وزارة المواصلات والاتصالات
MINISTRY OF TRANSPORT AND COMMUNICATIONS

Qatar National Cryptographic Standard

Version: 1.0

Author: Public Key Management and Digital identity Section

Document Classification: Public

Published Date: October 2019

Document History:

Date	Issue	Status	Author
11/06/2018	0.1	First draft for review	PKI Team
20/10/2018	0.2	First internal review	PKI Team
16/01/2019	0.3	Second internal review	PKI Team
03/05/2019	0.4	Update draft for review	PKI Team
20/06/2019	0.5	Update final draft for review	PKI Team
07/07/2019	0.9	Update draft	PKI Team
15/07/2019	BeTa Version	Request For Comments	CSP-PMA Director
02/10/2019	BeTA Version	Update Beta Version	PKI Team
02/10/2019	Beta Version	Document changed Name	PKI Team
07/10/2019	BeTA Version	Internal Review	PKI Team
27/10/2019	1.0	Final version: Approval	CSP-PMA Director

Content

I. Introduction

The Qatar National Cryptographic Technical Standard sets the basis for a sound and lawful use of cryptography in the State of Qatar. It aims, amongst other goals, to provide recommendations for the safe use of cryptography, and proposes algorithms and parameters suitable for defined periods.

The State of Qatar follows recommendations set forth in this document. This document rather addresses primarily, in a mandatory manner, by both the private and public sectors who plan to implement cryptographic systems.

Therefore, this document deliberately dispenses with the specification of current cryptographic mechanisms, which are still considered to be secure at the time of the publication of this document, but can not be recommended in the future, as they have theoretical vulnerabilities that can be exploited in the long term.

II. Audience

This standard is intended for all organizations in Qatar and others who are responsible for providing and using cryptographic services to meet identified security requirements. This document might be used by:

- Management responsible for selecting and integrating cryptographic mechanisms into a system.
- Technical specialist that has been requested to select one or more cryptographic methods/techniques to meet a specified requirement.
- Procurement specialist developing a solicitation for a system, network or service that will require cryptographic methods to perform security functionality.
- Users of cryptographic services.

The goal is to provide these organizations with sufficient information to allow them to make informed decisions about the cryptographic methods that will meet their specific needs to protect the confidentiality and integrity of data that is transmitted and/or stored in a system or network, as well as to obtain assurance of its authenticity.

This document is not intended to provide a technical discussion on the mathematics of cryptography and cryptographic algorithms.

III. Scope

This document covers cryptographic methods[1] (symmetric and asymmetric cryptographic scheme, Hash functions, Signature Algorithms and cryptographic protocols). While all organizations in Qatar is required to use this technical standard when applicable.

This document provides information on selecting and using cryptography in all systems.

- **Electronic signature, PKI Certificates and Key Management are not part of this document.**

[1] Electronic signature, PKI Certificates and Key Management are not part of this document.

IV. Terms and Definitions

The following terms and definitions are used in this document:

- **Algorithm**: A clearly specified mathematical process for computation; a set of rules that, if followed, will give a prescribed result.
- **Authentication**: A process that provides assurance of the source and integrity of information that is communicated or stored or the identity of an entity interacting with a system. Note that in common practice, the term "authentication" is used to mean either source or identity authentication only.
- **Block cipher algorithm**: A family of functions and their inverse functions that is parameterized by cryptographic keys; the functions map bit strings of a fixed length to bit strings of the same length.
- **Certificate**: A set of data that uniquely identifies an entity, contains the entity's public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity identified in the certificate. Additional information in the certificate could specify how the key is used and the validity period of the certificate.
- **Cipher text**: Data in its encrypted form.
- **Cryptographic algorithm**: A well-defined computational procedure that takes variable inputs, including a cryptographic key (if applicable), and produces an output.
- **Cryptographic boundary**: An explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software and/or firmware components of a cryptographic module.
- **Cryptographic hash function**: A function that maps a bit string of arbitrary length to a fixed-length bit string.
- **Cryptographic key**: A parameter used in conjunction with a cryptographic algorithm that determines its operation in such a way that an entity with knowledge of the key can reproduce or reverse the operation, while an entity without knowledge of the key cannot.
- **Cryptographic module**: The set of hardware, software and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within a cryptographic boundary.
- **Decryption**: The process of changing ciphertext into plaintext using a cryptographic algorithm and key.
- **Encryption**: The process of changing plaintext into ciphertext using a cryptographic algorithm for the purpose of security or privacy.
- **Integrity**: The property that data has not been modified or deleted in an unauthorized and undetected manner.
- **Public key**: A cryptographic key used with a public-key cryptographic algorithm, that is uniquely associated with an entity and that may be made public. In an asymmetric (public) key cryptosystem, the public key is associated with a private key.
- **Public Key Infrastructure (PKI)**: A framework that is established to issue, maintain and revoke public key certificates.

