NETFILES

End-to-end encryption

in netfiles Data Vault

Content

1. Introduction 3
2. Requirements 4
3. Overview 4
4. Implementation 6
5. Inviting users 11

1. Introduction

End-to-end encryption ("E2EE“) enables users to securely exchange information, regardless of the communication channel. Messages and data are encrypted and decrypted on the sender’s or recipient’s device – before and after transmission – ensuring continuous protection throughout the entire transfer.

This document provides a compact overview of how netfiles implements end-to-end encryption in netfiles Data Vault, ensuring that:

- document content, comments, and annotations are transmitted to netfiles servers in encrypted form only;
- encryption and decryption occur exclusively on the user’s device (in the browser or netfiles app);
- the encryption keys are never stored on netfiles servers in a way that would allow netfiles to decrypt content, comments, or annotations;
- all of this is in addition to the existing encryption of the data room and the TLS encryption used for communication with the netfiles servers.

2. Requirements

End-to-end encryption (E2EE) is a strong protective measure for a company's most sensitive data. Because encryption and decryption take place solely on the user's device, some productivity features may be limited or unavailable. This is due to the fact that such features – like virus scanning or online document editing – require temporary server-side access to unencrypted data.

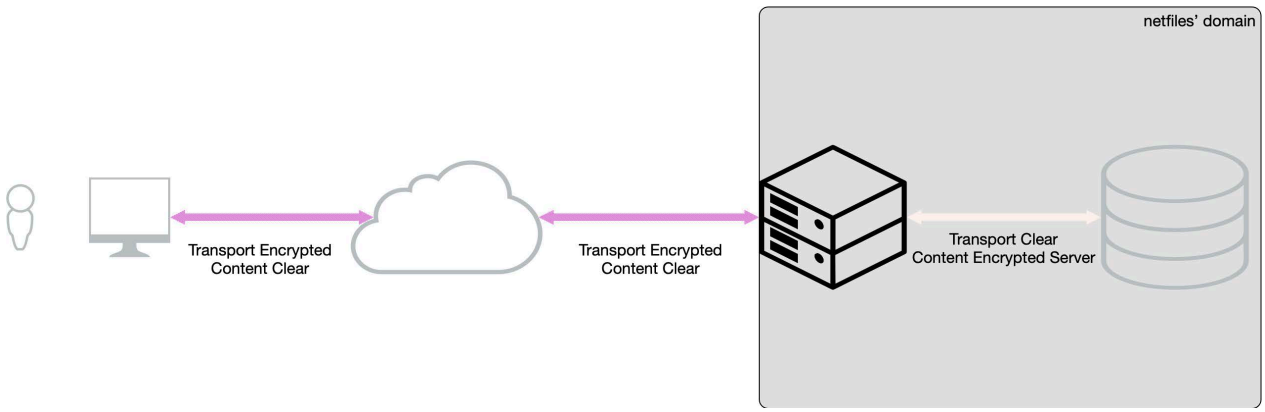
When should you use netfiles Data Vault?

Use of netfiles Data Vault is recommended when the data in question requires the highest level of protection. Even without E2EE, netfiles meets the highest security standards: it is ISO 27001 and SOC 2 certified, BSI C5-audited, and regularly penetration-tested by independent parties. All data is AES-256 encrypted during both transmission and storage on netfiles servers.

For further information, we recommend reviewing our detailed security concept.

3. Overview

End-to-end encryption in netfiles Data Vault complements the standard encryption already used in netfiles data rooms. Even without E2EE, all data is protected by transport encryption ("encryption-in-transit") and storage encryption ("encryption-at-rest"):



