GOOGLE WORKSPACE CLIENT-SIDE ENCRYPTION

Integration Guide

Applicable Devices:
KMES Series 3
Applicable Versions:
6.3.1.x

TABLE OF CONTENTS

[1] OVERVIEW OF THE GOOGLE WORKSPACE CSE / KMES SERIES 3 INTEGRATION 3
[1.1] About Google Workspace CSE 3
[1.2] Purpose of the Integration 3
[1.3] Basic Setup Steps for Google Workspace CSE 3
[1.4] Google Service-Level Requirements for CSE 4
[1.5] Client-side encryption process 5
[1.6] Personal Keys and Key Rotation on the KMES Series 3 5
[2] PREREQUISITES 7
[3] IDENTITY AND ACCESS MANAGEMENT (IAM) 8
[3.1] Connecting Google Workspace to an identity provider for client-side encryption 8
[3.2] Setup of IAM on the KMES Series 3 9
[3.3] Setup of IAM in Google Workspace 11
[4] EXTERNAL KEY SERVICE SETUP FOR GOOGLE WORKSPACE CSE 12
[4.1] Configurations in the KMES Series 3 application interface 12
[4.2] Configurations in the Google Admin Console 12
[5] VALIDATION & TESTING 15
[5.1] Validate successful connection from Google Workspace to the KMES Series 3 15
[5.2] Validate successful connection from Google Workspace to the configured Identity Provider (IdP) 15
[5.3] Test the creation of a blank encrypted Google Doc 15
[5.4] Test encrypting and uploading a file to Google Drive 16
[5.5] Viewing Personal Keys in the KMES Series 3 application interface 18
[5.6] Test sharing an encrypted Google Doc 18
APPENDIX A: XCEPTIONAL SUPPORT 20

[1] OVERVIEW OF THE GOOGLE WORKSPACE CSE / KMES SERIES 3 INTEGRATION

[1.1] ABOUT GOOGLE WORKSPACE CSE

From the Google Workspace Admin Help website: "You can use your own encryption keys to encrypt your organization's data, in addition to using the default encryption that Google Workspace provides. With Google Workspace Client-side encryption (CSE), content encryption is handled in the client's browser before any data is transmitted or stored in Drive's cloud-based storage. That way, Google servers can't access your encryption keys and, therefore, can't decrypt your data. To use CSE, you'll need to connect Google Workspace to an external encryption key service and an identity provider (IdP)."

[1.2] PURPOSE OF THE INTEGRATION

Google Workspace already uses the latest cryptographic standards to encrypt all data at rest and in transit between its facilities. With CSE, however, you have direct control of encryption keys and the identity provider used to access those keys to further strengthen the security of your data.

Your organization might need to use CSE for various reasons—for example:

- **Privacy**—Your organization works with extremely sensitive intellectual property.
- **Regulatory compliance**—Your organization operates in a highly regulated industry, like aerospace and defense, financial services, or government.

[1.3] BASIC SETUP STEPS FOR GOOGLE WORKSPACE CSE

Step 1: Set up your external encryption key service

First, you'll set up an encryption key service through one of Google's partner services (i.e., the Futurex KMES Series 3). This service controls the top-level encryption keys that protect your data.

Step 2: Connect Google Workspace to your external key service

Next, you'll specify the location of your external key service, so Google Workspace can connect CSE for supported apps to it.

Step 3: Connect Google Workspace to your identity provider

For this step, you'll need to connect to either a third-party IdP or Google identity, using either the Admin console or a .well-known file hosted on your server. Your IdP verifies the identity of users before allowing them to encrypt content or access encrypted content. Learn more

Note: In this integration guide we demonstrate using VirtuCrypt as the identity provider.

Step 4: Turn on CSE for users

You can turn on CSE for any organizational units or groups in your organization. Note, however, that you need to turn on CSE only for users that you want to create client-side encrypted content:

- **Google Drive**—You need to turn on CSE only for users who need to create client-side encrypted documents, spreadsheets, and presentations or upload client-side encrypted files to Drive. You don't need to turn on CSE for users who only view and edit files shared with them.
- **Google Meet**—You need to turn on CSE only for users who need to host client-side encrypted meetings. You don't need to turn on CSE for other participants in meetings.

For details about turning on CSE for users, see Create client-side encryption policies.

[1.4] GOOGLE SERVICE-LEVEL REQUIREMENTS FOR CSE

Administrator requirements

To set up Google Workspace Client-side encryption for your organization, you need to be a Super Admin for Google Workspace.

User requirements

- Users need a Google Workspace Enterprise Plus, Google Workspace for Education Plus, or Enterprise Essentials license to use CSE to:
  - Create or upload files
  - Host meetings
- Users can have any type of Google Workspace or Cloud Identity license to:
  - To view, edit, or download an existing file encrypted with CSE
  - Join a CSE meeting
- Users with a consumer Google Account (such as Gmail users) can't access CSE files or participate in CSE meetings.
- To view or edit encrypted files, users must use either the Google Chrome or Microsoft Edge browser.
- To join a CSE meeting, users must be invited or added during the meeting. Knocking isn't available for CSE meetings.
- Access to CSE files and meetings depends on your organization's CSE policies.

