

Server-side Encryption



- [Overview](#)
 - [Definitions](#)
 - [Encryption Method Comparison](#)
 - [Key Points & Limitations](#)
 - [Before You Enable Encryption](#)
 - [Key Management Modes](#)
 - [Enabling Encryption \(Step-by-Step\)](#)
 - [Backups](#)
 - [Encrypting External Mountpoints and Team Folders](#)
 - [Managing Encryption via occ Commands](#)
 - [Encrypting All Files](#)
 - [Decrypting Files / Disabling Encryption](#)
 - [Data Not Encrypted](#)
 - [User Keys: Sharing & Recovery](#)
 - [LDAP and External User Backends](#)
 - [Troubleshooting](#)
 - [Why don't I see the recovery key option in the Encryption settings?](#)
 - [Invalid private key for encryption app](#)
 - [Bad signature error](#)
 - [Encryption key cannot be found](#)
 - [Encryption key cannot be found with external storage or group folders](#)
 - [Further Reading](#)
-

Overview

Nextcloud provides several encryption methods to protect your data. These work at different layers of the system and address different security needs. This guide focuses on Nextcloud's built-in



Encryption and risk management is a complex and nuanced topic. Unless you are already an expert, it is recommended to consult a professional or perform a deep dive to ensure your approach targets your specific concerns. You may also find this Nextcloud Blog post helpful: [Data encryption methods in Nextcloud](#). It provides a solid high-level overview of the different approaches commonly considered when using Nextcloud.

Definitions

- **Server-Side Encryption (SSE):** Performed by the Nextcloud server, protecting files at rest on local and external storage. Encryption keys are stored on the server.
- **End-to-End Encryption (E2EE):** Performed by Nextcloud desktop or mobile clients before uploading files. Only the client can decrypt, making data inaccessible to server administrators and external storage providers.
- **Master Key:** A central key controlled by the server, used to encrypt all files.
- **User Keys:** Each user has their own key, protected by their password, to encrypt their files.
- **Recovery Key:** An admin-defined key to recover files if users lose their passwords.
- **Disk/Block Device Encryption:** A method of securing all data stored on a physical storage device by encrypting it at the hardware or filesystem level - typically using tools such as LUKS on Linux - so that data is only accessible after the device is unlocked with the correct key or password.

Encryption Method Comparison

Method	Encryption Location	Who Can Decrypt?	Protects Against
SSE (Master Key)	Server	Admins & users	External storage providers
SSE (User Keys)	Server	Users & malicious admins	External storage providers
SSE (User Keys w/ Recovery)	Server	Users & admins with recovery key	External storage providers
E2EE	Client	Users only	Admins, external storage providers

Method	Encryption Location	Who Can Decrypt?	Protects Against
Disk/Block Encryption	Server	OS admin	Physical tampering, theft

Key Points & Limitations

- Encryption methods are not interchangeable; each is designed for specific risks.
- **Server-Side Encryption (SSE)** is mainly for protecting files on external, third-party storage.
- **End-to-End Encryption (E2EE)** is for scenarios where server administrators must not access data.
- SSE does **not** encrypt filenames or folder structures, only file contents.
- SSE does not protect data from a compromised Nextcloud server or malicious administrator. Use E2EE for this threat.
- SSE cannot be reversed via the Nextcloud Web interface.
- Troubleshooting SSE generally requires `occ` command access. Make sure you have this before enabling SSE!
- Losing encryption keys or your instance secret results in permanent data loss.
- Nextcloud quotas are based on unencrypted file size; files encrypted with SSE may be ~1% larger (was 35% before Nextcloud 25).
- SSL/TLS (HTTPS) terminates before files are encrypted, so files may be exposed in memory between SSL/TLS and Nextcloud's SSE encryption code.
- When files on external storage are encrypted with SSE, you cannot share them directly from the external storage provider; sharing is only possible via Nextcloud, since the decryption key never leaves the Nextcloud server.
- For local storage, it may be better to use other encryption tools, such as disk/block device encryption (e.g., LUKS) provided by your operating system. This protects against other concerns, such as theft of your physical server, which is not SSE's goal.

“ ”

SSE does **not** encrypt filenames or folder structures, only file contents.

“ ”

Don't confuse Nextcloud's SSE with S3 SSE-C (also supported).

Changed in version 9.0.0: Nextcloud (since v9.0.0) supports Authenticated Encryption for all newly encrypted files. See <https://hackerone.com/reports/108082> for technical details.



For maximum security, configure external storage with "Check for changes: Never". This causes Nextcloud to ignore new files not added via Nextcloud, preventing unauthorized additions by external storage admins. Do not use this if your storage is subject to legitimate external changes.

Before You Enable Encryption

1. Read this guide fully and understand the risks.
2. Back up your instance configuration and all encryption keys in a safe location before proceeding.
3. Decide which key management mode suits your needs (see below).

Key Management Modes

Master Key (default):

- All files are encrypted with a central server-controlled key.
- Admins can decrypt any user's files.
- **Recovery keys are not available in master key mode.** Files remain accessible if a user forgets their password, as they are encrypted by the master key, not the user password.
- Recommended for most deployments.

