ISO/IEC JTC 1/SC 27 N16827

SD12

REPLACES: N16737

ISO/IEC JTC 1/SC 27

Information technology - Security techniques

Secretariat: DIN, Germany

DOC TYPE:	Standing Document
TITLE:	**Draft of ISO/IEC JTC 1/SC 27 Standing Document No. 12 (SD12) on the Assessment of Cryptographic Techniques and Key Lengths, 4th edition**
SOURCE:	SD12 Editors (G.Vidal, H. von Sommerfeld and B. Poletti)
DATE:	2016-12-25
PROJECT:	SC 27 SD12
STATUS:	Draft Revised SD12
PLEASE NOTE:	**This document is also freely accessible from the public SC 27 web site at: http://www.din.de/go/jtc1sc27 / Downloads**
ACTION ID:	FYI
DUE DATE:	
DISTRIBUTION:	P-, L-, O-Members W. Fumy, SC 27 Chairman M. De Soete, SC 27 Vice Chair E. Humphreys, T. Chikazawa, M. Bañón, J. Amsenga, K. Rannenberg, WG Convenors
MEDIUM:	http://isotc.iso.org/livelink/livelink/open/jtc1sc27
NO. OF PAGES:	1 + 10

Secretariat ISO/IEC JTC 1/SC 27 -
DIN Deutsches Institut für Normung e. V., Am DIN-Platz, Burggrafenstr. 6, D-10787 (postal D-10772) Berlin, Germany
Telephone: + 49 30 2601-2652; Facsimile:+ 49 30 2601-1723; E-Mail: krystyna.passia@din.de;
HTTP://www.din.de/go/jtc1sc27

Draft of ISO/IEC JTC 1/SC 27 Standing Document No. 12 (SD12) on the Assessment of Cryptographic Techniques and Key Lengths 4th edition

1 Disclaimer

The intention of this standing document is to provide general references to cryptographic techniques and key length selection sources as well as to present general background information that is applicable to the use of ISO/IEC cryptography standards. The references may be useful for users of the standards, but do not necessarily reflect the approval or disapproval of certain key-lengths or cryptographic techniques standardized within ISO/IEC, and particularly ISO/IEC JTC 1/SC 27. Applicable cryptographic techniques and key lengths are reflected by the International Standards developed within ISO/IEC JTC 1/SC 27 and should be referenced if clarification is required.

2 Security analysis of two-key and three-key Triple-DES

2.1 General

The following is an analysis of three-key Triple-DES (i.e. TDEA keying option 1) and two-key TripleDES (i.e. TDEA keying option 2) in reference to ISO/IEC 18033-3:2005 *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers* subclause 4.1.2 'TDEA keying options'

The commonly known Triple-DES (Data Encryption Standard) is addressed as Triple Data Encryption Algorithm (TDEA) by ISO/IEC 18033-3:2005 subclause 4.1 'TDEA' (see [7]). The TripleDES is a symmetric cipher that can process data blocks of 64 bits, using cipher keys with length of 128 (or 192) bits, of which 112 (or 168) bits can be chosen arbitrarily, and the rest may be used for error detection.

The effective strength of three-key Triple DES is at most 112 bits. The remainder of this clause focuses on the effective strength of two-keyTriple DES.

2.2 The security of two---key Triple-DES

It is well-known that two-key Triple-DES can be attacked in ways more efficient than a simple exhaustive search through the entire key space; such an exhaustive search attack would require of the order of 2^{112} encryption operations, and would be infeasible with current technology (and for many years to come). However, as described by van Oorschot and Wiener in [14], if 2^m plaintext/ciphertext pairs are known, all computed using the same secret key K, then K can be determined using of the order of 2^m storage and 2^{120-m} encryption/decryption operations. This is faster than the exhaustive search, as long as at least 2^8 (=256) plaintext/ciphertext pairs are known. As an example, in the case where as many as 2^{32} plaintext/ciphertext pairs are known (i.e. m=32),

the attack requires 'only' 2^{88} encryption/decryption operations. Since this is still well beyond practical reach (2010), it is debatable whether this has a major impact on the security of two-key Triple-DES in practical applications. If a single key is only ever used to encrypt, say, 1000 plaintext blocks, then the van Oorschot-Wiener attack is essentially no better than the brute force exhaustive search attack referred to above. Hence, depending on the required security level, the maximum number of plaintexts encrypted under a single key should be limited.