External user requirements

- During the beta, external users must have a Google Workspace license to access your content encrypted with CSE. Users with a consumer Google Account or a visitor account can't access files encrypted with CSE.
- External organizations must also set up CSE, either in the Admin console or with a .well-known file.
- Your external encryption service must allowlist the third-party IdP service that's used by the external domain or the individuals you want to use CSE. You can usually find the IdP service in their publicly available .well-known file, if they set up one. Otherwise, contact the external organization's Google Workspace admin for their IdP details.

[1.5] CLIENT-SIDE ENCRYPTION PROCESS

After an administrator enables CSE for their organization, users for whom CSE is enabled can choose to create encrypted documents using the Google Workspace collaborative content creation tools, like Docs and Sheets, or encrypt files they upload to Google Drive, such as PDFs.

After the user encrypts a document or file:

1. Google Workspace generates a DEK in the client browser to encrypt the content.
2. Google Workspace sends the DEK and authentication tokens to your third-party KACLS for encryption, using a URL you provide to the Google Workspace organization's administrator.
3. Your KACLS uses this API to encrypt the content, then sends the obfuscated, encrypted data back to Google Workspace.
4. Google Workspace stores the obfuscated, encrypted data in the cloud. Only users with CSE enabled and access to your KACLS are able to access the data.

For more details, see Encrypt and decrypt files.

[1.6] PERSONAL KEYS AND KEY ROTATION ON THE KMES SERIES 3

What are Personal Keys?

Personal Keys on the KMES Series 3 are used for encrypting data for Google CSE, and an individual key is generated for each user. The first time a user creates an encrypted document or encrypts and uploads a file to Google Drive, the KMES generates a new Personal Key Group and Personal Key for that user. Personal Keys created for CSE are AES-256 Data Encryption Keys. Personal Keys can be viewed and managed in the KMES application interface under *Key Management -> Personal Keys*.

Automatic key rotation

By default, newly-generated Personal Key Groups are assigned a **Regenerative** rotation policy with the **Validity Period** set to **1** month. At the time of writing, the default rotation policy cannot be modified, but this functionality will be added in a later release.

Note: Only one Personal Key can be active at a time for CSE users. After a key is rotated, it remains stored on the KMES and will be used for decrypting any documents that were encrypted using that key. Every document encrypted after a key is rotated will be encrypted using the new active key.

[2] PREREQUISITES

Supported Hardware:

- KMES Series 3, version 6.3.1.x and above, with the *Google CSE* license enabled

[3] IDENTITY AND ACCESS MANAGEMENT (IAM)

[3.1] CONNECTING GOOGLE WORKSPACE TO AN IDENTITY PROVIDER FOR CLIENT-SIDE ENCRYPTION

After you set up your external key service and connect it to Google Workspace, you need to connect Google Workspace to your **identity provider (IdP)**. Any IdP that supports **OAuth** can be utilized. Your external key service uses the IdP to authenticate users before they can encrypt files or access encrypted files.

[3.1.1] Choose your IdP for CSE

If you don't already use a third-party identity provider (IdP) with Google Workspace, you can set up your IdP for use with your key service in either of two ways:

- **Use a third-party IdP (recommended)**—Use a third-party IdP if your security model requires more isolation of your encrypted data from Google.
- **Use Google identity**—If your security model doesn't require additional isolation of your encrypted data from Google, you can use the default Google identity as your IdP.

[3.1.2] Choose how to connect to your IdP for CSE

You can set up your IdP—either a third party IdP or Google identity—using either a .well-known file that you host on your organization's website or the Admin console (which is your IdP fallback). There are several considerations for each method, as described in the table below.

Considerations	.well-known setup	Admin console setup (IdP fallback)
Isolation from Google	IdP settings are stored on your own server.	IdP settings are stored on Google servers.
Admin responsibilities	An IdP admin can manage your setup instead of a Google Workspace Super Admin.	Only a Google Workspace Super Admin can manage your IdP setup.
CSE availability	CSE availability (uptime) depends on availability of the server that hosts your .well-known file.	CSE availability corresponds to the general availability of Google Workspace services.
Ease of setup	Requires changing DNS settings for your server, outside of the Admin console.	Configure settings in the Admin console.
Sharing outside your organization	Your collaborator's external key service can easily access your IdP settings. This access can be automated and ensures your collaborator's service has immediate access to any changes to your IdP settings.	Your collaborator's external key service can't access your IdP settings in the Admin console. You must provide your IdP settings directly to your collaborator before you share encrypted files for the first time, as well as any time you change your IdP settings.

Please refer to the following Google Workspace knowledgebase article for further details on connecting Google Workspace to an identity provider (IdP):
https://support.google.com/a/answer/10743588?hl=en#zippy=%2Coption-to-connect-to-your-idp-using-a-well-known-file

[3.2] SETUP OF IAM ON THE KMES SERIES 3

Two different **Identity Providers** need to be created on the KMES Series 3. One will be configured with the **Authentication JSON Web Token (JWT)** issued by the identity partner (IdP) to attest a user's identity, and the other will be configured with the **Authorization JSON Web Token (JWT)** issued by Google to verify that the caller is authorized to encrypt or decrypt a resource. In addition to creating the identity providers, a new **Role** needs to be made for Google CSE, and **Identities** need to be created for all users in your organization that will use Google CSE.

