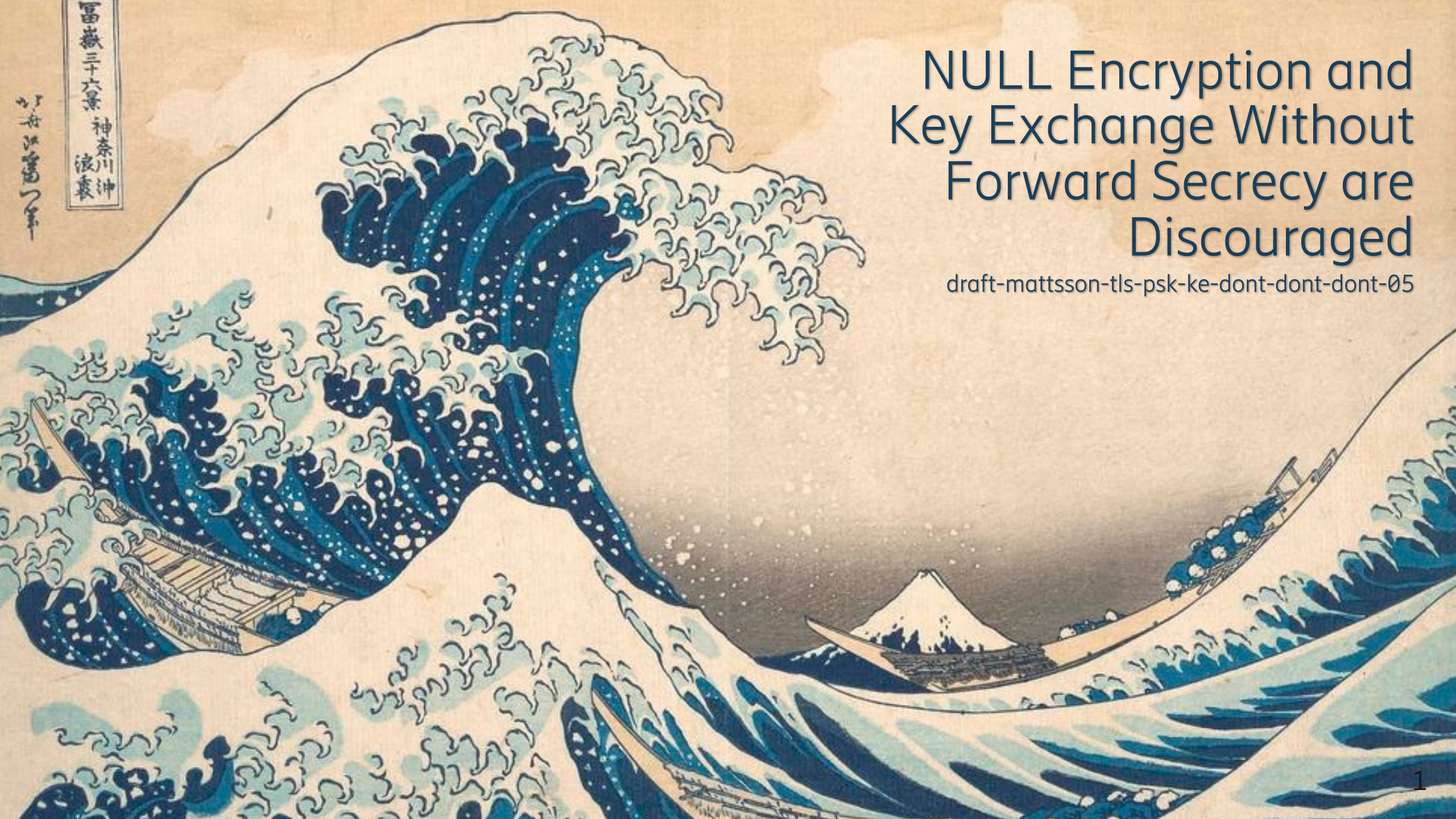

NULL Encryption and Key Exchange Without Forward Secrecy are Discouraged

draft-mattsson-tls-psk-ke-dont-dont-dont-05

draft-mattsson-tls-psk-ke-dont-dont-dont-05

- The draft evaluates TLS pre-shared key exchange modes, (EC)DHE groups, signature algorithms, and cipher suites and proposes to downgrade many entries to "N" and "D" where "D" indicates that the entries are "Discouraged".
- **Background:** RFC 8447 added a Recommended column to many of the TLS registries. The Recommended column did originally non-normatively indicate parameters that are generally recommended for implementations to support.