- **Symmetric key**: A single cryptographic key that is used with a symmetric (secret key) algorithm, is uniquely associated with one or more entities and is not made public, i.e., the key is kept secret.

V. Legal mandate

In order to achieve this objective, the State of Qatar has legislated on the use of Cryptography, particularly through:

- The **E-Commerce and E-Transactions law No. (16) of 2010**, applying to transactions between persons who have agreed to conduct transactions using electronic communications.

  The **E-Commerce law** defines general principles and obligations to be respected in order to ensure the protection of consumers in electronic transactions, including notably transparency and communication requirements, and termination rights granted to consumers in electronic transactions.
- The **Resolution No. (01) of 2012 of The Supreme Council of Information and Communication Technology (Now "MOTC" : Ministry of Transport and communications) regarding the Bylaw regulating the work of certification service providers**: The Bylaw regulates the work of certification service providers:
  - Maintaining an infrastructure for public keys.
  - Providing services related to electronic signatures.
  - Issuing electronic certification certificates.

  Similar to the above, the Bylaw also defines general principles and obligations to be respected in order to ensure the protection of consumers in electronic transactions, including notably privacy and data protection obligations, termination rights, and procedures to ensure that adequate complaint mechanisms are available.

This document lists the Cryptographic methods that are considered trustworthy in order to generate confidence in the use of information and communications systems.

VI. Security objectives and selection criteria

The security of cryptographic mechanisms depends primarily on the strength of the underlying algorithms and also the associated Process. For this reason, this document recommends only mechanisms which can be assessed accordingly based on the results available (As of the date of publishing) from long-standing discussions and analysis. Other factors relevant to security are the specific implementations of the algorithms and the reliability of potential background systems, such as required public key infrastructures for the secure exchange of certificates.

The security of the cryptographic mechanisms recommended in this document is assessed without taking the use case into consideration. For specific scenarios, other security requirements may arise.

VII. Protection of privacy and personal data

The fundamental rights of individuals to privacy, including secrecy of communications and protection of personal data, should be respected in national cryptography policies and in the implementation and use of cryptographic methods.

The legislation described in section III cares for this purpose. In addition, the State of Qatar decided that:

- The use of encryption, based on public or secret cryptographic key methods, is unrestricted in the State of Qatar. In order to ensure full effectiveness of encryption, ensuring user privacy and personal data protection, the private or secret key used for decryption MUST:
    - Be under the sole control of the encrypted data recipient (for PKI based encryption methods), and under the sole control of the encrypted data authorized recipient(s) and the user encrypting that data (for secret key based encryption methods), except when a key recovery scheme is implemented by a third party, in which case any key recovery:
        - Occurs on a need-to-know basis only,
        - Is fully controlled by the Third Party without disclosure of the key to external parties,
        - Occurs under conditions clearly defined in "General Terms and Conditions" of use of the Encryption within the context of use, and in the Certificate Policy for PKI based encryption methods,
        - Is necessarily notified to the key owner.
    - Reach a sufficient level of security, by fulfilling the technical requirements related to key management as specified section 6, further complemented by the Cryptographic Algorithms and Parameters.
    - Be unambiguously linked to its owner, by means of a certificate issued by a CSP dully licensed by CSPs-PMA (for PKI based encryption methods), and by the strict limitation of the distribution of secret key to the encrypted data authorized recipient(s) and to the user encrypting that data (for secret key based encryption methods).
- There is currently no mandatory key escrow or key recovery scheme that is enforced by the government for what regards **encryption keys**. However, service providers are encouraged to use confidentiality key recovery schemes themselves to recover encrypted data whenever necessary, so as to minimize possible disruptions to business operations, under the conditions stated above.