[3.2.1] Create the Authentication JWT Identity Provider

A JWT Identity Provider must be created to allow the identity partner (IdP) to attest a user's identity. In this example, VirtuCrypt is serving as the IdP.

1. Navigate to *Identity Management -> Identity Providers*, then right-click and select **Add** -> **Provider** -> **JSON Web Token**. This will open the *Identity Provider Editor* dialog.
2. In the *Info* tab, specify a name for the Identity Provider and de-select **Enforce Dual Factor**.
3. In the *JWT Options* tab, you can specify an issuer and set leeway and max validity values according to your requirements. The issuer field is optional, but if you are using VirtuCrypt as the IdP (as we are in this example), this field should be set to **vip**.
4. In the *JWT Key* tab, select the **JWKS** radio button (JWKS stands for JSON Web Key Set). Two new fields will populate in the dialog: **JWKS URL** and **TLS PKI**. The JWKS URL is a read-only endpoint URL that points to a list of public keys used to verify JSON Web Tokens (JWT). Configuring a CA certificate in the TLS PKI field is not required if the domain configured in the JWKS URL field can be verified using trusted public internet CAs. However, if you have a JWK setup on your LAN, you must select the custom CA certificate used to sign the domain specified in the JWKS URL field. For the VirtuCrypt use case, the TLS PKI field can be left blank because vip.virtucrypt.com has a certificate issued by a trusted public internet CA. If configuring a custom CA certificate is required for your use case, you must download and then copy that certificate to the storage medium configured on the KMES and import the certificate into a Certificate Container in the *PKI -> Certificate Authorities* menu. After that is completed, you will be able to browse and select the certificate in the TLS PKI field.
5. Click **OK** to save.
6. Right-click on the Identity Provider that was just created and select **Add** -> **Mechanism** - > **JSON Web Token**. This will open the *Authentication Mechanism Editor* dialog.
7. In the *Info* tab, specify a name for the authentication mechanism.
8. Leave the default settings in the *Identifiers* and *Claims* tabs, then click **OK** to save.

[3.2.2] Create the Authorization JWT Identity Provider

A JWT Identity Provider must be created to allow Google to verify that the caller is authorized to encrypt or decrypt a resource.

1. Navigate to *Identity Management* -> *Identity Providers*, then right-click and select **Add** -> **Provider** -> **JSON Web Token**. This will open the *Identity Provider Editor* dialog.
2. In the *Info* tab, specify a name for the Identity Provider and de-select **Enforce Dual Factor**.
3. In the *JWT Options* tab, you can specify an issuer and set leeway and max validity values according to your requirements. The issuer field is optional, but an appropriate value would be "gsuitecse-tokenissuer-drive@system.gserviceaccount.com".
4. In the *JWT Key* tab, select **JWKS** and then specify "https://www.googleapis.com/service_accounts/v1/jwk/gsuitecse-tokenissuer-drive@system.gserviceaccount.com" in the JWKS URL field. The TLS PKI field should be left blank because the www.googleapis.com domain can be verified using trusted public internet CAs; therefore, it is unnecessary to configure a custom CA certificate.
5. Click **OK** to save.
6. Right-click on the Identity Provider that was just created and select **Add** -> **Mechanism** - > **JSON Web Token**. This will open the *Authentication Mechanism Editor* dialog.
7. In the *Info* tab, specify a name for the authentication mechanism.
8. Leave the default settings in the *Identifiers* and *Claims* tabs, then click **OK** to save.

[3.2.3] Create Role definition for CSE

1. Navigate to the *Identity Management* -> *Roles* menu and click the **Add...** button. This will pull up the *Role Editor* dialog.
2. Specify a **Name** for the role, ensure that the **Role class** is set to **Principal**, and set **Logins Required** to **1**.

   Note: Principal roles are granted view permissions on any objects created using that principal role. This makes sharing encrypted documents possible within an organization because all CSE users are assigned the same principal role. To demonstrate the point further, suppose one CSE user in your organization shares a document with another CSE user. The document can then be decrypted in the second CSE user's browser using the first user's Personal Key since that Personal Key was created using the shared CSE principal role. However, all encrypted documents that the second user creates will be encrypted using their own Personal Key.
3. In the Permissions tab, select the following permissions:
   - **Cryptographic Operations** -> **Unwrap|Wrap**
   - **Keys** (only the top-level **Keys** permission)
4. In the *Advanced* tab, you can leave the default settings.
5. Click the **OK** button to finish creating the role.

[3.2.4] Create an Identity for the CSE user

1. Navigate to the *Identity Management -> Identities* menu, then right-click and select **Add** -> **User**.
2. Leave **Storage** set to **Application** and in the **Name** field, enter the CSE user's email address that they use to log in to Google Workspace.
3. In the *Assigned Roles* tab, select the **Role** that you just created.
4. In the *Device Info* tab, you can leave the default settings.
5. In the *Authentication* tab, two credentials need to be added. One for the **Authentication JWT Identity Provider** created in section 3.2.1, and one for the **Authorization JWT Identity Provider** created in section 3.2.2.

   Click the **Add** button to add the new credentials. After configuring the authentication and Authorization **JWT** credentials, remove the default **Password** credential.
6. Click **OK** to finish creating the identity.

