OFFICE OF THE CONTROLLER
OF CERTIFICATION AUTHORITIES

TECHNICAL REQUIREMENTS FOR AUDIT
OF CERTIFICATION AUTHORITIES

Table of contents

1.0 SOFTWARE 1

2.0 HARDWARE 2

3.0 TECHNICAL COMPONENTS 2

3.1 KEY MANAGEMENT 3

3.2 CERTIFICATE MANAGEMENT 5

3.3 CERTIFICATE AUTHORITY SYSTEM AND SECURITY 6

4.0 ALGORITHMS 8

5.0 STANDARDS 8

5.1 KEY MANAGEMENT 8

5.2 CERTIFICATE MANAGEMENT 9

5.3 CERTIFICATE AUTHORITY SYSTEM AND SECURITY 9

6.0 OTHER PERTINENT PARAMETERS 9

1. SOFTWARE

This section specifically refers to the design, functional description and performance of the CA system software. Software for operating system, network security and monitoring, utilities and all other software components that is not directly related will not be covered in this section.

1. The CA software architecture should be designed to operate in a distributed environment that permits secure exchange of information.

2. The CA software should be certified by accreditation body such as ISO, FIPS, ITSEC or its equivalent to ensure its quality and absence of malicious code.

3. The CA software comprehensive key and certificate management facility as providing functions described in **Section 4 – Technical Component** of this document.

4. The CA software should have intuitive and consistent human interfaces to enhance business processes.

5. The CA software should be written in high-level programming languages.

6. The CA software development should be performed in a structured way using internationally accepted software development methodology such as the Trusted Software Development Methodology (TSDM) level IV and V and the Software Engineering Institute's Capability Maturity Model (SEI-CMM)

7. The CA software should include documentation indicating performance specification and functional description.

8. The CA software should only be installed and configured by authorized personnel via strict access control procedures.

9. The CA software should have the flexibility to take advantage of new technologies and resources, and can be implemented in changing environments.

10. The CA software should be scaleable to support growing number of certificates.

11. The sources code of the CA software must be escrowed either by the Government of Malaysia or a trusted third party escrow service provider.

12. The CA software should be fully Year 2000 compliant.

2. HARDWARE

This section specifically provides the technical guidelines for the computer (server and workstation) in which the CA software components are installed, networking components and peripherals or devices such smart card, smart card reader and other hardware tokens that are used to enhance the security of the CA system. The configuration and setup of the hardware and networking equipment such as router, switch and hub and all other hardware components that is not directly related to the CA system will not be covered in this section.

1. The CA hardware components should have inherent reliability.

2. The CA hardware should offer an adequate level of security and tamper resistant functionality to ensure the integrity of its private key.

3. The cryptographic modules in use should be validated as meeting Federal Information Processing Standards (FIPS-140-1) security requirement standard or its equivalent.

4. The CA hardware should be designed and implemented to provide a high degree of reliability and to require minimum maintenance while providing a high level of operational availability.

5. The CA hardware (server) should be scaleable to migrate to machines of greater processing power.

6. The CA hardware should have the flexibility to take advantage of new technologies and resources, and can be implemented in changing environments.

7. The CA hardware should only be installed and accessed by authorized personnel via strict access control procedures.

8. The CA hardware should be fully year 2000 compliant.

3. TECHNICAL COMPONENTS

This section lists out the main functional requirements of a CA system. The technical components represent a list of noteworthy technical prescriptions or guidelines for a person to consider when implementing a CA system. A CA should be implemented and administered in a highly secured environment in order to:

- Safeguard the confidentiality, integrity and availability of services;

- Provide strong non-repudiation services for actions of certificate services;
- Prohibit Certification Authorities themselves from repudiating their own actions; and
- Prohibit subscribers and subscribers from repudiating their own actions.

3.1 KEY MANAGEMENT

The topic/matter related to digital signature key management are key generation, key registration, key storage, key recovery, key back-up and key replacement/update key revocation, key suspension and key termination. Confidentiality key management will not be covered in this document.

1. The key management component should provide seed keys to assist in ensuring that a generated key pair does not have attributes that might jeopardize the security of the signature mechanism.

2. The key management component should not allow the duplication of private key unless for a valid reason such as back-up purpose.

3. The key management component should only generate *public-key cryptosystem* (asymmetric) key pair where the private key cannot be derived from the corresponding public key.

4. The key management component should produce good pseudo-random sequence for the creation of quality key pair.

5. The key management component should only generate private key which cannot be derived from the digital signature it created.

6. The management component should allow the use of private key only after due identification of the subscriber through a strict access control mechanism.

7. The key management component should only generate *public-key cryptosystem* (asymmetric) key pair which is unique.

8. The key management component should have a facility for distribution of public keys (via certificate) to repositories.

