International Journal of Computer Communication and Information System (IJCCIS) – Vol2. No1. ISSN: 0976–1349 July – Dec 2010

Key Management Techniques for Controlling the Distribution and Update of Cryptographic keys.

[1]T.Lalitha, [2]R.Umarani and [3]G.M.Kadharnawaz

[1]Senior Lecturer, Department of MCA, Sona College of Technology, Salem, Tamilnadu., India.
lalithasrilekha@rediffmail.com
[2]Assoc.Prof., Department of Comp. Science, Sri Saradha College for Women, Salem, Tamilnadu, India.
umainweb@gmail.com
[3]Director,Department of MCA, Sona College of Technology, Salem, Tamilnadu., India nawazse@yahoo.co.in

***Abstract*-Key management plays a fundamental role in cryptography as the basis for securing cryptographic techniques providing confidentiality, entity authentication, data origin authentication, data integrity, and digital signatures. The goal of a good cryptographic design is to reduce more complex problems to the proper management and safe-keeping of a small number of cryptographic keys, ultimately secured through trust in hardware or software by physical isolation or procedural controls. Reliance on physical and procedural security (e.g., secured rooms with isolated equipment), tamper-resistant hardware, and trust in a large number of individuals is minimized by concentrating trust in a small number of easily monitored, controlled, and trustworthy elements.**

***Keyword*-Key Management, Cryptography, Algorithm, Secret Key.**

I. INTRODUCTION

Systems providing cryptographic services require techniques for initialization and key distribution as well as protocols to support on-line update of keying material, key backup/recovery, revocation, and for managing certificates in certificate-based systems.

Key management [1] is the set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties.

Key management encompasses techniques and procedures supporting:

- Initialization of system users within a domain
- Generation, distribution, and installation of keying material
- Controlling the use of keying material.
- Update, revocation, and destruction of keying material and
- Storage, backup/recovery, and archival of keying material.

II. CLASSIFYING KEYS BY ALGORITHM TYPE AND INTENDED USE

The terminology of Table I is used in reference to keying material. A symmetric cryptographic system is a system involving two transformations – one for the originator and one for the recipient – both of which make use of either the same secret key (symmetric key) or two keys easily computed from each other. An asymmetric cryptographic system is a system involving two related transformations – one defined by a public key (the public transformation), and another defined by a private key (the private transformation) with the property that it is computationally infeasible to determine the private transformation from the public transformation.

TABLE 1 PRIVATE, PUBLIC, SYMMETRIC AND SECRET KEYS

Term	Meaning
private key, public key	paired keys in an asymmetric cryptographic system
symmetric key	key in a symmetric (single-key) cryptographic system
secret	adjective used to describe private or symmetric key

Table II indicates various types of algorithms commonly used to achieve the specified cryptographic objectives. Keys associated with these algorithms may be correspondingly classified, for the purpose of controlling key usage .The classification given requires specification of both the type of algorithm (e.g., encryption vs. signature) and the intended use (e.g., confidentiality vs. entity authentication).

TABLE 1I TYPES OF ALGORITHMS COMMONLY USED TO MEET SPECIFIED OBJECTIVES

Cryptographic objective	Algorithm type	
	public-key	***symmetric-key***
confidentiality	encryption	encryption
data origin authentication	signature	MAC

key agreement	Diffie-Hellman	various methods
entity authenticatio	3.Customized	

III. KEY MANAGEMENT OBJECTIVES, THREATS AND POLICY

Keying relationships in a communications environment involve at least two parties (a sender and a receiver) in real-time. In a storage environment, there may be only a single party, which stores and retrieves data at distinct points in time.

The objective of key management is to maintain keying relationships and keying material in a manner which counters relevant threats, such as:

- Compromise of confidentiality of secret keys.
- Compromise of authenticity of secret or public keys. Authenticity requirements include knowledge or verifiability of the true identity of the party a key is shared or associated with.
- Unauthorized use of secret or public keys.

