Journal of Advance Management Research, ISSN: 2393-9664

Vol.05 Issue-05, (December 2017), Impact Factor: 4.598

CRYPTOGRAHIC KEY MANAGEMENT - A REVIEW

Dr.T.LOGESWARI

Dept of Computer Science, New Horizon College, Bangalore

Abstract

cryptography is an important and powerful tool for achieving secure communications. Key management, including key distribution and revocation, is a central part of any cryptographically protected secure communication and is one of the weakest links of system security in general and protocol design in particular. In most communication scenarios, cryptographic keys need to be established between the communicating network nodes prior to any service can be delivered. In this paper, we review the two broad categories of cryptographic keys, list the most commonly used key types, identify the key states and chart the resulting transition diagram.

KeyWords: Keystate, Encryption, Key Management.

1. INTRODUCTION

In cryptography, the most crucial and challenging step is Key management[5] . Cryptographic techniques make use of two types of keys, either symmetric or asymmetric. Symmetric cryptography relies on a shared secret key between two nodes to enable secure communication. Asymmetric cryptography, applies two different keys, a private key and a public key. The public key is used for encryption and can be published. The private key is used for decryption. From a computational point of view asymmetric cryptography requires orders of magnitude more resources than symmetric cryptography. In general, key management is considered of four sorts of keys as following: one-time session symmetric keys, public keys, private keys, passphrase-based symmetric keys. The session keys are used once and generated for each new message. The public keys are used in asymmetric encryption. On the other hand, private keys are also used in asymmetric encryption. Passphrase-based keys are used to protect private keys. A single node can have multiple public or private key pairs.

1.1 KEY TYPES

Cryptographic keys fall into two broad categories:

1. Secret key: A key that is normally used to 1) perform encryption/decryption using symmetric cryptographic algorithms; and/or 2) to provide data integrity using message authentication codes (i.e., Hash based Message Authentication Code or HMAC) or an encryption mode of operation that also provides data integrity[1]. A secret key is also Known as symmetric key, because the same key is required for encryption and decryption or for integrity value generation and integrity verification.

Figure 1. Types of Keys

2. Public/Private Key Pair: A pair of mathematically associated keys used in asymmetric cryptography for authentication, digital signature, or key establishment. As the name indicates, the private key is used by the owner of the key pair, is kept secret, and should be protected at all times, while the public key can be available and used by the relying party to complete the procedure or invert the operations performed with the private key. From these wide categories one can find out the most commonly used key types in a cloud computing environment.

1. Public/Private verification Key Pair: This key pair is used by one party (peer, client or server) to validate to the other party. Its characteristic apply entails combining a random challenge with the signer-generated random number and signing the result for the benefit of the challenger who wishes to authenticate the private-key holder. Examples of usage include client-authenticated Transport Layer Security (TLS), Virtual Private Network (VPN) authentication, and smart card-based logon. An authentication key pair is usually used in a network surroundings and is usually used for long-term use (e.g., up to 3 years)

2. Public/Private Signature Key Pair: The private key of the key pair is used by one party to digitally sign a message/data, while the corresponding public key is used to verify the signature. Examples of the usage of a signature key pair are signed Secure/Multipart Internet Mail Extensions (S/MIME) messages, signed electronic documents, and signed code. In several implementations, a key pair may be used for both verification and signature functions. A signature key pair is generally used in a network environment and is generally used for long-term use (e.g., up to 3 years). It may also be used to create and confirm signatures on stored data.

3. Public/Private Key Establishment Pair: This key pair is used to strongly found a key between parties. Examples of the use of a key pair for key establishment are encrypting the symmetric key for S/MIME

payload encryption/decryption and encrypting the random secret to be sent from a TLS client to a server[4]. It is suggested that key establishment key pairs be different from authentication and signature key pairs. However, it is recognized that some devices such as web servers use the same key pair for key establishment and authentication. A key concern key pair is usually used in a network environment, but some practice for stored data is also seen and can be envisioned. A key establishment key pair is generally used for a pre-defined period for encryption (e.g., up to 3 years), but is used for decryption for as long as the confidentiality of the data needs to be protected.

4. Symmetric Encryption/Decryption Key: A symmetric key is used to encrypt and decrypt data or messages. For data-in-transit, a symmetric encryption/decryption key may have a short life, typically for each message (e.g., S/MIME message) or for each session (for example a TLS session). For stored information, the symmetric life of the encryption/decryption key tends to be as extended as the secrecy of the data wants to be protected.

5. Symmetric Message Authentication Code (MAC) Key: A symmetric key is used to provide promise for the reliability of data. There are three techniques used to provide this assurance: 1) use a symmetric encryption algorithm and a MAC mode of operation (e.g., CMAC using AES); 2) use a symmetric encryption algorithm and an authenticated encryption mode of operation (e.g., GCM or CCM using AES); and 3) use a hash-based MAC (HMAC). For data-in-transit, a symmetric MAC key has a short life, typically for a single message or for a single session (for example a TLS session). For stored information, the life of a symmetric MAC key tends to be for as long as the data needs to be protected. message that when authenticated encryption mode is used, the same key is used for both the 4 MAC and encryption/decryption, since both objectives are achieved by invoking a single mode of operation.