An important aspect to keep in mind is that DES (and hence Triple-DES) is a 64-bit block cipher. Generally a block cipher, with n being the block length, should not be used with the same key for more than $2^{(n/2)}$ encryptions. This implies that DES (including two-key Triple-DES and three-key Triple-DES options) should not be used to encrypt more than 2^{32} bytes of data anyway. Recently proposed modes of operation for block ciphers have been proposed that causes less concern when encrypting more than $2^{(n/2)}$ blocks of plaintext (see for instance the article "New block cipher modes of operation with beyond the birthday bound security" [15]).

2.3 Summary

The security level of two-key Triple-DES may be assessed at less than the number of bits in the keys (112 bits) because of a "meet-in-the-middle" attack on the Triple-DES construction. An attacker with access to 2^{40} plaintext/ciphertext pairs encrypted under the same key will benefit from an exhaustive key search equivalent to 80 bits key search according to the attack described in [14]. The implication of this is that the effective key-length of two-key Triple-DES in specific applications can only be regarded as 80 bits (instead of 112 bits).

For many practical applications this degradability of the effective key-length is not necessarily a problem as access to 2^{40} plaintext/ciphertext pairs encrypted under a single key is quite unlikely. However, conservative system security design will not encrypt a too huge amount of data under the same key before re-keying.

3 General references to cryptographic algorithm and key length selection

The selection of cryptographic algorithms and/or key-lengths is not an exact science. For this reason one may find conflicting recommendations for cryptographic algorithm and key length selection. The cryptographic algorithm and key length selection amongst other things depend on:

- The specific area of application (for example government vs commercial applications),
- The period of time the protected data shall remain secure,
- The amount of data that is to be encrypted without rekeying,
- Cost-relevant aspects such as the period of time a specific piece of equipment is expected to be used before a planned equipment upgrade can take place,

- Additional security margin requirements.

The security analyst should determine the security requirements to determine an appropriate key length and an appropriate cryptographic algorithm that fit the cost-relevant requirements and meet the margins of derived security parameters. The documents and web-sites listed in the bibliography contain background information which can aid in the cryptographic algorithm and key length selection process (see 5 below).

References [1] and [2] are of a general theoretical nature, while reference [3], [8], [9] and [10] contain recommendations which are aimed at specific applications or sectors. Reference [4] contains more references itself to key length selection criteria and also provide implementations of calculations of the various references above (and more) which may aid in key length selection. Reference [8] is a technical report specifically aimed at the financial services sector and takes interoperability into account in its recommendations.

4 Block length selection

When selecting block ciphers, an often overlooked aspect is the block size. Ciphers have been suggested recently with block sizes as low as 32-bits, whilst 64-bit block sizes used to be the norm. Nowadays 128-bit blocks are the norm for general purpose block ciphers. Block sizes for modern block ciphers can go as high as 192-bit blocks and 256-bit blocks.

Currently there is no indication in academic literature that block length and key length has to be related. Therefore, assume that no attacks exist against block cipher *A* utilising a block size of e.g. 64-bits and key size of 128-bits. Also assume that no attacks exist against block cipher *B* utilising a block size of 128-bits and a key size of 128-bits. When implemented correctly, both block cipher *A* and block cipher *B* provide 128-bits of effective security, i.e. in both cases the only attack would be a brute force search through the entire space of 2^{128} keys.

The difference between the two block ciphers lies within the maximum amount of plaintext that can be encrypted with a single key before rekeying takes place. Generally, for a block cipher with block size of n-bits, the maximum amount of plaintext that can be encrypted before rekeying must take place is $2^{(n/2)}$ blocks, due to the birthday paradox. By this reason this number is called the birthday bound. As long as the implementation of a specific block cipher do not exceed these limits, using the block cipher will be safe.

