Google Workspace

How Google Workspace uses encryption to protect your data

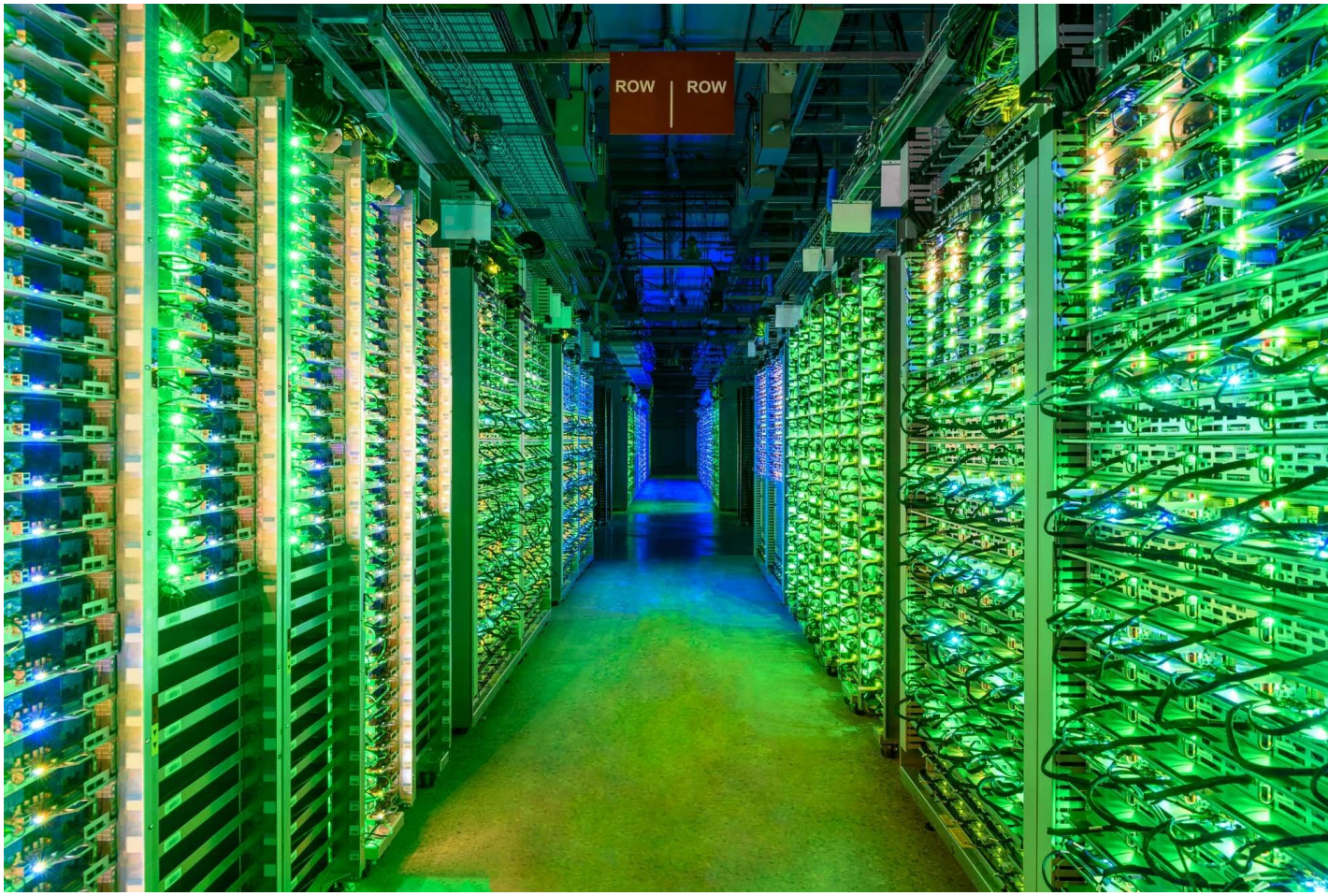


Table of contents

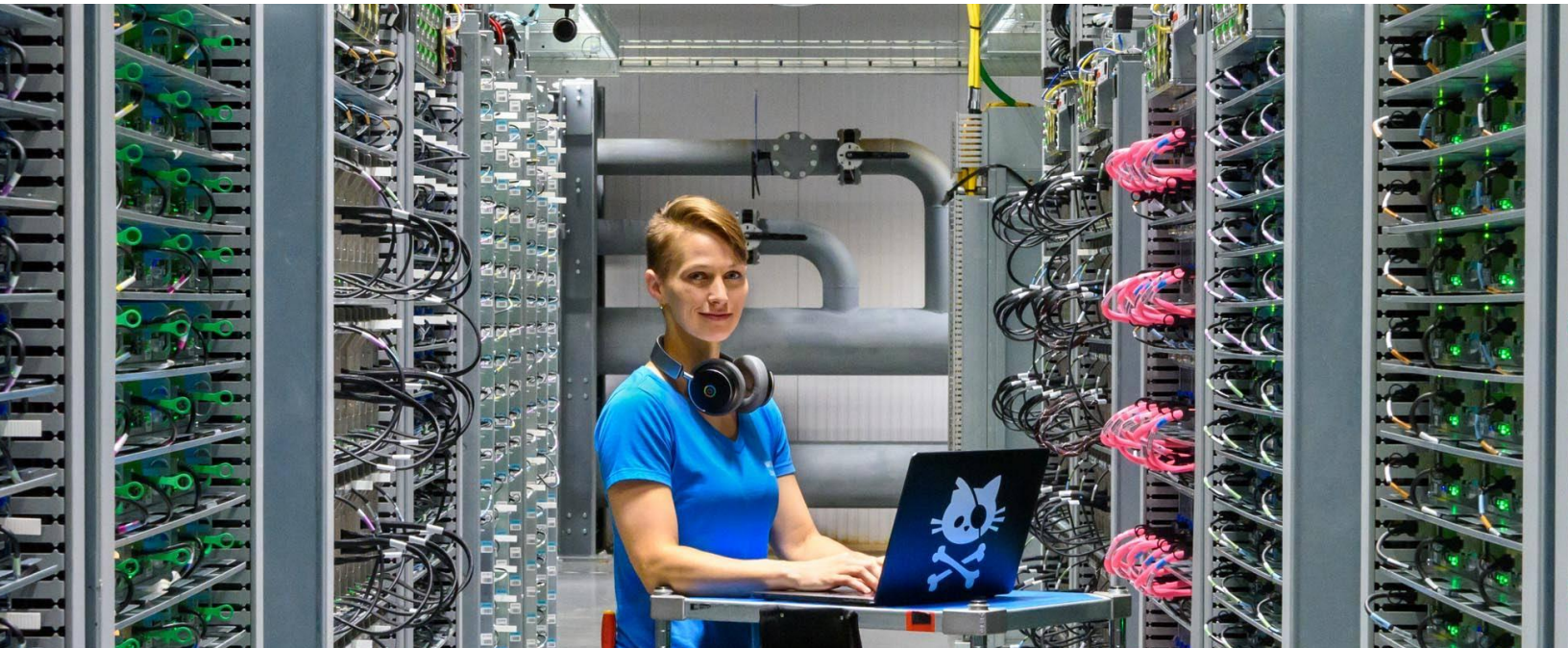
Introduction

Here at Google, we know that security is a key consideration for organizations that choose Google Workspace. This is why we work so hard to protect your data — whether it's traveling over the Internet, moving between our data centers or stored on our servers.

A central part of our comprehensive security strategy is encryption, which helps prevent information from being accessed in the event that it falls into the wrong hands. This paper will describe Google's approach to encryption and how it keeps your sensitive information safe.