  The meaning of the column is changed by RFC8447bis to indicate that the IETF has consensus that the item is RECOMMENDED, i.e., using normative RFC 2119 language. RFC8447bis also introduces a third value "D" indicating that an item is discouraged and SHOULD NOT or MUST NOT be used.

  This means that all current values need to be re-evaluated. The current values also need to be re-evaluated as attacks, government requirements, and best practices have changed in the more than 5 years since RFC 8446 and RFC 8447 were published.
- **This document is very much related to RFC8447bis and draft-ietf-tls-deprecate-obsolete-kex.**

Key exchange without forward secrecy enables passive monitoring

- Key exchange without forward secrecy enables passive monitoring.
- Malicious actors can get access to long-term keys in different ways: physical attacks, hacking, social engineering attacks, espionage, or by simply demanding access to keying material with or without a court order.
- Exfiltration attacks are a major cybersecurity threat.

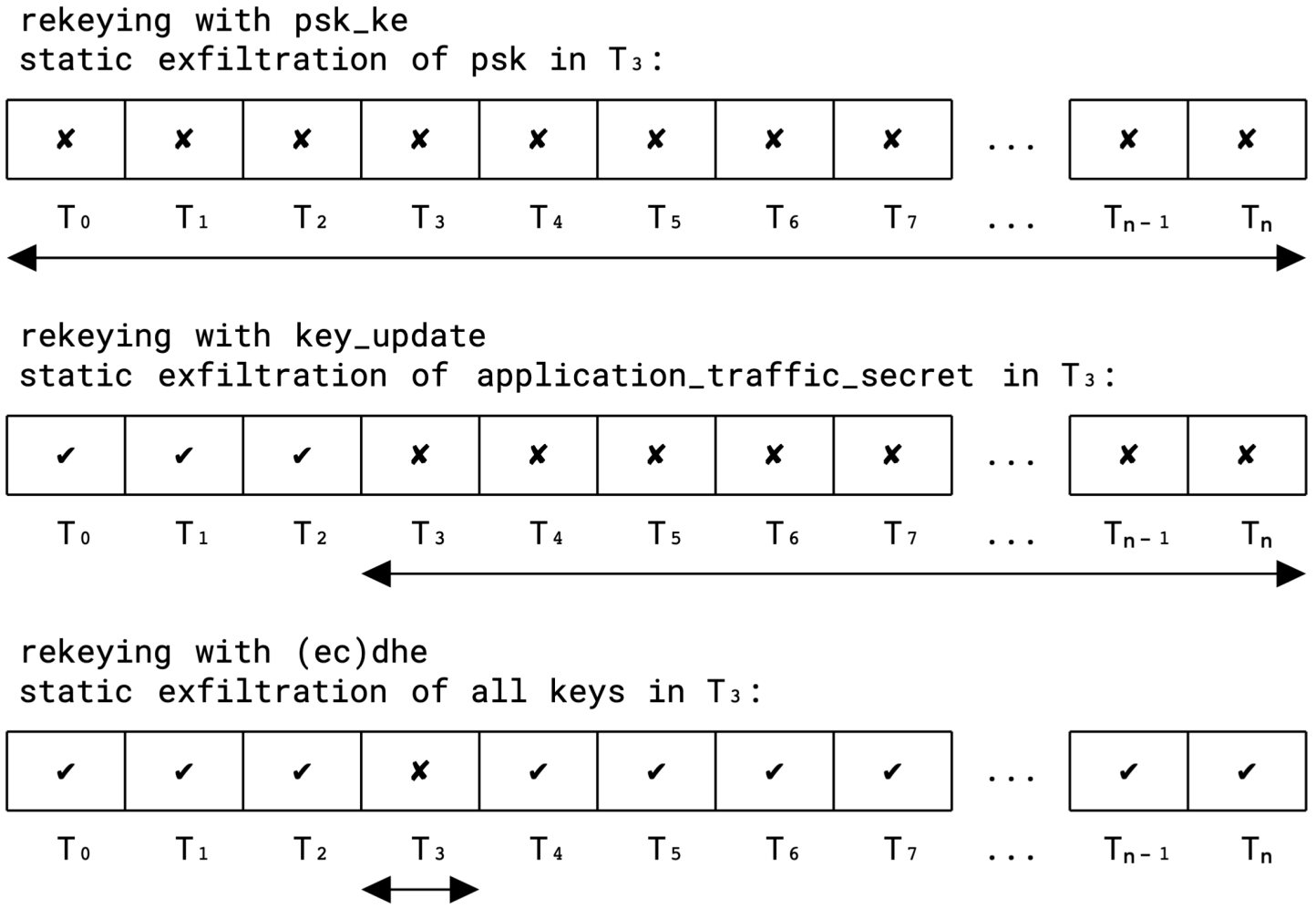

Figure 1: Impact of static key exfiltration in time period T3

Key exchange without forward secrecy

- Static RSA and DH key exchange are forbidden in TLS 1.3.
- ANSSI states that for all versions of TLS: *"The perfect forward secrecy property must be ensured".*
- BSI states regarding psk_ke that *"This mode should only be used in special applications after consultation of an expert."* and has set a deadline that use is only allowed until 2026.
- The HTTP/2 specification RFC 7540 from 2015 prohibits all cipher suites that do not offer ephemeral key exchange.
- TLS implementations like BoringSSL have chosen to not implement psk_ke for security reasons.
- Key exchange without forward secrecy is not adhering to the zero trust principles of assuming breach and to minimize impact when breach occur.
- Modern ephemeral key exchange algorithms like x25519 are very fast and have small message overhead. Kyber is even faster.

- **Proposal to set the "Recommended" value of psk_ke to "D".**
- **Proposal to set the "Recommended" value of all cipher suites that do not offer ephemeral key exchange to "D".**

