TECHNICAL REPORT

ISO/TR 14742

First edition
2010-07-01

Financial services — Recommendations on cryptographic algorithms and their use

Services financiers — Recommandations sur les algorithmes cryptographiques et leur utilisation

Reference number
ISO/TR 14742:2010(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In exceptional circumstances, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide by a simple majority vote of its participating members to publish a Technical Report. A Technical Report is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TR 14742 was prepared by Technical Committee ISO/TC 68, *Financial services*, Subcommittee SC 2, *Security management and general banking operations*.

Introduction

The financial services industry has a clear need for cryptographic algorithms for a number of different applications. ISO standards provide definitions for an extensive and comprehensive set of such algorithms. However, as the state of the art of cryptology progresses and the power of computers increases, cryptographic algorithms as well as cryptographic keys of a particular length all have a limited window of time in which they can be considered secure. Furthermore, as neither the development of cryptology nor the increase in computing power are entirely predictable, the collective wisdom of the cryptographic community as to which algorithms and key lengths are secure is constantly evolving. For this reason it was felt that there was an equally clear need in the financial services industry for guidance regarding the current and up-to-date view in the cryptographic community about the security of cryptographic algorithms and their keys. It was also felt that there was a need for appropriate guidance on migration from one algorithm or key length to another.

The ISO standards that define cryptographic algorithms for the financial services industry do not contain such guidance, and by the evolving nature of the field, it would be difficult for them to do so. Hence, the need was recognized for a document that could contain such guidance, and be updated more frequently than the five year review cycle for ISO standards. This Technical Report is intended to be that document. The intention is to update this Technical Report when the need arises, or at least every other year.

The strength requirements of a security mechanism can vary depending on the application(s) in which the mechanism is being used and the way it is being used. The recommendations given in this Technical Report are considered to be general purpose recommendations. Although it is accepted that there may exist low-risk applications that do not warrant the level of cryptographic strength recommended in this Technical Report, it is advisable that deviation from the recommendations only be made after appropriate analysis of the risks and in the context of any rules and policies that might apply.

A special case of the above relates to the lifetime of protection required by the application and its data. For example, if protection requirements are ephemeral (e.g. confidentiality is required only for one day, or authentication is one-time) then this may be cause for allowing a deviation from the recommendations. Conversely, if the data must remain protected for a very long period of time, then the keys and algorithms used to provide the protection must be good for that duration, even if the keys are no longer in active use.

Financial services — Recommendations on cryptographic algorithms and their use

1 Scope

This Technical Report provides a list of recommended cryptographic algorithms for use within applicable financial services standards prepared by ISO/TC 68. It also provides strategic guidance on key lengths and associated parameters and usage dates.

The focus is on algorithms rather than protocols, and protocols are in general not included in this Technical Report. However, in some cases, for example for some key agreement and some authentication protocols, there is no “underlying” algorithm, and in a sense it is the protocol that constitutes the algorithm. In this case, the mechanisms are included, in particular where they have security parameters that can be adjusted for higher or lower security.

Algorithmic vulnerabilities or cryptographic keys of inadequate lengths are less often the cause of security compromises in the financial industry than are inadequate key management or other procedural flaws, or mistakes in the implementation of cryptographic algorithms or the protocols that use them. However, compromises caused by algorithmic vulnerabilities are more systemic and harder to recover from than other kinds of compromises.

This Technical Report deals primarily with recommendations regarding algorithms and key lengths.

NOTE Key management is covered in ISO 11568-1, ISO 11568-2 and ISO 11568-4.

The categories of algorithms covered in this Technical Report are:

— block ciphers;

— stream ciphers;

— hash functions;

— message authentication codes (MACs);

— asymmetric algorithms:

  — digital signature schemes giving message recovery,

  — digital signatures with appendix,

  — asymmetric ciphers;

— authentication mechanisms;

— key establishment and agreement mechanisms;

— key transport mechanisms.

This Technical Report does not define any cryptographic algorithms; however, the standards to which this Technical Report refers may contain necessary implementation information as well as more detailed guidance regarding choice of security parameters, security analysis, and other implementation considerations.

2 Measuring bits of security

For both block ciphers (Clause 4) and hash algorithms (Clause 6) the notion of "n bits of security" is introduced (e.g. see NIST SP 800-57, 2007, 5.6.1). For a block cipher to have n bits of security means that an estimated 2^n operations are needed to break the block cipher. Given a few plaintext blocks and corresponding ciphertext, a block cipher with n bits of security would then require an average of $2^{n-1}T$ of time to recover the encryption key, where T is the amount of time needed to perform one encryption of a plaintext value and a comparison of the result against the corresponding ciphertext value. For a hash algorithm to have n bits of security with respect to collision resistance means that an estimated 2^n calls to the hash function are necessary to find a hash collision, that is, two messages that when hashed yield the same hash result.

Table 1 below reflects recommendations for when an algorithm with n bits of security can be used. The dates coincide, where applicable, with the recommendations in NIST SP 800-57.

Table 1 — Recommended usage periods for algorithms of varying bit-strength

Bits of security	Recommended usage period
80	until end 2010
96	until end 2020
112	until end 2030
$\geqslant$ 128	as from 2030

The recommendations from Table 1 reflect that it is estimated that there is an overwhelming likelihood that an algorithm of the indicated bit strength will remain secure (that is, unbroken) until at least the year indicated.

For other categories of algorithms, such as message authentication codes and asymmetric algorithms, the concept of n bits of security is more difficult to define because of the nature of compromises and the measurement of the work or cost required to accomplish a compromise. However, for each category of algorithm, their security is still expressed in terms of bits of security. The intended interpretation is that if an algorithm is listed as having n bits of security, then it is estimated that it will remain secure until the same year as a symmetric cipher with n bits of security.

The efforts of breaking ciphers of different categories may have very different "profiles". One algorithm may require a large amount of computing power and little storage, while another may use a large amount of storage and less computing power. One effort may be parallelizable, so that the main limitation is the number of computers that can be recruited to participate, whereas another may require a single computer with a very large amount of RAM. Lenstra and Verheul in Reference [52] estimate that the financial costs associated with breaking an asymmetric cipher are 2 500 times larger than those associated with breaking a symmetric cipher, if the computational efforts measured in MIPS years are the same. See also Reference [19] for comparisons of cryptographic strengths of symmetric and asymmetric algorithms.

For algorithms with an estimated security of 128 bits or more, a recommendation of "past 2030" is given, reflecting the view that any estimate beyond 2030 is so far into the future that it seems unwise to make the estimate any more precise at this time.

For symmetric algorithms, Grover's algorithm (see Reference [17]) means that if a quantum computer were to be implemented, key sizes should be roughly doubled to maintain the same level of security. All the asymmetric algorithms mentioned in this Technical Report are vulnerable to quantum computing algorithms (see Reference [69]), and hence any leaps in progress in the area of implementing quantum computers could render the recommendations in Table 1 void. However, the commonly established wisdom is currently that

quantum computing on the scale necessary, say to factor a 1 024-bit RSA modulus, is at least 20 to 25 years away. On the other hand, if and when quantum computers are realized, it would be expected that increases in key lengths would be much less a barrier to compromise than now, so that the mentioned asymmetric algorithms would quickly become obsolete.

3 Algorithm migration

As the state of the art of cryptology progresses and the power of computers increases, cryptographic algorithms and key lengths that once were secure may no longer be so. For algorithms that have security parameters, security can be improved by adjusting the security parameters rather than migrating to a new algorithm. Examples include RSA-based crypto systems where the RSA key length can be increased and AES where the choice is between key lengths of 128, 192 and 256 bits.

Migration where only the security parameters are changed is mostly less onerous than migration where the cryptographic algorithm itself changes, and although performance in general would be expected to deteriorate with a more secure choice of security parameters, improvements in computer performance may make up for such a deterioration.