- Key escrow or key recovery scheme are strictly forbidden by the government for what regards **signature and authentication keys**. The sole exception concerns licensed trust service providers managing electronic signature creation data on behalf of the signatory. Such trust service providers may duplicate the electronic signature creation data for back-up purposes provided the following requirements are met:
    - The security of the duplicated datasets must be at the same level as for the original datasets.
    - The number of duplicated datasets shall not exceed the minimum needed to ensure continuity of the service.

VIII. Cryptographic Mechanisms

1. Symmetric encryption schemes

Symmetric encryption schemes are used to ensure the confidentiality of data which is exchanged. The authenticity and/or integrity of the data is thus not ensured. In this context, it should be emphasized that even in cases in which the protection of the confidentiality of the data transmitted seems to be the dominating or even sole security objective at first glance, neglecting integrity-securing mechanisms can easily result in vulnerabilities in the overall cryptographic system, which make the system also prone to attacks on confidentiality. Vulnerabilities to some types of active side-channel attacks may arise in such a manner.

This section addresses symmetric methods, in which the encryption and decryption keys are identical (in contrast to asymmetric schemes in which the secret key practically cannot be calculated on the basis of the public key without additional information).

1.1. Block ciphers

A block cipher is an algorithm which encrypts plaintext with a fixed bit length (ex. 128 bits) by means of a key to ciphertext with the same key length. This bit length is also referred to as block size of the cipher. For the encryption of plaintexts with another length, so-called modes of operation are used, see V.1.1.1. For new applications, only block ciphers the block size of which is at least 128 bits should be used. The following block ciphers are recommended for use in new cryptographic systems:

Recommended block ciphers		
AES-128	AES-192	AES-256

Table 1: Recommended block cipher

1.1.1. Modes of operations

As has already been established in Section V.1.1, a block cipher by itself only provides a mechanism for the encryption of plaintexts of a single fixed length. In order to encrypt plaintexts of another length, an encryption scheme for plaintexts of any length must be constructed based on the block cipher by means of an adequate mode of operation. As a further effect of a cryptographically strong mode of operation, it must be mentioned that the resulting encryption scheme will be stronger in some respects than the underlying block cipher, for example if the mode of operation randomizes the encryption process and thus makes it difficult to recognize the same plaintexts that were encrypted several times.

Various modes of operation for block ciphers can initially only handle plaintexts the length of which is a multiple of the block size. In this case, the last block of a given plaintext may still be too short and must be padded accordingly, see Section V.1.1.3 for adequate schemes. Among the modes of operation recommended for block ciphers, however, only the CBC Mode needs a padding step.

The easiest way to encrypt a plaintext the length of which is a multiple of the block size is to encrypt each plaintext block with the same key (this mode of operation is referred to as Electronic Code Book (ECB)). This results, however, in same plaintext blocks being encrypted to the same ciphertext blocks. The ciphertext thus provides at least information on the structure of the plaintext and a reconstruction of parts of the plaintext by means of frequency analysis may become realistic if the entropy per block of plaintext is low. In order to avoid this, the nth cipher block should not only depend on the nth plaintext block and the key used, but also on an additional value, such as the (n − 1)th ciphertext block or a counter. The following modes of operation are adequate for the block ciphers listed under V.1.1:

Recommended modes of operation for block ciphers		
Galois/Counter Mode (GCM)	Cipher Block Chaining (CBC)	Counter Mode (CTR)

Table 2: Recommended mode of operation

1.1.2. Requirements of use

The modes of operation listed under V.1.1.1 require initialization vectors. Furthermore, certain constraints for secure operation must be complied with. These conditions are summarized as follows below.

- **For GCM**: Initialization vectors may not repeat themselves within a key change period. More precisely, two AES encryptions with the same input values (key, message) must not be carried out in the entire mechanism. Any non-compliance with this condition results in a potentially complete loss of confidentiality for the plaintext blocks affected if the repetition occurs as part of the generation of the GCM key stream. For general cryptographic applications, GCM with a length of the GCM tags of at least 96 bits should be used. For special applications, shorter tags can be used as well upon consultation with experts.