9. The key management component should provide a method for revocation of previously-distributed public keys, as a result of such events as change in authorization or suspected signing key compromise.

10. The key management component should provide a method for archival of verification keys. The key management component should provide support for the recovery of a verification key on request. The key recovery facility should have the following desirable characteristics:

    - The key recovery facility should support the protection and recovery of keys;
    - Only key recovery enabled systems should be usable within the CA solution;
    - The key recovery facility should provide a mean to verify the legitimacy of a key submitted to it for storage; and
    - A subscriber of the key recovery repository should be able to verify that it is an authorized repository.

11. Discretionary key fragmentation between key recovery facilities should be available.

12. The key management component should have the flexibility to allow:

    - The digital signature key pair to be generated by the CA (upon request by the subscriber) and securely transferring the private key to subscriber using a secure communication protocol or tokens (smart card, diskette or other secure device); or
    - The subscriber to generate its own digital signature key pair.

13. The key management component should have features to generate pair of variable length (512, 768, 1024, 2048 bit).

14. The key management component should support the management of key life-cycles.

15. The key management component should provide a permanent, non-repudiable and independently verifiable record of key retrieval operations.

3.2 CERTIFICATE MANAGEMENT

The topics/matter related to certificate management are certificate generation, certificate storage, certificate revocation, certificate suspension, certificate distribution/ publication. The management of certificate with confidentiality key will not be covered in this document.

1. The certificate management component should provide function to generate public key certificates.

2. The certificate management component should have the capability to distribute certificates to repository or repositories via a common networking protocol such as Transmission Control Protocol/Internet Protocol (TCP/IP).

3. The certificate management component should be able to manage and cope with certificates for a large distributed group of subscribers.

4. The certificate management component should have the capability to retain copy of the certificate for archival purposes.

5. The certificate management component should have the capability to generate multiple certificates for one identified subscriber and corresponding public-key, for different purposes (digital signature and confidential key).

6. The certificate management component should have the facility to revoke certificates for individual keys under the terms of the applicable policy.

7. The certificate management component support electronic renewal of expired certificate with proper notification sent to the subscriber where assurance of policy permits.

8. The certificate management component should support issuance of certificate to subscribers via electronic mail or support the retrieval of certificate by the subscribers via web browser such as Netscape browser (v2.2 and above) and Microsoft Internet Explorer (v3.0 and above).

9. The CA system or its components should to display or print in a way the subscribers will understand the content of the certificate before issuing and incurring legal responsibility to them.

10. The certificate management component should have the facility to suspend and reactivate certificates for individual keys under the terms of the applicable policy.

11. The certificate management component should have the function to customize or configure certificate profile under the terms of the applicable policy.

12. The certificate management component should have interface to electronically accept certificate request made by subscribers.

13. The certificate management component should be able to generate Certificate Revocation List (CLR) at regular intervals as specified in its certification policy to ensure subscribers cannot exploit expired or revoked certificates. During the time interval, the certificate will be suspended until the identity of the person who requested for revocation is confirmed.

14. The certificate management component should be able to post the most recently generated Certificate Revocation Lists (CRLs) immediately to a public repository server or a repository server at subscribers' premises upon request. This is to ensure subscribers have the most current certificate information status.

3.3 CA SYSTEM AND SECURITY

The topics/matter that are related to CA system and security are general system functionality and features such as audit trial, cryptographic tokens support, fault tolerance and access control.

1. The CA system and its components should provide security audit trial facilities. Operations related to key management, certificate management, token initialization activities, communication among the CA system components and publication of certificates to repository should be logged. Log files should be digitally signed.

2. The CA system and its components should maintain access control information, and enforce access control with strong authentication mechanism if necessary, to ensure that only properly authorized persons or systems can access, create or modify information in the system.

3. The CA system and its components should have the capability to digitally sign and optionally encrypt all messages passed among these components and other external components.

4. The CA system design should allow:

   - The separation of key generation component and certificate management component.

- The separation between client module for subscriber registration and server module.

5. The CA system and its components should be able to support hardware token, physical media, devices, initial key material or other relevant security technologies.

6. The CA system and its component should have a proper backup and disaster recovery protection measures to protect the system and critical application data, and off-site vaulting for long-term archival and disaster recovery purposes.

7. The CA system and its component should have process that allows renewal and re-enrollment in a manner similar to the process of the initial application or in addition to that, the subscriber needs to submit only new or changed information thereof. Furthermore, any up-to-date requirements for re-enrollment and renewal should be accessible from a repository.

8. The CA system and its component should provide API (Application Programming Interface) for applications development.

9. The CA system and its components should be able to generate notices of the following operations:

   - Notification of issuance of a certificate to the subscriber who is the subject of the certificate being issued.

   - Notification of issuance of a certificate to other than the subject of the certificate.

