ONLYOFFICE

Data encryption at rest 1.0

August 2020

Abstract

Out of voluntary intention to protect classified information or due to the effect of the industry-specific regulations, an organization might seek ways to secure parts of its data by means of encryption. ONLYOFFICE allows encrypting disk data at rest with Encrypt-then-MAC mechanism. Follow through this White Paper to study the technology, principles and applicability of this method of data encryption.
This White Paper describes the components and mechanics of the version 1.0 of encryption at rest in ONLYOFFICE Workspace. Before referring to the contents of this paper, it is important to compare the version of the encryption functionality relevant to user organization's software.

Contents

Introduction

ONLYOFFICE has implemented a mechanism of data encryption at rest in the private server solutions to offer an additional voluntary protection for the user data stored within the system. It allows fully encrypting the persistent disk data, including documents, files, and user information, to protect it from breach, alteration, and other malicious actions from outside of the user organization.

Encryption at rest servers the following purposes for the organization:

Protection of the data from physical access. In a scenario when the attacker gains physical access to the hardware that is mishandled and attempts to breach and compromise the bare data stored on the disk, securing the static data with encryption and remote key management is vital.
Data security regulation compliance. Governments and industries imply regulations that require the subject organizations to provide particular data protection safeguards some of which include the implementation of data encryption of specific level. Within HIPAA, for instance, data encryption is included in the list of technical safeguards for organizations that process protected medical patient information.

The chosen data encryption model allows performing swift encryption and decryption of large amounts of data to snap it to the dynamic form when processing and save back to the storage with seamless yet secure key management procedures. The key management scheme ensures that control over the keys used in the encryption belong to the user organization alone by design.

This model involves symmetric encryption and decryption of the data with AES-256 encryption standard and data integrity validation procedure to establish a secure authorized encryption and decryption.

ONLYOFFICE Control Panel is a component that provides encryption at rest functionality for the administrators of the system.

Encryption model

Overview

Encrypt-then-MAC scheme is a chosen model of data encryption and decryption in ONLYOFFICE. This method allows reaching the highest definition of security in the authenticated encryption of data because the validation key is computed from the data that was encrypted first and stops the decryption process if the encryption signature appears to be incorrect, therefore posing no risk of forging the alternative data.

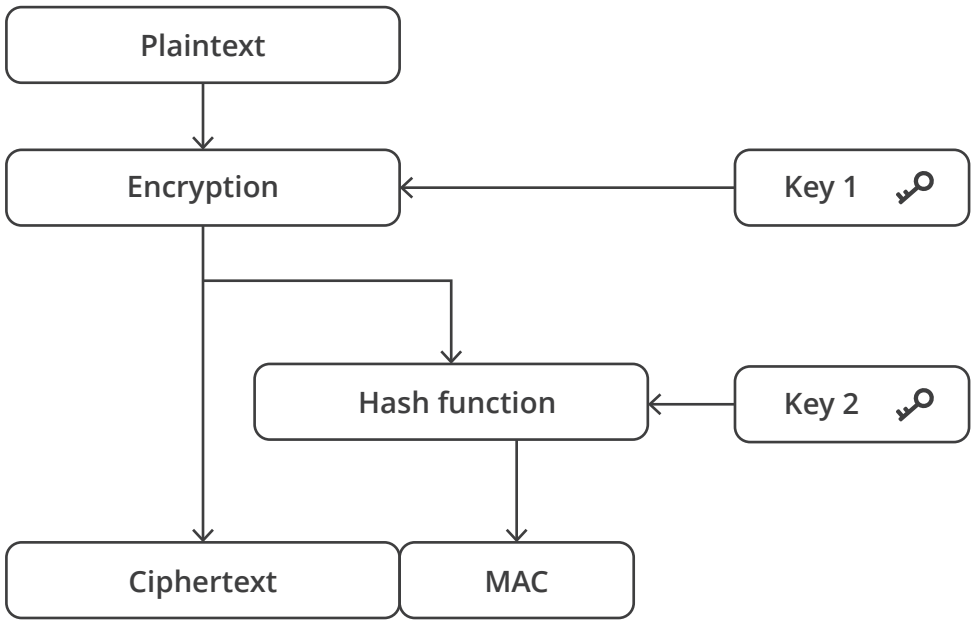
The applied scheme reliability is ensured with the following key provisions:

- It provides the integrity of the encrypted data. It is possible to deduce if the given encrypted text in the file (ciphertext) is authentic and terminate the decryption if it is invalid (has possibly been forged).

- It provides the initial data integrity as well because the signature is obtained after the text is encrypted.

