

Recommended TLS Cipher-Suits

(BSI TR-02102-2)

Content

1	Introduction	3
1.1	Initial Assessment	3
2	SSL/TLS versions	3
3	Recommendations for TLS 1.2	3
3.1	(EC)DHE cipher suites	3
3.2	Diffie-Hellman groups	4
3.3	Signature algorithms	4
3.4	Further recommendations	5
4	Recommendations for TLS 1.3	5
4.1	Diffie-Hellman groups	5
4.2	Signature algorithms	6
4.3	Cipher suites	7
5	Key lengths	7
6	References	7

1 Introduction

This document is a partly excerpt from the technical guidelines by the BSI. Parts of these technical recommendation have been left out because the specific technologies are not used in the product. Some commentary was added to explain the decisions that are based on the technical guidelines.

The recommendations in the Technical Guideline are suitable to reach the security level of at least 100 bit. If only algorithms and key lengths are used that are recommended until 2028 a security level of 120 bit is reached.

The prediction period for the recommendations at hand is 7 years. Appropriate recommendations for larger periods, as they can be found in other publicly available documents, are naturally very hard to make because future cryptographic developments cannot be predicted precisely for larger periods. In such cases, these recommendations contain parameters and key lengths that might exceed those given in this Technical Guideline.

1.1 Initial Assessment

- Only **TLS 1.2 and 1.3** should be used
- Only **key lengths and algorithms** that are **recommended until 2028** should be used
- Only **cipher suits** that support **perfect forward secrecy** should be used

2 SSL/TLS versions

- In general, **TLS 1.2** or **TLS 1.3** should be used
- TLS 1.0 and TLS 1.1 are not recommended
- SSL v2 and SSL v3 may not be used (see also RFC6176 and RFC7568).

3 Recommendations for TLS 1.2

We generally recommend using **only TLS cipher suits that provide perfect forward secrecy (PFS)** unless there is a technical reason to choose a cipher suit without PFS.

3.1 (EC)DHE cipher suites

Cipher suite	IANA no.	Specified in	Use up to
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	0xC0,0x23	RFC5289	2028+
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	0xC0,0x24	RFC5289	2028+
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xC0,0x2B	RFC5289	2028+
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	0xC0,0x2C	RFC5289	2028+
TLS_ECDHE_ECDSA_WITH_AES_128_CCM	0xC0,0xAC	RFC7251	2028+
TLS_ECDHE_ECDSA_WITH_AES_256_CCM	0xC0,0xAD	RFC7251	2028+
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	0xC0,0x27	RFC5289	2028+
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	0xC0,0x28	RFC5289	2028+
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	0xC0,0x2F	RFC5289	2028+
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	0xC0,0x30	RFC5289	2028+
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	0x00,0x40	RFC5246	2028+
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256	0x00,0x6A	RFC5246	2028+
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	0x00,0xA2	RFC5288	2028+
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	0x00,0xA3	RFC5288	2028+
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	0x00,0x67	RFC5246	2028+

TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	0x00,0x6B	RFC5246	2028+
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	0x00,0x9E	RFC5288	2028+
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	0x00,0x9F	RFC5288	2028+
TLS_DHE_RSA_WITH_AES_128_CCM	0xC0,0x9E	RFC6655	2028+
TLS_DHE_RSA_WITH_AES_256_CCM	0xC0,0x9F	RFC6655	2028+

1 Recommended cipher suites for TLS 1.2 with Perfect Forward Secrecy

Note: The use of cipher suites with CBC mode is only recommended in conjunction with the TLS extension “Encrypt-then-MAC” as soon as suitable implementations are available. (Greyed out cipher suits)

3.2 Diffie-Hellman groups

For cipher suites of type TLS_DHE_* or TLS_ECDHE_*, the client can use the extension “supported_groups” (formerly also called “elliptic_curves”) to inform the server about the Diffie-Hellman groups he wants to use (see RFC7919 for DHE and RFC8422 for ECDHE).

The use of the extension “supported_groups” for TLS_ECDHE_* cipher suites is recommended.

The use of the extension “supported_groups” for TLS_DHE_* cipher suites is recommended as soon as suitable implementations are available.