   - Notification of revocation or suspension of a certificate to the subscriber whose certificate is being revoked or suspended.

   - Notification of revocation or suspension of a certificate to others than the subject whose certificate is being revoked or suspended.

   - Notification of reactivation of a certificate to the subscriber whose certificate is being suspended.

   - Notification of reactivation of a certificate to others than the subject whose certificate is being suspended.

10. The CA system and its components should provide access as needed to external repository services used in supporting interoperation between the system and other CA system

4. ALGORITHMS

1. The CA system should support multiple public-key cryptosystem based digital signature scheme such as:

   - RSA (Rivest, Shamir, Adleman) with MD5
   - DSA with SHA-1
   - Elliptic curve cryptosystem (ECC)

2. The CA system should support symmetric key algorithm such as:

   - DES, Triple DES
   - RC2, RC4

3. The CA system should support one way hash algorithm such as:

   - MD5
   - SHA-1

5. STANDARDS

This section recommends the relevant standards and protocols associated with the technical component of the CA system. The standards and protocols provided in this guide are based on international and de facto standard. The principle of recommendation is to adopt existing standards and protocols wherever possible, and to invent new standards or protocols only as a last resort. New emerging standards shall be incorporated in later revision of this section.

5.1 KEY MANAGEMENT

1. The key management component should adopt ANSI X9.17 method of key generation or its equivalent.

2. The key management component should adopt RSA or DSA digital signature scheme for the creation of digital signature key pair.

3. The key management component should adopt MD5 or SHA-1 one way hash function for creation of digital signature.

5.2 CERTIFICATE MANAGEMENT

1. The certificate management component should issue certificates which conform to ITU-TX.509 Version 3 standards.

2. The certificate management component should support certificate request standard complying to PKCS#10.

3. The certificate management component should adopt Certificate Revocation List (CLR) that conforming Internet Engineering Task Force Working Group (IETF PKI WG) (PKIX) X.509 CRLv2 format.

5.3 CA SYSTEM AND SECURITY

1. The CA system should adopt TCP/IP related protocols (for example HTTP, PEM, S/MIME, SSL) as the transport mechanism for X.509v3 certificates for on-line system.

2. The CA system should be able to support secure electronic transaction protocol such as the Secure Electronic Transaction (SET) protocol.

3. The CA system should support cryptographic token interface standard such as the PKCS#11 standard.

6. OTHER PERTINENT PARAMETERS

The key usage extension defines the purpose (e.g., encipherment, signature, certificate signing) of the key contained in the certificate. The usage restriction might be employed when a key that could be used for more than one operation is to be restricted. For example, when an RSA key should be used only for signing, the digital signature and non-repudiation bits would be asserted. Likewise, when an RSA key should be used only for key management, the key Encipherment bit would be asserted. The profile recommends that when used, this be marked as a critical extension.

Id-ce-keyUsage OBJECT IDENTIFIER :: = {id-ce 15}
KeyUsage ::= BIT STRING {

digital signature	(0),
nonRepudiation	(1),
keyEncipherment	(2),
dataEncipherment	(3),
keyAgreement	(4),

keyCertSign (5),
cRLSign (6),
encipherOnly (7),
decipherOnly (8) }

Bits in the KeyUsage type are used as follows:

- The digital Signature bit is asserted when the subject public key is used to verifying digital signatures that have purposes other than non-repudiation, certificate signature, and CRL signature. For Example, the digital signature bit is asserted when the subject public key is used to provide authentication.

- The nonrepudiation bit is asserted when the subject public key is used to verifying digital signature used to provide a non-repudiation service which protects against the signing entity falsely denying some action, excluding certificate or CRL signing.

- The keyEncipherment bit is asserted when the subject public key is used for key transport. For example, when an RSA key is to be used exclusively for key management, then this bit must asserted.

- The dataEncipherment bit is asserted when the subject public key is used for enciphering user data, other than cryptographic keys.

- The keyAgreement bit is asserted when the subject public key is used for key agreement. For example, when a Diffie-Hellman key is to be used exclusively for key management, then this bit must asserted.

- The keyCertSign bit is asserted when the subject public key is used for verifying a signature on certificates. This bit may only be asserted in CA certificates.

- The cRLSign bit is asserted when the subject public key is used for verifying a signature on CRLs. This bit may only be asserter in CA certificates.

- When the encipherOnly bit is asserted and the keyAgreement bit is also set, the subject public key may be used only for enciphering data while performing key agreement. The meaning of the encipheronly bit is undefined in the absence of the keyAgreement bit.

- When the decipherOnly bit is asserted and the keyAgreement bit is also set, the subject public key may be used only for deciphering data while performing key agreement. The meaning of the decipherOnly bit is undefined in the absence of the keyAgreement bit.

This profile does not restrict the combinations the bits that may be set in an instantiation of the keyUsage extension.