Description	Recommended
psk_ke	D

Table 1: Downgraded TLS PSK Key Exchange Modes

Cipher suites with NULL encryption

- Cipher suites with NULL encryption were completely removed from TLS 1.3.
- Unfortunately, the independent stream document RFC 9150 reintroduced cipher suites with NULL Encryption in TLS 1.3 even though NULL encryption violates several of the fundamental TLS 1.3 security properties, namely "Protection of endpoint identities", "Confidentiality", and "Length concealment".
- Modern encryption algorithms like AES-GCM are very fast and have small message overhead. The upcoming algorithm AEGIS is even faster.
- Enterprise networks also require encryption. NIST states as the first basic assumption for network connectivity for any organization that utilizes zero trust is that: "*The entire enterprise private network is not considered an implicit trust zone. Assets should always act as if an attacker is present on the enterprise network, and communication should be done in the most secure manner available. This entails actions such as authenticating all connections and encrypting all traffic.*"

- **Proposal to set the "Recommended" value of all cipher suites with NULL encryption to "D".**

Discouraged cipher suites

- The HTTP/2 specification RFC 7540 from 2015 lists and prohibits all cipher suites that do not offer an ephemeral key exchange and those that are based on the TLS null, stream, or block cipher type.
- draft-ietf-tls-deprecate-obsolete-kex lists all _DHE_ cipher suites. The document does not analyze _DHE_ cipher suites. Should follow draft-ietf-tls-deprecate-obsolete-kex. But some draft needs to update the "Recommended" value.
- TLS_SHA256_SHA256 , TLS_SHA384_SHA384 use NULL TLS_PSK_WITH_CHACHA20_POLY1305_SHA256 was registered after RFC 7540 was published and are not listed in draft-ietf-tls-deprecate-obsolete-kex.
- **Proposal to set the "Recommended" value of all cipher suites listed in Appendix A of [RFC9113], all cipher suites listed in Appendix A of [draft-ietf-tls-deprecate-obsolete-kex], TLS_SHA256_SHA256 , TLS_SHA384_SHA384, TLS_PSK_WITH_CHACHA20_POLY1305_SHA256 to "D".**