Disclaimer

This whitepaper applies to Google Workspace products described at workspace.google.com. We are bringing Google Workspace to our education and nonprofit customers in the coming months. Education customers can continue to access our tools via G Suite for Education, which includes Classroom, Assignments, Gmail, Calendar, Drive, Docs, Sheets, Slides, and Meet. G Suite for Nonprofits will continue to be available to eligible organizations through the Google for Nonprofits program. This whitepaper applies to both consumer and enterprise data. The content contained herein is correct as of June 2021 and represents the status quo as of the time it was written. Google's security policies and systems may change going forward, as we continually improve protection for our customers.



How Google approaches encryption

Encryption is a way of scrambling data into unreadable content known as ciphertext that can only be descrambled by parties who possess a secret key. Attackers who want to circumvent encryption will typically try to steal the keys or exploit flaws in the encryption algorithms and their implementation. Encryption strength depends on a number of factors, such as the algorithm used, its implementation and the key size for that algorithm. It also depends on how keys are created, managed and secured. As computers get better and faster, it becomes easier to perform the complicated mathematical computations needed to break encryption. Even the mathematics behind this process — known as cryptanalysis — can improve over time, making it easier to break encryption. As a result, encryption algorithms that seemed strong a few years ago may no longer be as strong today.

To keep pace with this evolution, Google has a team of world-class security engineers tasked with following, developing and improving encryption technology. Our engineers take part in standardization processes and in maintaining widely used encryption software such as BoringCrypto — an SSL library in Chrome/Chromium, Android and a number of other apps/programs. We regularly publish our research in the field of encryption so that everyone in the industry — including the general public — can benefit from our knowledge.

Google has a team of world-class security engineers tasked with following, developing and improving encryption technology.

Encryption is an important piece of the Google Workspace security strategy, helping to protect your emails, chats, video meetings, files, and other data. First, we encrypt certain data as described below while it is stored “at rest” — stored on a disk (including solid-state drives) or backup media. Even if an attacker or someone with physical access obtains the storage equipment containing your data, they won’t be able to read it because they don’t have the necessary encryption keys. Second, we encrypt all data while it is “in transit” — traveling over the Internet and across the Google network between data centers. Should an attacker intercept such transmissions, they will only be able to capture encrypted data. We’ll take a detailed look at how we encrypt data stored at rest and data in transit below.

Encryption of data stored at rest

In Google's data centers, data belonging to Google Workspace customers is stored at rest in two types of systems: disks and backup media. Disks are used to write new data as well as store and retrieve data in multiple replicated copies. Google also stores data on offline backup media to help ensure recovery from any catastrophic error or natural disaster at one of our data centers. Data stored at rest is encrypted on both disks and backup media, but for each system we use a distinct approach for encryption to mitigate the corresponding security risks. These encryption mechanisms are detailed below.

Data on disks

Google encrypts customers' data stored at rest for the solutions in the Google Workspace product family (see Table 1). This encryption happens without the customer having to take any action. Core content is data created by the user, such as messages and attachments in Gmail.

To understand how this encryption works, it's important to understand how Google stores customer data. Data is broken into "chunks," which are stored on local disks and identified by unique chunk IDs.

Google encrypts data as it is written to disk with a per-chunk encryption key that is associated with a specific Access Control List (ACL). The ACL ensures that data in each chunk can only be decrypted by authorized Google services and employees.

This means that different chunks are encrypted with different encryption keys, even if they belong to the same customer. These chunks are encrypted using the Advanced Encryption Standard (AES) cipher with a 128-bit or stronger key.

Table 1 details what type of data is encrypted by each Google Workspace solution.

Google encrypts data as it is written to disk with a per-chunk encryption key that is associated with a specific Access Control List.