In the above example, block cipher *A* can only encrypt 2^{32} blocks of plaintext with a single key before rekeying should take place, whilst block cipher *B* can encrypt 2^{64} blocks of plaintext with a single key before rekeying should take place.

Recently McGrew [22] presented attacks against n-bit block ciphers in CBC, CFB, and CTR modes that can recover an unknown plaintext values when the birthday bound is not respected. The collision-based attacks against CBC and CFB are straightforward and relatively inexpensive to carry out against 64-bit block ciphers; attacks against CTR are more involved, but are still feasible.

Additionally, according to [23] the birthday attack for the 64-bit block cipher requires about 800GB data and about 20-40 hours. While the attack for 64-bit block ciphers requires still huge data and long time, similar attacks for the 32- or 48-bit block cipher require only a few MB data with a few seconds or a few GB data with a few minutes, respectively.

Consequently, in order to prevent birthday attack in real world, block ciphers which block size is less than 64 bits are not recommended.

Finally, it is important to remark that in the case that a message authentication code (MAC) is based on a symmetric key block cipher, as seen discussions in NIST SP-800 38B, the default recommendation is to limit the key to no more than 2^48 messages when the block size is 128 bits and 2^21 messages when the block size is 64 bits in order to satisfy that the collision probability is less than one in a billion.

5 Related key attacks

5.1 General

Related-key attacks against a block-cipher rely on the following assumption: it is possible for an attacker to encrypt or decrypt messages under several different keys whose values are initially unknown, but where some mathematical relationship connecting the keys is known to the attacker.

This is a strong assumption which is not relevant for a large number of practical applications such as encryption or MAC using a long-term key. However, for "ad-hoc" applications which make use of a block-cipher known to be vulnerable to one or several related-key attacks, it is strongly recommended to assess the impact on the security. Indeed, it should not be possible for an attacker to encrypt/decrypt messages under several different keys by controlling the mathematical relationship connecting the keys in such a way that the constraints imposed by the attack (e.g. the number of related-keys required to mount the attack, the mathematical relationship connecting the keys required to mount the attack, etc.) are realistic.

As an illustration, it is not recommended to use a block cipher *A* which is known to be vulnerable to one or several related-key attacks as a cryptographic primitive in an ad-hoc authentication protocol such that the encryption key of the block cipher *A* is equal to *K* XOR *R* where *K* is a long-term secret key and *R* is a random value under the control of an eventual attacker.

The original ideas of related key attacks were introduced by Biham [17] and Knudsen [18], and many more articles followed.

5.2 Example

As an example the following trivial related key attack is applicable to all block ciphers.

A block cipher encrypts one plaintext under an unknown key *K*. The attacker can modify the key *K* to form key K_i by setting K_i equal to the logical AND between *K* and the bit pattern containing all ones except in bit position *i*. This operation must be performed by the system (or in a more formal model, by an Oracle).

The same plaintext is encrypted again, and if the corresponding ciphertext differs from the sample ciphertext obtained from key *K,* the attackers knows that bit position *i* of *K* is a one, otherwise bit position *i* is a zero. By repeating the steps for each of the bits of *K,* the entire key can be recovered by the attacker. If *K* contains *n*-bits, the attacker requires *n* masks (or calls to the Oracle), and *n*+1 encryption calls.

The attack described above is described in terms of block ciphers, but is clearly applicable to more than just block ciphers. In many practical applications one does not expect such modifications to the encryption key to be allowable by the system. The described attack is the most basic form of related key attack, but many more exist that also rely on the structure of the particular block cipher.

6 Possible defects in International Standards

6.1 General

This clause deals with perceived defects that came to the attention of ISO/IEC JTC 1/SC 27 on its cryptography standards and possible ways to deal with these defects should there be concern for their continued use.