Diffie-Hellman group	IANA no.	Specified in	Use up to
secp256r1	23	RFC8422	2028+
secp384r1	24	RFC8422	2028+
secp521r1	25	RFC8422	2028+
brainpoolP256r1	26	RFC7027	2028+
brainpoolP384r1	27	RFC7027	2028+
brainpoolP512r1	28	RFC7027	2028+
ffdhe2048	256	RFC7919	2022
ffdhe3072	257	RFC7919	2028+
ffdhe4096	258	RFC7919	2028+

2 Recommended Diffie-Hellman groups for TLS 1.2

In general the technical guidelines by the BSI recommend to use the Brainpool curves.

3.3 Signature algorithms

In TLS 1.2, the client can use the extension “signature_algorithms” (see RFC5246) to inform the server about the signature algorithms he wants to use for key agreement and certificates. The algorithm has to be specified as combination of signature algorithm and hash function.

The use of the extension “signature_algorithms” is recommended.

The use of the following signature algorithms is recommended:

Diffie-Hellman group	IANA no.	Specified in	Use up to
rsa	1	RFC5246	2025
dsa	2	RFC5246	2028+
ecdsa	3	RFC5246	2028+

3 Recommended signature algorithms for TLS 1.2

The use of the signature algorithm rsa (IANA no. 1) is recommended only up to 2025, because it uses the PKCS #1 v1.5 padding scheme.

The use of the following hash functions (combined with a signature algorithm in the table above) is recommended:

Hash function	IANA no.	Specified in	Use up to
sha256	4	RFC5246	2028+
sha384	5	RFC5246	2028+
sha512	6	RFC5246	2028+

4 Recommended hash functions for signature algorithms in TLS 1.2

3.4 Further recommendations

- Session renegotiation
 - It is recommended to use session renegotiation only on the basis of RFC5746.
 - Renegotiation initiated by the client should be rejected by the server.
- Truncated HMAC output
 - The extension "truncated_hmac" defined in RFC6066 to truncate the HMAC output to 80 bits **should not be used**.
- TLS compression and the CRIME attack
 - In order to prevent this attack, it must be ensured that all data of a data packet come from correct and legitimate connection partners and that the attacker cannot perform a plaintext injection. If this cannot be ensured, it is recommended not to use TLS data compression.
- The Lucky 13 attack / The Encrypt-then-MAC extension (This should be implemented as soon as a trustworthy implementation is available)
 - The use of the TLS extension "encrypt-then-MAC" according to [RFC7366] will be recommended as soon as suitable implementations are available.
- The Heartbeat extension
 - It is **urgently recommended not to use** the Heartbeat extension. If it is still necessary, it should be ensured that the TLS implementation is not susceptible to the Heartbleed bug.
- The Extended Master Secret extension (This should be implemented as soon as a trustworthy implementation is available)
 - Using the TLS extension Extended Master Secret according to RFC7627 is recommended as soon as suitable implementations are available.

4 Recommendations for TLS 1.3

We generally recommend using **only TLS cipher suits that provide perfect forward secrecy (PFS)** unless there is a technical reason to choose a cipher suit without PFS.

4.1 Diffie-Hellman groups

In TLS 1.3, client and server can use the extension "supported_groups" to inform each other about the Diffie-Hellman groups they want to use for (EC)DHE.

The use of the following Diffie-Hellman groups is recommended:

Diffie-Hellman group	IANA no.	Specified in	Use up to
secp256r1	23	RFC8422	2028+
secp384r1	24	RFC8422	2028+
secp521r1	25	RFC8422	2028+
brainpoolP256r1tls13	31	RFC8734	2028+
brainpoolP384r1tls13	32	RFC8734	2028+
brainpoolP512r1tls13	33	RFC8734	2028+
ffdhe2048	256	RFC7919	2022
ffdhe3072	257	RFC7919	2028+
ffdhe4096	258	RFC7919	2028+

5 Recommended Diffie-Hellman groups for TLS 1.3

In general the technical guidelines by the BSI recommend to use the Brainpool curves.