[3.3] SETUP OF IAM IN GOOGLE WORKSPACE

You need to turn on Google Workspace Client-side encryption (CSE) for all users who need to do any of the following:

- Create or upload encrypted files to Google Drive
- Host encrypted meetings with Google Meet (beta)

Note: You don't need to turn on CSE for users who only need to view or edit encrypted files or attend meetings. However, external users need to use an identity provider (IdP) allowlisted by your domain. For details, see "External user requirements" in About client-side encryption.

To turn on CSE for users, you need to turn on CSE for the organizational units or configuration groups the users belong to.

At any time, you can disable CSE for users by turning CSE off for the organizational units or configuration groups they belong to. If you disable CSE for users, any existing client-side encrypted content remains encrypted and accessible.

Please refer to this Google Workspace knowledge base article for instructions on how to perform the following steps for setting up IAM for CSE in Google Workspace:

1. Set the default key service for your organization
2. Turn CSE on or off for users

[4] EXTERNAL KEY SERVICE SETUP FOR GOOGLE WORKSPACE CSE

This section will describe various steps required to configure the KMES Series 3 as an external key service for Google Workspace CSE. Some of the configurations will be completed in the KMES Series 3 application interface, and some will be completed in the Google Admin Console.

[4.1] CONFIGURATIONS IN THE KMES SERIES 3 APPLICATION INTERFACE

[4.1.1] Define KACL URL for Google Client-side encryption

1. Log in to the KMES Series 3 application interface with the default Admin identities.
2. Navigate to the *Administration -> Configuration* menu and select **Google API options**.
3. In the **KACL URL** field, enter the URL for your key service (i.e., https://<server ip>:<port>/kmes/v7/key-encrypt/client).

   Note: Google requires this connection to be TLS, with a publicly-trusted certificate. The connection can be through NAT or reverse proxy.
4. Click **Save**.

[4.1.2] Enable the required Host API commands

1. Navigate to the *Administration -> Configuration* menu and select **Host API Options**.
2. Select the **KACL** command, which enables Google client side key wrap and unwrap.
3. Click **Save**.

[4.2] CONFIGURATIONS IN THE GOOGLE ADMIN CONSOLE

[4.2.1] Configure KACLS and IdP for Client-side encryption

Before outlining the configuration steps, a couple of terms should be defined. **KACLS** stands for **Key Access Control List Service**, and this is your external key service (i.e., KMES Series 3) that uses this API to control access to encryption keys stored in an external system. IdP's were discussed extensively in the previous section, but to reiterate, **IdP** stands for **Identity Provider**, and it is the service that authenticates users before they can encrypt files or access encrypted files. This integration uses VirtuCrypt as the IdP for demonstration purposes, but any IdP that supports OAuth can be used.

KACLS Configuration

1. Sign in to your Google admin console.

   Note: Sign in using an account with super administrator privileges.

2. In the main menu, select *Security -> Access and data control -> Client-side encryption*.
3. Click the **External key service** card to open it.
4. Click **Add external key service**.
5. Enter a name for your key service.
6. Enter the URL for your key service (i.e., https://<server ip>:<port>/kmes/v7/key-encrypt/client).

   Note: Google requires this connection to be TLS, with a publicly-trusted certificate. The connection can be through NAT or reverse proxy.
7. To confirm that Google Workspace can communicate with the external key service, click **Test connection**.
8. To close the card, click **Continue**.

IdP Configuration

To connect Google Workspace to your identity provider (IdP), you can use a .well-known file or the Admin console. After establishing the connection, you need to allowlist your IdP in the Admin console.

This section will walk through connecting Google Workspace to your IdP using the Admin console. However, this method is meant to serve as a fallback method for the .well-known file method. Please refer to the following Google Workspace documentation instructions on connecting Google Workspace to your IdP using a .well-known file: https://support.google.com/a/answer/10743588#config_wellknown&zippy=%2Coption-to-connect-to-your-idp-using-a-well-known-file

1. Sign in to your Google admin console.

   Note: Sign in using an account with super administrator privileges.
2. In the main menu, select *Security -> Access and data control -> Client-side encryption*.
3. Under **Identity provider configuration**, click **Configure IdP fallback**.
4. Enter the details for your IdP.
   a. In the **Name** field, specify a descriptive name to help identify your IdP. It will be shown in IdP messages for users.
   b. In the **Client ID** field, you need to specify the OpenID Connect (OIDC) client ID that the CSE client application uses to acquire a JSON Web Token (JWT).

      If you're using a third-party IdP: You generate this ID using your IdP's admin console.

      If you're using Google identity: You generate this ID using the Google Cloud Platform (GCP) Admin console. For details, go to "Create a client ID for Google identity".
   c. In the **Discovery URI** field, specify the OIDC discovery URL, as defined in this OpenID specification.

      If you're using a third-party IdP: Your IdP provides you with this URL, which usually ends with /.well-known/openid-configuration.

