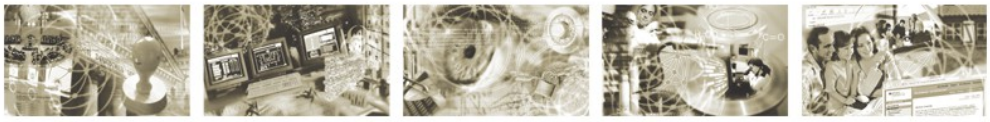
Bundesamt
für Sicherheit in der
Informationstechnik



Technische Richtlinie TR-02102-2

Kryptographische Verfahren: Empfehlungen und Schlüssellängen

Teil 2 – Verwendung von Transport Layer Security (TLS)

(Version 2018-01)

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
E-Mail: TR02102@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

1 Einleitung....4
2 Grundlagen....4
3 Empfehlungen....5
3.1 Allgemeine Hinweise....5
3.1.1 Verwendungszeiträume....5
3.1.2 Sicherheitsniveau....5
3.1.3 Schlüssellängen bei Verfahren mit elliptischen Kurven....5
3.2 SSL/TLS-Versionen....5
3.3 Cipher-Suiten....6
3.3.1 Empfohlene Cipher-Suiten....6
3.3.2 Übergangsregelungen....8
3.4 Weitere Hinweise und Empfehlungen zu TLS....9
3.4.1 Session Renegotiation....9
3.4.2 Verkürzung der HMAC-Ausgabe....9
3.4.3 TLS-Kompression und der CRIME-Angriff....9
3.4.4 Der Lucky 13-Angriff....10
3.4.5 Die Encrypt-then-MAC-Erweiterung....10
3.4.6 Die Heartbeat-Erweiterung....11
3.4.7 Die Extended Master Secret-Erweiterung....11
3.4.8 Die Supported Groups-Erweiterung....11
3.4.9 Die Signature Algorithms-Erweiterung....12
3.5 Authentisierung der Kommunikationspartner....12
3.6 Domainparameter und Schlüssellängen....12
3.6.1 Verwendung von elliptischen Kurven....14
4 Schlüssel und Zufallszahlen....14
4.1 Schlüsselspeicherung....14
4.2 Umgang mit Ephemer-Schlüsseln....14
4.3 Zufallszahlen....14
5 Quellenverzeichnis....15

Tabellenverzeichnis

Tabelle 1: Empfohlene Cipher-Suiten mit Perfect Forward Secrecy....7
Tabelle 2: Empfohlene Cipher-Suiten ohne Perfect Forward Secrecy....7
Tabelle 3: Empfohlene Cipher-Suiten mit Pre-Shared Key....8
Tabelle 4: Übergangsregelungen für TLS....9
Tabelle 5: Empfohlene Mindest-Schlüssellängen für das TLS-Handshakeprotokoll....13

1 Einleitung

Diese Technische Richtlinie enthält Empfehlungen für den Einsatz des kryptographischen Protokolls *Transport Layer Security (TLS)*. Es dient der sicheren Übertragung von Informationen in Datennetzwerken, wobei insbesondere die *Vertraulichkeit,* die *Integrität* und die *Authentizität* der übertragenen Informationen im Vordergrund stehen.

Die vorliegende Technische Richtlinie enthält Empfehlungen für die zu verwendende Protokollversion sowie die kryptographischen Algorithmen und Schlüssellängen als Konkretisierung der allgemeinen Empfehlungen in Teil 1 dieser Technischen Richtlinie [TR-02102-1].

Diese Technische Richtlinie enthält keine Empfehlungen für konkrete Anwendungen, keine Risikobewertungen sowie keine Angriffsmöglichkeiten, die sich aus Fehlern in der Implementierung des Protokolls ergeben.

Hinweis: Auch bei Beachtung aller Empfehlungen für die Verwendung von TLS können Daten in erheblichem Umfang aus einem kryptographischen System abfließen, zum Beispiel durch Ausnutzung von Seitenkanälen (Messung von Timing-Verhalten, Stromaufnahme, Datenraten etc.). Daher sollte der Entwickler eines kryptographischen Systems unter Hinzuziehung von Experten auf diesem Gebiet mögliche Seitenkanäle identifizieren und entsprechende Gegenmaßnahmen umsetzen. Je nach Anwendung gilt dies auch für Fault-Attacken.

Hinweis: Für Definitionen kryptographischer Begriffe in diesem Dokument siehe das Glossar in [TR-02102-1].

2 Grundlagen

Transport Layer Security (TLS), früher bekannt als Secure Socket Layer (SSL), ermöglicht die sichere Übertragung von Informationen aus der Anwendungsschicht (zum Beispiel HTTPS, FTPS oder IMAPS) über TCP/IP-basierte Verbindungen (insbesondere das Internet).