However, specific applications, implementations, data formats or indeed performance considerations may impose limits on the values of certain security parameters such that at some point it becomes impossible, infeasible or un-economical to maintain adequate security by only adjusting the security parameters.

It must further be assumed that no cryptographic algorithm will continue forever to provide adequate and cost-efficient security, regardless of the choice of security parameters. Hence it must be assumed that although increasing the security parameters for an existing algorithm may buy some time, eventually any application of a cryptographic algorithm will face a migration from that algorithm to a newer one.

Any such migration will be likely to incur both cost and disruption, but it is also an opportunity to take advantage of cryptographic and technical progress in modernizing the use of cryptographic algorithms, to what should be a faster, more secure and more cost-effective solution.

Experience gained in migrating from DES to TDEA has highlighted that the financial industry must establish a long-term and holistic (as opposed to piecemeal) approach to cryptographic algorithms. Lying as they do at the heart of all data security systems, changes to such algorithms are difficult, sensitive and expensive, and they take a long time to implement.

Thus, apart from identifying and preparing the financial industry for migration to other algorithms and longer key lengths with associated changes in key management, there is also a need to ensure that

— the structure of stored and transmitted data is suitable to be processed by generic cryptographic algorithms, and

— systems are designed to be sufficiently flexible to enable the negotiation of cryptographic algorithms and associated parameters.

For this reason, in order to create systems that are sufficiently flexible to withstand algorithm migration in the future, it is important to first start migrating to more flexible data structures and methods for processing such data structures. A good example of this is the adoption by ISO/TC 68 (e.g. see ISO 16609) of ISO/IEC 10116 in place of ISO 8372.

Because of the complexity of the task and the lifetime of relevant system components, a migration time of 10 to 15 years may well be realistic. Example steps that may need to be completed are:

a) development of flexible data structures;

b) agreement on algorithms and APIs;

c) development of plans to ensure interoperability through migration phase;

d) product development and test;

e) product implementation;

f) phased migration, including stopping the use of old algorithms;

g) protected data lifetime: this is the period after any new use of the old algorithms has ceased, but while data must still remain protected by the old algorithms.

See Figure 1.

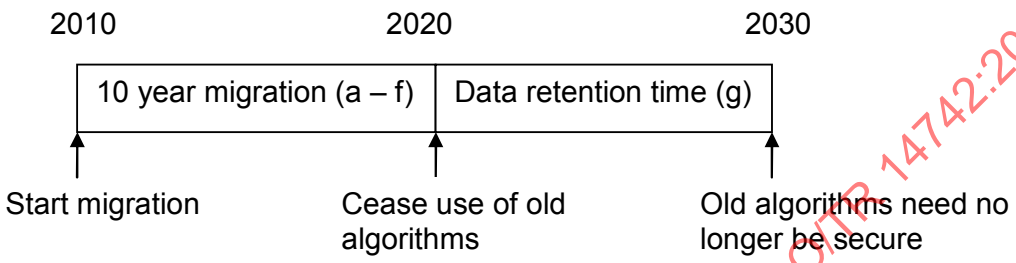

Figure 1 — Example of migration from old to new algorithms

The individual clauses below will highlight any particular migration issues there may be for the algorithms they discuss.

4 Block ciphers

4.1 General

This clause lists block ciphers that may be used within applicable ISO/TC 68 standards.

A block cipher maps a block of n plaintext bits to a block of n ciphertext bits using a key of k bits. The block ciphers listed in Table 2 below are defined in ISO/IEC 18033-3.

Table 2 — Block ciphers

Block length	Algorithm name	Key length
64 bits	TDEA	128 or 192 bits
	MISTY1	128 bits
	CAST-128	
128 bits	AES	128, 192 or 256 bits
	Camellia	
	SEED	128 bits

4.2 Keying options

4.2.1 Keying options for TDEA

Keying option 1, also known as 2-key Triple DES: 128-bit key represented as two 64-bit DEA keys, where for each DEA key 56 bits can be chosen arbitrarily and the rest can be used for error detection.

Keying option 2, also known as 3-key Triple DES: 192-bit key represented as three 64-bit DEA keys, where for each DEA key 56 bits can be chosen arbitrarily and the rest can be used for error detection.

4.2.2 Keying options for AES

Keying option 1: 128-bit, where all 128 bits can be chosen arbitrarily.

Keying option 2: 192-bit, where all 192 bits can be chosen arbitrarily.

Keying option 3: 256-bit, where all 256 bits can be chosen arbitrarily.

4.2.3 Keying options for Camellia

Keying option 1: 128-bit, where all 128 bits can be chosen arbitrarily.

Keying option 2: 192-bit, where all 192 bits can be chosen arbitrarily.

Keying option 3: 256-bit, where all 256 bits can be chosen arbitrarily.

4.3 Recommended block ciphers

Table 3 contains a list of recommended block ciphers and their current estimated security in bits. The recommendations are based on the analyses and recommendations provided in the ECRYPT yearly report on algorithms and key sizes (see References [13] and NIST SP 800-57).

Table 3 — Security of block ciphers

Algorithm	Keying option	Key length	Security in bits
TDEA	1	128 bits	80 – 112
	2	192 bits	112
AES	1	128 bits	128
	2	192 bits	192
	3	256 bits	256

Note that:

— 2-key Triple DES (TDEA with keying option 1) has effective strength $2^{\min(112,\ 120-t)}$ where 2^t is the number of plaintext ciphertext pairs available to an attacker.

— 3-key Triple DES (TDEA with keying option 2) has effective strength exceeding 2^{100}. Typically, its strength is cited as 2^{112}, but in general its strength is best expressed as in Reference [55], Note 7.38, as a trade-off between memory and computation, where a “meet-in-the-middle” attack requires 2^{57-s} space and 2^{112+s} time, for $1 \leqslant s \leqslant 56$.

The recommended end date for use of 2-key Triple DES (TDEA with keying option 1) ranges from 2010 to 2030. Which date is appropriate for a given implementation depends on the way in which the keys are being used in that implementation. If the key usage provides a potential attacker with a large number of plaintext-ciphertext pairs for the same key (e.g. 1 000 000 000 000 $\approx 2^{40}$ pairs), the security of the key is approximately 80 bits and hence the recommended use is until 2010. If only a few (fewer than 256) pairs are available, it may be acceptable to continue use until 2030.

Interpolating, by way of example, if an estimated maximum of 16 million plaintext-ciphertext pairs might become available to an attacker, the estimated security of the key would be approximately 96 bits (since $2^{24} \approx 16$ million), and the recommended use would be until 2020. Hence, proper use of session keys will

greatly extend the usable life of a 2-key Triple DES (TDEA with keying option 1) implementation, as will frequent change of keys. If it is not possible to estimate a limit on the number of plaintext-ciphertext pairs that may become available to an attacker, then the most conservative recommendation (to stop use by 2010) applies.

Notice also that in the absence of session keys, 64-bit MACs may provide an attacker with plaintext-ciphertext pairs (in particular for messages less than 8 bytes) and thus aid in reducing the security of the key used.

For example, consider PIN entry devices that use a fixed key versus those that use unique keys per transaction, such as DUKPT (Derived Unique Key Per Transaction, as specified in ANSI X9.24-1). Fixed-key devices could provide an attacker with a large number of plaintext-ciphertext pairs (one pair for each encryption), weakening the strength of the key, whereas devices that use a unique key per transaction provide at most one plaintext-ciphertext pair for each session key. Particular implementations or formats may also limit the availability of plaintexts to attackers (e.g. by including randomness in the encrypted values, such as in PIN block format 3), thereby protecting the strength of the encipherment key.

