Recite me®

Key Management Policy

www.reciteme.com

Version	Date	Details	Author
1.0.0	03/03/2022	Initial Version	Craig Tyers

Table of contents:

Introduction:

This policy will detail ways to securely handle keys for encryption of data, with a focus on security and preventing data loss.

Summary:

Management and handling of encryption and encryption keys is crucial to any data storage and encryption strategy.

An under prepared key management process could result in loss of data, loss of sensitive data and potentially result in severe penalties, including legal liabilities.

Purpose:

This document outlines Recite Me's Key management across our organisation which spans all electronic devices and methods of communication.

This policy will set out the minimum key management policy which should be adhered to in all cases.

Scope:

This document is applicable to all Recite Me employees especially with regard to those involved in the storing or retrieval of data within Recite Me and Recite Me's server architecture.

Policy Statement:

Data Transit:

Data transit between any elements of the Recite Me infrastructure should always take place via an approved, secured method.

In the case of data flowing into Recite Me from the web (via the toolbar or any other means) this should be via HTTPS.

In the case of transfer between server and workstations, this should be via a secured, public/private key pair over SSH.
In such cases, the private key should be unique to the user and it is the user's responsibility to maintain and securely store their unique private key.

Data Storage:

Not Stored:

The most secure method of storing data will always be to not store data.
Therefore the first step in any process to determine the required encryption level for the data you intend to store is to question if it really does need to be stored.

If we are able to achieve the goal without storing any additional data, then that is the route to take.

Key usage:

Any data that is to be stored should be assessed as to what level of encryption is required.

Publicly available:

Any data that we are receiving from a public facing system (such that the data is already available publicly) can be stored without the need for encryption.
Equally any data that is only available as a result of a look up with encrypted data as the source (such as the result of a translation) can also be stored unencrypted.
There is no benefit in encrypting data that is publicly available.

Hashed at Rest (One Way):

Any data that only needs to exist as a look up for comparison to a live value (for example a password) should be encrypted using a one way hashing algorithm; the key/salt for this algorithm should exist only on the system that needs to perform the encryption and subsequent comparisons.

The key should be unique to only this system.

The algorithm should be one of the approved hashing algorithms mentioned above.

Any loss or exposure of this key should not pose too large an issue as all the data should be for comparison use only, allowing for the current data to simply be dropped, a new key generated, and the system to start over.
In the case of passwords, this will require a user to reset their passwords.

In the case of text received from the web for translation/TTS processing, we should maintain the hashed result only; this is only for purposes of lookup, such that upon a subsequent request for the same secretion of text, we can re-hash and recognise that this is data we have already processed. In this way we use the hash of the data (as opposed to the data itself) as an identifier for any data relationships.

Hashed at Rest (Public/Private pair):

In cases where the data to be stored also needs to be retrieved (not just for a comparison value), the data should be stored using an approved Public key encryption algorithm.

The public key should be unique to the system.
Any private keys should also be unique.

It is very rare that we meet the necessity for any data that must be stored that is not A: public domain available or B: hashable one way as a reference. If data is falling into this category the first question we should face is - do we really need to store this data?

Key Dependency:

It stands to reason that the weakest link in any chain will be the source of problems, this is true for encryption algorithms too. Consider for example a system that encrypts all data in a public private key pair algorithm and a high bit depth. This is only ever as secure as the private key storage. If that private key storage is in an accessible system with only a short password protection, then that will be the weakness - not the data encryption itself.

For this reason it is important that any encryption of systems that takes place above the data storage layer be at least as strong as the data encryption both in terms of methodology and also in terms of bit depth.

Equally careful consideration should be given to cryptoperiod - a longer period of validity makes for a higher risk to a system, so as the levels are stacked the crypto period should shorten.

Generation:

Keys should be generated only by approved cryptographic algorithms, with additional attention paid to key dependency (any dependent keys should inform the choice of approved algorithm).

Access & Storage:

Keys should be protected such that access is only available to authorised users and applications.
Keys used to encrypt data should not be stored in the same media as the data.
Where keys and data are stored in physically or logically close proximity: all efforts should be made to protect that data, these efforts may include encryption of the keys. Other methods of data protection may also be suitable.

Removal:

When key removal is required - all symmetric and private keys must be removed in a secure and permanent manner from all systems (including from any back up systems)

Lifetime:

Key lifetime (cryptoperiod) should be defined as a function of the classification of the data the key is being used to encrypt along with the strength of the key's encryption.

Non-sensitive data encrypted with a strong key can have a longer crypto-period.
Whilst more sensitive data or data encrypted with a weaker key, should have shorter crypto periods, preferably with automated key replacement techniques (for example short lived GUID tokens).

It is the responsibility of the technical department to define and adhere to key crypto periods.