- **For CTR**: The counter values must not repeat themselves with the same key. Noncompliance with this condition results in a virtually complete loss of confidentiality.

- **For CBC**: Only unpredictable initialization vectors are to be used.

1.1.3. Padding schemes

As already explained in Section V.1.1.1, the CBC Mode requires an additional padding step: When partitioning a plaintext to be encrypted, it may occur that the last plaintext block is smaller than the block size of the cipher used. Formatting realized by filling this last block in order to

achieve the size required is also referred to as padding. The following padding schemes are recommended:

Recommended padding schemes for block ciphers		
ISO padding	Padding according to RFC 5652 (6.3)	ESP padding

Table 3: Recommended Padding Schemes

1.2. Legacy Block ciphers

Mechanisms which are not listed are not necessarily considered by QPKI to be insecure. There are no negative findings regarding these block ciphers, but the security of these block ciphers has been examined far less intensively than that of the AES.

- **3DES**: Comes in two variants; a two key version with a 112-bit key and a three key version with a 168-bit key. The effective key length of three key 3DES is 112 bits and not 168 bits as one would expect. The small block length (64-bits) is a problem in some applications.
  Observation: Due to meet-in-the-middle attacks, the security is not as strong as the key length would suggest. For both variants, related-key attacks with complexity 2^{88} are published.

- **Kasumi**: This cipher, used in 3GPP, has a 128-bit key and 64-bit block size is a variant of MISTY-1. Kasumi is called UIA1 in UMTS and is called A5/3 in GSM.
  Observation: Whilst some provable security against linear and differential cryptanalysis has been established, the cipher suffers from a number of problems. These attacks do not affect the practical use of Kasumi in applications such as 3GPP, however given them we do not advise to use Kasumi in further applications.

- **Blowfish**: This cipher has a 64-bit block size, which is too small for some applications and the reason we only advise it for legacy use. It also has a key size ranging from 32- to 448-bits, which we clearly only endorse using at 80-bits and above for legacy applications. The Blowfish block cipher is used in some IPsec configurations.
  Observation: There have been a number of attacks on reduced round versions but no attacks on the full cipher.

1.3. Stream ciphers

In the case of stream ciphers, a key and an initialization vector are used in the generation of a key stream, which is then bitwise added to the message to be encrypted. At the moment, no dedicated stream ciphers are recommended. AES in counter mode, however, can be understood as a stream cipher.

If a stream cipher is used, it is urgently recommended to protect the integrity of the information transmitted by means of separate cryptographic mechanisms. In the absence of such mechanisms, an adversary can make bit-by-bit changes to the plaintext.

2. Asymmetric encryption schemes

Due to their low efficiency as compared to widely used symmetric schemes, asymmetric encryption schemes are used in practice mostly for the transmission of symmetric keys. The message to be encrypted is encrypted with the public key of the recipient. The recipient can then reverse the encryption using the secret key associated to the public key. It has to be practically impossible to reconstruct the plaintext from the ciphertext without knowing the secret key. This implies in particular that the secret key practically cannot be derived from the public key. In order to safeguard the attribution of the public key to the owner of the corresponding secret key, a public key infrastructure is usually needed. For the specification of asymmetric encryption schemes, the following algorithms are to be defined:

- One algorithm for the generation of key pairs.
- One algorithm for the encryption of data and one algorithm for the decryption of data.

The following Table states the recommended asymmetric schemes and the key lengths:

Recommended asymmetric encryption algorithms			
Encryption Algorithm	ECIES	DLIES	RSA
Length (Bit)	250	2000	2048
Observation	The key length of 2048is expected to remain in conformance with the present document until 2022. For a period of use beyond 2022, it is recommended to use RSA/DLIES keys of a length of 3000 bits in order to achieve a uniform security level in all recommended asymmetric encryption schemes.		