A. Security policy and key management

Key management is usually provided within the context of a specific security policy. A security policy explicitly or implicitly defines the threats a system is intended to address. The policy may affect the stringency of cryptographic requirements, depending on the susceptibility of the environment in question to various types of attack. Security policies typically also specify:

- Practices and procedures to be followed in carrying out technical and administrative aspects of key management, both automated and manual responsibilities and accountability of each party involved and the types of records (audit trail information) to be kept, to support subsequent reports or reviews of security-related events.

IV. TRADE OFFS AMONG KEY ESTABLISHMENT PROTOCOLS

In selected key management applications, hybrid protocols involving both symmetric and asymmetric techniques offer the best alternative. More generally, the optimal use of available techniques generally involves combining symmetric techniques for bulk encryption and data integrity with public-key techniques for signatures and key management.

B. Public key vs. symmetric key techniques (in key

Primary advantages offered by public-key (vs. symmetric-key) techniques for applications related to key management include:

- Simplified key management. To encrypt data for another party, only the encryption public key of that party need be obtained. This simplifies key management as only authenticity of public keys is required, not their secrecy.
- On-line trusted server not required. Public-key techniques allow a trusted on-line server to be replaced by a trusted off-line server plus any means for delivering authentic public keys (e.g., public-key certificates and a public database provided by an entrusted on-line server). For applications where an on-line trusted server is not mandatory, this may make the system more amenable to scaling, to support very large numbers of users.
- Enhanced functionality. Public-key cryptography [2] offers functionality which typically cannot be provided cost-effectively by symmetric techniques (without additional online trusted third parties or customized secure hardware). The most notable such features are non-repudiation of digital signatures, and true (single-source) data origin authentication.

IV. PUBLIC KEY CERTIFICATES

Public-key certificates are a vehicle by which public keys may be stored, distributed or forwarded over unsecured media without danger of undetectable manipulation. The objective is to make one entity's public key available to others such that its authenticity (i.e., its status as the true public key of that entity) and validity are verifiable. In practice, X.509 certificates are commonly used.

C. Definition

A public-key certificate [4] is a data structure consisting of a data part and a signature part. The data part contains clear text data including, as a minimum, a public key and a string identifying the party (subject entity) to be associated there with. The signature part consists of the digital signature of a certification authority over the data part, thereby binding the subject entity's identity to the specified public key.

The Certification Authority (CA) is a trusted third party whose signature on the certificate vouches for the authenticity of the public key bound to the subject entity. The significance of this binding (e.g., what the key may be used for) must be provided by additional means, such as an attribute certificate or policy statement. Within the certificate, the string which identifies the subject entity must be a unique name within the system (distinguished name), which the CA typically associates with a real-world entity. The CA requires its own signature key pair, the authentic public key of which is made available to each party upon registering as an authorized system user. This CA public key allows any system user, through certificate acquisition and verification, to transitively acquire trust in the authenticity of the public key in any certificate signed by that CA.Certificates are a means for transferring trust, as opposed to establishing trust originally. The authenticity of the CA's public key may be originally provided by non-cryptographic means including personal acquisition, or through trusted couriers; authenticity is required, but not secrecy.

Examples of additional information which the certificate data part might contain include:

- A validity period of the public key.
- A serial number or key identifier identifying the certificate or key.
- Additional information about the subject entity (e.g., street or network address).
- Additional information about the key (e.g., algorithm and intended use).
- Quality measures related to the identification of the subject entity, the generation of the key pair, or other policy issues.
- Information facilitating verification of the signature (e.g., a signature algorithm identifier, and issuing CA's name).
- The status of the public key.

D. Creation of public-key certificates

Before creating a public-key certificate for a subject entity A, the certification authority should take appropriate measures (relative to the security level required, and customary business practices), typically non-cryptographic in nature, to verify the claimed identity of A and the fact that the public key to be certified is actually that of A. Two cases may be distinguished.

- Trusted party creates key pair. The trusted party creates a public-key pair, assigns it to a specific entity, and includes the public key and the identity of that entity in the Certificate. The entity obtains a copy of the corresponding private key over a secure (authentic and private) channel after proving its identity (e.g., by showing a passport or trusted photo-id, in person). All parties subsequently using this certificate essentially delegate trust to this prior verification of identity by the trusted party.
- Entity creates own key pair. The entity creates its own public-key pair, and securely transfers the public key to the trusted party in a manner which preserves authenticity.(e.g., over a trusted channel, or in person). Upon verification of the authenticity (source) of the public key, the trusted party creates the public-key certificate the signer.