Bevor Daten übertragen werden können, muss zunächst eine gesicherte Verbindung zwischen den beiden Verbindungspartnern (Client und Server) aufgebaut werden. Dieser Vorgang heißt *Handshake* und ist ein wichtiger Bestandteil des TLS-Protokolls. Hierbei werden zwischen Client und Server vereinbart:

1. Kryptographische Verfahren für die Ver*schlüsselung*, *Integritätssicherung, Schlüsseleinigung* und ggf. für die (ein- oder beidseitige) *Authentisierung.* Diese Verfahren werden durch die sogenannte *Cipher-Suite* festgelegt (siehe Abschnitt 3.3).
2. Ein gemeinsames Geheimnis, das *premaster secret*. Aus diesem wird von beiden Verbindungspartnern das *master secret* erzeugt, aus welchem wiederum die Sitzungsschlüssel für den Integritätsschutz und die Verschlüsselung abgeleitet werden.

Hinweis: Das TLS-Protokoll erlaubt auch Verbindungen, die nicht oder nur einseitig authentisiert sind (beispielsweise sind HTTPS-Verbindungen üblicherweise nur serverseitig authentisiert). Daher sollten Entwickler kryptographischer Systeme darauf achten, ob eine weitere Authentisierung innerhalb der Anwendungsschicht erforderlich ist (Beispiel: Authentisierung eines Homebanking-Benutzers durch Anforderung eines Passwortes). Bei besonders kritischen Operationen sollte dabei grundsätzlich eine Authentisierung durch Wissen und Besitz (Zwei-Faktor-Authentisierung) erfolgen, die

sich unter Ausnutzung kryptographischer Mechanismen auch auf die übertragenen Daten erstrecken sollte.

3 Empfehlungen

3.1 Allgemeine Hinweise

3.1.1 Verwendungszeiträume

Die Empfehlungen in dieser Technischen Richtlinie sind mit Verwendungszeiträumen versehen. Die Angabe der Jahreszahl bedeutet hierbei, dass das entsprechende Verfahren bis zum Ende des angegebenen Jahres empfohlen wird. Ist die Jahreszahl mit einem „+“-Zeichen gekennzeichnet, so bedeutet dies, dass dieser Verwendungszeitraum möglicherweise in einer zukünftigen Version dieser Technischen Richtlinie verlängert wird.

3.1.2 Sicherheitsniveau

Das Sicherheitsniveau für alle kryptographischen Verfahren in dieser Technischen Richtlinie richtet sich nach dem in Abschnitt 1.1 in [TR-02102-1] angegebenen Sicherheitsniveau. Es liegt zurzeit bei 100 Bit.

Hinweis: Ab dem Jahr 2023 wird ein Sicherheitsniveau von 120 Bit angestrebt. Siehe dazu auch Abschnitt 1.1 in [TR-02102-1].

3.1.3 Schlüssellängen bei Verfahren mit elliptischen Kurven

Für einen Einsatzzeitraum bis Ende 2022 sind die Schlüssellängen bei Verfahren, die auf elliptischen Kurven basieren, etwas größer (im Vergleich zu RSA) gewählt worden, um einen Sicherheitsspielraum für diese Verfahren zu erreichen (vgl. Abschnitt 3.6). Für eine Begründung und weitere Erläuterungen siehe Bemerkung 4, Kapitel 3 in [TR-02102-1].

3.2 SSL/TLS-Versionen

Das SSL-Protokoll existiert in den Versionen 1.0, 2.0 und 3.0, wobei die Version 1.0 nicht veröffentlicht wurde. TLS 1.0 ist eine direkte Weiterentwicklung von SSL 3.0 und wird in [RFC2246] spezifiziert. Des Weiteren gibt es die Versionen 1.1 und 1.2 des TLS-Protokolls, welche in [RFC4346] und [RFC5246] spezifiziert werden.

Empfehlungen für die Wahl der TLS-Version sind:

- Grundsätzlich wird TLS 1.2 empfohlen.
- TLS 1.1 wird **nicht mehr empfohlen** (siehe dazu Abschnitt 3.3.2).
- TLS 1.0 wird **nicht empfohlen.**
- SSL v2 ([SSLv2]) und SSL v3 ([SSLv3]) werden **nicht empfohlen** (siehe auch [RFC6176]).

Bemerkung: Zum Zeitpunkt der Erstellung der vorliegenden Technischen Richtlinie ist der Entwurf des RFC zu TLS 1.3 noch nicht verabschiedet. Daher enthält diese Technische Richtlinie (noch) keine Empfehlungen zu TLS 1.3.

3.3 Cipher-Suiten

Eine Cipher-Suite spezifiziert die zu verwendenden kryptographischen Algorithmen für

- die Schlüsseleinigung (und ggf. Authentisierung),
- die Verschlüsselung (Strom- bzw. Blockchiffre inkl. Betriebsmodus) der Daten, und
- eine Hashfunktion für die Integritätssicherung (HMAC-Algorithmus) der Datenpakete und für die Verwendung als Pseudozufallszahlengenerator (ab TLS 1.2).

