17.2. Approved Cryptographic Algorithms

Objective

17.2.1. Information is protected by a properly implemented, Approved Cryptographic Algorithm.

Context

Scope

17.2.2. This section covers cryptographic algorithms that the GCSB recognises as being approved for use within government. Implementations of the algorithms in this section need to have successfully completed an approved cryptographic evaluation before they can be approved to protect information. Correct implementations of cryptographic protocols are checked during system certification.

17.2.3. High assurance cryptographic are not covered in this section.

Approved cryptographic algorithms

17.2.4. There is no guarantee or proof of security of an algorithm against presently unknown attacks. However, the algorithms listed in this section have been extensively scrutinised by government, industry and academic communities in a practical and theoretical setting and have not been found to be susceptible to any feasible attacks. There have been some cases where theoretically impressive vulnerabilities have been found, however these results are not considered to be feasible with current technologies and capabilities.

17.2.5. Where there is a range of possible key sizes for an algorithm, some of the smaller key sizes do not provide an adequate safety margin against attacks that might be found in the future. For example, future advances in number factorisation could render the use of smaller RSA moduli a security vulnerability.

17.2.6. The approved cryptographic algorithms fall into three categories: asymmetric/public key algorithms, hashing algorithms and symmetric encryption algorithms. Collectively these were known as SUITE B and were first promulgated in 2006.

17.2.7. Suite B was superseded by the Commercial National Security Algorithm Suite in August 2015 and later supplemented by the Commercial Solutions for Classified (CSFC) Programme.

17.2.8. Some algorithms that were previously approved in earlier versions of the NZISM are now deprecated. These are still permitted to be used to decrypt or verify previously encrypted or signed files. These algorithms are described as 'for legacy use only' in the NZISM.

17.2.9. The approved asymmetric/public key algorithms are:

- ECDH for agreeing on encryption session keys.
- ECDSA for digital signatures.
- DH for agreeing on encryption session keys. This should only be used for interoperability with third parties where ECDH is not supported.
- RSA for digital signatures and passing encryption session keys or similar keys.
- DSA for digital signatures for legacy use only.

17.2.10. The approved hashing algorithms are:

- Secure Hashing Algorithm 2; and
- Secure Hashing Algorithm 1 for legacy use only.

17.2.11. The approved symmetric encryption algorithms are:

- AES; and
- 3DES for legacy use only.

17.2.12. SHA-1, 3DES and DSA MUST NOT be used for new implementations but are approved only for processing already protected information. These are legacy use only.

17.2.13. Summary Table

Function	Cryptographic algorithm or protocol	Applicable standards	Minimum
Encryption	Advanced Encryption Standard (AES)	FIPS 197	256-bit key
Hashing	Secure Hash Algorithm (SHA)	FIPS 180-4	SHA-384 (SHA-256 IN CONFIDENCE & BELOW only)
Digital signature	Elliptic Curve Digital Signature Algorithm (ECDSA)	FIPS 186-3	NIST P-384
	Rivest-Shamir-Adleman (RSA)	NIST SP 800-56B Rev. 2	3072-bit key (2048-bit key in PKI)
Key exchange	Elliptic Curve Diffie-Hellman (ECDH)	SP 800-56A ANSI X9.63	NIST P-384
	Rivest-Shamir-Adleman		3072-bit key
	Diffie-Helman (DH)	IETF RFC 3526 (Reference m)	3072-bit key

Salting

17.2.14. Salting is a technique of further modifying a hash by adding a value or character string to the start or end of a password. This improves the resistance of the hash to brute-force attacks. To further improve resistance the salt should be cryptographically strong and randomly generated as unique for each password.

17.2.15. The effectiveness of salts is reduced if implemented poorly. Common implementation errors are salts that are too short and the reuse of salts. To implement credential-specific salts the following principles should be followed:

- Generation of a unique salt every time a stored credential is created;
- Generate salts as cryptographically strong random data;
- Use a 32 or 64 bit salt as storage and system constraints permit;
- Implement a security schema that is not dependent on hiding, splitting or otherwise obfuscating the salt; and
- Do NOT apply salts per user or on a system wide basis.