The other symmetric algorithms from Table 2 (MISTY1, CAST-128, Camellia and SEED) should only be used when legacy applications require it. In this case, the maximum strength of the algorithm would be expected to be similar to that of AES with the same keying option, and hence the recommendations from Table 3 can be carried over for those key lengths. Consideration should however be given to the fact that these algorithms have received significantly less scrutiny in the cryptographic community than TDEA and AES. Note that there are recent research papers which propose theoretical related-key attacks against AES using keying options 2 and 3 (192-bit and 256-bit keys respectively). See Reference [75].

When evaluating the suitability of a particular block cipher for a given implementation, it is important to take into account the length of time it is necessary to protect the data that the block cipher is used to encrypt. For example, if a 3-key Triple DES (TDEA with keying option 2) implementation is used to encrypt data which needs to be protected for 10 years after it is encrypted, then encipherment of new data should stop in 2020 [because in terms of years, $2020 + 10 = 2030$, the last year where 3-key Triple DES (TDEA with keying option 2) is recommended].

4.4 Block size and key use

Besides key length, block size is an important security parameter, e.g. the French government IT Security Agency recommends against using 64-bit block ciphers for encryption or MAC-ing. The concern with small block size is mainly that text dictionary attacks and matching ciphertext attacks become feasible, as outlined in Reference [55], Note 7.8. A text dictionary attack builds a dictionary of known plaintext-ciphertext pairs (each 1 block), and a complete dictionary for a 64-bit block cipher can thus be built if 2^{64} plaintext-ciphertext pairs are available. Matching ciphertext attacks exploit that the birthday paradox implies that once the number of available ciphertexts for an n-bit block cipher reaches $2^{n/2}$, which for a 64-bit block cipher is approximately 4 290 000 000, one expects to find matching ciphertext blocks, which may reveal partial information about the plaintexts. For this reason it is recommended not to use the same key for more than $2^{n/2}$ times for an n-bit block cipher.

The use of session keys greatly reduces the risks of small block size.

4.5 Modes of operation

The modes of operation for block ciphers should follow ISO/IEC 10116. For TDEA, see also ISO/TR 19038. As stated in ISO/IEC 10116:2006, B.1.2, one property of the Electronic Code Book (ECB) mode is as follows:

> "(...) the same plaintext block always produces the same ciphertext block (for the same key) making it vulnerable to a "dictionary attack", where a dictionary is built up with corresponding plaintext and ciphertext blocks. The ECB mode is, in general, not recommended for messages longer than one block. The use of ECB may only be specified in future International Standards for those special purposes where the repetition characteristic is acceptable, blocks have to be accessed individually, or blocks have to be accessed randomly".

Hence, for plaintexts longer than the block size of the block cipher, Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) or Counter (CTR) mode are recommended. Which mode of operation to choose depends on the specific requirements of the application (such as ability of parallelizing encryption/decryption, error propagation, or requirements to initialization values). Please refer to ISO/IEC 10116:2006, Annex B, for properties of the modes of operation.

4.6 Enciphering small plaintexts

When enciphering small plaintexts such as customer PINs, special considerations apply when estimating the security of a particular mechanism. For example, if the entire plaintext to be enciphered is an 8 byte PIN block and no random data is added, dictionary attacks may be possible regardless of the size of the key used for encipherment. In this case, it is important to follow established standards, such as ISO 9564-2 for PIN encipherment, where the formats specified have been developed with these concerns in mind.

4.7 Migrating from TDEA to AES

In light of the experience of migrating from DES to TDEA in particular, any migration from TDEA to a replacement symmetric algorithm should be planned well in advance in order to reduce unnecessary costs.[1)]

Aside from the inherent challenges of migrating from one symmetric encipherment algorithm to another across the financial services industry, one important difference between TDEA and AES is the block length of each. TDEA is a 64-bit block cipher, whereas AES is a 128-bit block cipher. A consequence of this is that today, many of the data structures used for storage and transmission are 8-byte oriented, prime examples of this being TDEA keys themselves, and PIN blocks. Today, many payment networks would not be able to handle the 128-bit blocks that an AES-based PIN block would use. Migration would have to be staged and hence involve parts of the payment networks using TDEA and other parts using AES, which would lead to a temporary but long lasting requirement for translations both from TDEA encrypted messages to AES encrypted messages, and vice versa.

AES also supports longer key lengths than TDEA, and as a consequence a migration from TDEA to AES may have significant key management impact as well. New AES-based systems should be built flexibly, so that they can eventually employ all the available key lengths for AES (keying options 1, 2, and 3).

5 Stream ciphers

A stream cipher maps bits of plaintext to bits of ciphertext using a key of k bits. It does so by using the k-bit key to generate a key stream of the same length as the plaintext. The key stream is used (like a one-time pad) to encipher the plaintext, bit by bit (using the xor operation). Stream ciphers may be dedicated stream ciphers or based on modes of operations of block ciphers. For modes of operations of block ciphers, see ISO/IEC 10116. Stream ciphers are standardized in ISO/IEC 18033-4, including the dedicated stream ciphers MUGI and SNOW, and the stream ciphers CTR, OFB, and CFB, which are based on block ciphers. ISO/TC 68 recommends use of CTR, OFB or CFB, because these schemes are based on well-known modes of operations with block ciphers, whereas both MUGI and SNOW are of relatively recent design (2002).

6 Hash functions

6.1 Hash functions and their properties

For the purposes of this Technical Report, the terms “hash algorithm” and “hash function” are considered equivalent. A hash function maps plaintext of an arbitrary length into an h-bit hash result, processing the input in blocks of m bits. Hash functions are taken from ISO/IEC 10118-2 (hash functions based on block ciphers), ISO/IEC 10118-3 (dedicated hash functions) and ISO/IEC 10118-4 (hash functions using modular arithmetic). Applications rely on different properties of hash functions, such as:

1) Note that similar planning will be needed for the future migration from SHA-1 to its approved successor(s).

a) pre-image resistance — the property that given an h-bit bit-string H, it is infeasible to find a plaintext value the hash of which is H;

b) second pre-image resistance — the property that, given a plaintext A, it is infeasible to find a different plaintext B such that the hash of A is identical to the hash of B;

c) collision resistance — the property that it is infeasible to find any two different plaintexts A and B such that the hash of A is identical to the hash of B.

6.2 Hash functions based on block ciphers

The following is a list of hash functions based on a block cipher with block size n (see ISO/IEC 10118-2):

a) Hash algorithm 1 — size of hash result less than or equal to n;

b) Hash algorithm 2 — size of hash result less than or equal to $2n$;

c) Hash algorithm 3 — size of hash result equal to $2n$;

d) Hash algorithm 4 — size of hash result equal to $3n$.

The security of hash functions based on block ciphers depends on the security of the underlying block cipher and on the size of the hash result. If there are no known algorithmic vulnerabilities (such as, presently, for AES), the security in bits of a hash function

— based on a block cipher with m bits of security and

— with size of hash result equal to h bits

is expected to be $\min\left\{\frac{h}{2}, m\right\}$.

ISO/IEC 10118-2 mentions specifically using DEA ("single DES"; see ANSI X3.92 and ISO/IEC 18033-3:2005, Annex A) as the underlying block cipher, and provides examples for how to construct hash algorithms 1 to 4 with the size of hash result equal to 64, 128, 128 and 192 bits. If it is necessary to use a hash function based on a block cipher, AES should be used as the underlying block cipher. Otherwise, using a dedicated hash function is recommended.

6.3 Dedicated hash functions

Table 4 lists dedicated hash functions, together with their estimated security strength in bits. Their recommended last usage dates can then be inferred from Table 1, depending on whether collision resistance, pre-image resistance, or second pre-image resistance is required. If it is not clear if collision resistance is required or not, it must be assumed that it is required, and hence the more conservative estimate should be used for usage dates.

