zum Schlüsselaustausch verwendet werden: curve448-sha512, sntrup761x25519-sha512@openssh.com, curve25519-sha256@libssh.org, curve25519-sha256, ecdh-sha2-nistp521, ecdh-sha2-nistp384, cdh-sha2-nistp256, diffie-hellman-group18-sha512, diffie-hellman-group16-sha512, diffie-hellman-group-exchange-sha256 oder mlkem768x25519-sha256@openssh.com.

Nutzerrollen: Betrieb, Entwicklung, Integration

Für den Austausch von Schlüsseln dürfen die folgenden Algorithmen verwendet werden:

Algorithmus	Referenzspezifikationen
curve448-sha512	[RFC-7748]
sntrup761x25519-sha512@openssh.com	
curve25519-sha256@libssh.org	[RFC-7748]
curve25519-sha256	[RFC-7748]
ecdh-sha2-nistp521	[RFC-5656]
ecdh-sha2-nistp384	[RFC-5656]
ecdh-sha2-nistp256	[RFC-5656]
diffie-hellman-group18-sha512	[RFC-8268]
diffie-hellman-group16-sha512	[RFC-8268]
diffie-hellman-group-exchange-sha256	[RFC-4419]
mlkem768x25519-sha256@openssh.com	

Hinweis zu curve448 und curve25519:

curve448 und curve25519 werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Quellen:

[RFC-4419] Diffie-Hellman Group Exchange for SSH

[RFC-5656] Elliptic Curve Algorithm Integration

[RFC-7748] Elliptic Curves for Security

[RFC-8268] More MODP Diffie-Hellman Groups

[RFC-8731] Secure Shell (SSH) Key Exchange Method Using Curve25519 and Curve448

[TR 02102-4] BSI-Technische Richtlinie: Kryptographische Verfahren - Empfehlungen und Schlüssellängen - Teil 4 SSH, Version 2025-01

Motivation: Bei einem Angriff kann gegebenenfalls die Verschlüsselung von übertragenen Daten kompromittiert werden, wenn schwache Chiffren und Algorithmen verwendet werden.