  Hinweis: Ab TLS 1.2 wurde die Kombination von MD5 und SHA-1 in der Pseudozufallsfunktion (engl. pseudorandom function, kurz PRF) durch eine Cipher-Suite-spezifische PRF ersetzt (Beispiel: Die Cipher-Suite TLS_RSA_WITH_AES_256_GCM_SHA384 verwendet SHA-384 als PRF).

Eine vollständige Liste aller definierten Cipher-Suiten mit Verweisen auf die jeweiligen Spezifikationen ist verfügbar unter [IANA].

3.3.1 Empfohlene Cipher-Suiten

Grundsätzlich wird empfohlen, nur Cipher-Suiten einzusetzen, die die Anforderungen an die Algorithmen und Schlüssellängen der [TR-02102-1] erfüllen.

Als Konkretisierung der allgemeinen Empfehlungen in der [TR-02102-1] wird der Einsatz der folgenden Cipher-Suiten empfohlen (vgl. [RFC5246] Und [RFC5289]):

	Schlüsseleinigung und -authentisierung		*Verschlüsselung*	*Betriebs-modus*	*Hash*	*Verwendung bis*
TLS_	ECDHE_ECDSA_	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_ GCM_	SHA384	2024+
	ECDHE_RSA_	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_ GCM_	SHA384	2024+
	DHE_DSS_[1]	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_	SHA256	2024+
				GCM_	SHA384	2024+
	DHE_RSA_[1]	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_	SHA256	2024+
				GCM_	SHA384	2024+

Tabelle 1: Empfohlene Cipher-Suiten mit Perfect Forward Secrecy

Sofern die Verwendung von Cipher-Suiten mit Perfect Forward Secrecy[2] nicht möglich ist, können auch die folgenden Cipher-Suiten eingesetzt werden (vgl. [RFC5246] und [RFC5289]):

1 Da einige gängige Implementierungen von DH(E) in TLS zurzeit nur 1024 Bit unterstützen, sei hier auf Abschnitt 3.6 sowie Abschnitt 7.2.1 in [TR-02102-1] verwiesen, in welchen eine Mindestgröße von 2000 Bit für dieses Verfahren empfohlen wird.

2 Perfect Forward Secrecy (kurz PFS, auch Forward Secrecy) bedeutet, dass eine Verbindung auch bei Kenntnis der Langzeit-Schlüssel der Kommunikationspartner nicht nachträglich entschlüsselt werden kann. Bei der Verwendung von TLS zum Schutz personenbezogener oder anderer sensibler Daten wird Perfect Forward Secrecy grundsätzlich empfohlen.

	Schlüsseleinigung und -authentisierung		*Verschlüsselung*	*Betriebs-modus*	*Hash*	*Verwendung bis*
TLS_	ECDH_ECDSA_	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_ GCM_	SHA384	2024+
	ECDH_RSA_	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_ GCM_	SHA384	2024+
	DH_DSS_	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_	SHA256	2024+
				GCM_	SHA384	2024+
	DH_RSA_	WITH_	AES_128_	CBC_ GCM_	SHA256	2024+
			AES_256_	CBC_	SHA256	2024+
				GCM_	SHA384	2024+

Tabelle 2: Empfohlene Cipher-Suiten ohne Perfect Forward Secrecy

3.3.1.1 Schlüsseleinigung mit vorab ausgetauschten Daten

Sollen bei einer TLS-Verbindung zusätzliche vorab ausgetauschte Daten in die Schlüsseleinigung einfließen, können Cipher-Suiten mit einem Pre-shared Key (kurz PSK, siehe dazu [RFC5487] und [RFC5489]) verwendet werden. Grundsätzlich werden hierbei solche Cipher-Suiten empfohlen, bei denen neben dem Pre-shared Key weitere ephemere Schlüssel oder ausgetauschte Zufallszahlen in die Schlüsseleinigung eingehen.

Die Verwendung von Cipher-Suiten vom Typ `TLS_PSK_*`, das heißt ohne zusätzliche ephemere Schlüssel oder Zufallszahlen, wird **nicht empfohlen,** da bei diesen Cipher-Suiten die Sicherheit der Verbindung ausschließlich auf der Entropie und der Vertraulichkeit des Pre-shared Keys beruht.