Transport encryption

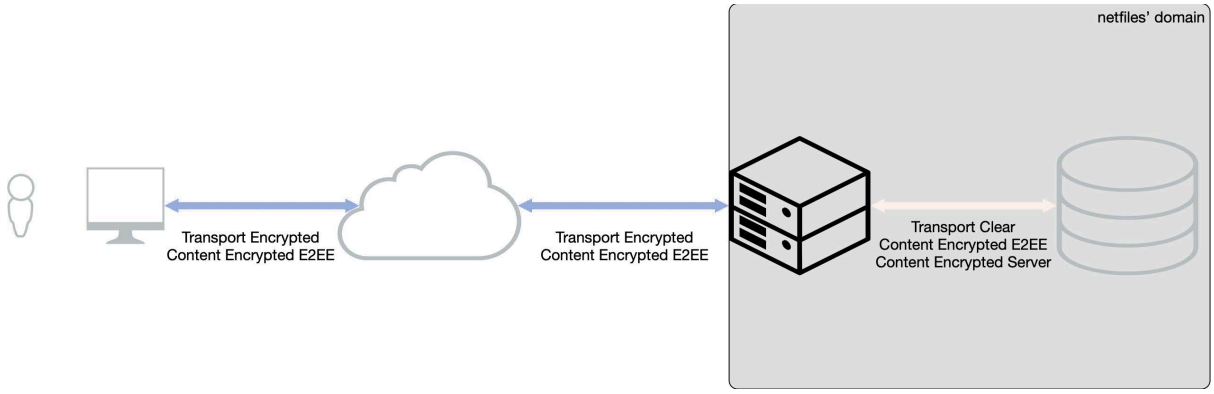
All data uploaded from or downloaded to a user's device is transmitted via a TLS-encrypted connection. TLS secures the entire communication with netfiles servers, preventing outsiders from reading or tampering with the transmitted data.

Encryption at rest

Data is stored on netfiles servers in encrypted form only. netfiles applies AES-256 encryption to all incoming data. Decryption occurs solely as part of specific server-side services (e.g. virus scanning or online document editing).

End-to-end encryption

With netfiles Data Vault, content is encrypted directly on the user's device before transmission. Transmission is still protected via TLS. Once the encrypted content reaches netfiles servers, it is encrypted again (as we have built E2EE on top of existing infrastructure) and stored in the database.



Encryption and decryption take place entirely in the background ("on-the-fly") and require no further user action after logging in and entering the personal "data key".

Important note: Because the encrypted content arrives at netfiles servers without any access to the E2EE key, server-side services requiring access to unencrypted content (e.g. virus scanning or online editing) are not available.

4. Implementation

All encryption processes – including key management – run in the background and require no technical knowledge on the user's side. When a user first creates or accesses a netfiles Data Vault, they are prompted to set a secure password for end-to-end encryption (E2EE) use. This password becomes the personal "data key" and serves as the foundation for all encryption activities.

To implement E2EE, netfiles performs the following tasks:

- Encrypts and decrypts content directly in the browser using an AES-256 key to ensure data confidentiality and integrity.
- Stores the encryption key securely in a location inaccessible to netfiles.
- Securely distributes encryption keys to newly invited data room members.

netfiles uses the following methods and technologies:

- RSA-4096: Key exchange (public/private key pair)
- Argon2id: Key derivation
- AES-GCM: Symmetric encryption (data encryption), including integrity check
- WebCrypto: For generating an asymmetric RSA 4096-bit key pair, for key management and for encryption/decryption

For encryption, decryption and key generation, netfiles uses WebCrypto and the Argon2id library. Since WebCrypto (as of March 2025) does not support Argon2id, netfiles uses a WebAssembly implementation of the Argon2id library.

RSA key pair: Public and private key (“data key”)

To share E2EE-protected data with others, users require an RSA key pair: a private key (only known to the user, used for decryption) and a public key (freely available, used for encryption). Both administrators and invited users need such a key pair. The key pair is created when a Data Vault is set up (administrator) or upon joining via invitation (user).

When the administrator creates a Data Vault for the first time, netfiles generates a 4096-bit RSA key pair using WebCrypto (crypto.subtle.generateKey(...)). The private RSA key is protected with a password that must be defined by the administrator. This password, which is used exclusively for E2EE, is the personal “data key”.

The private key is encrypted using an AES-256 “password key” derived from the administrator’s “data key” and an initialization vector (IV). The encrypted private key and public key are then stored in the user account.

Security note: RSA-4096 and Argon2id require modern hardware due to their computational intensity. netfiles deliberately chose these methods: RSA-2048 has been classified as insecure by the German BSI as of 2025. While RSA-3072 is currently recommended, it is considered a transitional solution. RSA-4096 is expected to remain secure into the 2040s. Our implementation allows for a future transition to stronger algorithms if necessary.

Important: The personal “data key” is independent of the netfiles user password, which is required for login. For security reasons, they should never be identical. The “data key” is the only encryption-related password administrators and users need to remember in addition to their netfiles login credentials.