Umsetzungsbeispiel: sshd_config enthält:

```
KexAlgorithms curve448-sha512, curve25519-sha256@libssh.org, curve25519-sha256, ecdh-sha2-nistp521, ecdh-sha2-nistp384, ecdh-sha2-nistp256, diffie-hellman-group18-sha512, diffie-hellman-group16-sha512, diffie-hellman-group-exchange-sha256
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.50-48/8.0

Req 49 Es dürfen nur die folgenden Chiffren verwendet werden: aes256-gcm@openssh.com, aes128-gcm@openssh.com, chacha20-poly1305@openssh.com, aes256-ctr, aes192-ctr oder aes128-ctr.

Nutzerrollen: Betrieb, Entwicklung, Integration

Veraltete und unsichere Chiffren und Algorithmen dürfen nicht benutzt werden. Die folgenden Chiffren können für SSH genutzt werden:

Cipher	Referenzspezifikationen
aes256-gcm@openssh.com	[RFC-5647]
aes128-gcm@openssh.com	[RFC-5647]
chacha20-poly1305@openssh.com	[RFC-8439]
aes256-ctr	[RFC-4344]
aes192-ctr	[RFC-4344]
aes128-ctr	[RFC-4344]

Quellen:

[RFC-4344] The Secure Shell (SSH) Transport Layer Encryption Modes
[RFC-5647] AES Galois Counter Mode for the Secure Shell Transport Layer Protocol
[RFC-8439] ChaCha20 and Poly1305 for IETF Protocols

Motivation: Bei einem Angriff kann gegebenenfalls die Verschlüsselung von übertragenen Daten kompromittiert werden, wenn schwache Chiffren und Algorithmen verwendet werden.

Umsetzungsbeispiel: sshd_config enthält:

Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten
- Unberechtigte Modifikation von Daten
- Unberechtigte Nutzung von Diensten oder Ressourcen
- Abstreiten von durchgeführten Aktionen

ID: 3.50-49/8.0

Req 50 Es dürfen nur die folgenden MAC-Algorithmen verwendet werden: hmac-sha2-512-etm@openssh.com, hmac-sha2-512, hmac-sha2-256-etm@openssh.com oder hmac-sha2-256.

Nutzerrollen: Betrieb, Entwicklung, Integration

Es ist wichtig die Nutzung von unsicheren MAC-Algorithmen zu vermeiden. Beispiele für solche veralteten Algorithmen sind MD5 und SHA1. Die folgenden MAC-Algorithmen sind zugelassen und müssen entsprechend für den SSH-Server konfiguriert werden:

Algorithmus	Referenzspezifikationen
hmac-sha2-512-etm@openssh.com	[RFC-6688]
hmac-sha2-512	[RFC-6688]
hmac-sha2-256-etm@openssh.com	[RFC-6688]
hmac-sha2-256	[RFC-6688]

References:

[RFC-6688] SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol

Motivation: Bei einem Angriff kann gegebenenfalls die Verschlüsselung von übertragenen Daten kompromittiert werden, wenn schwache Chiffre und Algorithmen verwendet werden.

Umsetzungsbeispiel: `sshd_config` enthält:

MACs `hmac-sha2-512-etm@openssh.com, hmac-sha2-512, hmac-sha2-256-etm@openssh.com, hmac-sha2-256`

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigte Modifikation von Daten

ID: 3.50-50/8.0

Req 51 Nur die folgenden Host Key Algorithmen (alias "public key signature algorithms" oder "server authentication algorithms") dürfen verwendet werden: `ssh-ed448`, `ssh-ed25519`, `ssh-ed25519-cert-v01@openssh.com`, `sk-ssh-ed25519@openssh.com`, `sk-ssh-ed25519-cert-v01@openssh.com`, `ecdsa-sha2-nistp521`, `ecdsa-sha2-nistp521-cert-v01@openssh.com`, `ecdsa-sha2-nistp384`, `ecdsa-sha2-nistp384-cert-v01@openssh.com`, `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp256-cert-v01@openssh.com`, `sk-ecdsa-sha2-nistp256@openssh.com` oder `sk-ecdsa-sha2-nistp256-cert-v01@openssh.com`. Ein Host Key muss einen Host eindeutig identifizieren. (1:1 Beziehung).

Nutzerrollen: Betrieb, Entwicklung, Integration

Veraltete und unsichere Algorithmen dürfen nicht benutzt werden. Die folgenden Algorithmen können für SSH genutzt werden. Die Bezeichnungen können von Software zu Software variieren:

Algorithmus	Referenzspezifikationen
<code>ssh-ed448</code>	[RFC-8709]
<code>ssh-ed25519</code>	[RFC-8709]
<code>ssh-ed25519-cert-v01@openssh.com</code>	[RFC-8709]
<code>sk-ssh-ed25519@openssh.com</code>	[RFC-8709]
<code>sk-ssh-ed25519-cert-v01@openssh.com</code>	[RFC-8709]
<code>ecdsa-sha2-nistp521</code>	[RFC-5656]
<code>ecdsa-sha2-nistp521-cert-v01@openssh.com</code>	[RFC-5656]
<code>ecdsa-sha2-nistp384</code>	[RFC-5656]
<code>ecdsa-sha2-nistp384-cert-v01@openssh.com</code>	[RFC-5656]
<code>ecdsa-sha2-nistp256</code>	[RFC-5656]
<code>ecdsa-sha2-nistp256-cert-v01@openssh.com</code>	[RFC-5656]
<code>sk-ecdsa-sha2-nistp256@openssh.com</code>	[RFC-5656]
<code>sk-ecdsa-sha2-nistp256-cert-v01@openssh.com</code>	[RFC-5656]
<code>rsa-sha2-512</code>	[RFC-8332]
<code>rsa-sha2-512-cert-v01@openssh.com</code>	[RFC-8332]
<code>rsa-sha2-256</code>	[RFC-8332]
<code>rsa-sha2-256-cert-v01@openssh.com</code>	[RFC-8332]

Hinweis zu ed448 und ed25519:

ed448 und ed25519 werden nicht (explizit) vom BSI empfohlen, aber bisher sind keine Schwachstellen bekannt, so dass sie als sicher einzustufen sind.

Quellen:

[RFC-5656] Elliptic Curve Algorithm Integration

[RFC-8709] Ed25519 and Ed448 Public Key Algorithms for the Secure Shell (SSH) Protocol

[RFC-8332] Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol

Motivation: Bei einem Angriff kann gegebenenfalls unbemerkt ein Host imitiert werden.

Umsetzungsbeispiel: `sshd_config` enthält:

```
HostKeyAlgorithms ssh-ed448,ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,sk-ssh-ed25519@openssh.com,sk-ssh-ed25519-cert-v01@openssh.com,ecdsa-sha2-nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,sk-ecdsa-sha2-nistp256@openssh.com,sk-ecdsa-sha2-nistp256-cert-v01@openssh.com
```

Für diese Anforderung sind folgende Bedrohungen relevant:

- Unberechtigter Zugang zum System
- Unberechtigte Einsichtnahme oder Abgriff von Daten

ID: 3.50-51/8.0