If you're using Google identity: Use https://accounts.google.com/.well-known/openid-configuration

Note: Configure your discovery URI to allow origin URLs for Cross-Origin Resource Sharing (CORS) calls, as follows:

- Methods: GET
- Allowed origins:
  - https://admin.google.com
  - https://client-side-encryption.google.com
  - https://krahsc.google.com/callback
  - https://krahsc.google.com/oidc/cse/callback
  - https://krahsc.google.com/oidc/drive/callback
  - https://krahsc.google.com/oidc/gmail/callback
  - https://krahsc.google.com/oidc/meet/callback
  - https://krahsc.google.com/oidc/calendar/callback
  - https://krahsc.google.com/oidc/docs/callback
  - https://krahsc.google.com/oidc/sheets/callback
  - https://krahsc.google.com/oidc/slides/callback
  - https://client-side-encryption.google.com/callback
  - https://client-side-encryption.google.com/oidc/cse/callback
  - https://client-side-encryption.google.com/oidc/drive/callback
  - https://client-side-encryption.google.com/oidc/gmail/callback
  - https://client-side-encryption.google.com/oidc/meet/callback
  - https://client-side-encryption.google.com/oidc/calendar/callback
  - https://client-side-encryption.google.com/oidc/docs/callback
  - https://client-side-encryption.google.com/oidc/sheets/callback
  - https://client-side-encryption.google.com/oidc/slides/callback

d. In the **Grant type** field, select the OAuth flow you want to use for OIDC.

   If you're using a third-party IdP: You can use either the **Implicit** or **Authorization code with PKCE** grant type.

   If you're using Google identity: You can use only the **Implicit** grant type.

e. Click **Test connection**.

   If Google Workspace can connect to your IdP, the "Connection success" message appears.

f. Click **Add provider** to close the card.

[5] VALIDATION & TESTING

In this section, we will do the following:

1. Validate that Google Workspace can successfully connect to the external key service (i.e., the KMES Series 3)
2. Validate that Google Workspace can successfully connect to the configured Identity Provider (IdP)
3. Test the creation of a blank encrypted Google Doc
4. Test encrypting and uploading a file to Google Drive
5. View Personal Keys in the KMES Series 3 application interface
6. Test sharing an encrypted Google Doc

[5.1] VALIDATE SUCCESSFUL CONNECTION FROM GOOGLE WORKSPACE TO THE KMES SERIES 3

1. Sign in to your Google admin console.

   Note: Sign in using an account with super administrator privileges.
2. In the main menu, select *Security -> Access and data control -> Client-side encryption*.
3. Click **Test connection**.

   If Google Workspace can connect to the KMES Series 3, a green checkmark and the "Your external key service is active" message appears.

[5.2] VALIDATE SUCCESSFUL CONNECTION FROM GOOGLE WORKSPACE TO THE CONFIGURED IDENTITY PROVIDER (IDP)

1. Sign in to your Google admin console.

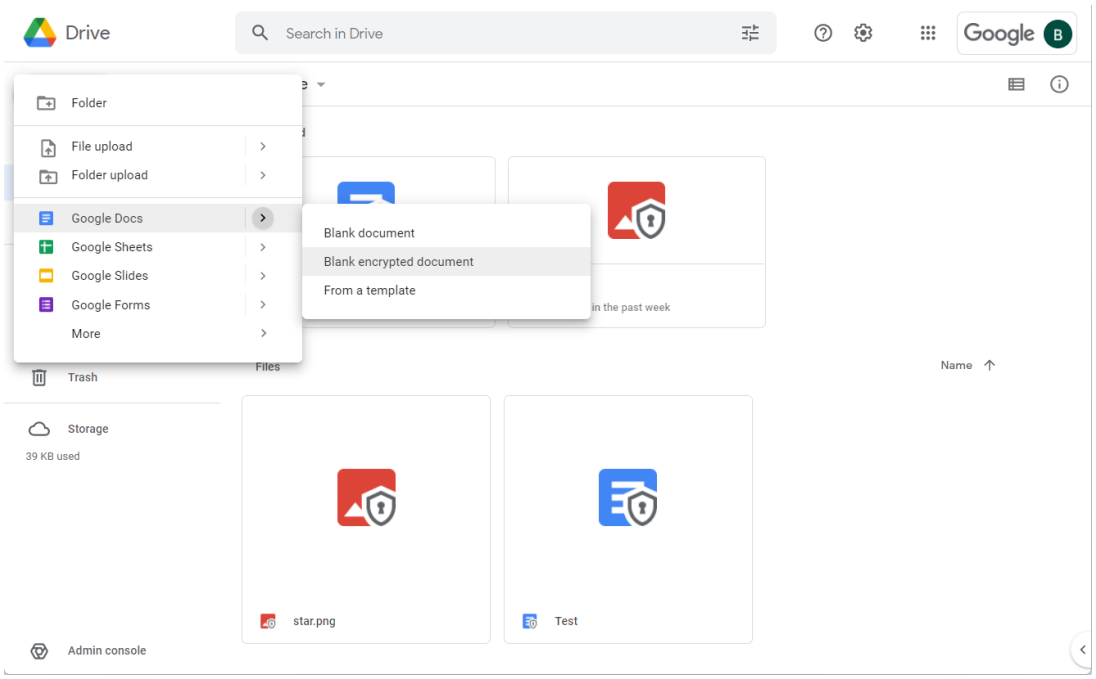
   Note: Sign in using an account with super administrator privileges.
2. In the main menu, select *Security -> Access and data control -> Client-side encryption*.
3. Click the **Identity provider configuration** card to open it.
4. Click **Test connection**.

   If Google Workspace can connect to your IdP, the "Connection success" message appears.

