opentext™ | Cybersecurity

WHITE PAPER

Endpoint Backup

An in-depth look at the encryption technology behind our premium endpoint backup solution

As one of our core pillars, addressing the end-to-end security and privacy of data is a primary requirement of Endpoint Backup. By utilizing our automated key management and encryption technology in conjunction with our unique data deduplication, both efficiency of data deduplication and security of data can be accomplished. This document describes the end-to-end encryption process and encryption key life cycle management.

Company account creation

During creation of a company account (tenant), a cryptographically random asymmetric RSA 4096-bit encryption key is generated and stored in the vault. Company encryption keys are stored separately from any data that will be stored in the vault and are NOT used to encrypt data. Company encryption keys are used to wrap and escrow device keys as described below. Company encryption keys can be rotated as required by contacting support.

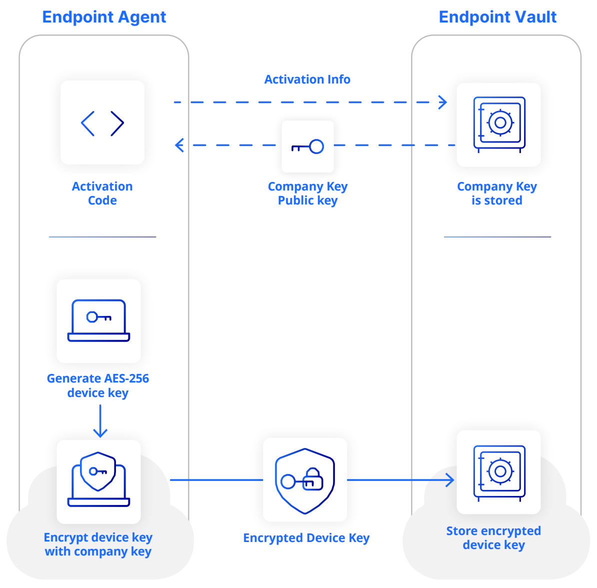
During company key rotation, a new cryptographically random key is generated. Each device key is re-wrapped with the new company key. Since no data is encrypted with the company key, rotation does not require any data to be reuploaded or any devices to be reactivated.

Device activation

Device activation is the process of having a new device register and activate itself with the vault. During device activation, the following key activities are performed:

1. First, an authenticated and encrypted session is established to the server over HTTPS using TLS 1.2.
2. Over this HTTPS session, the device will exchange its activation code with the server and once validated, it will be assigned a device identifier.
3. Additionally, the device will generate a cryptographically random AES 256-bit key known as the device key.
4. This key is stored locally on the device within DPAPI on Windows and within keychain on Mac.
5. The device key is also encrypted using the public key of the asymmetric company key and escrowed in the vault. The escrowed device key can then be securely supplied if a device requires a reset.

Device Activation Process



Device reset (reactivation)

Device reset is a process performed when a device needs to either be:

1. Uninstalled and reinstalled on the same device. During the uninstallation process, the device key is removed from the device and needs to be obtained again from the vault in order to continue to function.
2. Transferred to a new device. During installation on the new device, the device key needs to be retrieved from the vault for the new device to function.

The device reset process is as follows:

1. An administrator will use two-factor authentication (2FA) to login to the vault dashboard. Authentication and authorization methods – which are discussed separately – allow for enterprise controls such as role-based access control (RBAC) and single sign-on.

2. Once authenticated and authorized to reset a device, the administrator will choose the option to "reset" a device.
3. The administrator is prompted to enter a passphrase.
4. An AES 256-bit passphrase key is deterministically generated from this passphrase and a salt using PBKDF2 hashing with multiple rounds.
5. The device key that was escrowed in the vault is then encrypted with this passphrase key and stored in the vault. A reset activation code is generated and displayed to the administrator.

Once reset, the device can be reactivated and the device key retrieved as follows:

Once the software is installed on the new target device, the device will execute an activation process known as "reactivation." During reactivation, two pieces of information are required to be entered. The administrator must enter the activation code to associate this device to the reset device. The activation code is used to retrieve the encrypted device key from the vault over an HTTPS session using TLS 1.2 encryption. Additionally, the passphrase must be entered. The entered passphrase is used to deterministically generate the decryption key for the device key. Once the device key is decrypted on the device, it is stored using DPAPI on Windows or keychain on Mac. Once successfully reactivated, the reset activation code is no longer able to be used again. Any further reinstallations or transfers to a new device require this process to be executed again.