Description	Recommended
TLS_SHA256_SHA256	D
TLS_SHA384_SHA384	D
TLS_PSK_WITH_CHACHA20_POLY1305_SHA256	D

Table 2: Downgraded TLS Cipher Suites

Key exchange with less than 128-bit security

— Following NIST SP 800-57 cryptographic protection with 112-bit algorithms is only allowed if the application data does not have to be protected after 2030.

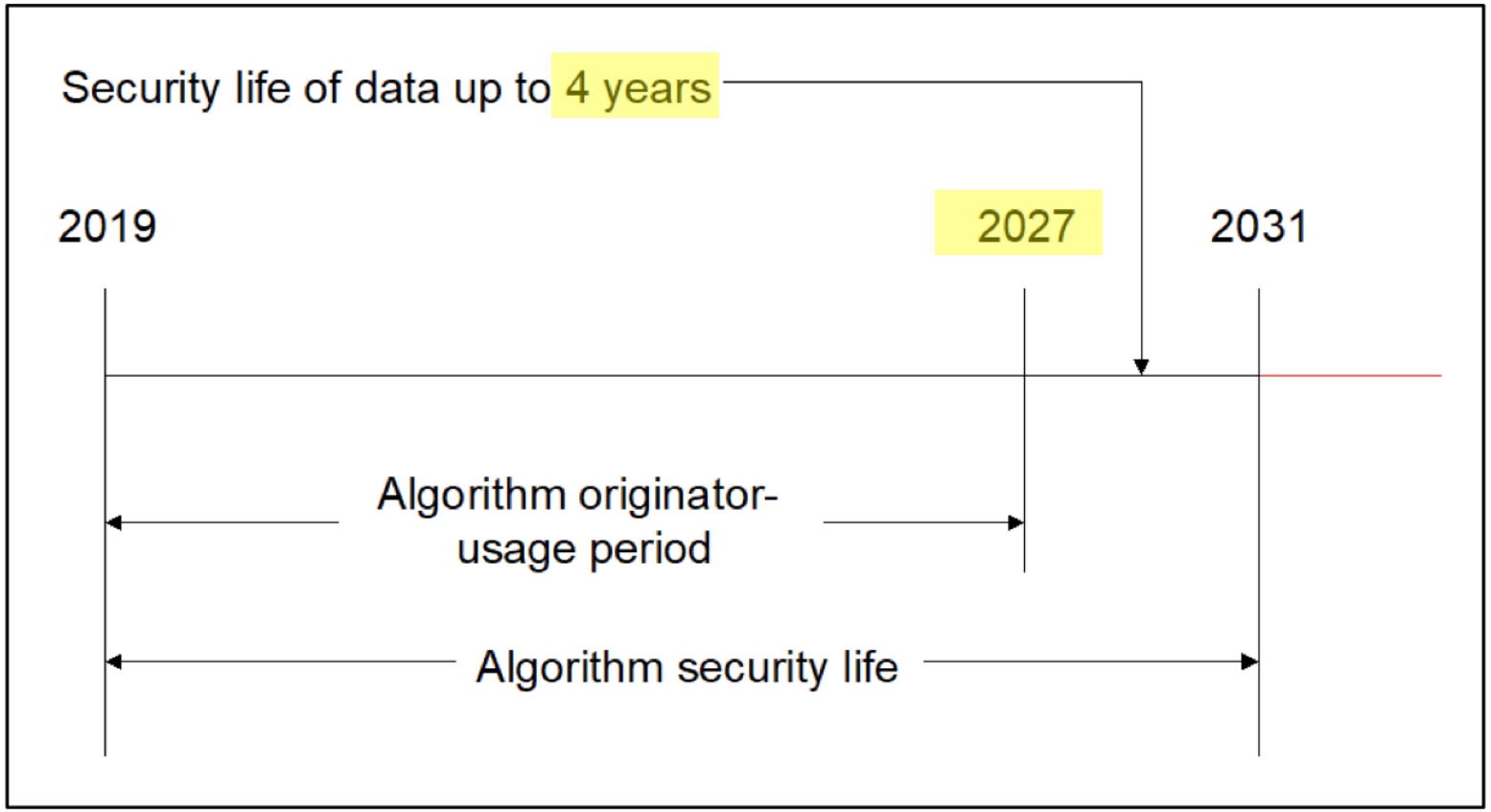

Figure 2: Algorithm originator-usage period example

Key exchange with less than 128-bit security

- Government organizations like NIST, ANSSI, BSI, and NSA have already produced recommendations regarding the deprecation of key exchange algorithms with less than 128-bit security such as ffdhe2048:
  - NIST [NIST-Lifetime] and ANSSI [ANSSI-TLS] only allow 2048-bit Finite Field Diffie-Hellman if the application data does not have to be protected after 2030. If the application data had a security life of ten years, NIST and ANSSI allowed use of ffdhe2048 until December 31, 2020.
  - [BSI] allowed use of ffdhe2048 up to the year 2022.
  - The Commercial National Security Algorithm Suite (CNSA) [RFC9151] forbids the use of ffdhe2048.
  - ECDHE groups that offer less than 128-bit security are forbidden to use in TLS 1.3.
  - **Proposal to set the "Recommended" value of secp160k1, secp160r1, secp160r2, sect163k1, sect163r1, sect163r2, secp192k1, secp192r1, sect193r1, sect193r2, secp224k1, secp224r1m sect233k1, sect233r1, and sect239k1, and ffdhe2048 to "D".**

Description	Recommended
sect163k1	D
sect163r1	D
sect163r2	D
sect193r1	D
sect193r2	D
sect233k1	D
sect233r1	D
sect239k1	D
secp160k1	D
secp160r1	D
secp160r2	D