References

17.2.16. The following references are provided for the approved asymmetric/public key algorithms, hashing algorithms and encryption algorithms. Note that Federal Information Processing Standards (FIPS) are standards and guidelines that are developed by the US National Institute of Standards and Technology (NIST) for US Federal computer systems.

Reference	Title	Publisher	Source
	W. Diffie and M. E. Hellman, "New Directions in Cryptography" IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, November 1976, doi: 10.1109/TIT.1976.1055638.	IEEE	https://ee.stanford.edu/~hellman/publications/24.pdf [PDF, 2.1 MB]
RFC 3447	PKCS #1 Public Key Cryptography Standards #1 RSA Laboratories	IETF	https://datatracker.ietf.org/doc/html/rfc3447
RFC 8624	Algorithm Implementation Requirements and Usage Guidance for DNSSEC June 2019	IETF	https://datatracker.ietf.org/doc/html/rfc8624
RFC 3602	The AES-CBC Cipher Algorithm and Its Use with IPsec September 2003	IETF	https://datatracker.ietf.org/doc/html/rfc3602
RFC 5288	AES Galois Counter Mode (GCM) Cipher Suites for TLS August 2008	IETF	https://datatracker.ietf.org/doc/html/rfc5288
RFC 8492	Secure Password Ciphersuites for Transport Layer Security (TLS) February 2019	IETF	https://datatracker.ietf.org/doc/html/rfc8492
RFC 2898	PKCS #5: Password-Based Cryptography Specification Version 2.0 September 2000	IETF	https://datatracker.ietf.org/doc/html/rfc2898
RFC 8018	PKCS #5: Password-Based Cryptography Specification Version 2.1 January 2017	IETF	https://datatracker.ietf.org/doc/html/rfc8018
FIPS 186-4	Digital Signature Standard (DSS) July 2013	NIST	https://csrc.nist.gov/publications/detail/fips/186/4/final
FIPS 197	Advanced Encryption Standard (AES) November 2001 This publication is currently under review (10 June 2021)	NIST	https://csrc.nist.gov/publications/detail/fips/197/final
	Key Management	NIST	https://csrc.nist.gov/projects/key-management/key-management-guidelines
SP 800-56A Rev. 3	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography	NIST	https://doi.org/10.6028/NIST.SP.800-56Ar3 Also ANSI x9.63 and ANSI X9.42
	Key Establishment	NIST	https://csrc.nist.gov/Projects/Key-Management/Key-Establishment Also ANSI X9.63 and ANSI X9.42
FIPS Pub 180-4	Secure Hash Standard (SHS) August 2015	NIST	FIPS 180-4, Secure Hash Standard (SHS) | CSRC (nist.gov)
SP 800-67 Rev. 2	Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher November 2017	NIST	https://csrc.nist.gov/publications/detail/sp/800-67/rev-2/final
FIPS 140-3	Security Requirements for Cryptographic Modules March 2019	NIST	https://csrc.nist.gov/publications/detail/fips/140/3/final

SP 800-56C Rev. 2	Recommendation for Key-Derivation Methods in Key-Establishment Schemes August 2020	NIST	https://csrc.nist.gov/publications/detail/sp/800-56c/rev-2/final
	Block Cipher Techniques	NIST	https://csrc.nist.gov/projects/block-cipher-techniques/bcm
SP 800-38D	Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC November 2007 This publication is under review, August 2021	NIST	https://csrc.nist.gov/publications/detail/sp/800-38d/final
	McGrew, David A. and Viega, John (2005) "The Galois/Counter Mode of Operation (GCM)"	NIST	https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-spec.pdf [PDF, 1 MB]
	Cryptographic Algorithm Validation Program CAVP	NIST	https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program
SP 800-38A	Recommendation for Block Cipher Modes of Operation: Methods and Techniques December 2001 This publication is under review (May 2021)	NIST	https://csrc.nist.gov/publications/detail/sp/800-38a/final
FIPS 180-4	Secure Hash Standard (SHS) August 2015	NIST	https://csrc.nist.gov/publications/detail/fips/180/4/final
SP 800-63	Digital Identity Guidelines	NIST	https://pages.nist.gov/800-63-3/
SP 800-106	Randomized Hashing for Digital Signatures February 2009	NIST	https://csrc.nist.gov/publications/detail/sp/800-106/final
SP 800-107 Rev. 1	Recommendation for Applications Using Approved Hash Algorithms August 2012 This publication is under review (6 August 2021)	NIST	https://csrc.nist.gov/publications/detail/sp/800-107/rev-1/final
SP 800-132	Recommendation for Password-Based Key Derivation: Part 1: Storage Applications December 2021	NIST	https://csrc.nist.gov/publications/detail/sp/800-132/final
	Commercial National Security Algorithm (CNSA) Suite	NSA	CSA_CNSA_2.0_ALGORITHMS_.PDF (defense.gov)
	Commercial National Security Algorithm (CNSA) Suite Factsheet	NSA	CSI_CNSA_2.0_FAQ_.PDF (defense.gov)
	Commercial Solutions for Classified (CSfC) FAQ 2018	NSA	https://www.nsa.gov/Portals/70/documents/resources/everyone/csfc/csfc-faqs.pdf [PDF, 1.15 MB]