Table 4: Recommended asymmetric encryption algorithms

2.1. ECIES encryption scheme

ECIES stands for **Elliptic Curve Integrated Encryption Scheme**. It is a hybrid encryption scheme. The security of the asymmetric component is based on the Diffie-Hellman problem in the respective elliptic curve used. Below, we describe a version of ECIES that is consistent with the other recommendations of this document. The description of ECIES provided in this document is almost entirely identical to the description of the closely related DLIES scheme in Section V.2.2. The main reason for addressing these two schemes separately is the difficulties which could result from differences in the notations as well as the recommendations regarding secure key lengths which differ for the two schemes. As a normative reference, ECIES-HC in **ISO/IEC 18033-2** is recommended [1].

2.2. DLIES encryption scheme

DLIES stands for **Discrete Logarithm Integrated Encryption Scheme**. It is a hybrid encryption scheme which, in the asymmetric component, is based on the difficulty of solving instances of the Diffie-Hellman problem in a suitable subgroup of F_p^*. A normative description can be found in **IEEE P1363** [2].

2.3. RSA encryption scheme

RSA is an asymmetric cryptographic algorithm. This algorithm has been specified in standards [**ANSI X9.31**] and [**PKCS 1**] and has been adopted for the calculation of electronic signatures in the standard [**FIPS 186-4**].

3. Hash Functions

3.1. Hash function properties

A hash function **H** is a function which, from a message **x** of any size, computes a bit string **H (x)** of fixed size called fingerprint (or hash, or condensed). A hash function is a mathematical function that controls the integrity of a message. A hash function h must verify the following three properties:

- **Collision resistance**: It is computationally infeasible to find two different inputs to the hash function that have the same hash value. That is, if **H** is a hash function, it is computationally infeasible to find two different inputs **x** and **x'** for which **H (x) = H (x')**.
- **Preimage resistance**: Given a randomly chosen hash value, **H_value**, it is computationally infeasible to find an **x** so that **H (x)** = **H_value**. This property is also called the one-way property.
- **Second preimage resistance**: It is computationally infeasible to find a second input that has the same hash value as any other specified input. That is, given an input **x**, it is computationally infeasible to find a second input **x'** that is different from x, such that **H (x) = H (x')**.

3.2. Hash Function recommendation

The list of hash functions in table 5 **shall** be used. The functions shall be implemented as per the reference listed in table 5.

Short hash function name	References
SHA-256	FIPS Publication 180-4 [3]
SHA-384	FIPS Publication 180-4 [3]
SHA-512	FIPS Publication 180-4 [3]
SHA-512/256	FIPS Publication 180-4 [3]
SHA3-256	FIPS Publication 202 [4]
SHA3-384	FIPS Publication 202 [4]
SHA3-512	FIPS Publication 202 [4]

Table 5: Recommended Hash Function

4. Signature Algorithms

In signature algorithms[2], the data to be signed is hashed first and, based on this hash value, the tag and/or the signature is then calculated with the secret key of the prover. The verifier then checks the signature using the public key. As was already the case for asymmetric encryption schemes, it must not be possible in practice to calculate the signature without knowing the private key. This implies in particular that the private key cannot be practically derived from the public key. For the distribution of the public keys to verifiers, a public key infrastructure is usually used.

The list of signature algorithms given in table 6 **shall** be used. The algorithms shall be implemented as per the reference listed in table 6.

[2] For more detail about electronic signature (specification, algorithms and formats), refer to our documents "Electronic Signature: Overview & Specification", "Electronic Signature Algorithms Standard" and "Electronic Signature Formats Standards"

Signature algorithm name	References
RSA-PKCS#1v1_5	IETF RFC 3447 [5]
RSA-PSS	IETF RFC 3447 [5]
DSA (FF-DLOG DSA)	FIPS Publication 186-4 [3] ISO/IEC 14888-3 [6]
EC-DSA (EC-DLOG EC-DSA)	FIPS Publication 186-4 [3]
EC-SDSA-opt	ISO/IEC 14888-3 [6]

Table 6: Recommended Signature Algorithms

The parameters defined in following tables should be used.