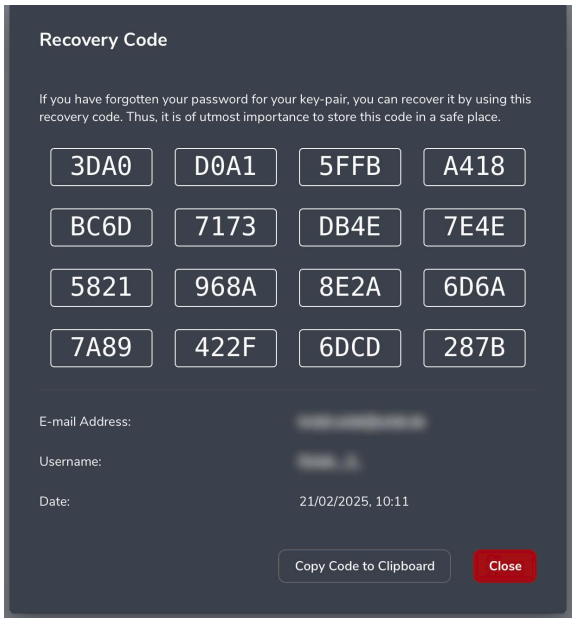
All other processes required for working with E2EE data are handled by netfiles in the background. The “data key” is unique for each netfiles user and can be used for several different data rooms. It therefore only needs to be created the first time a netfiles Data Vault is created/entered.

Recovery code

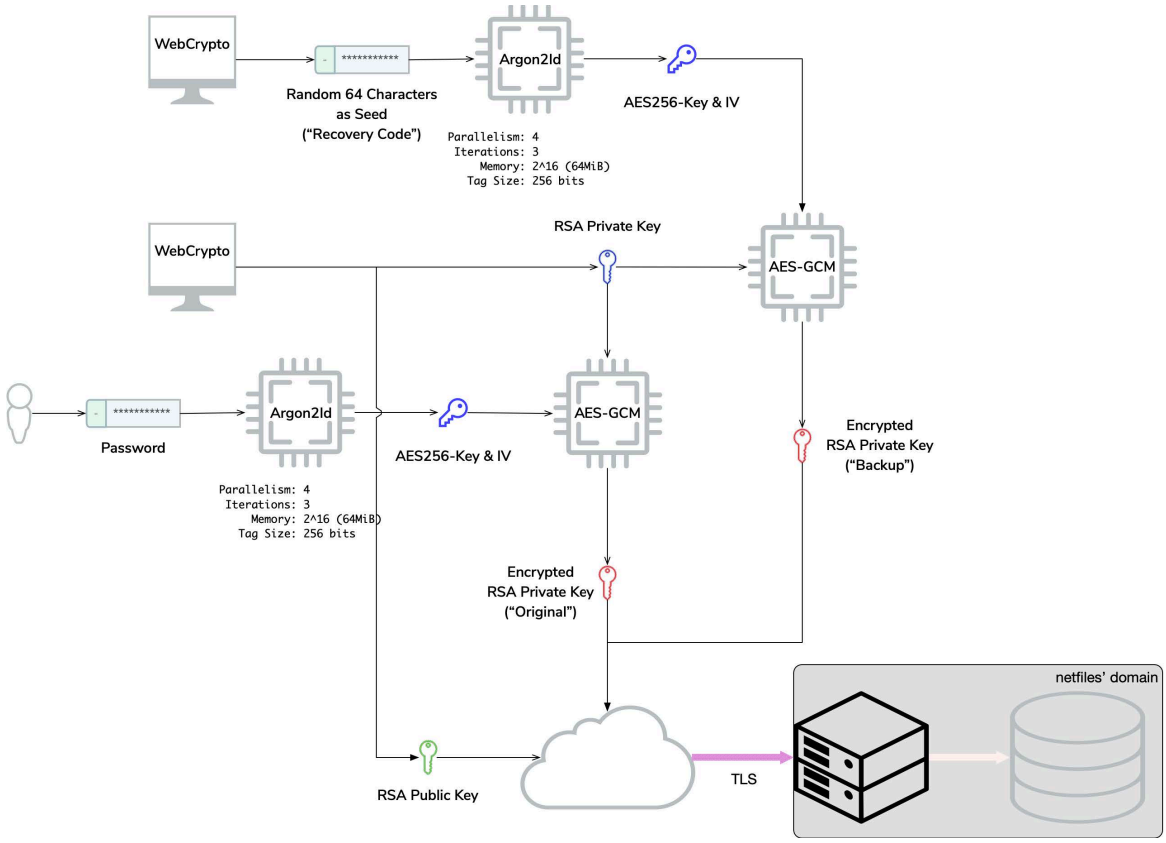
Because netfiles never knows a user's personal "data key" – the password required to decrypt E2EE-protected data – losing this key would make it impossible to access the user's encrypted data. To mitigate this risk, netfiles generates a "recovery key" for each user during the setup of E2EE.