Die folgenden Cipher-Suiten mit PSK werden empfohlen:

	Schlüsseleinigung und -authentisierung		*Verschlüsselung*	*Betriebs-modus*	*Hash*	*Verwendung bis*
TLS_	ECDHE_PSK_	WITH_	AES_128_	CBC_	SHA256	2024+
			AES_256_		SHA384	2024+
	DHE_PSK_	WITH_	AES_128_	CBC_	SHA256	2024+
				GCM_		2024+
			AES_256_	CBC_	SHA384	2024+
				GCM_		2024+
	RSA_PSK_	WITH_	AES_128_	CBC_	SHA256	2024+
				GCM_		2024+
			AES_256_	CBC_	SHA384	2024+
				GCM_		2024+

Tabelle 3: Empfohlene Cipher-Suiten mit Pre-Shared Key

Hinweis: Die Cipher-Suiten der Form TLS_RSA_PSK_* aus Tabelle 3 bieten *keine* Perfect Forward Secrecy, alle anderen Cipher-Suiten aus Tabelle 3 hingegen bieten Perfect Forward Secrecy.

3.3.2 Übergangsregelungen

SHA-1 ist keine kollisionsresistente Hashfunktion; die Erzeugung von SHA-1-Kollisionen ist zwar mit einigem Aufwand verbunden, aber praktisch machbar [SBK17]. Gegen die Verwendung in Konstruktionen, die keine Kollisionsresistenz benötigen (zum Beispiel als Grundlage für einen HMAC oder als Komponente eines Pseudozufallsgenerators) spricht aber nach gegenwärtigem Kenntnisstand sicherheitstechnisch nichts. Es wird empfohlen, auch in diesen Anwendungen als grundsätzliche Sicherungsmaßnahme eine Hashfunktion der SHA-2-Familie oder der SHA-3-Familie einzusetzen. Prinzipiell ist eine Verwendung von SHA-1 in der HMAC-Konstruktion oder in anderen kryptographischen Mechanismen mit vergleichbaren kryptographischen Anforderungen an die genutzte Hashfunktion (zum Beispiel im Rahmen eines Pseudozufallsgenerators oder als Teil der Mask Generation Function in RSA-OAEP) bis 2018 konform zu der vorliegenden Technischen Richtlinie.

Daher kann, abweichend zu den Empfehlungen in diesem Kapitel und in Teil 1 dieser Technischen Richtlinie [TR-02102-1], in *bestehenden* TLS-Anwendungen als Hashfunktion für die Integritätssicherung mittels HMAC übergangsweise noch SHA-1 eingesetzt werden (das heißt Cipher-Suiten der Form *_SHA). Unabhängig von dem in Tabelle 4 angegebenen *maximalen* Verwendungszeitraum wird eine schnellstmögliche Migration zu SHA-256 bzw. SHA-384 und TLS 1.2 empfohlen.

Hinweis: Da TLS 1.1 die Hashfunktion SHA-1 als Komponente für die Signaturerstellung verwendet (und keine Unterstützung der SHA-2-Familie bietet), wird der Einsatz von TLS 1.1 nicht mehr empfohlen.

Der Verschlüsselungsalgorithmus RC4 in TLS weist erhebliche Sicherheitsschwächen auf. Seine Verwendung wird daher nicht empfohlen.

Abweichung	*Verwendung maximal bis*	*Empfehlung*
SHA-1 zur HMAC-Berechnung und als Komponente der PRF in TLS	2018+	Migration zu SHA-256/-384
SHA-1 als Komponente für die Signaturerstellung in TLS	2015	Migration zu SHA-256/-384
TLS 1.0 **bei Bestandssystemen** zusammen mit geeigneten Schutzmaßnahmen gegen Chosen-Plaintext-Angriffe auf die CBC-Implementierung wie oben beschrieben (z.B. BEAST)	2014	Migration zu TLS 1.2 mit AES
RC4 als Verschlüsselungsfunktion	2013	

Tabelle 4: Übergangsregelungen für TLS

Anmerkung: In der vorliegenden Technischen Richtlinie wurde der Einsatz von SHA-1 bis 2015 als Komponente für die Signaturerstellung *ausschließlich im Rahmen von TLS* empfohlen (vgl. Tabelle 4), damit TLS 1.0 noch übergangsweise bis Ende 2015 eingesetzt werden konnte. Da SHA-1 für den Handshake bei TLS 1.0 erforderlich ist und eine SHA-1-Kollision sicherlich nicht in Echtzeit (also während des Handshakes) berechnet werden konnte, war der Einsatz von SHA-1 in diesem einzigen Spezialfall noch etwas länger möglich.

Grundsätzlich wird jedoch die Verwendung von SHA-1 (z.B. für die Erstellung von Signaturen) in der TR-02102-1 seit 2013 nicht mehr empfohlen.

3.4 Weitere Hinweise und Empfehlungen zu TLS

3.4.1 Session Renegotiation

Es wird empfohlen, Session Renegotiation nur auf Basis von [RFC5746] zu verwenden. Durch den Client initiierte Renegotiation sollte vom Server abgelehnt werden.