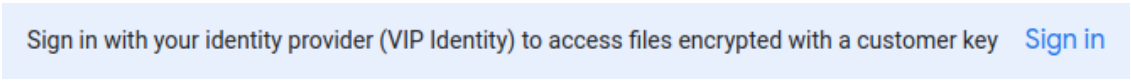
[5.3] TEST THE CREATION OF A BLANK ENCRYPTED GOOGLE DOC

1. Sign in to Google Drive with your CSE user.

2. Click the **New** button, then select **Google Docs** -> **Blank encrypted document**.



3. A message will appear warning you that intelligent features such as spelling and grammar won't work with encrypted files, collaboration features will be limited, and only certain people can access encrypted files due to admin settings. Click **Create**.
4. If this is the first encryption operation you have attempted with Google Workspace CSE, the following message will appear at the top of the page prompting you to sign in with your identity provider.

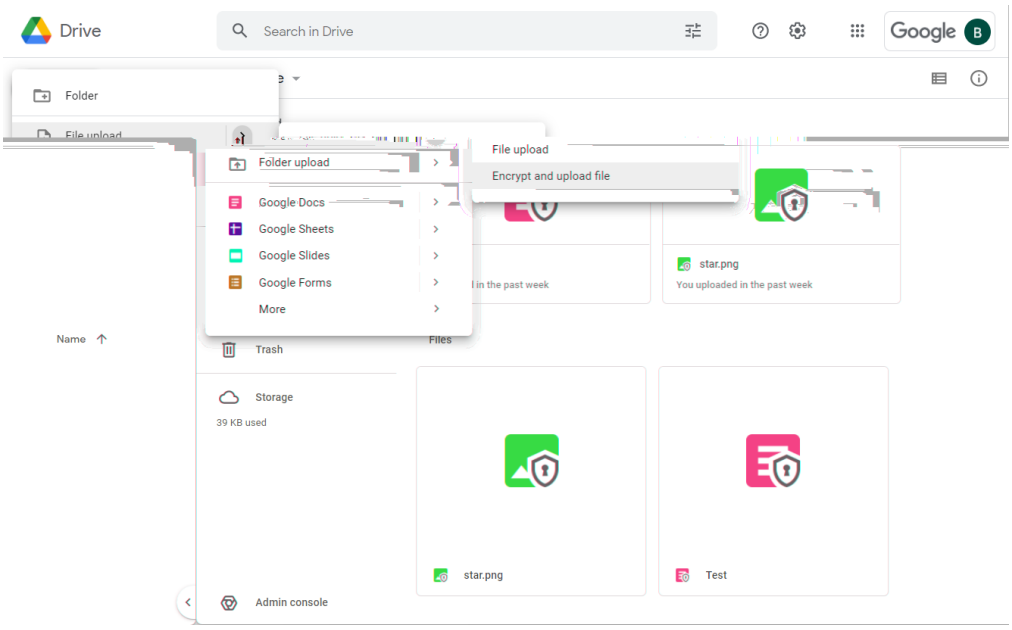


Click **Sign In**, which will redirect you to your IdP's website to sign in. After signing in and allowing your IdP access to your Google Account, you will be redirected back to the Google Doc, which should now be encrypted. A confirmation message will appear if encryption is successful. Then you can edit and save the document per the normal process.