DSA Algorithm	1 year (2018)	3 years (2020)	6 years (2023)
Key Length	**2048**	**2048**	**3072**

Table 7: DSA Algorithm parameters

RSA Algorithm	1 year (2018)	3 years (2020)	6 years (2023)
Key Length	**≥ 1900**	**≥ 1900**	**≥ 3000**

Table 8: RSA Algorithm parameters

ECDSA Algorithm	1 year (2018)	3 years (2020)	6 years (2023)
Key Length	256, 384 or 512	256, 384 or 512	256, 384 or 512

Table 9: ECDSA Algorithm parameters

5. Cryptographic protocols

5.1. TLS

The TLS (Transport Layer Security) protocol (current version v1.3) is primarily aimed at securing traffic between an unauthenticated web browser and an authenticated web site, although the protocol is now often used in other applications due in part to the availability (and ease of use) of a variety of libraries implementing TLS. The TLS protocol suite aims to provide a confidential channel rather than simply a key agreement protocol. The protocol has been standardized by the IETF in RFC 8446 [7].

The list of protocol version given in table 10 **shall** be used. The protocols shall be implemented as per the reference listed in table 10.

TLS Protocol version	Reference
V1.2	IETF RFC 5246 [8]
V1.3	IETF RFC 8446 [7]

Table 10: TLS Protocol implementation parameters

The parameters defined the following table states the cipher suites for use with TLS 1.3.

Cipher suite specification
TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256

Table 11: TLS Cipher suite specification

Cipher suite names follow the naming convention "**TLS_AEAD_HASH = VALUE**". Where:

- AEAD: The AEAD algorithm used for record protection.
- HASH: The hash algorithm used.
- VALUE: The two-byte ID assigned for this cipher suite.

5.2. SSH

Secure Shell (SSH) was originally designed as a replacement for insecure remote shell protocols such as telnet. It has now become a more general purpose tool that is used to provide a secure channel between two networked computers for applications such as secure file transfer. In general the host one is connecting to is authenticated, whereas the client is not (although some corporations do insist on client side authentication for SSH usage). SSHv2 was standardized in a collection of RFCs [9] [10] [11] in 2006, and **shall be used**. **OpenSSH** [12] is one of the most widely used implementations of the protocol.

The purpose of this protocol is to perform client user authentication. It assumed that this runs over a secure transport layer protocol, which has already authenticated the server machine, established an encrypted communications channel, and computed a unique session identifier for this session. The transport layer provides forward secrecy for password authentication and other methods that rely on secret data. Full security considerations for this protocol are provided in RFC 4251 [9], RFC 4252 [10] and RFC 4253 [11].

5.3. IPSEC

IPsec is designed to provide security at the IP network layer of the TCP/IP protocol stack. This differs from protocols such as TLS and SSH, above, which provide security at higher layers such as the application layer. The main use of IPsec has been to create virtual private networks (VPNs) which facilitates secure communication over an untrusted network such as the Internet. The IPsec protocols can be deployed in two basic modes:

- Tunnel mode: Cryptographic protection is provided for entire IP packets. In essence, a whole packet (plus security fields) is treated as the new payload of an outer IP packet, with its own header, called the outer header. The original, or inner, IP packet is said to be encapsulated within the outer IP packet. In tunnel mode, IPsec processing is typically performed at security gateways (e.g. perimeter firewalls or routers) on behalf of endpoint hosts.
- Transport mode: The header of the original packet itself is preserved, some security fields are inserted, and the payload together with some header fields undergo cryptographic processing. Transport mode is typically used when end-to-end security services are needed, and provides protection mostly for the packet payload.

Full security considerations for this protocol are provided from RFC 4301 to RFC 4309. There are two main IPsec protocols which specify the actual cryptographic processing applied to packets. These are called Authentication Header (AH) and Encapsulating Security Payload (ESP). AH provides integrity protection, data origin authentication and anti-replay services for packets through the application of MAC algorithms and the inclusion of sequence numbers in packets. ESP provides similar services to AH and in addition provides confidentiality and traffic flow confidentiality services through symmetric key encryption and variable length padding of packets. ESP allows both encryption-only and authenticated encryption modes. We only recommend the following algorithms for future use within IPsec:

- If only authentication is required then either AH or ESP may be used with one of the following MAC algorithms as defined in RFC 4868 [13]:
    - HMAC-SHA2-256.
    - HMAC-SHA2-384.
    - HMAC-SHA2-512.

- If confidentiality is required then ESP should be used by combining one of the following encryption algorithms with one of the MAC algorithms described above:
    - AES-CTR.