secp192k1	D
secp192r1	D
secp224k1	D
secp224r1	D
ffdhe2048	D

Table 3: Downgraded TLS Supported Groups

Signature algorithms with PKCS #1 v1.5 padding or SHA-1

- **SHA-1 signature algorithms:**
  - [RFC8446] labels rsa_pkcs1_sha1 and ecdsa_sha1 as legacy algorithms which are being deprecated and that endpoints SHOULD NOT or MUST NOT negotiate.
  - Proposal to set the "Recommended" value of rsa_pkcs1_sha1 and ecdsa_sha1 to "D".
- **RSASSA-PKCS1-v1_5 in signed TLS handshake messages:**
  - [RFC8446] forbids the use of RSASSA-PKCS1-v1_5 in signed TLS handshake messages.
  - [I-D.davidben-tls13-pkcs1] registered new RSASSA-PKCS1-v1_5 signature algorithms for use in signed TLS 1.3 handshake messages.
  - Proposal to set the "Recommended" value of rsa_pkcs1_sha256_legacy, rsa_pkcs1_sha384_legacy, and rsa_pkcs1_sha512_legacy to "D".
- **RSASSA-PKCS1-v1_5 in certificates:**
  - The RSA Cryptography Specifications [RFC8017] specifies that "RSASSA-PSS is REQUIRED in new applications: "RSASSA-PKCS1-v1_5 is included only for compatibility with existing applications."
  - [BSI] allows use of the PKCS #1 v1.5 padding scheme in certificates up to the year 2025.
  - The Commercial National Security Algorithm (CNSA) [RFC9151] requires offer of rsa_pkcs1_sha384 in certificates.
  - Proposal to set the "Recommended" value of rsa_pkcs1_sha256, rsa_pkcs1_sha384, and rsa_pkcs1_sha512 to "N".

Description	Recommended
rsa_pkcs1_sha1	D
ecdsa_sha1	D
rsa_pkcs1_sha256	N
rsa_pkcs1_sha256_legacy	D
rsa_pkcs1_sha384	N
rsa_pkcs1_sha384_legacy	D
rsa_pkcs1_sha512	N
rsa_pkcs1_sha512_legacy	D

Table 4: Downgraded TLS Signature Schemes