3.4.2 Verkürzung der HMAC-Ausgabe

Die in [RFC6066] definierte Extension „truncated_hmac“ zur Verkürzung der Ausgabe des HMAC auf 80 Bit sollte *nicht* verwendet werden.

3.4.3 TLS-Kompression und der CRIME-Angriff

TLS bietet die Möglichkeit, die übertragenen Daten vor der Verschlüsselung zu komprimieren. Dies führt zu der Möglichkeit eines Seitenkanalangriffes auf die Verschlüsselung, und zwar mit Hilfe der Länge der verschlüsselten Daten (siehe [CRIME]).

Um dies zu verhindern, muss sichergestellt werden, dass alle Daten eines Datenpakets von dem korrekten und legitimen Verbindungspartner stammen und keine Plaintext-Injection durch einen

Angreifer möglich ist. Kann dies nicht sichergestellt werden, so darf wird empfohlen, die TLS-Datenkompression nicht zu verwenden.

3.4.4 Der Lucky 13-Angriff

Lucky 13 ist ein Seitenkanalangriff (Timing) auf TLS, bei dem der Angreifer sehr geringe Zeitdifferenzen bei der Verarbeitung des Paddings auf Seiten des Servers ausnutzt. Für diesen Angriff muss der Angreifer sehr genaue Zeitmessungen im Netzwerk machen können. Er schickt manipulierte Chiffrate an den Server und misst die Zeit, die der Server benötigt, um das Padding dieser Chiffrate zu prüfen bzw. einen Fehler zu melden. Durch Netzwerk-Jitter können hier aber sehr leicht Fehler bei der Zeitmessung entstehen, so dass ein Angriff grundsätzlich als schwierig realisierbar erscheint, denn der Angreifer muss im Netzwerk „sehr nahe“ am Server sein, um genau genug messen zu können.

Der Angriff kann abgewehrt werden, wenn

- Authenticated Encryption, wie zum Beispiel AES-GCM (erst ab TLS 1.2 verfügbar), oder
- Encrypt-then-MAC (siehe auch nächster Abschnitt)

eingesetzt wird.

3.4.5 Die Encrypt-then-MAC-Erweiterung

Gemäß TLS-Spezifikation (siehe [RFC5246]) werden die zu übertragenen Daten zunächst mit einem Message Authentication Code (MAC) gesichert und dann mit einem Padding versehen; danach werden die Daten und das Padding verschlüsselt. Diese Reihenfolge („MAC-then-Encrypt“) war in der Vergangenheit häufig der Grund für Angriffe auf die Verschlüsselung, da das Padding nicht durch den MAC geschützt ist.

Bei den sogenannten Padding-Oracle-Angriffen werden die verschlüsselten TLS-Pakete durch einen Man-in-the-Middle-Angreifer manipuliert, um die Prüfung des Paddings als Seitenkanal zu missbrauchen. Dies kann beispielsweise dazu führen, dass der Angreifer ein HTTPS-Sitzungs-Cookie entschlüsseln kann und somit die Sitzung des Opfers übernehmen kann.

In RFC 7366 wird die TLS-Erweiterung „Encrypt-then-MAC“ spezifiziert. Hierbei werden die zu übertragenen Daten zuerst mit einem Padding versehen, dann verschlüsselt und danach mit einem MAC gesichert. Damit sind Manipulationen des Paddings ausgeschlossen, da es auch durch den MAC gesichert ist.

Der Einsatz der TLS-Erweiterung „Encrypt-then-MAC“ gemäß RFC 7366 wird empfohlen, sobald geeignete Implementierungen zur Verfügung stehen.

Hinweis: Ab TLS 1.2 gibt es Cipher-Suiten mit Authenticated Encryption. Dabei werden Verschlüsselung und MAC-Sicherung kombiniert. Die oben beschriebenen Angriffe können durch den Einsatz von Authenticated Encryption ebenfalls abgewehrt werden. Ein Beispiel für die Verschlüsselung mit Authenticated Encryption ist die Kombination von AES mit dem Galois Counter Mode (AES-GCM).

Die Verwendung von Authenticated Encryption (ab TLS 1.2) ist eine Alternative zur oben genannten Encrypt-then-MAC Extension.

3.4.6 Die Heartbeat-Erweiterung

Die Heartbeat-Erweiterung wird in RFC 6520 spezifiziert; sie ermöglicht es, eine TLS-Verbindung über einen längeren Zeitraum aufrecht zu halten, ohne eine Renegotiation der Verbindung durchfüh-

ren zu müssen. Durch den sogenannten Heartbleed-Bug ist es einem Angreifer möglich, bestimmte Speicherbereiche des Servers auszulesen, die möglicherweise geheimes Schlüsselmaterial enthalten. Dies kann zu einer vollständigen Kompromittierung des Servers führen, falls der private Schlüssel des Servers bekannt wird.