As part of the password key derivation process, netfiles additionally creates a random sequence of 64 hexadecimal characters, known as the "recovery code". This code serves as a seed for a separate Argon2id key derivation process. From this seed, a second AES-256 key (the actual "recovery key") and initialization vector (IV) are generated.

The user's (unencrypted) private RSA key is then encrypted using this "recovery key" and IV and stored as a backup. The "recovery code" is provided to the user for secure storage. Should the user ever forget or lose their personal "data key" – i.e., the password that protects their private RSA key – the "recovery code" can be used to decrypt the backup and regain access to the encrypted data.



Overview: Key creation and storage with netfiles Data Vault

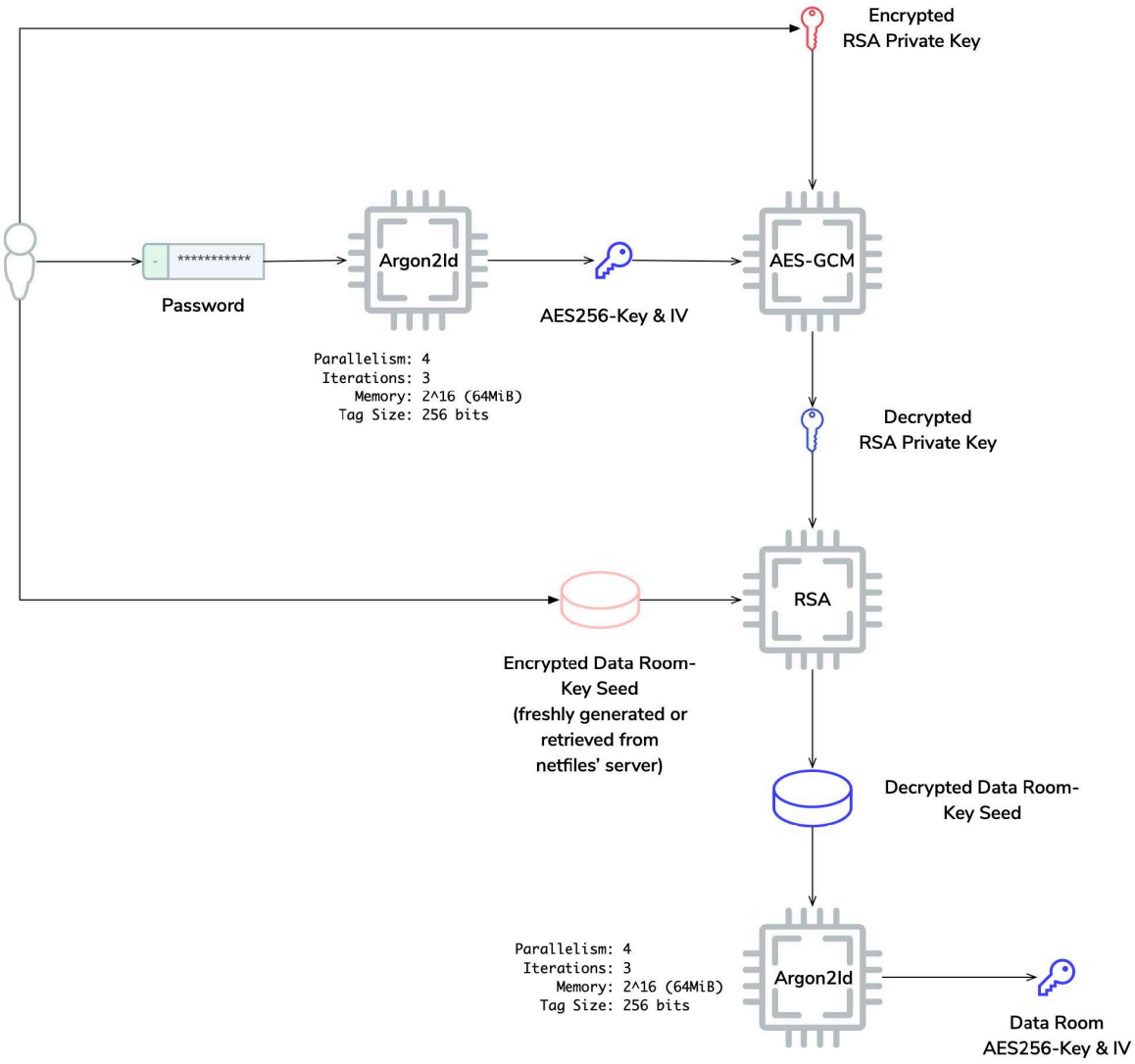


AES data room key

To encrypt data room content, netfiles generates a random 32-byte seed using WebCrypto (crypto.get-RandomValues(new Uint8Array(32))). The hexadecimal representation of this seed, referred to as the "data room key seed", is then used in Argon2id to derive the actual **AES-256 encryption key ("data room key")** and initialization vector (IV).

netfiles encrypts the "data room key seed" immediately using the administrator's public RSA key and stores it in the administrator's user account for this data room. The original, unencrypted seed is deleted from memory immediately afterwards, so that it is not kept in the working memory for longer than necessary. For each session, a session-specific "data room key" is created locally in the browser and remains valid only for that session.

To use the “data room key seed” again in a future session, netfiles follows a defined process:



Please note: All cryptographic keys – including the generated AES data room key and IV as well as the user’s decrypted private RSA key – are imported into WebCrypto with exportable: false. This ensures they cannot be exported or accessed outside the secure browser context. After import, any plaintext key material is immediately deleted from memory.

From that point on, all content, comments, and annotations are encrypted in the browser (before upload) and decrypted locally (after download). The entire process runs automatically in the background and is seamless to the user.

Note: If a user reloads or refreshes the data room browser tab, the “data key” (i.e. the password that protects their private RSA key) must be re-entered. This is because WebCrypto automatically deletes all previously imported keys when a page is refreshed and requires the data room key to be generated again.