6. Key Management Security Controls Requirements for Secret Keys

This section describes the key management requirements that shall be met for the use of a certificate service

6.1. Key Management life-cycle

Key security controls must be in place for each step of the life of a certificate service, including:

- Key generation.
- Key distribution.
- Key backup and recovery.
- Key usage.
- Key change.
- Key archival.
- Key destruction.

6.1.1. Key Generation

All keys used in encryption must be generated and stored in systems meeting the following standards:

- Hardware Security device should be certified FIPS 140-1 Level 3, FIPS 140-2 Level 3 or Common Criteria (CC) EAL 4 or above.
- Smartcards should be certified FIPS 140-1 Level 2, FIPS 140-2 Level 2 or CC EAL 3 or above.

The secure cryptographic modules MUST be evaluated and certified to fulfil **ALL** of the following requirements:

- The module MUST ensure the confidentiality and integrity of the keys during their whole life time;
- The module MUST be able to identify and authenticate its users;
- The module MUST restrict access to its services, depending on the user and his role, to those services explicitly assigned to this user and his role;
- The module MUST be able to run a suite of tests to verify that it is operating correctly, and to enter a secure state when it detects an error;
- The module MUST detect attempts of physical tampering and enter a secure state when a tampering attempt is detected;
- The module MUST be able to create audit records for any security-relevant changes;
- The module MAY optionally support backup and restore of keys, but MUST then protect the confidentiality and integrity of the backup data, and require at least dual control for both backup and restore operations.

Additional controls around the management of such module MUST be taken such as dual controls, implemented on Trusted Systems set of controls

6.1.2. Key Distribution

Secret keys MUST NOT be distributed in plain text. Symmetric keys can be distributed manually by having the key custodians physically travel with the key components to the other parties for key installation. The other option is to send the key electronically as a cipher text.

6.1.3. Key Backup and Recovery

Encryption Keys shall be backed up within a secured container (i.e., safe). The key will remain stored until the key needs to be recovered and hence taken out from the secure container.

6.1.4. Key Usage

Access controls MUST be implemented for all secure cryptographic modules used for encryption, infrastructure and control keys. Some level of controls might be necessary for certificate used for identity purpose for example such as:

- Having dual control mechanism when accessing control-keys.
- Separate infrastructure keys generation for separate functions, limiting single key compromise.

Authorized key usage MUST only occur within the operational life of the key.

6.1.5. Key Change

Infrastructure and Control Keys SHOULD be changed on a regular basis. Key change MUST be carried out securely and MAY be online, or processed physically at the Local RA office.

6.1.6. Key Archival

Encryption keys archival is allowed for the purpose of decrypting encrypted data.

6.1.7. Key Destruction

All cryptographic keys have a limited lifetime. A crypto period is the operational life of a key, and is determined by a number of factors based on:

- The sensitivity of the data or keys to be protected
- How much data or how many keys are being protected.

Therefore the final step in the key life cycle is key destruction, where all instances of the key are erased. Service providers SHALL provide the capability to wipe the keys stored in both hardware and software.

References

[1] ISO/IEC 18033-2. Information Technology { Security techniques { Part 2: Asymmetric Ciphers, 2006.

[2] IEEE P1363. Standard Specifications for Public Key Cryptography, 2000.

[3] FIPS 186-4. Digital Signature Standard (DSS), July 2013.

[4] FIPS PUB 202. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015.

[5] RFC 3447. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, February 2003.

[6] ISO/IEC 14888-3. Information technology -- Security techniques -- Digital signatures with appendix -- Part 3: Discrete logarithm based mechanisms, August 2017.

[7] RFC 8446. The Transport Layer Security (TLS) Protocol Version 1.3, August 2018.

[8] RFC 5246. The Transport Layer Security (TLS) Protocol Version 1.2, August 2008.

[9] RFC 4251. The Secure Shell (SSH) Protocol Architecture, January 2006.

[10] RFC 4252. The Secure Shell (SSH) Authentication Protocol, January 2006.

[11] RFC 4253. The Secure Shell (SSH) Transport Layer Protocol, January 2006.

[12] Open SSH Project. OpenSSH project. http://www.openssh.org/.

[13] RFC 4868. Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA512 with IPsec, May 2007.