International Standards may contain cryptographic techniques for which, after publication, concerns are raised as to possible defects that may exist in the mechanisms. In some cases, these perceived defects are in fact not defects, but rather concerns expressed by the community (such as constants used in the mechanism which were generated in an unknown way). ISO/IEC JTC 1/SC 27 usually initiates a study period to which its liaison organizations, experts and National Bodies contribute. The outcome of such a study period can then either

1. Confirm the defect.

2. Prove the defect to be invalid.

3. Neither prove nor confirm the defect, but propose mitigation techniques.

In the case of 1. and 3. a Technical Corrigendum to the respective International Standard will be published. In all the cases further information can be made available in this document if appropriate.

6.2 MASH-1 hash function

ISO/IEC 10118-4:1998 *Information technology – Security techniques – Hash functions using an n-bit block cipher* contains a mechanism called MASH-1. To the knowledge of ISO/IEC JTC 1/SC 27 experts a paper was published at the pre-proceedings of CTCrypt 2013 (also available on ePrint [16]). According to the paper it is possible to select weak moduli for use in MASH-1 which may lead to a weakening of the collision resistance of MASH-1. It is however possible to choose the moduli carefully so that this does not happen. A Technical Corrigendum is currently being produced by ISO/IEC JTC 1/SC 27 which will contain more detail on how to avoid choosing weak moduli.
Alternatively, users can study the original paper (see [16]) on how to avoid generating weak moduli.

6.3 Dual elliptic curve deterministic random bit generator (Dual_EC_DRBG)

ISO/IEC 18031:2011 *Information technology – Security techniques – Random bit generation* [6] contains a number of random bit generator mechanisms, one of which is known as the Dual_EC_DRBG. The same mechanism with identical application specific constants is specified in

U.S. National Institute of Standards and Technology (NIST) Special Publication 800-90A, which was a major contribution in establishing the International Standard.

Recent community commentary has called into question the trustworthiness of this mechanism. In particular these comments relate to the default elliptic curve points i.e. the default application specific constants that are provided in Annex D of this International Standard.

To the knowledge of ISO/IEC JTC 1/SC 27 experts no confirmation has surfaced to date that the default application specific constants actually lead to a compromise of the mechanism. However, there is sufficient evidence of a security issue. NIST has released a supplemental bulletin [12] giving advice about use of the mechanism and further U.S. federal government recommendations. In the outcome of the SC 27 Korea 2013 meeting ISO/IEC JTC 1/SC 27 has initiated a study period to carefully review the security issues and possible ramifications for the International Standard. This process runs in parallel to the NIST public comment period on the same mechanism and includes all national bodies and their experts subscribing that are members of ISO/IEC JTC 1/SC 27. The outcome of this study period may lead to a future revision of the International Standard.

ISO/IEC JTC 1/SC 27 is aware that Dual_EC_DRBG is included in many cryptographic libraries. Implementers and users of this mechanism should conduct a risk analysis of their security products, services and/or systems and may decide to take one of the following actions:

1. Continue using Dual_EC_DRBG if the perceived risk does not impact the security of their products, services and/or systems.

2. Generate their own application specific constants for application in Dual_EC_DRBG, and continue to use this random bit generator mechanism with the new constants.

3. Stop using Dual_EC_DRBG and replace it with another pseudo random bit generator from ISO/IEC 18031 *Information technology – Security techniques – Random bit generation* [6].

However, the second option (i.e. generate new specific constants) would require the appropriate level of technical expertise and some specific knowledge of Elliptic Curve Cryptography, and is only possible if the application is designed for allowing new constants to be specified.

ISO/IEC JTC 1/SC 27 expert discussions on [6], [20] and [21] during the SC 27 Korea 2013 meeting perceived, there is sufficient evidence of a security issue. The security of the Dual_EC_DRBG is based entirely on the property of two specific Dual_EC_DRBG system parameters (i.e. the a.m. application specific constants which are points on a given elliptic curve) to be chosen independently and random. As long as there is no confidence that the two parameters are actually chosen in this way, the Dual_EC_DRBG should be considered 'compromised'.