Table 4 — Dedicated hash functions and their security

Hash function	Block length m in bits	Hash result length h in bits	Collision resistance strength in bits	Pre-image resistance strength in bits	Second pre-image resistance strength in bits
RIPEMD-160	512	160	80	160	105-160
RIPEMD-128	512	128	<60	128	73-128
SHA-1	512	160	63*	160	105-160
SHA-224	512	224	112	224	201-224
SHA-256	512	256	128	256	201-256
SHA-512	1024	512	256	512	394-512
SHA-384	1024	384	192	384	384
WHIRLPOOL	512	512	256	512	265-512

NOTE Recent progress in the research on finding hash function collisions has resulted in the estimated collision resistance security of SHA-1 decreasing from 80 bits to 63 or 69 bits during the period from 2004 to 2006 (see Reference [73]). For this reason, NIST has published the following recommendation regarding the SHA family of hash functions on their website: "March 15, 2006: The SHA-2 family of hash functions (i.e. SHA-224, SHA-256, SHA-384 and SHA-512) may be used by Federal agencies for all applications using secure hash algorithms. Federal agencies should stop using SHA-1 for digital signatures, digital time stamping and other applications that require collision resistance as soon as practical, and must use the SHA-2 family of hash functions for these applications after 2010. After 2010, Federal agencies may use SHA-1 only for the following applications: hash-based message authentication codes (HMACs); key derivation functions (KDFs); and random number generators (RNGs). Regardless of use, NIST encourages application and protocol designers to use the SHA-2 family of hash functions for all new applications and protocols." Even more recent work suggests that the collision resistance of SHA-1 may be as low as 52 bits.

NIST has initiated a competition to identify alternatives to the SHA family of hash functions. The competition is scheduled to run from 2007 to the end of 2011, ending with an announcement of the alternative hash function(s) at the end of 2011. It is then anticipated that the augmented and revised Hash Function Standard will be published by 2012. Assuming that the NIST competition and subsequent standard end up nominating a hash algorithm which is not among the SHA family of hash algorithms, then the recommendations in this Technical Report would change accordingly.

Whether a particular hash function is fit for a particular application depends on which properties of the hash function the application relies. As the three columns for security strength in Table 4 show, pre-image and second pre-image resistance are much easier to achieve than collision resistance. Hence, if an application uses a hash function and relies only on the pre-image resistance of the hash function, it can continue to use that hash function longer than if it relies also on the collision resistance of the hash function.

For a practical example, collisions have been found for several hash functions (such as MD4 and MD5), where it still seems infeasible for the moment to find second pre-images in general (for the best results so far on finding second pre-images for MD4, see Reference [74]).

For all hash functions except SHA-384, second pre-image resistance strength varies according to the size of the input string, with the strength decreasing with increasing input length (see Reference [49]). The range indicated in Table 4 above is calculated using the formula given in the appendix of NIST SP 800-107, and using the definitions of maximum input length specified in ISO/IEC 10118-3. The minimum strength indicated in the column for second pre-image resistance strength in bits corresponds to the maximum input length allowed by ISO/IEC 10118-3, and the maximum strength corresponds to input strings of length at most the same as the block length of the hash function.

Whether an application relies on collision resistance or only on pre-image or second pre-image resistance may well require expert input. If it cannot be determined whether collision resistance is required or not, it should be assumed that it is required.

For applications that currently use SHA-1 and rely on collision resistance, migration to another algorithm would have to take place before 2010 in order to comply with the recommendations in Table 4, which coincides with the recommendations regarding use of SHA-1 given in NIST SP 800-57.

6.4 Hash functions using modular arithmetic

ISO/IEC 10118-4 defines two hash functions, MASH-1 and MASH-2, that are based on modular arithmetic, and where their basic properties such as pre-image and collision resistance depend on the difficulty of factoring a product of two large primes. Neither MASH-1 nor MASH-2 has seen any significant practical deployment and for this reason they are not recommended.

6.5 Migrating from one hash function to another

The industry has already seen a sizable migration from MD-5 and other obsolete hash algorithms to mainly SHA-1, but also to some extent to SHA-256. With the gradual demise of SHA-1 and with the NIST competition to find an alternative by 2012 we can expect a migration to the new hash algorithm starting even before 2012, but lasting many years.

If migrating to a hash algorithm with a larger hash result, the migration will likely result in a change of data formats and will, as such, require larger changes than just switching one hash algorithm for another with the same size of hash result. In order to facilitate more flexibility in the future, data formats should, therefore, be designed with the consideration in mind that not only might the hash algorithm change, but so may the size of the hash result.

For many financial applications, a migration away from SHA-1 before 2010 is not feasible for financial and logistical reasons having to do with a large installed base of equipment with a long shelf life (e.g. Point-of-Sale devices). In addition, a quick migration may not be necessary, as outlined above.

As a worst case, if a migration prior to 2010 is not feasible, but on the other hand a compromise will result in significant financial loss or loss in reputation, and collision resistance is indeed required, a migration away from SHA-1 to one of the recommended hash algorithms from Table 4 should be planned, and mitigating security measures (such as procedural controls) should be introduced to compensate for the risk in the period from 2010 until a migration has been completed, and no further reliance is placed on the value of SHA-1 hash results. If satisfactory mitigating controls can be maintained until a migration to the new NIST recommended hash algorithm can be completed, this may be preferable.

If collision resistance is not required, SHA-1 can be used well beyond 2010, and a migration can therefore take place after NIST has nominated an alternative. In this case, migration should take place as soon as practical, after the new hash algorithm has been identified. It should be noted that a migration of hash algorithms may be easier than a migration of encryption algorithms, in particular if the data structures involved can accommodate the new size of hash result.

When evaluating system security in connection with a migration away from SHA-1, it is necessary to take into account the time it takes to perform the migration. Hence, for example, if it is decided to wait for the nomination of an alternative hash algorithm to SHA-1, assuming the replacement is announced by 2012 and a migration takes 6 years to complete, then SHA-1 needs to be deemed adequate for the particular application until 2019, or compensating controls need to be put in place.

The situation with applications that require collision resistance is further complicated by the fact that for some applications, any one SHA-1 collision will suffice to compromise the security of the application. Hence, there is no individual task for a perpetrator to calculate a hash collision; once the first genuine SHA-1 collision has been published, it can be used to compromise a number of applications. Application owners should take extra care to determine whether their applications belong to this category of particularly vulnerable applications. For example, Reference [11] presents two postscript files resulting in two completely different views, but having the same MD-5 hash result, Reference [51] presents two different X.509 certificates (using MD-5 as the hash algorithm) with the same hash result, and Reference [71] shows how to apply for one X.509 certificate (that uses MD-5 as the hash algorithm) and change the received certificate to one with different basic constraints, e.g. from an end-entity certificate to a Certification Authority certificate, but with the same hash result.

7 Message authentication codes

7.1 Recommended MAC algorithms

ISO/IEC 9797-1 and ISO/IEC 9797-2 list MAC algorithms based on block ciphers and hash functions respectively. ISO 16609 recommends a subset of those MAC algorithms for the financial industry, provides a list of approved block ciphers, and defines the method for including further block cipher algorithms.

7.2 MAC algorithms based on block ciphers

The following options from ISO/IEC 9797-1 are recommended:

a) MAC Algorithm 1 using AES (Keying Option 1) or TDEA (Keying Option 1) using Padding Method 3;

b) MAC Algorithm 3 using DEA ("Single DES").

Single DES is defined in ANSI X3.92 (and also specified in ISO/IEC 18033-3). The recommendations above agree with those in ISO 16609, except for the addition of AES as a possible block cipher. The rationale why Single DES is acceptable as used in option b) above is that the mode of operation here amounts to iterated applications of Single DES followed by an application of TDEA (Keying Option 1).

