[Company]

CRYPTOGRAPHIC KEY MANAGEMENT POLICY

The management of encryption keys

1 Document Version Control

	Last Modified	Last Modified By	Document Changes
0.1	[DATE]		Document first created

2 Document Contents Page

3 Cryptographic Key Management Policy

3.1 Purpose

The purpose of this policy is to ensure the proper lifecycle management of encryption keys to protect the confidentiality and integrity of confidential information.

3.2 Scope

Confidential and personal information processed, stored, or transmitted on or in company owned, managed, and controlled systems and applications deemed in scope by the ISO 27001 scope statement.

All employees and third-party users.

3.3 Principle

Cryptographic Key Management is based on the OWASP guidelines - https://cheatsheetseries.owasp.org/cheatsheets/Key_Management_Cheat_Sheet.html

Cryptographic keys are classified as Confidential.

3.4 Generation

Cryptographic keys shall be generated within cryptographic module with at least a FIPS 140-2 compliance. For explanatory purposes, consider the cryptographic module in which a key is generated to be the key-generating module.

Any random value required by the key-generating module shall be generated within that module; that is, the Random Bit Generator that generates the random value shall be implemented within cryptographic module with at least a FIPS 140-2 compliance that generates the key.

Hardware cryptographic modules are preferred over software cryptographic modules for protection.

3.5 Distribution

The generated keys shall be transported (when necessary) using secure channels and shall be used by their associated cryptographic algorithm within at least a FIPS 140-2 compliant cryptographic modules. For additional detail for the recommendations in this section refer to NIST Special Paper 800-133.

3.6 Storage

Developers must understand where cryptographic keys are stored within the application. Understand what memory devices the keys are stored on.

Keys must be protected on both volatile and persistent memory, ideally processed within secure cryptographic modules.

Keys should never be stored in plaintext format.

Ensure all keys are stored in cryptographic vault, such as a hardware security module (HSM) or isolated cryptographic service.

If you are planning on storing keys in offline devices/databases, then encrypt the keys using Key Encryption Keys (KEKs) prior to the export of the key material. KEK length (and algorithm) should be equivalent to or greater in strength than the keys being protected.

Ensure that keys have integrity protections applied while in storage (consider dual purpose algorithms that support encryption and Message Code Authentication (MAC)).

Ensure that standard application-level code never reads or uses cryptographic keys in any way and use key management libraries.

Ensure that keys and cryptographic operation is done inside the sealed vault.

All work should be done in the vault (such as key access, encryption, decryption, signing, etc).

3.7 Escrow and Backup

Data that has been encrypted with lost cryptographic keys will never be recovered. Therefore, it is essential that the application incorporate a secure key backup capability, especially for applications that support data at rest encryption for long-term data stores.

When backing up keys, ensure that the database that is used to store the keys is encrypted using at least a FIPS 140-2 validated module. It is sometimes useful to escrow key material for use in investigations and for re-provisioning of key material to users if the key is lost or corrupted.

Never escrow keys used for performing digital signatures but consider the need to escrow keys that support encryption. Oftentimes, escrow can be performed by the Certificate Authority (CA) or key management system that provisions certificates and keys, however in some instances separate APIs must be implemented to allow the system to perform the escrow for the application.

3.8 Accountability and Audit

Accountability involves the identification of those that have access to, or control of, cryptographic keys throughout their lifecycles. Accountability can be an effective tool to help prevent key compromises and to reduce the impact of compromises once they are detected.

Although it is preferred that no humans can view keys, as a minimum, the key management system should account for all individuals who are able to view plaintext cryptographic keys.

In addition, more sophisticated key-management systems may account for all individuals authorized to access or control any cryptographic keys, whether in plaintext or ciphertext form.

Accountability provides three significant advantages:

1. It aids in the determination of when the compromise could have occurred and what individuals could have been involved.
2. It tends to protect against compromise, because individuals with access to the key know that their access to the key is known.
3. It is very useful in recovering from a detected key compromise to know where the key was used and what data or other keys were protected by the compromised key.

Certain principles have been found to be useful in enforcing the accountability of cryptographic keys. These principles might not apply to all systems or all types of keys.

Some of the principles that apply to long-term keys controlled by humans include:

- Uniquely identifying keys.
- Identifying the key user.
- Identifying the dates and times of key use, along with the data that is protected.
- Identifying other keys that are protected by a symmetric or private key.

Two types of audit should be performed on key management systems:

1. The security plan and the procedures that are developed to support the plan should be periodically audited to ensure that they continue to support the Key Management Policy (NIST SP 800-57 Part 2).
2. The protective mechanisms employed should be periodically reassessed with respect to the level of security that they provide and are expected to provide in the future, and that the mechanisms correctly and effectively support the appropriate policies.