Rationale & Controls

Using Approved Cryptographic Algorithms

17.2.17.R.01. **Rationale**

Inappropriate configuration of a product using an Approved Cryptographic Algorithm can inadvertently select relatively weak implementations of the cryptographic algorithms. In combination with an assumed level of security confidence, this can represent a significant security risk.

17.2.17.R.02. **Rationale**

When configuring unevaluated products that implement an Approved Cryptographic Algorithm, agencies should disable any non-approved algorithms. Correct implementation of cryptographic protocols and disabling of non-approved algorithms is checked during system certification.

A less effective control is to advise system users not to use them via a written policy.

17.2.17.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2128]

Agencies MUST ensure that only Approved Cryptographic Algorithms can be used when using an unevaluated product that implements a combination of approved and non-approved Cryptographic Algorithms.

Approved asymmetric/public key algorithms

17.2.18.R.01. **Rationale**

Over the last decade DSA and DH cryptosystems have been subject to increasingly successful sub-exponential factorisation and index-calculus based attacks. ECDH and ECDSA offer more security per bit increase in key size than either DH or DSA and are considered more secure alternatives.

17.2.18.C.01. **Control System Classifications(s): All Classifications; Compliance: Should** [CID:2131]

Agencies SHOULD use ECDH and ECDSA for all new systems, version upgrades and major system modifications.

Using DH

17.2.19.R.01. **Rationale**

While ECDH should be used in preference to DH, there are instances where DH is still in use. A modulus of at least 3072 bits for DH is now considered good practice by the cryptographic community.

17.2.19.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2134]

Agencies using DH, for the approved use of agreeing on encryption session keys, MUST use a modulus of at least 3072 bits.

Equipment using DH

17.2.20.R.01. **Rationale**

If a network device is NOT able to support the required cryptographic protocol, algorithm and key length, the system will be at risk of a cryptographic compromise. In such cases, the longest feasible key length must be implemented and the device scheduled for replacement as a matter of urgency.

17.2.20.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2137]

Devices which are NOT capable of implementing required key lengths MUST be reconfigured with the longest feasible key length as a matter of urgency.

17.2.20.C.02. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2138]

Devices which are NOT capable of implementing required key lengths MUST be scheduled for replacement as a matter of urgency.

Using DSA (for legacy use only)

17.2.21.R.01. **Rationale**

A modulus of at least 1024 bits for DSA is considered good practice by the cryptographic community.

17.2.21.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:7189]

Agencies using DSA, for the approved use of digital signatures, MUST use a modulus of at least 1024 bits.

Using ECDH

17.2.22.R.01. **Rationale**

A field/key size of at least 384 bits for ECDH is now considered good practice by the cryptographic community.

17.2.22.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2144]

Agencies using ECDH, for the approved use of agreeing on encryption session keys, MUST implement the curve P-384 (prime moduli).

17.2.22.C.02. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2145]

All VPN's using an ECDH key length less than 384 MUST replace all Pre-Shared Keys with keys of at least 384 bits, as soon as possible.

Using ECDSA

17.2.23.R.01. **Rationale**

An equivalent symmetric key security strength of at least 160 bits for ECDSA is considered good practice by the cryptographic community. Not all legacy systems support a modulus of this length, in which case significant risk is being carried.