6. Symmetric Key Wrapping Key: A symmetric key is used to encrypt a symmetric key or an asymmetric private key. A Key Wrapping Key is also called a Key Encrypting Key.

1.2 KEY STATES

A symmetric key or public/private key pair can go through the following states[2]. The key management implementation may not have additional states. Alternatively, a key management implementation may have a subset of these states.

• Generation: A symmetric key or public/private key pair is generated when required.

• Activation: A symmetric key or private key is activated when it is necessary to be used. A public key is activated when it is made available or on the date indicated in its associated metadata (e.g., notBefore date in an X.509 public key certificate).

• Deactivation: A symmetric key or private key is deactivated while it is no longer necessary for applying cryptographic protection to data. Deactivation of these keys may be followed by destruction or archival. A public key is not deactivated. It may terminate (e.g., at the notAfter date in an X.509 public key certificate), or may be suspended (e.g., via certificate revocation list (CRL) in X.509 standard) or revoked (e.g., via CRL in X.509 standard).

• Suspension: A key may be suspended from use for a variety of reasons, such as an unknown status of the key or due to the key owner being temporarily away. In the case of the public key, suspension

of the companion private key is communicated to the relying parties. This may be communicated as an "On hold" revocation reason code in a CRL and in an Online Certificate Status Protocol (OCSP) response

• Expiration: A key may expire due to the end of its crypto period [refer RFC 4949]. In the case of a public key, an termination date is indicated in the linked metadata (e.g., notAfter date in X.509 certificates).

• Destruction: A key is destroyed when it is no longer needed.

• Archival: A key may be archived when it is no longer necessary for normal use, but may be needed after the key's crypto period. An example for secret or private keys is the possible decryption of archived data. An example for public keys is the verification of archived signed documents.

• Revocation: A revocation is explicitly stated with respect to public keys; however, the revocation also applies to the corresponding private key. Revocation information is securely communicated to the relying parties, for example, as CRLs or OCSP responses, 5 in the case of X.509 public key certificates. Secret keys are also "revoked", often by together with them on lists, such as a compromised key list.

The following is the state diagram for the key states.

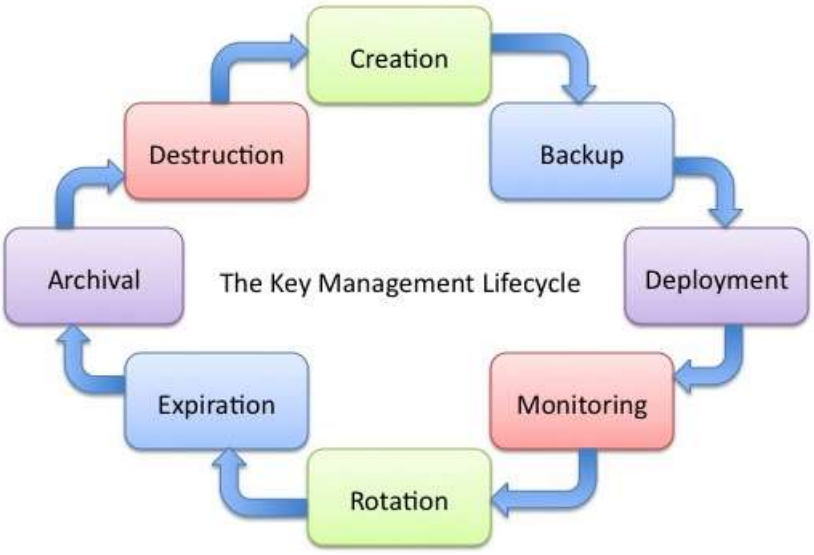

Figure 2. Key Management Lifecycle

1.3 KEY MANAGEMENT FUNCTIONS

The following are the significant key management functions:

• Generate Key: The creation of excellent keys is significant to security. Keys for a cryptographic algorithm should be generated in cryptographic modules that have been approved for the generation of keys for that algorithm[3].

• Generate Domain Parameters: Discrete Logarithm-based algorithms need the creation of domain parameters prior to the generation of the keys; the keys are generated using those domain parameters. The domain parameters for an algorithm shall be generated in approved cryptographic modules that have been approved for their 6 generation. while domain parameters can be familiar to a broad community of users, key generation need not entail domain parameter generation. For example, defining Suite B P-256 curve defines all the domain parameters for the attendant ECDSA and ECDH algorithms.

• Bind Key and Metadata: A key may have connected information, such as the time period of utilize, procedure constraints (such as authentication, encryption, and/or key establishment), domain parameters, and security services for which they are used, such as source authentication, integrity, and confidentiality protection. This function provides assurance that the key is associated with the correct metadata.

• Bind a Key to an Individual: The identifier of the individual or other entity that owns a key is considered as part of the key's metadata, but this association is sufficiently critical to be listed as a distinct function.