E. Use and verification of public-key certificates

The overall process whereby a party B uses a public-key certificate to obtain the authentic public key of a party A may be summarized as follows:

- (One-time) acquire the authentic public key of the certification authority.
- Obtain an identifying string which uniquely identifies the intended party A.
- Acquire over some unsecured channel (e.g. from a central public database of certificates, a public-key certificate corresponding to subject entity A and agreeing with the previous identifying string.

1) (a) Verify the current date and time against the validity period (if any) in the certificate, relying on a local trusted time/day-clock.

(b) Verify the current validity of the CA's public key itself.

(c) Verify the signature on A's certificate, using the CA's public key.

(d) Verify that the certificate has not been revoked .

If all checks succeed, accept the public key in the certificate as A's authentic key.

F. Attribute certificates

Public-key certificates bind a public key and an identity, and include additional data fields necessary to clarify this binding, but are not intended for certifying additional information. Attribute certificates are similar to public-key certificates, but specifically intended to allow specification of information (attributes) other than public keys (but related to a CA, entity, or public key), such that it may also be conveyed in a trusted (verifiable) manner. Attribute certificates may be associated with a specific public key by binding the attribute information to the key by the method by which the key is identified, e.g., by the serial number of a

Corresponding public-key certificate, or to a hash-value of the public key or certificate. Attribute certificates may be signed by an attribute certification authority, created in conjunction with an attribute registration authority, and distributed in conjunction with an attribute directory service More generally, any party with a signature key and appropriate recognizable authority may create an attribute certificate. One application is to certify authorization information related to a public key. More specifically, this may be used, for example, to limit liability resulting from a digital signature, or to constrain the use of a public key (e.g., to transactions of limited values, certain types, or during certain hours).consulted.

VII KEY LIFE CYCLE ISSUES

Key management is simplest when all cryptographic keys are fixed for all time. Crypto periods [3] necessitate the update of keys. This imposes additional requirements, e.g., on certification authorities which maintain and update user keys. The set of stages through which a key progresses during its existence, referred to as the life cycle of keys, is discussed in this section.

G Lifetime protection requirements

Controls are necessary to protect keys both during usage and storage. Regarding long-term storage of keys, the duration of protection required depends on the cryptographic function (e.g., encryption, signature, data origin authentication/integrity) and the time-sensitivity of the data in question.

Security impact of dependencies in key updates: Keying material should be updated prior to crypto period expiry. Update involves use of existing keying material to establish new keying material, through appropriate key establishment protocols and key layering .To limit exposure in case of compromise of either long term secret keys or past session keys, dependencies among keying material should be avoided. For example, securing a new session key by encrypting it under the old session key is not recommended (since compromise of the old key compromises the new).

Lifetime storage requirements for various types of keys: Stored secret keys must be secured so as to provide both confidentiality and authenticity. Stored public keys must be secured such that their authenticity is verifiable. Confidentiality and authenticity guarantees, respectively countering the threats of disclosure and modification,may be provided by cryptographic techniques, procedural (trust-based) techniques, or physical protection (tamper-resistant hardware). Signature verification public keys may require archival to allow signature verification at future points in time, including possibly after the private key ceases to be used. Some applications may require that signature private keys neither be backed up nor archived: such keys revealed to any party other than the owner potentially invalidates the property of nonrepudiation. Note here that loss (without compromise) of a signature private key may be addressed by creation of a new key, and is non-critical as such a private key is not needed for access to past transactions; similarly, public encryption keys need not be archived. On the other hand, decryption private keys may require archival, since past information encrypted there under might otherwise be lost. Keys used for entity authentication need not be backed up or archived. All secret keys used for encryption or data origin authentication should remain secret for as long as the data secured there under requires continued protection (the *protection lifetime*), and backup or archival is required to prevent loss of this data or verifiability should the key be lost.