5. Inviting users

Every user who is invited to a netfiles Data Vault must have an RSA key pair in their user account. This means that the user must have already set a personal “data key”. Depending on whether this key pair already exists, two invitation scenarios are possible:

User has an RSA key pair (“data key”)

If a user is invited and already has a key pair consisting of a private and public RSA key, the invitation is seamless: The administrator enters their personal “data key” when accessing the Data Vault. This decrypts the data room key seed locally in the browser and makes it available. netfiles then re-encrypts this seed – also locally in the browser – using the invited user’s public RSA key. The newly encrypted data room key seed is stored in the invited user’s account for this Data Vault.

After the user accepts the invitation, they must enter their personal “data key”. This decrypts the data room key seed locally in the browser, from which a data room key is generated. The user can then immediately join the Data Vault.

User does not yet have an RSA key pair (“data key”)

If an invited user does not yet have a key pair consisting of a private and public RSA key, a second confirmation by the administrator is required in the invitation process:

1. The administrator invites the user to a data room. When sending the invitation, the administrator is informed that they are inviting a user without an RSA key pair and must therefore make a second (final) confirmation as soon as the user has accepted the invitation and created a key pair.
2. If the invited user accepts the invitation, netfiles prompts the user to specify a data key and generates an RSA key pair (the procedure is the same as described in chapter 4, „Implementation“).
3. Once the user keys have been generated, netfiles informs the administrator that the user is ready to enter the data room. netfiles informs the user that a second, final confirmation by the administrator is required and has been arranged. The user will be informed by e-mail as soon as the administrator has issued the confirmation and access to the encrypted data room is possible.
4. The administrator is informed by e-mail that the invited user now has an RSA key pair and is awaiting final confirmation. When the administrator approves the user, netfiles performs the process for users who already have an RSA key pair (as described above).

Overview: Sharing a netfiles Data Vault with invited users