EUROPEAN COMMISSION

DIRECTORATE-GENERAL FOR INFORMATICS

eDelivery AS4 Security Profile

Cryptography Related Recommendations (January 2022)

Commission européenne/Europese Commissie, 1049 Bruxelles/Brussel, BELGIQUE/BELGIË – Tel. +32 22991111

DOCUMENT HISTORY

Date	Version	Modification	Author
7 January 2022	1.0	Initial text	Prof. Panagiotis Rizomiliotis, Harokopio University of Athens

EXECUTIVE SUMMARY

In this deliverable, we propose a portfolio of cryptographic algorithms and protocols that can be used to configure signing and encryption of eDelivery AS4 messages ([1]). We also investigate the compatibility of our proposals with the cryptographic recommendations from:

- Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446)
- eIDAS Cryptographic Requirements for the Interoperability Framework (Version 1.2, 8/2019)

Our goal is to propose one scheme per category, when possible. Table 1 summarizes our recommendations.

Table 1. Recommendations summary

Algorithm/Scheme	Proposal	TLS 1.3	eIDAS
Block Cipher	AES	✓	✓
Hash Function	SHA-2 family	✓	✓
AEAD	GCM/AES	✓	✓
Key Derivation Function	HKDF	✓	×
MAC	HMAC/SHA-2	Part of HKDF only	×
Digital Signature	EdDSA	✓	×
Certificate verification	RSASSA-PSS, RSA PKCS1-v1_5, ECDSA	✓	✓
Key Agreement	ECDHE	✓	×
Key Transport	RSAES-OAEP	×	✓
Key Encapsulation	ECIES-KEM	×	×

([1]) https://ec.europa.eu/digital-building-blocks/wikis/x/RqbXGw

1. SYMMETRIC CRYPTOGRAPHIC PRIMITIVES AND SCHEMES

1.1. Block Ciphers

A block cipher is the main symmetric key primitive. It is modelled as a keyed pseudo-random permutation. The Advanced Encryption Standard (AES) is by far the most acceptable solution for current and future systems.

Proposal:

- *AES*
    - Key length: *at least 128 bits*

Related Standard:

- Federal Information Processing Standards Publication 197. Advanced encryption standard (AES). National Institute of Standards and Technology, 2001. https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

Compliance:

	TLS 1.3	eIDAS
AES	✓	✓

1.2. Hash functions

A hash function supports a variety of cryptographic properties (pre-image security, 2nd pre-image security, collision resistance) without using a secret key. It is modelled as a pseudo-random function (PRF). The security of the scheme must solely depend on the output length. While there are nice schemes, like the SHA-3 competition finalist, Blake2, the old NIST standard SHA-2 is dominant in the field and it is expected to remain for at least a decade. However, we cannot ignore the new standard SHA-3 as a secure and efficient alternative.

Proposal:

- *SHA2 and SHA3 families* of hash functions
    - Output Length: *256, 384, 512 bits*

Related Standards:

- Federal Information Processing Standards Publication FIPS 202. SHA-3 standard: Permutation-based hash and extendable-output functions (draft). National Institute of Standards and Technology, 2014. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf
- Federal Information Processing Standards Publication FIPS 180-4, Secure Hash Standard (SHS), NIST. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

Compliance:

	TLS 1.3	eIDAS
SHA2	✓	✓
SHA3	✓	✖

1.3. Authenticated Encryption (with associated data)

Authenticated Encryption with Associated Data (AEAD) schemes are symmetric key protocols that aim to provide data confidentiality and integrity. It has been proposed to accelerate the more modular Encrypt-then-MAC solution in which the confidentiality and the integrity protection protocols are applied sequentially (and in order to avoid common mistakes, like MAC-then Encrypt or Encrypt and MAC). There is a specific block cipher mode of operation that has been extensively used by the industry and it is expected to be the main choice for the following years. It is called Galois-counter mode (GCM).

As an alternative solution, ChaCha20 with Poly1305 has been included in the latest TLS standard. It is considered more secure and more efficient than the GCM mode. However, GCM will remain the main choice for the following years as it is widely deployed.

Note: *Confidentiality-only schemes, like block cipher-based counter or CBC modes, must be avoided*.

Proposal:

- Block-cipher based: *GCM with AES*
    - Key length: *at least 128 bits*

Related Standards:

- NIST Special Publication 800-38D. Recommendation for block cipher modes of operation – Galois/Counter Mode (GCM) and GMAC. National Institute of Standards and Technology, 2007. https://csrc.nist.gov/publications/detail/sp/800-38d/final

Compliance:

	TLS 1.3	eIDAS
GCM/AES	✓	✓

1.4. Key derivation function

Key Derivation Functions (KDFs) are cryptographic schemes that are used to derive cryptographic keys from a source of keying material. This keying material can be either the result of a key agreement protocol or the output of a key generation source (it can be shared using key encapsulation or transport). There are several options, mainly leveraging a message authenticating code (MAC) scheme. The main state-of-the-art solution is HKDF that it is based on HMAC.