Compromise:

Detection:

Key usage should be logged such that an audit trail is available to assist in detecting potential compromise. Additional system monitoring tools may be used to assist in recognising that a compromise may have occurred. Logged usage is a bare minimum to enable detection.

Recovery:

Recovery following a compromise should be in line with any existing disaster recovery procedures.
Key recovery from back-up media is acceptable only in instances where compromise of the keys can be ruled out.
If key compromise can not be ruled out the key replacement should take place.

Replacement:

When key replacement is required, all previous symmetric or public keys relating to this encryption should be removed according to the removal policy set out in this document.

New keys should be generated and distributed again following the process set out in this document.

Any data affected should be re-encrypted with the new keys.

In the case of one way hashed at rest data (used only for comparative purposes). The data should be considered lost and removed from all systems.
If this is password hash values, then users will need to be requested to reset their passwords.
This should be achieved via a system which incorporates a second factor security process (such as requesting the password request via a hashed-check value delivered to a known safe source such as the user's email) or via a similar physical security methodology.

Responsibilities:

Recite Me technical staff (Developers / System administrators) are responsible jointly to uphold all requirements of this policy during daily activity.

Special attention should be given at project design development and execution stages to guarantee that all decisions taken are in line with this policy.

Breaches:

Following any detected breach of this policy, the data which has been stored in a manner that is deemed to be not in line with this policy will be removed.
This is to include backups and any reference or dependent data.
This purge of non conformant data is to happen within 24 hours of the breach being discovered (wherever possible - see Responsibilities).

Any system failing to comply will be removed from any production environments until such time as it can be updated to meet the requirements herein.

Following this, the system architects will be offered the opportunity to revisit this policy, to ensure that all team members are fully aware of the policy and its implications.

A breach should also trigger a review of this policy itself: to keep it in line with accepted current standards and expectations.

Review:

This policy will be reviewed following any substantial changes to infrastructure or related business policies, legislation changes and or any non-compliance complaints. The document will be automatically reviewed after annually.

Definitions:

Where reasonable and applicable, these definitions are taken from Recommendation for Key Management (Barker).

Access control	Restricts access to resources to only privileged entities.
Accountability	A property that ensures that the actions of an entity may be traced uniquely to that entity.
Archive	To place information into long-term storage. Also, see Key management archive.
Asymmetric key algorithm	See Public-key cryptographic algorithm.
At rest / rest	Pertains to data when in storage (see also In Transit)
Authentication	A process that establishes the source of information, provides assurance of an entity's identity or provides assurance of the integrity of communications sessions, messages, documents or stored data.
Authorization	Access privileges that are granted to an entity; conveying an "official" sanction to perform a security function or activity.
Backup	A copy of information to facilitate recovery during the cryptoperiod of the key, if necessary.
Ciphertext	Data in its encrypted form.
Collision	Two or more distinct inputs produce the same output. Also see hash function.
Compromise	The unauthorised disclosure, modification, substitution or use of sensitive data (e.g., keying material and other security-related information).
Confidentiality	The property that sensitive information is not disclosed to unauthorised entities.
Cryptographic algorithm	A well-defined computational procedure that takes variable inputs, including a cryptographic key, and produces an output.
Cryptographic hash function	See Hash function.
Cryptographic key (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. Examples include: 1. The transformation of plaintext data into ciphertext data, 2. The transformation of ciphertext data into plaintext data, 3. The computation of a digital signature from data, 4. The verification of a digital signature, 5. The computation of an authentication code from data, 6. The verification of an authentication code from data and a received authentication code, 7. The computation of a shared secret that is used to derive keying material.
Cryptoperiod	The time span during which a specific key is authorised for use or in which the keys for a given system or application may remain in effect.
Data integrity	A property whereby data has not been altered in an unauthorised manner since it was created, transmitted or stored. In this Recommendation, the statement that a cryptographic algorithm "provides data integrity" means that the algorithm is used to detect unauthorised alterations.
Decryption	The process of changing ciphertext into plaintext using a cryptographic algorithm and key.
Digital signature	The result of a cryptographic transformation of data that, when properly implemented with a supporting infrastructure and policy, provides the services of: 1. Origin authentication, 2. Data integrity, and 3. Signer non-repudiation.
Domain parameter	A parameter used in conjunction with some public-key algorithms to generate key pairs, to create digital signatures, or to establish keying material.
Encrypted key	A cryptographic key that has been encrypted using an approved security function with a key-encrypting key in order to disguise the value of the underlying plaintext key.
Encryption	The process of changing plaintext into ciphertext using a cryptographic algorithm and key.
Entity	An individual (person), organisation, device or process.
Ephemeral key	A cryptographic key that is generated for each execution of a key establishment process and that meets other requirements of the key type (e.g., unique to each message or session). In some cases, ephemeral keys are used more than once within a single session (e.g., broadcast applications) where the sender generates only one ephemeral key pair per message, and the private key is combined separately with each recipient’s public key.