Empfehlung: Es wird dringend empfohlen, die Heartbeat-Erweiterung nicht zu verwenden. Sollte es trotzdem erforderlich sein, so sollte sichergestellt sein, dass die verwendete TLS-Implementierung über Schutzmaßnahmen gegen den Heartbleed-Bug verfügt.

3.4.7 Die Extended Master Secret-Erweiterung

Um Angriffe, wie zum Beispiel den Triple Handshake-Angriff (siehe [BDF14]) abzuwehren, ist es sehr sinnvoll, weitere Verbindungsparameter in den TLS-Handshake einfließen zu lassen, damit unterschiedliche TLS-Verbindungen auch unterschiedliche Master Secrets (aus welchem die symmetrischen Schlüssel abgeleitet werden) benutzen.

In [RFC7627] wird die TLS-Erweiterung *Extended Master Secret* spezifiziert, die bei der Berechnung des „erweiterten" Master Secrets einen Hashwert über alle Nachrichten des TLS-Handshakes mit in dieses einfließen lässt.

Der Einsatz der TLS-Erweiterung *Extended Master Secret* gemäß [RFC7627] wird empfohlen, sobald geeignete Implementierungen zur Verfügung stehen.

3.4.8 Die Supported Groups-Erweiterung

Für Cipher-Suiten mit DHE (das heißt ephemeres Diffie-Hellman in endlichen Körpern) wird in [RFC7919], Kapitel 3 die Supported Groups-Erweiterung vorgeschlagen:

> *„The compatible client that wants to be able to negotiate strong FFDHE sends a Supported Groups extension (identified by type elliptic_curves(10) in [RFC4492]) in the ClientHello and includes a list of known FFDHE groups in the extension data, ordered from most preferred to least preferred."*

Mit dieser TLS-Erweiterung kann sichergestellt werden, dass nur Gruppen verwendet werden, die bereits bekannt und in Hinblick auf Sicherheit untersucht wurden, ähnlich der Empfehlung in Abschnitt 3.6.1, nur *named curves* für Verfahren mit elliptischen Kurven zu verwenden.

Die zur Zeit spezifizierten FFDHE-Gruppen finden sich im Abschnitt „Supported Groups Registry" in [IANA]. Die mathematischen Parameter (Modulus, Erzeuger, Größe der Gruppe sowie Sicherheitsniveau in Bit) der FFDHE-Gruppen finden sich in Appendix A in [RFC7919].

Der Einsatz der Supported Groups-Erweiterung gemäß [RFC7919] wird empfohlen, sobald geeignete Implementierungen zur Verfügung stehen.

3.4.9 Die Signature Algorithms-Erweiterung

Ab TLS 1.2 kann der Client dem Server mitteilen, welche Kombination aus Signatur- und Hash-Algorithmus er verwenden möchte. Er kann dies dem Server mit der Signature Algorithms-Erweiterung in Form eines Paares mitteilen, dass jeweils einen Wert für das Signaturverfahren und einen für die Hashfunktion enthält. Siehe Abschnitt 7.4.1.4.1 in [RFC5246] für eine Liste der erlaubten Signatur- und Hashverfahren.

Durch diese Erweiterung, die erst ab TLS 1.2 zur Verfügung steht, ist man nicht mehr gezwungen, ausschließlich SHA-1 zu verwenden, sondern kann beispielsweise ECDSA mit SHA-256 kombinieren. Dies ist eine signifikante Veränderung gegenüber den vorigen TLS-Versionen.

Für weitergehende Informationen hierzu, siehe Abschnitt 7.4.1.4.1 in [RFC5246].

3.5 Authentisierung der Kommunikationspartner

Das TLS-Protokoll bietet die folgenden drei Möglichkeiten zur Authentisierung der Kommunikationspartner:

- Authentisierung beider Kommunikationspartner
- Nur serverseitige Authentisierung
- Keine Authentisierung

Die Notwendigkeit einer Authentisierung ist abhängig von der jeweiligen Anwendung. Bei der Verwendung von TLS im Web ist im Allgemeinen zumindest eine Authentisierung des Servers notwendig. Bei der Verwendung in geschlossenen Systemen (VPN o. ä.) ist zumeist eine beidseitige Authentisierung notwendig.

Für die Authentisierung innerhalb von Projekten des Bundes sind die Vorgaben in [TR-03116-4] zu beachten.