In addition, the algorithm CMAC (see NIST SP 800-38B) is being incorporated into ISO/IEC 9797-1, and is recommended as well.

Padding Method 3 requires that the length of the plaintext to be MAC-ed be known prior to the start of the MAC calculation. If this is not the case, option a) above cannot be used. If Padding Method 3 is used, the length in bits of the data string that is input to the MAC algorithm needs to be less than 2^n. Padding Method 1 requires that the length of the message to be MAC-ed be known to the receiver in advance, since otherwise an attacker can insert text into the message while keeping the MAC the same.

7.3 MAC algorithms based on hash functions

The following MACs from ISO/IEC 9797-2 are based on hash functions:

a) MAC Algorithm 1 — MDx-MAC;

b) MAC Algorithm 2 — HMAC;

c) MAC Algorithm 3 — Short input variant of MDx-MAC.

The hash functions that according to ISO/IEC 9797-2 can be used with MAC Algorithms 1 – 3 are:

— MAC Algorithm 1: RIPEMD-160, RIPEMD-128, SHA-1;

— MAC Algorithm 2: RIPEMD-160, RIPEMD-128, SHA-1, SHA-224, SHA-256, SHA-512, SHA-384, WHIRLPOOL;

— MAC Algorithm 3: RIPEMD-160, RIPEMD-128, SHA-1.

MAC Algorithms 1 and 3 require that both the IV and the round functions for the underlying hash functions be modified. MAC Algorithm 2 requires that the IV be modified.

MAC Algorithm 3 requires that the message to be MAC-ed be less than 256 bits.

Reference [8] shows that MAC Algorithms 1 to 3 are vulnerable to collisions in the underlying hash functions, and therefore it is recommended not to use them beyond the recommended usage of the underlying hash function according to Table 4.

7.4 Length of the MAC

The length of the MAC is an important security parameter, as it determines the difficulty of carrying out a guessing attack on the MAC. In general, without regard for the underlying MAC algorithm, an attacker has a 1 in 2^m chance of guessing a particular MAC of length m. Thus, any application of MACs should take into account that an attacker may attempt to systematically guess many different MACs for a message, or for different messages, thereby increasing the probability that one or more are accepted as valid. For this reason, if m is less than or equal to 40, measures should be in place that limit the number of unsuccessful verifications using the same key. Short sessions or low bandwidth of the communications channel may achieve this. In accordance with NIST SP 800-38B, general guidance on the length of the MAC can be quantified in terms of two bounds:

a) The highest acceptable probability p for an inauthentic message to pass the verification process.

b) A limit on the number of times i that invalid MACs can be validated and rejected across all implementations of the verification process for a given key, before that key is retired.

Given estimates of p and i, m should then satisfy $2^m \geqslant \frac{i}{p}$. For example, suppose that the MAC verification process within a system will validate and reject at most 1 024 messages with an invalid MAC before a key is retired (that is, $i = 2^{10}$), and that it is acceptable that there is about a one in a million chance that the system will accept an inauthentic message (so $p = 2^{-20}$). In this case, any value of m greater or equal to 30 satisfies the inequality.

On the other hand, with a very large number of available MACs, full-length MACs may be vulnerable to key recovery attacks. Refer to ISO/IEC 9797-1:1999, Annex B, for more information.

7.5 Message span of the key

The message span of a key is the total number of messages for which MACs have been generated with that key. The message span of a key is relevant because it affects the vulnerability to attacks that exploit MAC collisions. For general-purpose applications for MACs based on block ciphers, it is recommended to limit use of any single MAC key to no more than 2^{48} messages ($\approx$ 281 billion messages) when the block size of the underlying block cipher is 128 bits and to no more than 2^{21} messages ($\approx$ 2 million messages) when the block size of the underlying block cipher is 64 bits. If these limits are exceeded there is a risk of key recovery attacks, as described in References [65], [66] and [58].

8 Asymmetric algorithms

8.1 General

Asymmetric algorithms are taken from ISO/IEC 9796-2, ISO/IEC 9796-3, ISO/IEC 14888-3 and ISO/IEC 18033-2. The asymmetric algorithms described here are used to support three different basic security mechanisms:

— digital signatures with message recovery;

— digital signatures with appendix;

— encipherment.

Each of the asymmetric algorithms in this Technical Report are based on one of the following “hard problems”:

— factoring integers of the form pq, where p and q are primes. In this case, the security parameter is k, the size in bits of the modulus pq;

— factoring integers of the form $p^d q$, where p and q are primes, and d is a natural number greater than 1. This Technical Report will be concerned only with the case $d = 2$. In this case, the security parameters are k, the size in bits of the modulus $p^d q$, and t, the size in bits of the least of the two primes p and q;

— integer-based DL: Finding discrete logarithms in finite subgroups of the field of integers. In this case, the security parameters are L, the size in bits of the public key, and N, the size in bits of the private key;

— EC-based DL: Finding discrete logarithms in subgroups of elliptic curves over finite fields. In this case, the security parameter is f, the size in bits of the order of the base point. The value of f is commonly considered to be the key size.

Table 5 specifies the recommended usage periods for asymmetric algorithms based on the hard problems above.

NOTE The papers "On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography" (2009/389 from IACR e-archive) and "Factorization of a 768-bit RSA modulus" (2010/006 from IACR e-archive) by Kleinjung et al provide state-of-the-art information relating to the hard problems underlying these asymmetric algorithms.

Table 5 — Recommended usage for asymmetric algorithms

Recommended until end of	Factorization $n = pq$	Factorization $n = p^2 q$	Integer-based DL	EC-based DL
2010	$k = 1\,024$ bits	$k = 1\,344$ $t = 448$	$L = 1\,024$ $N = 160$	$f = 160 - 191$
2020	$k = 1\,536$ bits	$k = 1\,776$ $t = 448$	$L = 1\,536$ $N = 192$	$f = 192 - 224$
2030	$k = 2\,048$ bits	$k = 2\,304$ $t = 768$	$L = 2\,048$ $N = 224$	$f = 224 - 255$
2030+	$k = 3\,072$ bits	$k = 3\,264$ $t = 1\,088$	$L = 3\,072$ $N = 256$	$f = 256$

Each of the underlying hard problems has applications for each of the basic security mechanism as outlined in Table 6.

Table 6 — Asymmetric algorithms and hard problems

Security mechanism	Hard problem			
	Factorization $n = pq$	Factorization $n = p^2 q$	Integer-based DL	EC-based DL
Digital signatures with message recovery	Digital signature scheme 1, 2, 3, options 1 and 2	None	Signature scheme on a prime field	Signature scheme on an elliptic curve over prime field Signature scheme on an elliptic curve over binary field
Digital signature with appendix	Digital signature mechanisms based on ISO/IEC 9796 Identity-based signature mechanism (Guillou and Quisquater)	ESIGN	DSA, KCDSA Pointcheval/Vaudenay	ECDSA
Encipherment	RSA-HC RSAES	HIME(R)	ECIES-HC, PSEC-HC, ACE-HC	ECIES-HC, PSEC-HC, ACE-HC

8.2 Factorization-based security mechanisms

In addition to selecting keys of sufficient size, as described in Table 5, a public exponent e must also be chosen (see ISO/IEC 9796-2:2002, A.3.1). $e = 3$ is not recommended. Although a choice of $e = 3$ or other low values has performance benefits, there have been a number of attacks on low-exponent RSA. See References [6], [7] and [9]. Most of the attacks exploit implementation errors or padding issues with factorization-based security mechanisms not recommended in this Technical Report.

Some of the attacks on low-exponent RSA recover plaintext and some recover the entire private key from small parts of the private key, and hence, in environments where attacks are possible that may recover parts of the private key (e.g. a quarter of the bits), small public exponents should be avoided. In such cases, $e = 2^{16} + 1$ or greater should be used. In cases where the performance penalty of using $e = 2^{16} + 1$ over $e = 3$ is not significant, $e = 2^{16} + 1$ or greater is recommended.