User Keys:

- Each user's files are encrypted with a password-protected key.
- Admins cannot (readily) decrypt files without the user's password, unless a recovery key is defined.
- If a user forgets their password and no recovery key exists, their files are lost.
- This mode does not work with all authentication methods (e.g., app passwords, single sign-on) and is only recommended for compatibility with older setups.

How to choose:

- If you trust your server administrators, use master key mode.
- If you need to prevent admins from accessing files, use E2EE.
- User key mode offers some protection against malicious server administrators, but has limitations.

To select user key mode:

Run:

```
occ encryption:disable-master-key
```

before enabling encryption.

Enabling Encryption (Step-by-Step)

1. Go to the **Server-side encryption** section of your Admin page.
2. Check **Enable server-side encryption**.
3. You'll see a message: "No encryption module loaded." Go to your Apps page and enable the Nextcloud Default Encryption Module.
4. Return to your Admin page. The module will appear and be auto-selected.
5. Log out and log back in to initialize your encryption keys.
6. Optional: Un-check the box for encrypting home storage if you wish to keep local files unencrypted.

Server-side encryption ⓘ

Server-side encryption makes it possible to encrypt files which are uploaded to this server. This comes with limitations like a performance penalty, so enable this only if needed.

☒ Enable server-side encryption

Please read carefully before activating server-side encryption:

- Once encryption is enabled, all files uploaded to the server from that point forward will be encrypted at rest on the server. It will only be possible to disable encryption at a later date if the active encryption module supports that function, and all pre-conditions (e.g. setting a recover key) are met.
- Encryption alone does not guarantee security of the system. Please see documentation for more information about how the encryption app works, and the supported use cases.
- Be aware that encryption always increases the file size.
- It is always good to create regular backups of your data, in case of encryption make sure to backup the encryption keys along with your data.

This is the final warning: Do you really want to enable encryption?

Enable encryption

Server-side encryption

Server-side encryption makes it possible to encrypt files which are uploaded to this server. This comes with limitations like a performance penalty, so enable this only if needed.

☒ Enable server-side encryption

Select default encryption module:

☒ Default encryption module

Server-side encryption

Server-side encryption makes it possible to encrypt files which are uploaded to this server. This comes with limitations like a performance penalty, so enable this only if needed.

☒ Enable server-side encryption

Select default encryption module:

☒ Default encryption module

Default encryption module

☒ Encrypt the home storage

Enabling this option encrypts all files stored on the main storage, otherwise only files on external storage will be encrypted

Backups

Encryption keys are stored in:

- `data/<user>/files_encryption` (per-user keys)
- `data/files_encryption` (system-wide/external storage keys)

Encrypting External Mountpoints and Team Folders

- You and your users can encrypt external mountpoints.

“

 - Set encryption options in the mount configuration for each external storage.
 - See [Mount Options](#) in [External Storage](#).
- To encrypt Team Folders, run:

```
occ config:app:set groupfolders enable_encryption --value=true
```

“

Only new or updated files in team folders will be encrypted.

Managing Encryption via occ Commands

Here is a reference table for common occ commands:

Command	Description
occ encryption:status	Show encryption status and module
occ encryption:enable	Enable server-side encryption
occ encryption:list-modules	List available encryption modules
occ encryption:set-default-module [Module ID]	Select default encryption module
occ encryption:encrypt-all	Encrypt all files for all users
occ encryption:decrypt-all [user]	Decrypt all files (or for one user)
occ encryption:show-key-storage-root	Show key storage location
occ encryption:change-key-storage-root [dir]	Move key storage directory

Command	Description
occ encryption:enable-master-key	Enable master key mode
occ encryption:disable-master-key	Disable master key mode
occ encryption:fix-encrypted-version	Fix bad signature errors
occ encryption:fix-key-location [user]	Fix key not found errors

“ ”

See the [Encryption commands](#) section of the `occ` reference guide for further `encryption` command examples and details.

Example: Move keys to a new directory (Ubuntu Linux):

```
cd /your/nextcloud/data
mkdir keys
chown -R root:www-data keys
chmod -R 0770 keys
occ encryption:change-key-storage-root keys
```

Encrypting All Files

By default, only new and changed files are encrypted when you enable SSE. To encrypt all files for all users run:

```
occ encryption:encrypt-all
```

- **Make sure you have backups before running.**
- The command creates a key pair for each user and encrypts their files.
- Progress is displayed until all files are encrypted.
- **Make sure no users access files during this process.**

Decrypting Files / Disabling Encryption

- Only possible via occ.
- First, decrypt all files:

```
occ encryption:decrypt-all
```

- **Make sure you have backups before running.**
- Server enters maintenance mode. If interrupted, rerun until complete.
- If some files remain encrypted, rerun the command after resolving issues.
- **Warning:** Disabling encryption without decrypting all files will cause unpredictable errors.