New technology developments and attacks should be taken into consideration. On a more frequent basis, the actions of the humans that use, operate, and maintain the system should be reviewed to verify that the humans continue to follow established security procedures.

Strong cryptographic systems can be compromised by lax and inappropriate human actions. Highly unusual events should be noted and reviewed as possible indicators of attempted attacks on the system.

3.9 Key Compromise and Recovery

The compromise of a key has the following implications:

In general, the unauthorized disclosure of a key used to provide confidentiality protection (i.e., via encryption) means that all information encrypted by that key could be exposed or known by unauthorized entities. The disclosure of a Certificate of

Authorities' private signature key means that an adversary can create fraudulent certificates and Certificate Revocation Lists (CRLs).

A compromise of the integrity of a key means that the key is incorrect - either that the key has been modified (either deliberately or accidentally), or that another key has been substituted; this includes a deletion (non-availability) of the key. The substitution or modification of a key used to provide integrity calls into question the integrity of all information protected by the key. This information could have been provided by, or changed by, an unauthorized entity that knows the key. The substitution of a public or secret key that will be used (later) to encrypt data could allow an unauthorized entity (who knows the decryption key) to decrypt data that was encrypted using the encryption key.

A compromise of a key's usage or application association means that the key could be used for the wrong purpose (e.g., for key establishment instead of digital signatures) or for the wrong application and could result in the compromise of information protected by the key.

A compromise of a key's association with the owner or other entity means that the identity of the other entity cannot be assured (i.e., one does not know who the other entity really is) or that information cannot be processed correctly (e.g., decrypted with the correct key).

A compromise of a key's association with other information means that there is no association at all, or the association is with the wrong "information". This could cause the cryptographic services to fail, information to be lost, or the security of the information to be compromised. Certain protective measures may be taken in order to minimize the likelihood or consequences of a key compromise. Similar affect as ransomware, except that you can't pay the ransom and get the key back.

The following procedures are usually involved:

- Limiting the amount of time, a symmetric or private key is in plaintext form.
- Preventing humans from viewing plaintext symmetric and private keys.
- Restricting plaintext symmetric and private keys to physically protected containers. This includes key generators, key-transport devices, key loaders, cryptographic modules, and key-storage devices.
- Using integrity checks to ensure that the integrity of a key or its association with other data has not been compromised. For example, keys may be wrapped (i.e., encrypted) in such a manner that unauthorized modifications to the wrapping or to the associations will be detected.
- Employing key confirmation (see NIST SP 800-57 Part 1 Section 4.2.5.5) to help ensure that the proper key was, in fact, established.
- Establishing an accountability system that keeps track of each access to symmetric and private keys in plaintext form.
- Providing a cryptographic integrity check on the key (e.g., using a MAC or a digital signature).
- The use of trusted timestamps for signed data.
- Destroying keys as soon as they are no longer needed.

- Creating a compromise-recovery plan, especially in the case of a CA compromise. A compromise-recovery plan is essential for restoring cryptographic security services in the event of a key compromise. A compromise-recovery plan shall be documented and easily accessible.

The compromise-recovery plan should contain:

- The identification and contact info of the personnel to notify.
- The identification and contact info of the personnel to perform the recovery actions.
- The re-key method.
- An inventory of all cryptographic keys and their use (e.g., the location of all certificates in a system).
- The education of all appropriate personnel on the recovery procedures.
- An identification and contact info of all personnel needed to support the recovery procedures.
- Policies that key-revocation checking be enforced (to minimize the effect of a compromise).
- The monitoring of the re-keying operations (to ensure that all required operations are performed for all affected keys).

Any other recovery procedures, which may include:

- Physical inspection of the equipment.

- Identification of all information that may be compromised as a result of the incident.
- Identification of all signatures that may be invalid, due to the compromise of a signing key.
- Distribution of new keying material, if required.

3.10 Trust Stores

Design controls to secure the trust store against injection of 3rd party root certificates. The access controls are managed and enforced on an entity and application basis.

Implement integrity controls on objects stored in the trust store.

Do not allow for export of keys held within the trust store without authentication and authorization.

Setup strict policies and procedures for exporting key material from applications to network applications and other components.

Implement a secure process for updating the trust store.

3.11 Cryptographic Key Management Libraries

Use only reputable crypto libraries that are well maintained and updated, as well as tested and validated by 3rd party organizations (e.g., NIST/FIPS)

4 Policy Compliance

4.1 Compliance Measurement

The information security management team will verify compliance to this policy through various methods, including but not limited to, business tool reports, internal and external audits, and feedback to the policy owner.

4.2 Exceptions

Any exception to the policy must be approved and recorded by the Information Security Manager in advance and reported to the Management Review Team.

4.3 Non-Compliance

An employee found to have violated this policy may be subject to disciplinary action, up to and including termination of employment.

4.4 Continual Improvement

The policy is updated and reviewed as part of the continual improvement process.