Hash function	A function that maps a bit string of arbitrary length to a fixed-length bit string. Approved hash functions satisfy the following properties: 1. (One-way) It is computationally infeasible to find any input that maps to any pre-specified output, and 2. (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output.
Hash value	The result of applying a hash function to information.
Identifier	A bit string that is associated with a person, device or organisation. It may be an identifying name, or may be something more abstract (for example, a string consisting of an IP address and timestamp), depending on the application.
Identity	The distinguishing character or personality of an entity.
In transit / Transit	Pertains to data in transit, see also At Rest.
Key	See Cryptographic key.
Key destruction	To remove all traces of keying material so that it cannot be recovered by either physical or electronic means.
Key distribution	The transport of a key and other keying material from an entity that either owns or generates the key to another entity that is intended to use the key.
Key establishment	A function in the life-cycle of keying material; the process by which cryptographic keys are securely established among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key-transport and/or key-agreement protocols), or a combination of automated and manual methods (consists of key transport plus key agreement).
Key length	Used interchangeably with “Key size”.
Key management	The activities involving the handling of cryptographic keys and other related security parameters (e.g., passwords) during the entire lifecycle of the keys, including their generation, storage, establishment, entry and output, use and destruction.
Key pair	A public key and its corresponding private key; a key pair is used with a public-key algorithm.
Key revocation	A function in the life cycle of keying material; a process whereby a notice is made available to affected entities that keying material should be removed from operational use prior to the end of the established cryptoperiod of that keying material.
Key size	The length of a key in bits; used interchangeably with “Key length”.

Non-repudiation	A service that is used to provide assurance of the integrity and origin of data in such a way that the integrity and origin can be verified by a third party as having originated from a specific entity in possession of the private key of the claimed signatory.
Owner	For a static key pair, the entity that is associated with the public key and authorised to use the private key. For an ephemeral key pair, the owner is the entity that generated the public/private key pair. For a symmetric key, any entity that is authorised to use the key.
Password	A string of characters (letters, numbers and other symbols) that are used to authenticate an identity, to verify access authorization or to derive cryptographic keys.
Period of protection	The period of time during which the integrity and/or confidentiality of a key needs to be maintained.
Personal Data	Any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.
Plaintext	Intelligible data that has meaning and can be understood without the application of decryption.
Private key	A cryptographic key, used with a public-key cryptographic algorithm, which is uniquely associated with an entity and is not made public. In an asymmetric (public) cryptosystem, the private key is associated with a public key. Depending on the algorithm, the private key may be used, for example, to: 1. Compute the corresponding public key, 2. Compute a digital signature that may be verified by the corresponding public key, 3. Decrypt keys that were encrypted by the corresponding public key, or 4. Compute a shared secret during a key-agreement transaction.
Proof of possession (POP)	A verification process whereby assurance is obtained that the owner of a key pair actually has the private key associated with the public key.
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) cryptosystem, the public key is associated with a private key. The public key may be known by anyone and, depending on the algorithm, may be used, for example, to: 1. Verify a digital signature that is signed by the corresponding

	private key, 2. Encrypt keys that can be decrypted using the corresponding private key, or 3. Compute a shared secret during a key-agreement transaction.
Public-key (asymmetric) cryptographic algorithm	A cryptographic algorithm that uses two related keys: a public key and a private key. The two keys have the property that determining the private key from the public key is computationally infeasible.
Retention period	The minimum amount of time that a key or other cryptographically related information should be retained in the archive.
Secret key	A cryptographic key that is used with a secret-key (symmetric) cryptographic algorithm that is uniquely associated with one or more entities and is not made public. The use of the term “secret” in this context does not imply a classification level, but rather implies the need to protect the key from disclosure.
Secure communication protocol	A communication protocol that provides the appropriate confidentiality, authentication and content-integrity protection.
Security strength	A number associated with the amount of work (that is, the number of operations) that is required to break a cryptographic algorithm or system. In this Recommendation, the security strength is specified in bits and is a specific value from the set {80, 112, 128, 192, 256}.
Static key	A key that is intended for use for a relatively long period of time and is typically intended for use in many instances of a cryptographic key establishment scheme. Contrast with an ephemeral key.
Symmetric key	A single cryptographic key that is used with a secret (symmetric) key algorithm.
Unauthorised disclosure	An event involving the exposure of information to entities not authorised access to the information.

Bibliography:

Barker, E. *Recommendation for Key Management, Part 1: Genera*. Revision 3 ed., July 2012. *nvlpubs.nist.gov*, Computer Security Division (Information Technology Laboratory), https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-57p1r3.pdf. Accessed 3 Feb 2022.