To the knowledge of the experts there is no proof positive that the default application specific constants given in [6] meet the requirements.

In the outcome of the SC 27 Hong Kong 2014 meeting the Dual_EC_DRBG mechanism (and associated material) will be removed from the text of ISO/IEC 18031, and a corrigendum will be prepared to achieve this objective. Further information the Dual_EC_DRBG mechanism will be made available in a new edition of SD 12 in parallel to publication of the corrigendum if appropriate.

6.4 Multivariate quadratic deterministic random bit generator (MQ_DRBG)

Another mechanism which is specified in ISO/IEC 18031:2011 *Information technology – Security techniques – Random bit generation* is multivariate quadratic deterministic random bit generator – MQ_DRBG. Particularly, this mechanism specifies restrictions on parameters of multivariate quadratic equations that are used for bit generation. However, the proof of security of the generator strongly relies on the randomness of the choice of the set multivariate quadratic equations. As it is shown in [19], the security level of the generator could be lower than specified in the International

Standard, if the equations are not randomly selected. It is therefore required to randomly choose the equations in order to ensure the security of generator.

7 Bibliography

[1] Arjen K. Lenstra and Eric R. Verheul, Selecting Cryptographic Key Sizes, PKC2000: p. 446-465, 01/2000.

[2] European Network of Excellence in Cryptology, ECRYPT2 Yearly Report on Algorithms and Keysizes (2009-2010), D.SPA.13.

[3] H. Orman and P. Hoffman, Determining Strengths for Public Keys Used for Exchanging Symmetric Keys, RFC 3766, 04/2004.

[4] <http://www.keylength.com>.

[5] ISO/IEC 10118-4:1998, Information technology – Security techniques – Hash functions using an n-bit block cipher

[6] ISO/IEC 18031:2011 Information technology – Security techniques – Random bit generation

[7] ISO/IEC 18033-3:2005, Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers.

[8] ISO TR 14742, Financial services — Recommendations on cryptographic algorithms and their use, (ISO TC 68, Technical Report).

[9] National Security Agency (US), Fact Sheet Suite B Cryptography, 08/2009.

[10] NIST Special Publication 800-57, Recommendation for Key Management, Part 1: General (Revised), March 2007.

[11] NIST Special Publication 800-90A.

[12] NIST Special Bulletin <http://csrc.nist.gov/publications/nistbul/itlbul2013_09_supplemental.pdf>.

[13] NIST online document <http://csrc.nist.gov/groups/STM/cavp/documents/drbg/drbgval.html>.

[14] P. C. van Oorschot and M. J. Wiener, 'A known-plaintext attack on two-key triple encryption'. In I. B. Damgård, ed., Proc. Eurocrypt '90, Springer-Verlag LNCS 473 (1996) page 318-325.

[15] Robshaw, M (Ed), Fast Software Encryption 2006.

[16] V. G. Antipkin, "Smashing MASH-1", *Math. Asp. of Crypt.*, **5**:2 (2014), 21–28

[17] Biham E, New types of cryptanalytic attacks using related keys." Journal of Cryptology, 4, Springer.

[18] Knudsen L.R., Cryptanalysis of LOKI91, Advances in Cryptography, Asiacrypt '92, LNCS 718, Springer-Verlag.

[19] Vladimir Drelikhov, Grigory Marshalko, Alexey Pokrovsky, On the security of MQ_DRBG, <eprint.iacr.org/2011/548>.

[20] Dan Shumow, Niels Ferguson, On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng. CRYPTO Rump Session 2007. Microsoft, <http://rump2007.cr.yp.to/15-shumow.pdf>.

[21] United States Patent Application Publication, US 2007/0189527 A1, Aug. 16, 2007, “Elliptic Curve Random Number Generation”.

[22] David McGrew, Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64 bit block cipher modes" <eprint.iacr.org/2012/623>

[23] Karthikeyan Bhargavan and Gaëtan Leurent, On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN, <eprint.iacr.org/2016/798>