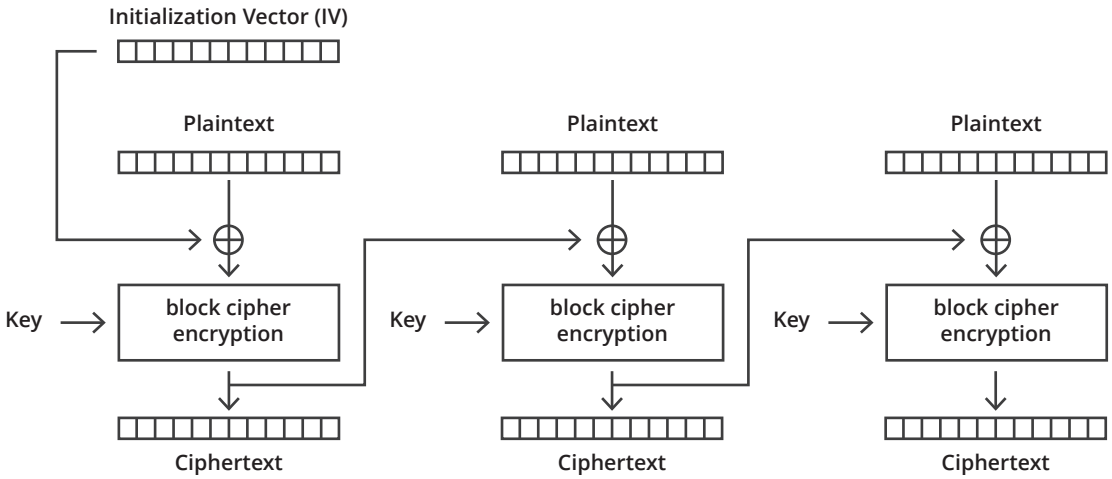
- It is not possible to compute any parts of the initial data from the MAC because it is generated based on the randomized input and no part of the original data is stored in the hash.

- While we are using the same password to access the encrypted files, each file's encryption metadata such as salt and the initialization vector are unique and random, and so are other encryption components derived from them.



Encryption algorithms

The files are encrypted using symmetric encryption algorithm called Advanced Encryption Standard (AES), also known as Rijndael. The size of the encryption key here is 256 bit, and the block size is 128 bit.
The algorithm for the cipher mode is Cipher Block Chaining (CipherMode.CBC)



Cipher Block Chaining (CBC) mode encryption

When encrypting the text, each final data block is filled until the integral number of bytes is reached, using PKCS#7 padding mode. In comparison to other padding modes applicable in AES encryption, It happens to be more reliable because it fills the blocks with inputs indistinguishable from the encrypted data.

Encryption password generation

The encryption Password is generated automatically using Membersip.GeneratePassword mechanism and saved to the database in the encrypted form. It is is executable on any ONLYOFFICE portal. The password length is set to 32 characters, out of which at least 16 are not alphanumeric, making it effective against brute force.

Key generation

The encryption Key is generated using the encryption password and a random Salt. The Salt is additional random data input to the function that hashes the data, which is in this case the encryption password. The salt is generated using RNGCryptoServiceProvider. The generated salt is unique for each file.
To generate a Key based on the Password and the Salt, the Rfc2898DeriveBytes function with SHA256 hashing algorithm is used, where the implied number of iterations is set to 4096 by default. This number can be changed in the “storage.encryption.iterations” setting of the configuration file. The resulting Key is 32 bytes (256 bits).

Cryptographic transformation of the data

To perform cryptographic transformation of the text, the Key and the initialization vector (IV) are required. The latter is defined by a random value, like the Salt, using RNGCryptoServiceProvider. The IV of each file is also unique.
Using the IV and the Key, it is possible to acquire the encryptor/decryptor needed for data encryption or decryption accordingly. The initial data and the encryptor/decryptor are processed in CryptoStream which results in cryptographically transformed data.

Encrypted data integrity validation

Data integrity validation is one of the key stages of protecting the encrypted data from the unauthorized intervention. HMAC is one of the existing methods of data integrity validation using MAC that allows to not only protect the initial data's integrity, but also its authenticity.
HMACSHA256 hash function calculates the SHA256 hash of the input data and signs it with the secret key (HmacKey), the size of which is recommended to be 64 bytes. The input data for HMAC is the IV combined with the encrypted data.
SHA516 hash (64 bytes) of the encryption Key is used as HmacKey. The output of the procedure is an encrypted data Signature.

Encrypted file structure

The final result of the encryption is the file that consists of the initial file's encrypted content (the encrypted data) and the metadata.The metadata of the encrypted file is the sequence of bytes that carries the information necessary for correct file decryption.
Encrypted file metadata used for decryption includes the following key components:

Component	Type	Description
Encryption version number	Number	Indicates the encryption version
Initial file size	Number	Allows retrieving the initial file size without decryption
Salt	Array	Ensures the ecryption key uniqueness. Necessary for file decryption
Signature	Array	Allows confirming data authenticity. Necessary for file decryption
Initialization vector	Array	Ensures the encryption uniqueness. Necessary for file decryption