8.3 Integer discrete logarithm-based security mechanisms

For integer discrete logarithm-based security mechanisms such as DSA, KCDSA, Pointcheval/Vaudenay, or any of the hybrid encipherment mechanisms implemented over integers and not elliptic curves, the security parameters to establish are:

— P a prime number, of length L bits;

— Q a prime divisor of P-1;

— F an integer such that $1 < F < P-1$ and $F^{\frac{P-1}{Q}} \bmod P > 1$.

For DSA, a hash algorithm has to be selected as well. For DSA, ISO/IEC 14888-3 provides the alternatives given in Table 7.

Table 7 — DSA parameter choice

L bits	Hash algorithm	Security bits
1 024	SHA-1	< 80
2 048	SHA-224	112
2 048	SHA-256	112
3 072	SHA-256	128

Due to recent attacks on SHA-1, the security of the option with $L = 1\ 024$ and hash function SHA-1 is less than 80 bits and is not recommended. For more specific recommendations regarding parameter choice, please refer to ISO/IEC 14888-3. The other options are all recommended until the date implied by their bit-security (see Table 1).

ISO/IEC 14888-3 mentions two other discrete log-based signature mechanisms, which are both variations of DSA, namely KCDSA and Pointcheval/Vaudenay signatures. They both allow flexibility in the choice of hash algorithm. For the purposes of this Technical Report their security can be taken to be the minimum of the security indicated by the choice of asymmetric key length as per Table 5 and the choice of hash algorithm as per Table 4. For more specific recommendations regarding parameter choice, see ISO/IEC 14888-3.

8.4 Elliptic curve discrete logarithm-based security mechanisms

For elliptic curve-based security mechanisms, a choice has to be made regarding the field over which the elliptic curve is defined, namely whether to use a prime field (F_p, where p is a prime) or a binary field (F_{2^m}, where m is a natural number). It is recommended to use prime fields (performance can be improved by choosing the prime to be a Mersenne prime, that is, of the form $2^n - 1$). NIST has defined a suite of elliptic curves that can be used. See FIPS PUB 186-3 and ANSI X9.62 for definitions of curves.

8.5 Algorithm or key expiry

The situation when an algorithm or a key length expires (passes the date beyond which reliance is not recommended) is very different for encipherment mechanisms than for digital signature mechanisms (with appendix or with message recovery). When a key length is no longer considered secure, data that was enciphered with keys of that length must now be considered exposed. Even prior to the "expiration" date there is little that can be done about this, short of keeping the ciphertext confidential by other means (e.g. procedural). Hence, at the time of encipherment, it is important to make a decision on when it will be acceptable to expose the data that is being enciphered. With digital signature algorithms, the signer has an option of re-signing the data using larger keys or improved signature algorithms. Provided a trusted time-stamp is used, relying parties can verify that the new signature was applied before the security of the original digital signature was considered unsecure. Hence, the binding of the signed data to the identity of the signer can be maintained beyond the expiry of the original signature.

8.6 Digital signature schemes giving message recovery

Digital signature schemes giving message recovery are taken from ISO/IEC 9796-2 and ISO/IEC 9796-3. They are:

a) digital signature scheme 1 (ISO/IEC 9796-2, option 1 or 2);

b) digital signature scheme 2 (ISO/IEC 9796-2, option 1 or 2);

c) digital signature scheme 3 (ISO/IEC 9796-2, option 1 or 2);

d) signature scheme on a prime field (ISO/IEC 9796-3);

e) signature scheme on an elliptic curve (ISO/IEC 9796-3) over prime field;

f) signature scheme on an elliptic curve (ISO/IEC 9796-3) over binary field.

Digital signature scheme 2 (both options) have as additional security parameter salt length L_S in bits. It is recommended that $L_S = L_H$, the length of hash code produced by the hash function.

Digital signature scheme 2 (ISO/IEC 9796-2), is compatible with the scheme known as IFSSR specified in Reference [18]. It is closely based on a scheme known as PSS-R.

Each digital signature scheme in this clause uses a hash function from ISO/IEC 10118-2 or ISO/IEC 10118-3 (see Clause 5). Apart from the security parameters listed, the security of a signature scheme with recovery also depends on the hash function used.

The digital signature schemes should not be used beyond the recommended usage date for the hash function they use (see Clause 5)[2]. With this restriction, the recommended usage dates are as specified in Table 5.

2) Digital signature schemes with message recovery may not require collision resistance, but only pre-image resistance of the hash function they use, and in certain circumstances there may therefore be arguments for using them beyond the usage dates listed for hash functions in Clause 5.

Digital signature scheme 2 is recommended over digital signature schemes 1 and 3. In environments where generation of random variables by the signer is deemed infeasible, and digital signature scheme 2 is therefore not possible, digital signature scheme 3 is recommended. Digital signature scheme 1 can only be used in environments where compatibility is required with systems implementing the first edition of ISO/IEC 9796-2 (ISO/IEC 9796-2:1997) and is only compatible with systems implementing ISO/IEC 9796-2:1997 that use hash-codes of at least 160 bits.

8.7 Digital signatures with appendix

These algorithms are taken from ISO/IEC 14888-2 and ISO/IEC 14888-3.

They are:

a) digital signature mechanisms based on ISO/IEC 9796;

b) ESIGN;

c) DSA;

d) Pointcheval/Vaudenay;

e) ECDSA;

f) identity-based digital signature algorithm by Guillou and Quisquarter.

In addition, ISO/IEC 14888-3 specifies two digital signature schemes, EC-GDSA and EC-KCDSA, which are recommended only for national or legacy applications.

The recommended usage dates are as specified in Table 5, except for algorithm f). The identity-based signature algorithm by Guillou and Quisquater comes in three variants. For each of the three variants it is recommended to choose the public key modulus and the randomizer size according to Table 5, using the column for factorization, $n = pq$. For example, if the algorithm is to be used until 2030, a public modulus size and randomizer size of 2048 is recommended.

8.8 Asymmetric ciphers

8.8.1 Overview

The recommended asymmetric ciphers are taken from ISO/IEC 18033-2.

The main security criterion for asymmetric ciphers is resistance to adaptive chosen ciphertext attacks as described in ISO/IEC 18033-2:2006, B.4.

The asymmetric ciphers from ISO/IEC 18033-2 are:

a) hybrid ciphers;

b) RSAES;

c) HIME(R).

8.8.2 Hybrid ciphers

8.8.2.1 General

8.8.2.1.1 A hybrid cipher as defined in ISO/IEC 18033-2 employs a symmetric encryption algorithm by encrypting a symmetric encryption key using an asymmetric algorithm (the Key Encapsulation Mechanism, "KEM"). The symmetric encryption key is then used to encrypt the actual message using symmetric cryptographic techniques (the Data Encapsulation Mechanism, "DEM"). A hybrid cipher is parameterized by a choice between four KEMs and three DEMs.

The KEMs are:

a) ECIES-KEM — which is parameterized by a choice of concrete group, key derivation function, "KDF", the security parameter *KeyLen*, as well as various other parameters that play no role in the recommendations of this Technical Report;

b) PSEC-KEM — which is parameterized by a choice of concrete group, KDF, and the security parameters *SeedLen* and *KeyLen*;

c) ACE-KEM — which is parameterized by a choice of concrete group, KDF, a hash function, the security parameter *KeyLen*, and other parameters that play no role in the recommendations of this Technical Report;

d) RSA-KEM — which is parameterized by KDF and the security parameter *KeyLen*.