3.6 Domainparameter und Schlüssellängen

Die Domainparameter und Schlüssellängen für

- statische Schlüsselpaare der Kommunikationspartner,
- ephemere Schlüsselpaare bei der Verwendung von Cipher-Suiten mit Perfect Forward Secrecy, und
- Schlüsselpaare für die Signatur von Zertifikaten

müssen den Vorgaben in Teil 1 dieser Technischen Richtlinie (siehe [TR-02102-1]) entsprechen. Es wird empfohlen, mindestens die folgenden Schlüssellängen zu verwenden:

Algorithmus	*Minimale Schlüssellänge*	*Verwendung spätestens ab*	*Verwendung bis*
Signaturschlüssel für Zertifikate und Schlüsseleinigung			
ECDSA	224 Bit		2015
	250 Bit[3]		2024+
DSS	2000 Bit		2022
	3000 Bit	2023	2024+
RSA	2000 Bit		2022
	3000 Bit	2023	2024+
Statische und ephemere Diffie-Hellman-Schlüssel			
ECDH	224 Bit		2015
	250 Bit[3]		2024+
DH	2000 Bit		2022
	3000 Bit	2023	2024+

Tabelle 5: Empfohlene Mindest-Schlüssellängen für das TLS-Handshakeprotokoll

Hinweis: Ist ein Schlüsselpaar *statisch,* so wird es mehrfach für neue Verbindungen wiederverwendet. Im Gegensatz dazu bedeutet *ephemer,* dass für jede neue Verbindung auch ein neues Schlüsselpaar erzeugt und verwendet wird. Ephemere Schlüssel müssen nach Verbindungsende unbedingt sicher gelöscht werden, siehe dazu auch Abschnitt 4.2. Soll eine Verbindung die Eigenschaft *Perfect Forward Secrecy* erfüllen, müssen ausschließlich ephemere Schlüsselpaare verwendet werden.

Wichtiger Hinweis: Es ist sinnvoll, eine Schlüssellänge von 3000 Bit zu nutzen, um ein gleichartiges Sicherheitsniveau für alle asymmetrischen Verfahren zu erreichen. Eine Schlüssellänge von mindestens 3000 Bit ist damit ab dem Jahr 2023 für kryptographische Implementierungen verbindlich, wenn sie zur vorliegenden Technischen Richtlinie konform sein sollen.

Jede Schlüssellänge von mindestens 2000 Bit bleibt aber für Systeme mit einer Lebensdauer bis zum Jahr 2022 konform zur vorliegenden Technischen Richtlinie. Es handelt sich dabei um die empfohlene Mindest-Schlüssellänge für RSA, DH und DSS. Weitere Informationen finden sich in den Bemerkungen 4 und 5 in Kapitel 3 in [TR-02102-1].

Bemerkung: Die Empfehlungen in dieser Technischen Richtlinie sind geeignet, um das in Abschnitt 3.1.2 genannte Sicherheitsniveau von zurzeit 100 Bit zu erreichen.

Der Vorhersagezeitraum für die vorliegenden Empfehlungen beträgt 7 Jahre. Geeignete Empfehlungen für deutlich größere Zeiträume, wie sie in anderen öffentlich verfügbaren Dokumenten zu finden sind, sind naturgemäß sehr schwierig, da zukünftige kryptographische Entwicklungen über längere Zeiträume nicht oder zumindest nicht präzise vorausgesagt werden können. In solchen Fällen

3 Hier werden 250 Bit (statt 256 Bit) empfohlen, um kleine Co-Faktoren bei elliptischen Kurven zu ermöglichen.

umfassen diese Empfehlungen Parameter und Schlüssellängen, die über die in der vorliegenden Technischen Richtlinie hinausgehen können.

3.6.1 Verwendung von elliptischen Kurven

Bei der Verwendung von elliptischen Kurven werden stets kryptographisch starke Kurven über endlichen Körpern der Form F_p (p prim) empfohlen. Zusätzlich wird empfohlen, nur *named curves* (siehe Abschnitt „Supported Groups Registry“ in [IANA]) einzusetzen, um Angriffe über nicht verifizierte schwache Domainparameter zu verhindern. Die folgenden *named curves* werden empfohlen:

- brainpoolP256r1, brainpoolP384r1, brainpoolP512r1 (siehe [RFC5639] und [RFC7027])

Sollten diese Kurven nicht verfügbar sein, so können auch die folgenden Kurven eingesetzt werden:

- secp256r1, secp384r1

Hinweis: Gemäß den Empfehlungen in Tabelle 5 wird die Kurve secp224r1 für einen Einsatz nach 2015 nicht mehr empfohlen.

Hinweis: Standardmäßig wird bei (EC-basierten) Signaturverfahren, wie zum Beispiel ECDSA, die Hashfunktion SHA-1 verwendet. Ab TLS 1.2 können die Signaturverfahren auch mit anderen Hashfunktionen verwendet werden. Siehe dazu Abschnitt 3.4.9 über die Signature Algorithms-Erweiterung.

4 Schlüssel und Zufallszahlen

4.1 Schlüsselspeicherung

Private kryptographische Schlüssel, insbesondere statische Schlüssel und Signaturschlüssel, müssen sicher gespeichert und verarbeitet werden. Dies bedeutet u. a. den Schutz vor Kopieren, missbräuchlicher Nutzung und Manipulation der Schlüssel. Eine sichere Schlüsselspeicherung kann zum Beispiel durch die Verwendung zertifizierter Hardware (Chipkarte, HSM) gewährleistet werden.