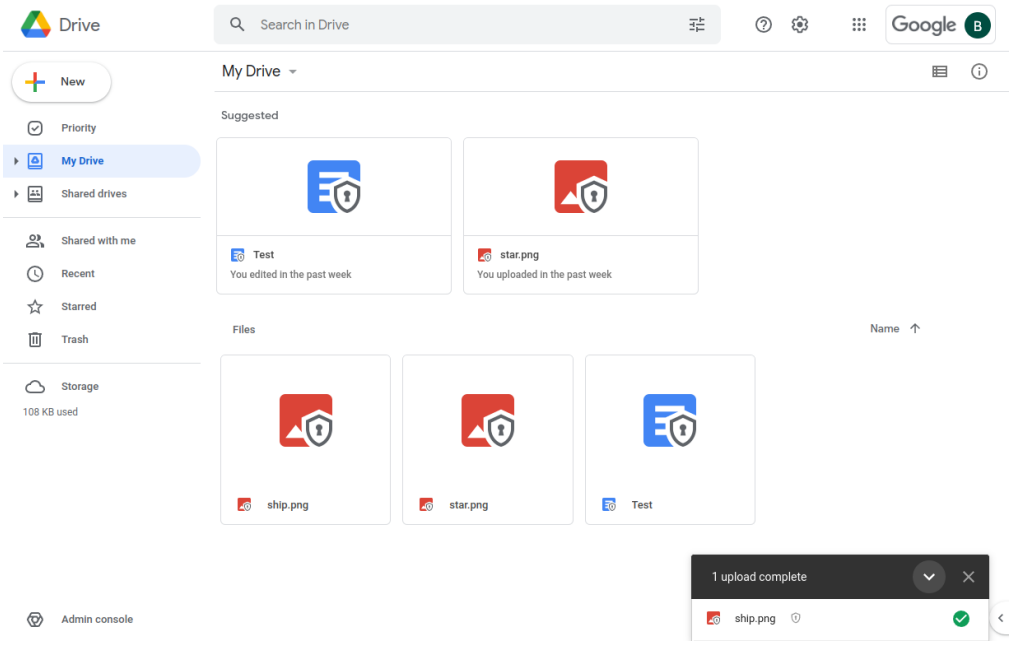
[5.4] TEST ENCRYPTING AND UPLOADING A FILE TO GOOGLE DRIVE

1. Sign in to Google Drive with your CSE user.

2. Click the **New** button, then select **File upload** -> **Encrypt and upload file**.



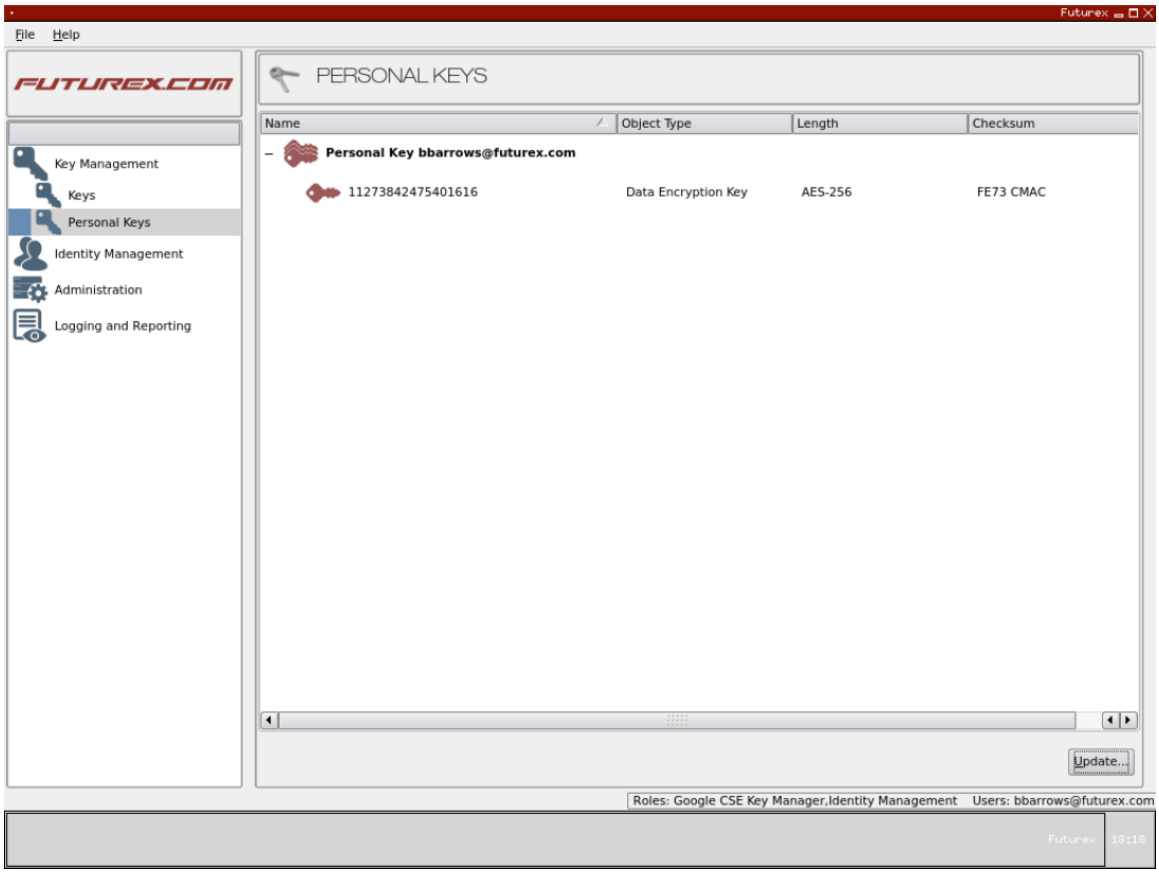
3. A message will appear warning you that some features, such as full-text search and file preview, will be unavailable and that only certain people can access encrypted files due to admin settings. Click **Select file**.
4. If this is the first encryption operation you have attempted with Google Workspace CSE, you will be prompted to sign with your identity provider. If this is the case, click **Sign In**, which will redirect you to your IdP's website to sign in. After signing in and allowing your IdP access to your Google Account, you will be redirected back to Google Drive, and the encrypted file upload will commence. Uploads are displayed in the bottom-right corner of the page, and once the upload completes, you will see a green checkmark and an updated status message similar to the image below:



[5.5] VIEWING PERSONAL KEYS IN THE KMES SERIES 3 APPLICATION INTERFACE

As mentioned in the Overview section at the beginning of this guide, the first time that a Google CSE user creates an encrypted document or encrypts and uploads a file to Google Drive, a new **Personal Key Group** and **Personal Key** are generated on the KMES for that user. That Personal Key is then used for all CSE operations performed by that user in Google Workspace until an automatic key rotation occurs and a new Personal Key becomes active.