• Activate Key: This function transitions a key to the active state. It is frequently done in combination with key generation.

• Deactivate Key: This function is normally done when a key is no longer required for applying cryptographic security. For example, when a key has expired, or is replaced by another key.

• Backup Key: A key is backed by the owner, the key management communications, or a third party in order to reconstitute the key when it is accidentally destroyed or otherwise unavailable. When a private or secret key is backed up by the key management infrastructure or by a third party, the function is also referred to as "key escrow".

• Recover Key: This function is parallel to the key backup function and is invoked when the key is unavailable for some reason and is required by the authorized parties. Key backup and recovery generally applies to the symmetric and private keys.

• Modify Metadata: This function is invoked when metadata bound to a key needs to change. The renewal of a public key certificate is an example of this function where the validity period for the public key is changed.

• Rekey: This function is used to replace the existing key with a new key. in general, the existing key (the key being replaced) plays a role in verification and approval for replacement.

• Suspend a Key: This function is used to temporarily cease the use of a key. It is similar to reversible revocation. This function may have to to be invoked if the status of a key is undecided or if the key owner wishes to temporarily suspend its use (e.g., for extended leave). For secret keys, this can also be accomplished via key deactivation. For public keys and the companion private key, this is usually done using suspension notification of the public key.

• Restore a Key: This function is used to restore a suspended key once its secure status is ascertained. For secret keys, this can also be accomplished via key activation. For public keys and the companion private keys, this is usually done using a revocation notification where the revoked public key entry is deleted implying the key is valid.

• Revoke a Key: This function is used to inform the relying parties to stop using a public key. There may be a mixture of reasons for this, as well as the compromise of companion private key, and the owner having stopped using the companion private key.

• Archive a Key: This function is used to store a key in long-term storage after it has been deactivated, expired, and/or compromised.

• Destroy a Key: This function is used to zeroize a key while it should no longer be used.

• Manage TA Store: This function is used by the relying party to determine what trust anchors to trust for what purpose. A trust secure is a public key and its associated metadata that the relying party openly trusts and uses to establish trust in other public keys via transitive trust, such as a public-key certification path that is a series of public key certificates where the digital signature in one certificate can be used to verify the digital signature on the next certificate.

1.4 KEY MANAGEMENT - Generic Security Requirements

The following are general key management security requirements:

1. Parties performing key management functions are correctly authenticated and their authorizations to make the key management functions for a given key are accurately verified.

2. All key management information and linked data are protected from spoofing, i.e., source authentication is performed prior to executing a command.

3. All key management instructions and linked data are protected from hidden, unauthorized modifications, i.e., integrity protection is provided.

4. Secret and private keys are protected from unauthorized disclosure.

5. All keys and metadata are confined from spoofing, i.e., source authentication is performed prior to accessing keys and metadata.

6. All keys and metadata are protected from undetected, illegal modifications, i.e., reliability protection is provided.

7. When cryptography is used as a protection mechanism for any of the above, the security strength of the cryptographic mechanism used is at least as strong as the security strength required for the keys being managed.

8 There are significant challenges to implementing these key management security requirements in cloud computing over unsecure public networks.

1.5 CONCLUSION

The paper discussed Encode significance with firmly protected key which is known just by sending and receiver end, is a huge angle to obtain powerful protection in cloud. The secure trade of key amongst sender and collector is an vital task. The key administration keeps up arrangement of mystery information from unapproved clients. It can similarly verify the respectability of the traded significance to validate the authenticity. display security covers the operation of cryptographic calculations in system conventions and system applications. This paper quickly presents the idea of key state, Key Function and key management to key circulation and administration and also ideal cryptography result for in sequence protection over mists

REFERENCES

[1] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, NIST Cloud Computing Reference Architecture (NIST SP 500-292), National Institute of Standards and Technology, U.S. Department of Commerce (2011). http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505

[2] P. Mell and T. Grance, The NIST definition of cloud computing (NIST SP 800-145), National Institute of Standards and Technology, U.S. Department of Commerce (2011) http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[3] L. Badger, D. Berstein, R. Bohn, F. de Valux, M. Hogan, J. Mao, J. Messina, K. Mills, A. Sokol, J. Tong, F. Whiteside, and D. Leaf, US government cloud computing technology roadmap volume 1: High-priority requirements to further USG agency cloud computing adoption (NIST SP 500-293, Vol. 1), National Institute of Standards and Technology, U.S. Department of Commerce (2011). http://www.nist.gov/itl/cloud/upload/SP_500_293_volumeI-2.pdf

[4] W. Jansen and T. Grance, Guidelines on Security and Privacy in Public Cloud Computing (NIST SP 800- 144). National Institute of Standards and Technology, U.S. Department of Commerce (2011). http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf.

[5] Wu, B.; Wu, J.; Cardei, M. A Survey of Key Management in Mobile Ad Hoc Networks. In Handbook of Research on Wireless Security; IGI Global: Hershey, PA, USA, 2010.