Ebenso müssen die öffentlichen Schlüssel von als vertrauenswürdig erkannten Stellen (Vertrauensanker) manipulationssicher gespeichert werden.

4.2 Umgang mit Ephemer-Schlüsseln

Wenn eine Cipher-Suite mit Perfect Forward Secrecy verwendet wird, sollte sichergestellt werden, dass alle Ephemer-Schlüssel nach ihrer Verwendung unwiderruflich gelöscht werden, und keine Kopien dieser Schlüssel erzeugt wurden. Ephemer- bzw. Sitzungsschlüssel sollten nur für *eine* Verbindung benutzt werden und grundsätzlich nicht persistent abgespeichert werden.

4.3 Zufallszahlen

Für die Erzeugung von Zufallszahlen, zum Beispiel für kryptographische Schlüssel oder Signaturerzeugung, müssen geeignete Zufallszahlengeneratoren eingesetzt werden.

Empfohlen wird ein Zufallszahlengenerator aus einer der Klassen DRG.3, DRG.4, PTG.3 oder NTG.1 gemäß [AIS 20/31], vgl. auch Kapitel 9 in Teil 1 dieser Technischen Richtlinie [TR-02102-1].

5 Quellenverzeichnis

Abk.	*Quelle*
AIS 20/31	BSI: AIS 20/31 – A proposal for: Functionality classes for random number generators, September 2011
BDF14	K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Pironti, P.-Y. Strub: Triple Handshake and Cookie Cutters: Breaking and Fixing Authentication over TLS, IEEE Symposium on Security and Privacy, 2014
CRIME	J. Rizzo, Th. Duong: The CRIME attack, http://www.ekoparty.org/2012/thai-duong.php, September 2012
IANA	IANA: http://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml, abgerufen am 08.12.2016
RFC2246	IETF: T. Dierks, C. Allen: RFC 2246, The TLS Protocol Version 1.0, Januar 1999
RFC4346	IETF: T. Dierks, E. Rescorla: RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1, April 2006
RFC5246	IETF: T. Dierks, E. Rescorla: RFC 5246, The Transport Layer Security (TLS) Protocol Version 1.2, August 2008
RFC5289	IETF: E. Rescorla: RFC 5289, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), August 2008
RFC5487	IETF: M. Badra: RFC 5487, Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode, März 2009
RFC5489	IETF: M. Badra, I. Hajjeh: RFC 5289, ECDHE_PSK Cipher Suites for Transport Layer Security (TLS), März 2009
RFC5639	IETF: M. Lochter, J. Merkle: RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, März 2010
RFC5746	IETF: E. Rescorla, M. Ray, S. Dispensa, N. Oskov: RFC 5746, Transport Layer Security (TLS) Renegotiation Indication Extension, Februar 2010
RFC6066	IETF: D. Eastlake 3rd: RFC 6066, Transport Layer Security (TLS) Extensions: Extension Definitions, Januar 2011
RFC6176	IETF: S. Turner, T. Polk: RFC 6176, Prohibiting Secure Sockets Layer (SSL) Version 2.0, März 2011
RFC7027	IETF: M. Lochter, J. Merkle: RFC 7027, Elliptic Curve Cryptography (ECC) Brainpool Curves for Transport Layer Security (TLS), Oktober 2013
RFC7627	IETF: K. Bhargavan, A. Delignat-Lavaud, A. Pironti, A. Langley, M. Ray: RFC 7627, Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension, September 2015
RFC7919	IETF: D. Gillmor: RFC 7919, Negotiated Finite Field Diffie-Hellman Ephemeral

	Parameters for Transport Layer Security (TLS), August 2016
SBK17	M. Stevens, E. Bursztein, P. Karpman, A. Albertini, Y. Markov: The first collision for full SHA-1. IACR Cryptology ePrint Archive, Report 2017/190, 2017.
SKP15	Marc Stevens, Pierre Karpman, Thomas Peyrin: Freestart collision for full SHA-1, EUROCRYPT 2016, Lecture Notes in Computer Science, vol. 9665, Springer, 2016, pp. 459-483
SSLv2	Netscape: Hickman, Kipp: "The SSL Protocol", April 1995
SSLv3	Netscape: A. Frier, P. Karlton, P. Kocher: "The SSL 3.0 Protocol", 1996
TR-02102-1	BSI: Technische Richtlinie TR-02102-1, Kryptographische Verfahren: Empfehlungen und Schlüssellängen
TR-03116-4	BSI: TR-03116-4, Kryptographische Vorgaben für Projekte der Bundesregierung, Teil 4: Kommunikationsverfahren in Anwendungen