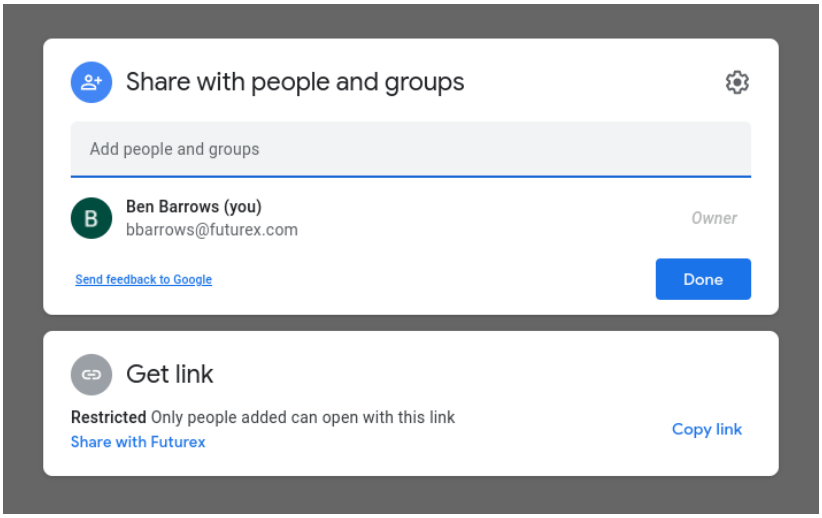
CSE users can view their Personal Keys by logging in to the application interface and navigating to the *Key Management -> Personal Keys* menu. An example is shown below:



In addition to individual CSE users being able to view their own Personal Keys, users with the **Personal Keys Managed** permission can manage the Personal Keys of all CSE users on the KMES.

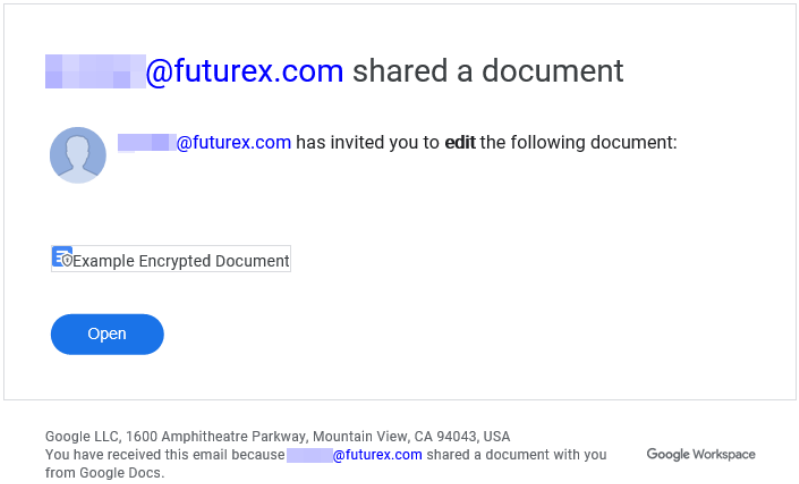
[5.6] TEST SHARING AN ENCRYPTED GOOGLE DOC

1. Sign in to Google Drive with your CSE user.
2. Right-click the encrypted document you would like to share and select **Share**, or, if you have the document open, you can click the **Share** button in the upper-right corner of the page.
3. In the following dialog, add people and groups you would like to share the encrypted document with and then click **Done**.

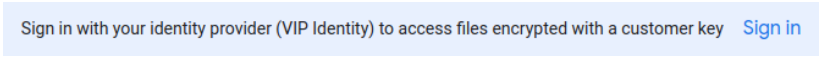


Note: Only share encrypted documents with other Google CSE users that your company administrator has set up with an account on the KMES Series 3. If they do not have a user configured on the KMES, they will not be able to decrypt, view, and edit the file you are sharing.

4. Users you shared the encrypted file with will receive an email that looks similar to the image below:



5. After the user clicks **Open** in the email they received, their browser will be redirected to sign in to Google. After signing in to Google (using the same email configured for their user on the KMES), they will be redirected to the shared Google Doc.
6. After a few seconds, the following message will appear at the top of the page. Click **Sign in**.



The user will be redirected to the configured Identity Provider (IdP) to sign in. After signing in and allowing the IdP access to the Google Account, the user will be redirected back to the Google Doc, which should now be encrypted. A confirmation message will appear if encryption is successful. Then the document can be edited and saved per the normal process.

APPENDIX A: XCEPTIONAL SUPPORT

In today's high-paced environment, we know you are looking for timely and effective resolutions for your mission-critical needs. That is why our Xceptional Support Team does whatever it takes to ensure you have the best experience and support possible. Every time. Guaranteed.

- 24x7x365 mission critical support
- Level 1 to level 3 support
- Extremely knowledgeable subject matter experts

At Futurex, we strive to supply you with the latest data encryption innovations as well as our best-in-class support services. Our Xceptional Support Team goes above and beyond to meet your needs and provide you with exclusive services that you cannot find anywhere else in the industry.

- Technical Services
- Onsite Training
- Virtual Training
- Customized Consulting
- Customized Software Solutions
- Secure Key Generation, Printing, and Mailing
- Remote Key Injection
- Certificate Authority Services

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

ENGINEERING CAMPUS

864 Old Boerne Road

Bulverde, Texas, USA 78163

Phone: +1 830-980-9782

+1 830-438-8782

E-mail: info@futurex.com

XCEPTIONAL SUPPORT

24x7x365

Toll-Free: 1-800-251-5112

E-mail: support@futurex.com

SOLUTIONS ARCHITECT

E-mail: solutions@futurex.com