Proposal:

- *HKDF*
    - Hash function: *collision resistant*
    - Hash function output: *at least 256 bits*

Related Standards:

- HMAC-based Extract-and-Expand Key Derivation Function (HKDF), RFC 5869, May 2010, DOI 10.17487/RFC5869. https://tools.ietf.org/html/rfc5869

Compliance:

	TLS 1.3	eIDAS
HKDF	✓	✖

1.5. Message Authentication Codes

Message Authentication Codes (MACs) are symmetric key protocols that aim to protect data integrity. The protected message is appended with a tag produced by the MAC function. The security of a MAC is determined by the secret key size and the tag's length.

While integrity only channels are not usually supported, a MAC scheme can serve several different goals in a secure protocol. It can be part of a key derivation function, it can bind an endpoint's identity to the exchanged keys after key agreement or it can authenticate a session in a pre-shared key (PSK) scenario.

Proposal:

- Block-cipher based: *CMAC with AES*
- Hash-based: *HMAC with SHA1, SHA2 and SHA3 families* of hash functions.
- Key length: *at least 128 bits*
- Output Length: *at least 128 bits*

Related Standards:

- ISO/IEC 9797-1:2011. Information technology – Security techniques – Digital signatures giving message recovery – Part 1: Mechanisms using a block cipher. International Organization for Standardization, 2011.
- ISO/IEC 9797-2:2011. Information technology – Security techniques – Digital signatures giving message recovery – Part 2: Mechanisms using a dedicated hash-function. International Organization for Standardization, 2011.
- NIST Special Publication 198-1. The keyed-hash message authentication code (HMAC). National Institute of Standards and Technology, 2008.

Compliance:

	TLS 1.3	eIDAS
CMAC/AES	✖	✖
HMAC/SHA-2, SHA-3	As part of the key derivation function	✖

2. Asymmetric (Public Key) Cryptographic Primitives and Schemes

2.1. Digital Signatures

Digital signatures are used both for message and user authentication. The user authentication protocol is usually combined with the key exchange protocol. When the public key distribution is based on PKI, the signature of the certificate must be verified. Regarding the EC-based schemes, EdDSA is more efficient and more secure than ECDSA. However, Certification Authorities tend to use more legacy solutions, like RSA-based schemes. For compatibility, we must consider supporting alternative signature schemes for certificate verification only (see below).

Proposal:

- *EdDSA*
    - Key length: *at least 256 bits* (*security at least 128 bits*)
    - Elliptic Curves: *ed25519* (*128 bits security*), *ed448* (*224 bits security*)
    - Hash function: —
    - Usage: certificate verification and message authentication

Related Standards:

- Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017. https://www.rfc-editor.org/info/rfc8032

Compliance:

	TLS 1.3	eIDAS
EdDSA	✓	✖

2.2. Certificate Verification Only

Digital signatures are used for digital certificate verification. These certificates are used for server/client authentication and in some cases for achieving a more ambitious security goal, like non-repudiation. Many certification authorities (CA) are still using more traditional signature schemes for signing the certificates that they produce, like legacy RSA-based solutions.

Unfortunately, several CAs haven't replaced vulnerable hash functions (like SHA-1). All the certificates that are using weak hash functions must be rejected. Ideally, only the hash functions that belong in the SHA-2 and SHA-3 families must be accepted with the recommended output sizes (see above). Regarding the signature scheme, we provide a list of popular and secure schemes that are still supported by the CAs.

Certificate management is a crucial and complicated issue and it is out of the scope of this document. NIST's Cybersecurity Guide provides best practices in the field:

- SP 1800-16. Securing Web Transactions: TLS Server Certificate Management. June 2020. https://csrc.nist.gov/publications/detail/sp/1800-16/final

Proposal:

- *RSASSA-PSS* and *RSASSA-PKCS1-v1_5*
    - Key length: *at least 3072 bits* (*security at least 128 bits*)
    - Hash function: *SHA-2*
    - Hash function output: *at least 256* (*384*, *512*)
    - Usage: mainly certificate verification (exclude message authentication)
- *ECDSA*
    - Key length: *at least 256 bits* (*security at least 128 bits*)
    - Elliptic Curves: *secp256r1* (*128 bits security*), *secp384r1* (*192 bits security*), and *secp521r1* (*256 bits security*)
    - Hash function: *SHA-2*
    - Hash function output: *at least 256*
    - Usage: certificate verification and message authentication

Related Standards:

- PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016. https://www.rfc-editor.org/info/rfc8017
- American National Standards Institute, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)",

ANSI ANS X9.62-2005, November 2005. ANSI X9.62 - Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA) | Engineering360 (globalspec.com)

- FIPS 186-4, Digital Signature Standard (DSS)
- FIPS 186-5 (Draft), Digital Signature Standard (DSS)

Compliance:

	TLS 1.3	eIDAS
RSASSA-PSS and PKCS1-v1_5	✓	✓
ECDSA	✓	✓

2.3. Key Agreement/Exchange

The elliptic curve-based version of the Diffie-Hellman protocol is used for key agreement. The DH protocol must be combined with an authentication protocol and must be combined with a key derivation function.