The choices for concrete group are:

— integer-based (a finite subgroup of the field of integers): in this case, the relevant key length parameters are L, the size of the public key, and N, the size of the private key;

— elliptic curve-based (a subgroup of an elliptic curve over a finite field): in this case, the relevant key length parameter is f, the size of the order of the base point; the value of f is commonly considered to be the key size.

8.8.2.1.2 RSA-KEM is based on integer factorization. The relevant key length parameter is k, the size of the modulus.

The choices for KDF are KDF1 and KDF2, which both depend on a choice of hash function (see Clause 5).

The parameter *KeyLen* is the length of the secret key which is encapsulated by the KEM mechanism and used for data encipherment in the DEM mechanism.

The choices for DEM are:

a) DEM1 – which is parameterized by a choice of symmetric cipher, "SC", and MAC;

b) DEM2 – which is parameterized by a choice of SC, MAC, and the security parameter *LabelLen*;

c) DEM3 – which is parameterized by a choice of MAC and *MsgLen*, the length in bits of the message to be encrypted.

The choices for SC are:

— SC1 — which uses a block cipher (BC) in CBC mode with padding according to ISO/IEC 18033-2:2006, 6.5.2.2. The available choices for BC are those block ciphers defined in ISO/IEC 18033-3 (see Table 2).

— SC2 — which uses a key derivation function (KDF) and has a security parameter *KeyLen*, the length of the key. The available choices for KDF are KDF1 and KDF2, both of which use a hash function (H). The available choices for H are those defined in ISO/IEC 10118-2 and ISO/IEC 10118-3 (see Clause 5).

8.8.2.1.3 The choices for MAC are those defined in ISO/IEC 9797-1 and ISO/IEC 9797-2 (see Clause 7).

The families of hybrid ciphers are:

a) ECIES-HC (uses ECIES-KEM);

b) PSEC-HC (uses PSEC-KEM);

c) ACE-HC (uses ACE-KEM);

d) RSA-HC (uses RSA-KEM).

Each family gives rise to a number of specific asymmetric ciphers depending on the choice of KEM, DEM and associated subsidiary choices such as for block ciphers, key derivation functions, MACs, hash functions, etc.

8.8.2.2 Security of hybrid ciphers

The recommended usage period of a hybrid cipher cannot extend the recommended usage period of any subsidiary algorithms such as block cipher, hash function, etc., that the particular hybrid cipher uses. So, for example, if the hybrid cipher uses DEM1, with the choice of SC being SC1 and with the choice of block cipher TDEA with keying option 1 (2 key TDEA), then referring to Table 3, the usage period of that hybrid cipher is limited to 2010 – 2030 depending on how many plaintext – ciphertext pairs would be available to an attacker.

With the limitation from the previous paragraph, the recommended usage periods for hybrid ciphers are as specified in Table 5.

ACE-KEM has an advantage of only relying on second pre-image resistance of the hash function, but it is slower than ECIES-KEM and PSEC-KEM.

8.8.3 RSAES

RSAES has the severe restriction on the size of the message that can be encrypted, that is, encoded as a hexadecimal number, must be smaller than the RSA modulus used for encipherment. This in general means it can only be used to encipher very small plaintexts. As a key encipherment mechanism, RSA-KEM is considered more secure and efficient than RSAES, which is therefore not recommended except for legacy applications that are already using it. The recommended usage in this case is as specified in Table 5.

8.8.4 HIME(R)

HIME(R) has received less scrutiny in the cryptographic community than RSA-HC, and is therefore not recommended except for legacy applications that are already using it (see also Reference [56]). The general HIME(R) algorithm allows the modulus $p^d q$, where p and q are primes, and d is a natural number greater than 1. However, the authors strongly recommend $d = 2$. The recommended usage is as specified in Table 5.

9 Random number generation

Central to most cryptographic algorithms is the generation of random numbers. Random numbers are generated as part of key generation in practically all cryptographic algorithms that use private or secret keys, but are used in many other contexts, such as when generating challenges, random padding, salt values, etc.

The specific cryptographic requirements to random numbers may vary with the application, but in the absence of evidence to the contrary, it should be assumed that the requirements are as specified in ISO/IEC 18031.

The principles and mechanisms for random number generation defined in ISO/IEC 18031 are appropriate. NIST SP 800-90 is also relevant, and parts 2 to 4 of ANSI X9.82 may prove helpful when published. These documents are not published at the same time, and so the latest of them may provide the most up-to-date information and guidance.

Annex A
(informative)

Entity authentication and key management mechanisms

A.1 General

This annex describes entity authentication, key establishment, agreement, and transport mechanisms.

An authentication mechanism is used to corroborate that an entity is the one that is claimed. An entity to be authenticated shows its identity by proving knowledge of a secret.

Recommended authentication mechanisms are taken from ISO/IEC 9798-1, ISO/IEC 9798-2, ISO/IEC 9798-3, ISO/IEC 9798-4, ISO/IEC 9798-5 and ISO/IEC 9798-6.

The kinds of authentication mechanisms are:

a) authentication mechanisms using symmetric encipherment algorithms;

b) authentication mechanisms using digital signature techniques;

c) authentication mechanisms using a cryptographic check function;

d) authentication mechanisms using zero knowledge techniques;

e) authentication mechanisms using manual data transfer.

The choice of mechanism depends on several factors, including:

— which cryptographic mechanisms are available, such as symmetric ciphers, digital signature algorithms, MAC algorithms etc.,

— the specific security requirements, such as for unilateral or mutual authentication,

— the environment, or components involved, e.g. the devices used when performing authentication using manual data transfer,

— if the entities share a secret key in advance,

— the availability or not of a trusted third party.

A.2 Authentication mechanisms using symmetric encipherment algorithms

The authentication mechanisms using symmetric encipherment algorithms are taken from ISO/IEC 9798-2. They are:

a) mechanisms not involving a third party:

   1) unilateral authentication:

      i) one pass authentication,

      ii) two pass authentication;

   2) mutual authentication:

      i) two pass authentication,

      ii) three pass authentication;

b) mechanisms involving a trusted third party:

   1) four pass authentication,

   2) five pass authentication.

All mechanisms depend on a choice of symmetric cipher. The symmetric cipher should be taken from the list of recommended block ciphers provided in 4.3, consistent with the recommendations for usage dates.

Several of the mechanisms also depend on the use of time-variant parameters, such as time-stamps, sequence numbers, or random numbers. In these cases, the time-variant parameters should be chosen such that it is unlikely that they will repeat within the lifetime of the secret authentication key that is used.

A.3 Authentication mechanisms using digital signature techniques

The authentication mechanisms using digital signature techniques are taken from ISO/IEC 9798-3. They are:

a) unilateral authentication:

   1) one pass authentication,

   2) two pass authentication;

b) mutual authentication:

   1) two pass authentication,

   2) three pass authentication,

   3) two pass parallel authentication.

The mechanisms in the list above depend on a choice of digital signature scheme. The digital signature scheme should be taken from those listed in 8.5 and 8.7, consistent with the recommendations for usage dates.

Several of the mechanisms also depend on use of time-variant parameters such as time-stamps, sequence numbers, or random numbers. In these cases, the time-variant parameters should be chosen such that it is unlikely that they will repeat within the lifetime of the private authentication key(s) used.

A.4 Authentication mechanisms using a cryptographic check function

The authentication mechanisms using a cryptographic check function are taken from ISO/IEC 9798-4. They are:

a) unilateral authentication:

   1) one pass authentication,

   2) two pass authentication;

b) mutual authentication:

   1) two pass authentication,

   2) three pass authentication.

The mechanisms in the list above depend on a choice of MAC function. The MAC function should be taken from those listed in Clause 7, consistent with the recommendations for usage dates.

The choice of length of MAC should follow the recommendations in 7.4.