H. Key management life cycle

Except in simple systems where secret keys remain fixed for all time, crypto periods associated with keys require that keys be updated periodically. Key update necessitates additional procedures and protocols, often including communications with third parties in public-key systems. The sequence of states which keying material progresses through over its lifetime is called the key management life cycle.

Life cycle stages may include:

- User registration – an entity becomes an authorized member of a security domain. This involves acquisition, or creation and exchange, of initial keying material such as shared passwords or PINs by a secure, one-time technique (e.g., personal exchange, registered mail, trusted courier).
- User initialization – an entity initializes its cryptographic application (e.g., installs and initializes software or hardware), involving use or installation (see below) of initial keying material obtained during user registration.
- Key generation – generation of cryptographic keys should include measures to ensure appropriate properties for the intended application or algorithm and randomness in the sense of being predictable (to adversaries) with negligible probability . An entity may generate

its own keys, or acquire keys from a trusted system component.

- Key installation – keying material is installed for operational use within an entity's software or hardware, by a variety of techniques including one or more of the following: manual entry of a password or PIN, transfer of a disk, read-only-memory device, chip card or other hardware token or device (e.g., key-loader). The initial keying material may serve to e
whic

subsequent updates, new keying material is installed to replace that in use, ideally through a secure on-line update technique.

- Key registration – in association with key installation, keying material may be officially recorded (by a registration authority) as associated with a unique name which distinguishes an entity. For public keys, public-key certificates may be created by a certification authority (which serves as guarantor of this association), and made available to others through a public directory or other means
- Normal use – the objective of the life cycle is to facilitate operational availability of keying material for standard cryptographic purposes. Under normal circumstances, this state continues until crypto period expiry; it may also be subdivided – e.g., for encryption public-key pairs, a point may exist at which the public key is no longer deemed valid for encryption, but the private key remains in (normal) use for decryption.
- Key backup – backup of keying material in independent, secure storage media provides a data source for key recovery (point 11 below). Backup refers to short-term storage during operational use.
- Key update – prior to crypto period expiry, operational keying material is replaced by new material. This may involve some combination of key generation, key derivation execution of two-party key establishment protocols, or communications with a trusted third party. For public keys, update and registration of new keys typically involves secure communications protocols with certification authorities.
- Archival – keying material no longer in normal use may be archived to provide a source for key retrieval under special circumstances (e.g., settling disputes involving repudiation). Archival [5] refers to off-line long-term storage of post-operational keys.
- key de-registration and destruction – once there are no further requirements for the value of a key or maintaining its association with an entity, the key is de-registered (removed from all official records of existing keys), and all copies of the key are destroyed. In the case of secret keys, all traces are securely erased.

equipment failure or forgotten passwords), it may be possible to restore the material from a secure backup copy.

- Key revocation – it may be necessary to remove keys from operational use prior to their originally scheduled expiry, for reasons including key compromise. For public keys distributed by certificates, this involves revoking certificates.

VIII. CONCLUSION

Key management plays a fundamental role in cryptography as the basis for securing cryptographic techniques providing confidentiality, entity authentication, data origin authentication, data integrity, and digital signatures. The goal of a good cryptographic design is to reduce more complex problems to the proper management and safe-keeping of a small number of cryptographic keys, ultimately secured through trust in hardware or software by physical isolation or procedural controls. Reliance on physical and procedural security (e.g., secured rooms with isolated equipment), tamper-resistant hardware, and trust in a large number of individuals is minimized by concentrating trust in a small number of easily monitored, controlled, and trustworthy elements.

REFERENCES

[1] National Institute of Standards and Technology.

[2] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, Vol. 21 (2), pp.120–126. 1978. Previously released as an MIT "Technical Memo" in April 1977, and published in Martin Gardner's Scientific American.

[3] http://www.discretix.com/PDF/Using%20Public%20Key%20Cryptography%20in%20Mobile%20Phones.pdf

[4] http://dlc.sun.com/pdf/316194901A/316194901A.pdf

[5] http://www.safecomprogram.gov/NR/rdonlyres/7C31664D-5B0B-4128-A85B-DC79B1D734ED/0/Security_Issues_Analysis_Report.pdf