Proposal:

- *ECDHE*
    - Key length: *at least 256 bits*
    - Elliptic Curves:
        - *secp256r1* (*128 bits security*), *secp384r1* (*192 bits security*), and *secp521r1* (*256 bits security*)
        - *X25519* (*128 bits security*) and *X448* (*224 bits security*)

Related Standards:

- "IEEE Standard Specifications for Public Key Cryptography", IEEE Std. 1363-2000, DOI 10.1109/IEEESTD.2000.92292. (inactive)
- Recommendations for Discrete Logarithm-Based Cryptography: Elliptic Curve Domain Parameters. SP 800-186 (Draft). https://doi.org/10.6028/NIST.SP.800-186-draft
- Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016. Information on RFC 7748 » RFC Editor (rfc-editor.org)

Compliance:

	TLS 1.3	eIDAS
ECDHE	✓	✖

2.4. Key Transport and Key Encapsulation

As an alternative to key exchange, either a key transport or a key encapsulation scheme is used. However this approach is not recommended. When key transport is used, it doesn't offer forward security as the session key is protected by a long-term public key.

Key transport proposal:

- RSAES-OAEP
    - Key length: *at least 3072 bits* (*security at least 128 bits*)
    - Hash function: *SHA-2*
    - Hash function output: *at least 256* (*384, 512*)

Related Standards:

- PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016. https://www.rfc-editor.org/info/rfc8017

Compliance:

	TLS 1.3	eIDAS
RSAES-OAEP	✖	✓

Key encapsulation proposal:

- *ECIES-KEM*
    - Key length: *at least 256 bits* (*security at least 128 bits*)
    - KDF function: *HKDF with SHA-2*
    - Hash function output: *at least 256* (*384, 512*)

Related Standards:

- ANSI X9.63. Public key cryptography for the financial services industry – Key agreement and key transport using elliptic curve cryptography. American National Standard Institute, 2011.

Compliance:

	TLS 1.3	eIDAS
ECIES-KEM	✖	✖

3. REFERENCES

American National Standards Institute, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI ANS X9.62-2005, November 2005, ANSI X9.62 - Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA) | Engineering360 (globalspec.com)

ANSI X9.63. Public key cryptography for the financial services industry – Key agreement and key transport using elliptic curve cryptography. American National Standard Institute, 2011.

"IEEE Standard Specifications for Public Key Cryptography", IEEE Std. 1363-2000, DOI 10.1109/IEEESTD.2000.92292. (inactive)

eIDAS Cryptographic Requirements for the Interoperability Framework (Version 1.2, 8/2019)

ENISA, Threat Landscape report, European Union Agency for Network and Information Security, 2014.

HMAC-based Extract-and-Expand Key Derivation Function (HKDF), RFC 5869, May 2010, DOI 10.17487/RFC5869, https://tools.ietf.org/html/rfc5869

ISO/IEC 9797-1:2011. Information technology – Security techniques – Digital signatures giving message recovery – Part 1: Mechanisms using a block cipher. International Organization for Standardization, 2011.

ISO/IEC 9797-2:2011. Information technology – Security techniques – Digital signatures giving message recovery – Part 2: Mechanisms using a dedicated hash-function. International Organization for Standardization, 2011.

Federal Information Processing Standards Publication FIPS 180-4, Secure Hash Standard (SHS), NIST, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

Federal Information Processing Standards Publication 197. Advanced encryption standard (AES). National Institute of Standards and Technology, 2001, https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

Federal Information Processing Standards Publication FIPS 202. SHA-3 standard: Permutation-based hash and extendable-output functions (draft). National Institute of Standards and Technology, 2014, https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, https://www.rfc-editor.org/info/rfc8032

Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, Information on RFC 7748 » RFC Editor (rfc-editor.org).

NIST Special Publication 198-1. The keyed-hash message authentication code (HMAC). National Institute of Standards and Technology, 2008.

NIST Special Publication 1800-16. Securing Web Transactions: TLS Server Certificate Management. June 2020, https://csrc.nist.gov/publications/detail/sp/1800-16/final

NIST Special Publication 800-38D. Recommendation for block cipher modes of operation – Galois/Counter Mode (GCM) and GMAC. National Institute of Standards and Technology, 2007, https://csrc.nist.gov/publications/detail/sp/800-38d/final

PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, https://www.rfc-editor.org/info/rfc8017

Recommendations for Discrete Logarithm-Based Cryptography: Elliptic Curve Domain Parameters. SP 800-186 (Draft), https://doi.org/10.6028/NIST.SP.800-186-draft

Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446)