4.2 Signature algorithms

In TLS 1.3, client and server can use the extensions “signature_algorithms” and “signature_algorithms_cert” to inform each other about the signature algorithms they want to use for certificate-based authentication. The extension “signature_algorithms” refers to signatures which are generated by client or server for their CertificateVerify message and the extension “signature_algorithms_cert” refers to signatures in certificates.

The use of the following signature algorithms for the extension “signature_algorithms” is recommended:

Signature algorithm	IANA no.	Specified in	Use up to
rsa_pss_rsae_sha256	0x0804	RFC8446	2028+
rsa_pss_rsae_sha384	0x0805	RFC8446	2028+
rsa_pss_rsae_sha512	0x0806	RFC8446	2028+
rsa_pss_pss_sha256	0x0809	RFC8446	2028+
rsa_pss_pss_sha384	0x080A	RFC8446	2028+
rsa_pss_pss_sha512	0x080B	RFC8446	2028+
ecdsa_secp256r1_sha256	0x0403	RFC8446	2028+
ecdsa_secp384r1_sha384	0x0503	RFC8446	2028+
ecdsa_secp521r1_sha512	0x0603	RFC8446	2028+
ecdsa_brainpoolP256r1tls13_sha256	0x081A	RFC8734	2028+
ecdsa_brainpoolP384r1tls13_sha384	0x081B	RFC8734	2028+
ecdsa_brainpoolP512r1tls13_sha512	0x081C	RFC8734	2028+

6 Recommended signature algorithms for TLS 1.3 (client/server signatures)

The use of the following signature algorithms for the extension “signature_algorithms_cert” is recommended:

Signature algorithm	IANA no.	Specified in	Use up to
rsa_pkcs1_sha256	0x0401	RFC8446	2025
rsa_pkcs1_sha384	0x0501	RFC8446	2025
rsa_pkcs1_sha512	0x0601	RFC8446	2025
rsa_pss_rsae_sha256	0x0804	RFC8446	2028+
rsa_pss_rsae_sha384	0x0805	RFC8446	2028+
rsa_pss_rsae_sha512	0x0806	RFC8446	2028+
rsa_pss_pss_sha256	0x0809	RFC8446	2028+
rsa_pss_pss_sha384	0x080A	RFC8446	2028+
rsa_pss_pss_sha512	0x080B	RFC8446	2028+
ecdsa_secp256r1_sha256	0x0403	RFC8446	2028+
ecdsa_secp384r1_sha384	0x0503	RFC8446	2028+
ecdsa_secp521r1_sha512	0x0603	RFC8446	2028+
ecdsa_brainpoolP256r1tls13_sha256	0x081A	RFC8734	2028+
ecdsa_brainpoolP384r1tls13_sha384	0x081B	RFC8734	2028+
ecdsa_brainpoolP512r1tls13_sha512	0x081C	RFC8734	2028+

7 Recommended signature algorithms for TLS 1.3 (signatures in certificates)

The use of the signature algorithms rsa_pkcs1_* (IANA no. 0x0401, 0x0501, and 0x0601) is recommended only up to 2025, because they use the PKCS #1 v1.5 padding scheme.

4.3 Cipher suites

Cipher suite	IANA no.	Specified in	Use up to
TLS_AES_128_GCM_SHA256	0x13,0x01	RFC8446	2028+
TLS_AES_256_GCM_SHA384	0x13,0x02	RFC8446	2028+
TLS_AES_128_CCM_SHA256	0x13,0x04	RFC8446	2028+

8 Recommended cipher suites for TLS 1.3

5 Key lengths

Only algorithms and key lengths that are recommended until 2028 should be used.

Algorithm	Minimum key length	Use up to
Signature keys for certificates and key agreement		
ECDSA	250 bit	2028+
DSS	2000 bit	2022
	3000 bit	2028+
RSA	2000 bit	2022
	3000 bit	2028+
Static und ephemeral Diffie-Hellman keys		
ECDH	250 bit	2028+
DH	2000 bit	2022
	3000 bit	2028+

9 Recommended minimum key lengths for the TLS handshake protocol

6 References

BSI: TR 02102-2, Cryptographic Mechanisms: Recommendations and Key Lengths - Part 2 Use of Transport Layer Security (TLS), Version 2019-01, 22.02.2019, Bundesamt für Sicherheit in der Informationstechnik.