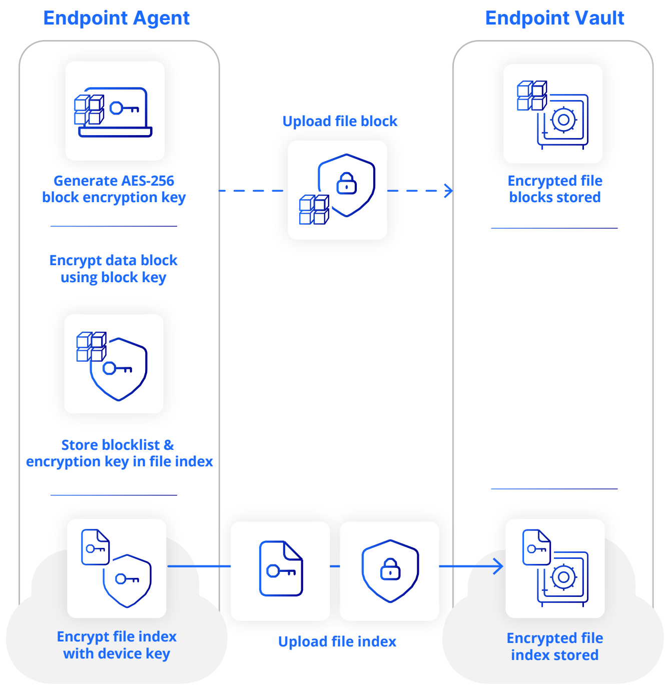
File / block encryption

Our client processes every file into congruent data blocks. From the server's point of view, each block is an opaque unit that it simply stores in its datastore, keyed by the block's unique cryptographic SHA256 hash of its contents. A block's integrity can be confirmed at any stage by computing the block data's SHA256 hash and comparing with its "block hash."

Each file is disassembled into a set of variable length blocks that are then processed on the client. Every block of data receives an encryption key, which is derived from the cryptographic SHA256 hash of the block content and a salt. The salt is scoped to a company by default but can be further limited to a user group or device. Changing the salt will prevent deduplication from occurring outside of the scope. The AES 256-bit encryption key is then used to encrypt the block. These block keys are then placed into the file index and the file index is encrypted using the AES 256-bit device key and uploaded to the vault. At no point are the block keys written to disk. The encryption of file index containing the block keys by the device keys prevents a different device from accessing the block encryption keys.

The encryption process is then repeated for each additional file configured for backup.

File/block Encryption Process


Endpoint Agent
Endpoint Vault
Generate AES-256 block encryption key
Upload file block
Encrypted file blocks stored
Encrypt data block using block key
Store blocklist & encryption key in file index
Encrypt file index with device key
Upload file index
Encrypted file index stored

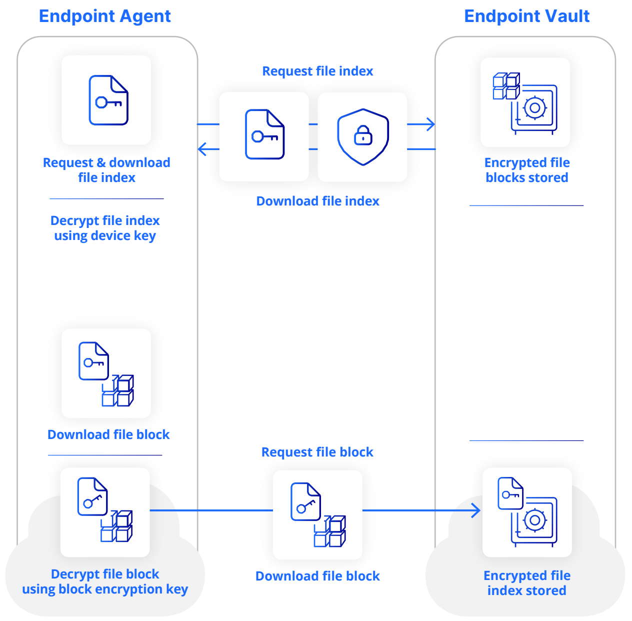
Restore process

To execute a restore from the device, it must be activated. This could either be the initial activation or a successful reactivation process as described above. Once a file (or set of files) has been selected for restore, the steps executed during each restore process include:

1. An encrypted file index is retrieved from the vault.
2. The encrypted file index is decrypted using the device key. This means that only a device with the appropriate device key can access the block keys and thereby the block data.
3. The file index is parsed to obtain the list of blocks and the associated block encryption key.
4. Each block of data is downloaded from the vault over an HTTPS session using TLS 1.2.
5. Once downloaded, every block's integrity is checked.
6. On the device, the block of data is decrypted using the block encryption key referenced in the file index.
7. The download and decryption process are repeated for each block of data referenced in the file index.
8. The file is written to disk.
9. This process is repeated for each file selected for restore.