The parties involved in the authentication (claimant and verifier) share an authentication key. Several of the mechanisms also depend on use of time-variant parameters such as time-stamps, sequence numbers, or random numbers. In these cases, the time-variant parameters should be chosen such that it is unlikely that they will repeat within the lifetime of the authentication key.

A.5 Authentication mechanisms using zero knowledge techniques

A.5.1 General

The authentication mechanisms using zero knowledge techniques are taken from ISO/IEC 9798-5.

Zero knowledge authentication mechanisms aim to enable a claimant to prove possession of a private key to a verifier who has a trusted copy of a corresponding public key, without giving the verifier or anybody else any knowledge about the private key.

One typical application would be that the public key is certified by a trusted Certification Authority, and proof of possession of the corresponding private key implies that the claimant has successfully passed the validation that the Certification Authority conducts prior to issuing a certificate.

The zero-knowledge authentication mechanisms from ISO/IEC 9798-5 are:

a) FS (based on identities);

b) GQ1 (based on identities);

c) GQ2 (based on integer factorization);

d) SC (based on discrete logarithms with respect to prime numbers);

e) GPS1 (based on discrete logarithms with respect to composite numbers);

f) GPS2 (based on discrete logarithms with respect to composite numbers);

g) GPS3 (based on discrete logarithms with respect to composite numbers);

h) mechanisms based on asymmetric encipherment systems.

A.5.2 FS

The recommendations cover both the first and second mode of use.

A hash function must be chosen. It must be chosen for collision resistance (see Clause 5) among the hash functions in Table 4.

An implementation must also select a format mechanism. PSS without salt is recommended (see ISO/IEC 14888-2).

Security parameters that must also be chosen are:

a) the modulus size, a;

b) the pair multiplicity parameter, m;

c) the exchange multiplicity parameter, t.

The modulus size a should be chosen according to the column for $n = pq$ from Table 5. If an adversary can factor a modulus, the adversary can impersonate a claimant for the public key with that modulus.

The likelihood of an adversary randomly guessing the correct response(s) to a verification challenge is 2^{-mt}, and so m and t should be chosen such that this probability is acceptable.

The total number of possible challenges should be limited to 2^{40}. If this recommendation is not followed, then special care should be taken to prevent the verifier using the claimant as a signing oracle.

A.5.3 GQ1

The recommendations cover both the first and second mode of use.

A hash function must be chosen. It must be chosen for collision resistance (see Clause 5) among the hash functions in Table 4.

An implementation must also select a format mechanism. PSS without salt is recommended (see ISO/IEC 14888-2). The format mechanism must use the selected hash function.

Security parameters that must also be chosen are:

a) the modulus size, a;

b) the exchange multiplicity parameter, t.

The modulus size a should be chosen according to the column for $n = pq$ from Table 5. If an adversary can factor a modulus, the adversary can impersonate a claimant for the public key with that modulus.

The likelihood of an adversary randomly guessing the correct response(s) to a verification challenge is 2^{-t}, and so t should be chosen such that this probability is acceptable.

The total number of possible challenges should be limited to 2^{40}. If this recommendation is not followed, then special care should be taken to prevent the verifier using the claimant as a signing oracle.

Additionally, an RSA verification exponent must be chosen. Recommended values are 257, $(2^{16} + 1)$, $(2^{72} + 15)$, $(2^{36} + 2^{13} + 1)$, $(2^{40} + 15)$. In particular, where speed of transaction is an issue, the first two values are recommended.

A.5.4 GQ2

The recommendations cover both the first and second mode of use.

Depending on the implementation choices, an implementation may or may not use a hash function. If a hash function is used, the hash function should be chosen for collision resistance (see Clause 5) among the hash functions in Table 4.

Security parameters that must also be chosen are:

a) the modulus size, a;

b) the security parameter, k;

c) the pair multiplicity parameter, m.

The modulus size a should be chosen according to the column for $n = pq$ from Table 5. If an adversary can factor a modulus, the adversary can impersonate a claimant for the public key with that modulus.

The likelihood of an adversary randomly guessing the correct response(s) to a verification challenge is 2^{-km}, and so k and m should be chosen such that this probability is acceptable. Recommended maximum values are $k = 5$ and $m = 8$.

The total number of possible challenges should be limited to 2^{40}. If this recommendation is not followed, then special care should be taken to prevent the verifier using the claimant as a signing oracle.

A.5.5 SC

Depending on the implementation choices, an implementation may or may not use a hash function. If a hash function is used, the hash function should be chosen for collision resistance (see Clause 5) among the hash functions in Table 4.

Security parameters that must also be chosen are:

a) size L of the modulus p;

b) size N of the prime number q;

c) δ from the interval {8, 9, …, 40}, possibly greater depending on the application.

It is recommended to choose L and N in accordance with Table 5, using the column for integer-based DL.

The number δ denotes the number of bits for challenges. A cheater can succeed in a masquerade by guessing the challenge in advance. Thus, if all challenges are equally probable, the chance of success of a cheater is $2^{-\delta}$. It depends on the application what is a reasonable level confidence that a successful verification is not the successful guess by a cheater. ISO/IEC 9798-5:2009, Annex C, has more advice on parameter selection.

The total number of possible challenges should be limited to 2^{40}. If this recommendation is not followed, then special care should be taken to prevent the verifier using the claimant as a signing oracle.

A.5.6 GPS1 and GPS2

Depending on the implementation choices, an implementation may or may not use a hash function. If a hash function is used, the hash function should be chosen for collision resistance (see Clause 5) among the hash functions in Table 4.

Security parameters that must also be chosen are:

a) α, the size of the modulus, in bits;

b) σ, the size of private numbers, in bits;

c) ρ, the size of random numbers, in bits;

d) δ from the interval {8, 9, …, 40}, possibly greater depending on the application.

In addition, a base g has to be chosen. The value $g = 2$ has some practical advantages.

The number α should be chosen according to the recommendations for the values of L in Table 5, using the column for integer-based DL.

The numbers σ and ρ should be chosen according to the recommendations for the values of N in Table 5, using the column for integer-based DL.

The number δ denotes the number of bits for challenges. A cheater can succeed in a masquerade by guessing the challenge in advance. Thus, if all challenges are equally probable, the chance of success of a cheater is $2^{-\delta}$. It depends on the application what is a reasonable level confidence that a successful verification is not the successful guess by a cheater. ISO/IEC 9798-5:2009, Annex C, has more advice on parameter selection.

The total number of possible challenges should be limited to 2^{40}. If this recommendation is not followed, then special care should be taken to prevent the verifier using the claimant as a signing oracle.

A.5.7 GPS3

Depending on the implementation choices, an implementation may or may not use a hash function. If a hash function is used, the hash function should be chosen for collision resistance (see Clause 5) among the hash functions in Table 4.

Security parameters that must also be chosen are:

a) α, the size of the modulus, in bits;

b) σ, the size of private numbers, in bits;

c) ρ, the size of random numbers, in bits;

d) δ from the interval {8, 9, …, 40}, possibly greater depending on the application.

In addition, a public number G and a verification exponent v have to be chosen. The value $G = 2$ has some practical advantages. The number v must be a prime, and be greater than 2^{δ}, for example, if $\delta = 40$, then a recommended value for v is $2^{40} + 15$ (a prime number).

The number α should be chosen according to the recommendations for the values of L in Table 5, using the column for integer-based DL.

The numbers σ and ρ should be chosen according to the recommendations for the values of N in Table 5, using the column for integer-based DL.

The number δ denotes the number of bits for challenges. A cheater can succeed in a masquerade by guessing the challenge in advance. Thus, if all challenges are equally probable, the chance of success of a cheater is $2^{-\delta}$. It depends on the application what is a reasonable level confidence that a successful verification is not the successful guess by a cheater. ISO/IEC 9798-5:2009, Annex C, has more advice on parameter selection.