You can decrypt for individual users:

```
occ encryption:decrypt-all <user-id>
```

Data Not Encrypted

Only file contents are encrypted. The following are **not** encrypted:

Not Encrypted
Filenames and folder structures
Existing trash bin files
Existing historical file versions
Image thumbnails
Image previews
Full text search index
Application data that isn't file-based (e.g., Deck, Tables)

User Keys: Sharing & Recovery

Sharing encrypted files:

- After enabling user key mode, users must log out and log in to generate keys.
- Users who see "Encryption App is enabled but your keys are not initialized..." must log out and back in.
- Shared files may need to be re-shared after encryption is enabled.
 - For individual shares: un-share and re-share the file.
 - For group shares: share with any individuals who cannot access the share, then remove their individual shares.

Can not decrypt this file, probably this is a shared file. Please ask the file owner to reshare the file with you.

Enabling file recovery keys:

Recovery keys are only available in per-user key mode (not the default master key mode).

- If you lose your Nextcloud password, you lose access to your encrypted files.
- If a user loses their password, their files are unrecoverable unless a recovery key is enabled (per-user key mode only).
- To enable recovery (in per-user key mode), go to Encryption in Admin page and set a recovery key password.
- Users must enable password recovery in their Personal settings for the Recovery Key to work.
- For users who have enabled password recovery, admins can reset passwords and recover files using the Recovery Key.



The recovery process can be slow and resource-intensive, especially for instances with large amounts of encrypted data. Test recovery procedures before relying on them in production.

Server-side encryption *i*

☒ Enable server-side encryption

Select default encryption module:

☒ Default encryption module

Enable recovery key

The recovery key is an extra encryption key that is used to encrypt files. It allows recovery of a user's files if the user forgets his or her password.

●●●●●●●●

●●●●●●●●

Disable recovery key

Encryption

Enable password recovery:

Enabling this option will allow you to reobtain access to your encrypted files in case of password loss

☒ Enabled

☐ Disabled

File recovery settings updated

Group Admin for

Quota

no group

Unlimited

Change recovery key password:

Old Recovery key password

New Recovery key password

Repeat New Recovery key password

Change Password

LDAP and External User Backends

- If using LDAP/Samba and changing passwords on the backend, users will need both their old and new passwords on next login.
- If recovery key is enabled, admins can reset the password via Nextcloud and notify users.

Troubleshooting

Why don't I see the recovery key option in the Encryption settings?

Recovery keys are only available in per-user key mode. Since Nextcloud 13, the default encryption mode uses master keys (system-wide encryption). Master key mode does not expose recovery key options in the Admin settings because recovery keys are not needed—admins can reset user passwords and files remain accessible.

If you are using master key mode (the default and recommended mode), you do not need recovery keys. Recovery keys are only relevant for per-user key setups, which are maintained for compatibility with older deployments.

See [Key Management Modes](#) for guidance on the differences between master key and per-user key modes, and [GitHub Issue #8283](#) for technical context on this design decision.

Invalid private key for encryption app

See [GitHub Issue #8546](#) and [workaround](#).

Bad signature error

In some rare cases, encrypted files cannot be downloaded and return a "500 Internal Server Error." If the Nextcloud log contains an error about "Bad Signature," run the following command to repair affected files:

```
occ encryption:fix-encrypted-version userId --path=/path/to/broken/file.txt
```

Replace "userId" and the path accordingly. The command will perform a test decryption for all files and automatically repair those with a signature error.

Encryption key cannot be found

If the logs contain an error stating that the encryption key cannot be found, you can manually search the data directory for a folder that has the same name as the file name. For example, if a file "example.md" cannot be decrypted, run:

```
find path/to/datadir -name example.md -type d
```

Then check the results located in the `files_encryption` folder. If the key folder is in the wrong location, move it to the correct folder and try again.

The `data/files_encryption` folder contains encryption keys for group folders and system-wide external storages, while `data/$userid/files_encryption` contains the keys for specific user storage files.

“ ”

This can happen if encryption was disabled at some point but the [occ command for decrypt-all](#) was not run. If someone then moved the files to another location, the keys did not get moved.

Encryption key cannot be found with external storage or group folders

To resolve this issue, run the following command:

```
sudo -E -u www-data php occ encryption:fix-key-location <user-id>
```

This will attempt to recover keys that were not moved properly.

If this doesn't resolve the problem, refer to the section [Encryption key cannot be found](#) for a manual procedure.

“ ”

There were two known issues where:

- moving files between an encrypted and non-encrypted storage like external storage or group folder [would not move the keys with the files](#).
- putting files on system-wide external storage would store the keys in the [wrong location](#).

Further Reading

- [occ Command Reference: Encryption](#)
 - [How Nextcloud uses encryption to protect your data](#)
 - [Technical impact of Authenticated Encryption](#)
 - [Nextcloud SSE Implementation Details](#)
 - [Nextcloud Encryption \(SSE & E2EE\) Recovery Tools](#)
 - [Nextcloud E2EE Server API App \(required for E2EE usage\)](#)
-

📦 #3

📦 cocobook-sync 📦 2026-10-28 16:28:03 UTC

📦 cocobook-sync 📦 2026-10-29 01:09:19 UTC