This restore process is followed for all restores whether initiated from the device itself or initiated by an administrator using the web dashboard. If an authenticated administrator, with the appropriate authorization, initiates a restore to another device other than the source device, an additional step is executed that will trigger the target device to download the escrowed device key for the source device from the vault to enable the restore. This key is stored using DPAPI on Windows and keychain on Mac. It is removed when the restore has finished.

File/block Encryption Process


Endpoint Agent
Endpoint Vault
Request file index
Request & download file index
Decrypt file index using device key
Download file index
Encrypted file blocks stored
Download file block
Request file block
Decrypt file block using block encryption key
Download file block
Encrypted file index stored

Web access restore

Web access is a feature that may be enabled by administrators. It allows users to retrieve files through a web browser or mobile device application. Web access may be enabled or disabled on a user basis. When using web access, the following process is utilized to access the data:

1. A user must authenticate to the web access portal. Enterprise single sign-on capabilities can be enabled as the authentication mechanism. A user must be authorized for web access and can only access devices assigned to their user account. Users cannot access devices assigned to other users via the web access feature.
2. On the web server backend, the encrypted file index is retrieved. The file signature block keys are decrypted using the device key that was escrowed in the vault and held in memory.
3. The file signature is parsed to obtain the list of blocks and the associated block encryption key.
4. Each block of data is retrieved from the datastore and held in memory on the web server.
5. Each block of data is decrypted using the block encryption key referenced in the file index.
6. The retrieval and decryption of data is repeated for each block of data referenced in the file index.
7. The fully assembled file is streamed to the user's browser over a TLS 1.2 session.
8. The device and block keys are removed from memory.

During web restore, no encryption keys are stored on disk within the vault.

If enabled by the administrator, the fully assembled file may be attached to an email and sent to the specified address. This feature may be enabled or disabled by administrators to comply with their enterprise controls.

Key management summary

This table provides a summary of the various keys discussed above and how each key is protected

Header	Encrypted with	Key type	Managed by	Notes
Block of data	Block key	AES 256	Carbonite	
Block key	Device key	AES 256	Carbonite	Unique key for each block
Device key	Carbonite or enterprise key controller	AES 256	Carbonite	Unique key for each device
Company key		RSA 4096	Carbonite or customer (EKC)	Unique key for each company

Authentication and authorization

All connections to the dashboard require an authenticated session. We allow users to create passwords within the dashboard that are used for authentication. All passwords are hashed using a one-way SHA-256 hashing algorithm with salt. Complex passwords are required and must contain eight characters, an upper case and lower case letter, and at least one number or symbol. After the first login, users are required to use either single sign-on or set up two-factor authentication.

Two-factor authentication is a security process in which a user provides two different factors to verify their identity. This helps ensure that users are who they claim to be and provides an additional layer of security to protect against unauthorized access. Endpoint Backup's 2FA time-based one-time password (TOTP) algorithm generates a one-time password that changes every 30 seconds based on the current time. 2FA passcodes, unlike conventional passwords, are only used once. 2FA works with authenticator apps such as Google Authenticator, Microsoft Authenticator and others.

Customers may already have an enterprise single sign-on system in place that imposes additional requirements such as two-factor authentication, password rotation periods, and even domain or IP restrictions. We support single sign-on via a SAMLv2 protocol for authenticating a session. When using a single sign-on provider, the user will be challenged to authenticate with the SSO provider and then, upon successful login to the SSO provider, they are redirected to the dashboard. This allows you to easily extend your enterprise controls to work with the dashboard. With SSO, It helps to reduce the number of passwords that users need to remember, which can improve security by reducing the risk of weak or easily guessable passwords. It also streamlines the login process, which can be especially useful in organizations where users need to access multiple applications and services.

Once authenticated into the application, all users must also be authorized to perform actions. We support different roles and permission scope for each user. A user must be explicitly granted permissions and must be granted a scope that allows them access.