17.2.23.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2148]

Agencies using ECDSA, for the approved use of digital signatures, MUST implement the curve P-384 (prime moduli).

Using RSA

17.2.24.R.01. **Rationale**

A modulus of at least 3072 bits for RSA is considered good practice by the cryptographic community.

17.2.24.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2151]

Agencies using RSA, for the approved use of digital signatures and passing encryption session keys or similar keys, MUST use a modulus of at least 3072 bits.

17.2.24.C.02. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2152]

Agencies using RSA, for the approved use of digital signatures and passing encryption session keys or similar keys, MUST ensure that the public keys used for passing encrypted session keys are different to the keys used for digital signatures.

17.2.24.C.03. **Control System Classifications(s): All Classifications; Compliance: Should** [CID:7181]

Agencies using RSA, for the approved use of digital signatures and passing encryption session keys or similar keys, SHOULD use a modulus of at least 4096 bits.

Public key infrastructure using RSA

17.2.25.R.01. **Rationale**

A modulus of at least 2048 bits for RSA is considered good practice by the cryptographic community for use within X.509 based Public Key Infrastructure (PKI) systems.

17.2.25.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:7186]

Agencies using RSA keys within internet X.509 Public Key Infrastructure certificates MUST use a modulus of at least 2048 bits.

Approved hashing algorithms

17.2.26.R.01. **Rationale**

Recent research conducted by cryptographic community suggests that SHA-1 may be susceptible to collision attacks. While no practical collision attacks have been published for SHA-1, they may become feasible in the near future.

17.2.26.R.02. **Rationale**

SHA-1 has been deprecated and the use of SHA-1 is permitted ONLY for legacy systems to validate existing hashes previously generated using SHA-1.

17.2.26.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:2155]

Agencies MUST use the SHA-2 family for new systems. Use of SHA-1 is permitted ONLY for legacy systems.

17.2.26.C.02. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:5905]

Agencies MUST use a minimum of SHA-384 when using hashing algorithms to provide integrity protection for information classified as RESTRICTED/SENSITIVE or above.

17.2.26.C.03.

Control System Classifications(s): All Classifications; Compliance: Must [CID:7187]

In all other cases when information requires integrity protection using hashing algorithms, Agencies MUST use a minimum of SHA-256.

Salts

17.2.27.R.01. **Rationale**

The use of salts strengthens the resistance of hash values to a variety of attacks, including brute-force, rainbow table, dictionary and lookup table attacks.

17.2.27.R.02. **Rationale**

Key derivation functions use a password, a salt, then generate a password hash. Their purpose is to make password guessing by an attacker who has obtained a password hash file expensive and therefore the cost of a guessing attack high and prohibitive.

17.2.27.C.01. **Control System Classifications(s): All Classifications; Compliance: Must** [CID:6560]

Memorised secrets such as passwords MUST be stored in a form that is resistant to offline attacks.

17.2.27.C.02. **Control System Classifications(s): All Classifications; Compliance: Should** [CID:6561]

Memorised secrets such as passwords SHOULD be salted and hashed using a suitable one-way key derivation function. See 17.2.14.

17.2.27.C.03. **Control System Classifications(s): All Classifications; Compliance: Should** [CID:6562]

The salt SHOULD be at least 32 bits in length, be chosen arbitrarily, and each instance is unique so as to minimise salt value collisions among stored hashes.

Approved symmetric encryption algorithms

17.2.28.R.01. **Rationale**

The use of Electronic Code Book (ECB) mode in block ciphers allows repeated patterns in plaintext to appear as repeated patterns in the ciphertext. Most cleartext, including written language and formatted files, contains significant repeated patterns. An attacker can use this to deduce possible meanings of ciphertext by comparison with previously intercepted data. In other cases they might be able to determine information about the key by inferring certain contents of the cleartext. The use of other modes such as Cipher Block Chaining, Cipher Feedback, Output Feedback or Counter prevents such attacks.

17.2.28.C.01. **Control System Classifications(s): All Classifications; Compliance: Should Not** [CID:2158]

Agencies using approved symmetric encryption algorithms (e.g. AES) SHOULD NOT use Electronic Code Book (ECB) mode.