File encryption and decryption

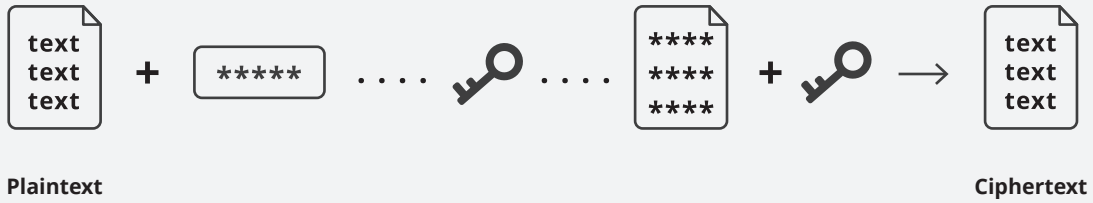
File encryption process

1. ONLYOFFICE checks the presence of file's encryption metadata.
   If there is any encryption metadata then the file is defined as already encrypted. Further process stops.
   If there's no encryption metadata, ONLYOFFICE proceeds to the next step.
2. The file size is estimated.
3. A random Salt is generated.
4. Using the encryption Password and the Salt, a Key is created to encrypt the file.
5. A random IV is is generated.
6. SHA515 is extracted from the Key to create HmacKey for the signature.
7. The file is encrypted using the Key and the IV.
8. The Signature of the IV and the encrypted file is calculated using the HmacKey.
9. In the «tmp» folder, a new file is created, containing the encrypted content and the encryption metadata. It is possible to assign another default folder in the configuration file, using the «storage.encryption.tempdir». If this setting is absent, the default folder is «tmp»; if the setting is empty, the current file location is made a default folder.
10. The initial file is replaced with the resulting encrypted file.

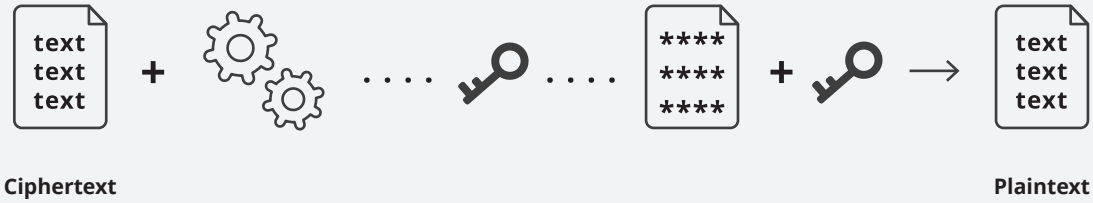
File decryption process

1. A file to decrypt is selected
2. ONLYOFFICE reads the the encryption metadata:
   - The encryption prefix to compare it with «AscEncrypted» string to confirm its validity.
   - The encryption version to compare it with the current version.
   - The initial file size to confirm it is more than zero.
   - The Salt, whole 32 bytes of which should be readable.
   - The Signature, whole 32 bytes of which should be readable.
   - The IV, whole 16 bytes of which should be readable.
3. If any of the verification stages is failed the file is defined as unencrypted and the decryption process stops.
4. Encrypted data integrity is checked. The Key is calculated using the Salt and and the file password. SHA512 is extracted from the Key and the HmacKey is acquired. The Computed Signature is calculated using HmacKey and the IV. The Computed Signature is compared to the Signature: if they are equal, the data integrity is confirmed.
5. Using the Key and the IV, the encrypted data is decrypted. It is then sent to the «tmp». It is possible to assign another temporary folder in the configuration file, using the «storage.encryption.tempdir». If this setting is absent, the default folder is «tmp»; if the setting is empty, the current file location is made a default folder.
6. The encrypted file is replaced with its decrypted version.

Encryption process



Decryption process



Limitations

To ensure the correct performance of the encryption process and the positive result, it is important to understand the limitations and prerequisites associated with the functionality:

- Backup copies cannot be encrypted by default. When disk encryption has been performed, each new backup archive created automatically or manually will contain unencrypted data. However, if you restore the data from the backup it will be encrypted automatically when the restoration is completed.
- The portal is blocked once a critical error occurs. For security reasons, if any emergency situation occurs (critical errors, service failure/termination, power outage), the portal will be blocked. It will be possible to restore access manually via the database.
- Encryption cannot be performed simultaneously with backup process. Backup processes on all portals must be finished before starting the disk encryption.
- «Static data» and «cdn» must be stored locally (Local Storage option).
- Disk encryption is available only for ONLYOFFICE Workspace since version **[.....]** of the Control Panel.

Make sure to study the instructions in **ONLYOFFICE Help Center** before performing full disk encryption.

Conclusion

ONLYOFFICE provides full disk encryption functionality to help the users of standalone self-hosted solutions secure data at rest and stay compliant with the regulations that require data encryption. Key value propositions in the technology used by ONLYOFFICE are the secure encryption scheme that ensures strong protection of data, guards its authenticity and integrity, and full control over the encrypted data and the encryption key management. This White Paper explains key methods and components of the technology behind the encryption functionality, as well as the applicable limitations.

References and further reading

(2000). Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1976.

(2001). Unforgeable encryption and chosen ciphertext secure modes of operation. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1978.

(2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. International Journal of Computer Applications, 67(19).

(1994). The security of cipher block chaining. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 839 LNCS.