Term	Definition
Authentication	Authentication challenges the user to prove that they are who they say they are by meeting challenges such as a complex password or multi-factor authentication.
Scope	Scope is the breadth of objects a user can have access to. This allows users to have visibility into only certain objects.
Permission	A permission is an explicit grant of a capability such as the ability to read information, add new information or delete information.
Authorization	Authorization is a check that the authenticated user meets the scope and permission requirement to execute a requested action.

We support several user roles. Each role can be assigned to a specific user and scope. Roles can be scoped at a partner, company or user group level.

	Admin	Backup admin	Legal admin	Support	Read only
Companies	All	No add/edit/delete	Read only	Read only	Read only
Users	All	All	Read only	No add/edit/delete. No changing permissions.	Read only
Devices	All	All	Read only, admin can restore	Can reset device. No add/edit/ delete. No data delete.	Read only
Legal Hold	All	None	All	None	Read only
QuickCache	All	Can add/edit, no delete	None	Read only	Read only
Reporting	All	Can add/edit, no delete	Read only	Read only	Read only

Permissions at each role are dependent on the entity on which the role is applied. For example, a user with the Administrator role in a company cannot add another company since adding companies can only be done by a Partner.

Device Reset Process with Enterprise Key Controller

Customers may also choose to host their own company key, in which case the private company key will be retained on premise for customers and not stored in the vault. The enterprise key controller (EKC) is software provided by OpenText to the customer.

When installing the EKC software, a cryptographically random asymmetric RSA-4096 key will be generated on the EKC The private key will be stored locally on the EKC and the public key will be stored in the configured vault. The public key in the vault is used to encrypt the device keys; the private key on the EKC is used to decrypt the device keys. When the device key is created on the device, it is encrypted on the device by the public key and then escrowed to the vault. The device key does not leave the device in a decrypted state.

When utilizing the EKC, during the device reset process, the authenticated and authorized administrator will login to the Dashboard and select the device to be reset. The Dashboard will redirect the administrator to the web service running on the EKC, where a pop-up window will prompt the administrator to enter a passphrase. From the passphrase and a salt, an AES 256-bit encryption key is created using PBKDF2 hashing.

The enterprise key controller will retrieve the encrypted device key from the vault, decrypt the device key using its locally stored private key, re-encrypt using the passphrase key, and store the re-encrypted key in the vault. The original device key is maintained in the vault, so while the device is in a reset state both the original and passphrase-encrypted device keys are kept in the vault. The AES encryption key generated from the passphrase and the passphrase are not stored.

When utilizing the EKC, during reactivation, the device will retrieve the device key that is encrypted with the passphrase key from the vault over TLS 1.2. The same passphrase must be entered into the agent and used to decrypt the device key. The backup agent software uses the same process as the EKC to derive a key from the passphrase and then uses that key to decrypt the downloaded device key. Again, neither the passphrase nor the derived key is sent to the vault. Once decrypted and the client is activated, the device key will be stored within DPAPI on Windows and keychain on Mac to perform decryption and encryption functions. The passphrase-encrypted key is also deleted from the vault.

It is important to note that when using an EKC, since there is no company private key stored in the vault, certain functions are limited. The following restore functions are supported when using an EKC:

1. Restore via the client
2. Admin restore of files to the currently activated device
3. Reset and reactivate a device, followed by restore via client or admin restore (to support a transfer of files to a new device)

The following functions are NOT supported when using an EKC:

1. Web access
2. Cross-device admin restore
3. SSO into the dashboard cannot be used when an EKC is in use

For customers hosting an EKC, they are responsible for maintaining the key controller in a highly available way to support their reactivation needs. Additionally, the company private key stored on this device must be backed up and protected because if the private key is lost, there is no way for it to be retrieved from OpenText. Without this key, no device can be reset or reactivated. When using an EKC, we are not able to retrieve the private key.

Customer responsibilities when using an EKC:

1. Provide the server infrastructure to meet the system requirements.
2. Back up the company encryption key. If lost, no device reactivations can be performed and Carbonite cannot retrieve this key.
3. Maintain the EKC hardware and software including OS and application patches.
4. Ensure network access is properly configured – inbound access on port 443 is required from administrator endpoints so the passphrase can be entered. The EKC also requires outbound access on port 443 to the vault. The backup agents do not require access to the EKC.

opentext™ | Cybersecurity