Table 1

Google Workspace: Encryption of data stored at rest

Solution	Core content data that is encrypted
Calendar	Events and descriptions of events.
Chat	Chat conversations that happen between individuals or in groups, while chat history is on, including images, videos, links, and uploaded files. This excludes all conversations when history is off.
Cloud Search	All search indices and indexable content from third party repositories.
Contacts	Content of end users' address books.
Currents	Posts, comments, photos.
Drive	Files uploaded to Drive via Drive File Stream, Backup and Sync, via the Drive web interfaces, Drive Mobile apps, Gmail, Google Drive API, and third-party services uploading files via the Drive API. File, folder, shared drive, and workspace metadata such as title, description, creator, and owner. Search indexes, Drive activity and audit logs, Drive metadata categories and values, Drive workspaces, and Drive approvals.
Docs, Sheets, Forms, and Slides	Content authored by the owner or collaborators as well as Forms responses (except, in some cases, content embedded into the file that is hosted on other Google products not referenced in this list; e.g. YouTube).
Gmail	Messages and attachments.
Groups	Group message archives.
Jamboard	Jam contents and embedded items (images, Drive files).
Keep	Note contents and attachments (images, drawings, recordings).
Meet	Recordings stored in Drive, text captions for recordings stored in Drive. See the Meet security help center for more information.
Sites	Content authored by the owners or collaborators of the site; except (i) content embedded into the site that is hosted on other Google products not referenced in this list (e.g., YouTube) (ii) content embedded into the site that remains hosted on other third-party websites, via Sites, Gadgets or image hotlinking.
Vault	Content created by Vault Admins, saved queries, and audit logs. Vault's exports of Gmail messages and attachments, Hangouts Classic conversations, Hangouts Chat, and Drive files.
Voice	Call history, messages and attachments, voicemails and transcriptions.

Key management and decryption process

Managing keys safely and reliably, while allowing access to the keys only to authorized services and individuals, is central to encrypted data security. Google has built a robust proprietary service for the distribution, generation, rotation and management of cryptographic keys using industry standard cryptographic algorithms that are in alignment with strong industry practices. In the following sections, we'll outline our approach to managing the encryption keys used to protect Google Workspace customers' information.

Google's key management service

As described in the previous section, files or data structures with customer-created content written by Google Workspace are subdivided into chunks, each of which is encrypted with its own chunk data encryption key ("chunk key"). Each chunk key is encrypted by another key known as the wrapping key, which is managed by a Google-wide key management service (KMS). The result is a "wrapped" (encrypted) chunk key, which is stored alongside the encrypted data. The wrapping keys, needed to decrypt wrapped chunk keys, and therefore to decrypt the chunk, are known only to the KMS and are never stored at rest in unencrypted form.

Customer-created content written by Google Workspace is subdivided into chunks, each of which is encrypted with its own chunk data encryption key.

Decryption and encryption operations on chunk keys are performed within the KMS. The wrapped chunk key is sent by a storage system to the KMS as a request to be unwrapped (decrypted) in order to access the encrypted data. The KMS authenticates the requesting system and checks the request against both system-level and per-wrapped-key ACLs.

If this request is authorized, the chunk key is decrypted in the KMS and returned to the storage system, which can now use that chunk key to decrypt that specific chunk of data. These chunk keys are encrypted in transit, as described below. This process is repeated until all the chunks that compose a specific file or data structure are decrypted, making the data available to the requesting application.

Data cannot be decrypted without both the wrapping key and the wrapped chunk key. Decrypting data therefore requires the cooperation of the storage system (which holds the encrypted data and wrapped chunk key) and the KMS (which holds the wrapping key). The KMS wrapping keys that encrypt the chunk keys are 128-bit or stronger AES keys.

All encryption and decryption operations by the KMS are controlled by ACLs which are checked against the cryptographic identity of the caller as provided by Google production identity management. Access is restricted to specific applications that require access and a limited number of individuals. Individuals are only provided access after demonstrating a recorded need and access requests to the KMS by employees are logged for auditing.

Rotating keys to limit risk

Google has built a proprietary system to manage key rotation. Keys are rotated or replaced regularly, so that if a key were compromised it wouldn't remain useful for decrypting new data indefinitely. Keys are typically rotated at least every 90 days. This process reduces data exposure in the event of a key compromise or cryptanalytic attack by limiting the time window in which any given key is used.

The key management server

The KMS, like other Google production services, runs on custom, purpose-built servers that we design and manufacture ourselves. These servers run a custom-designed operating system based on a stripped-down and hardened version of Linux, and are designed for the sole purpose of providing Google services.

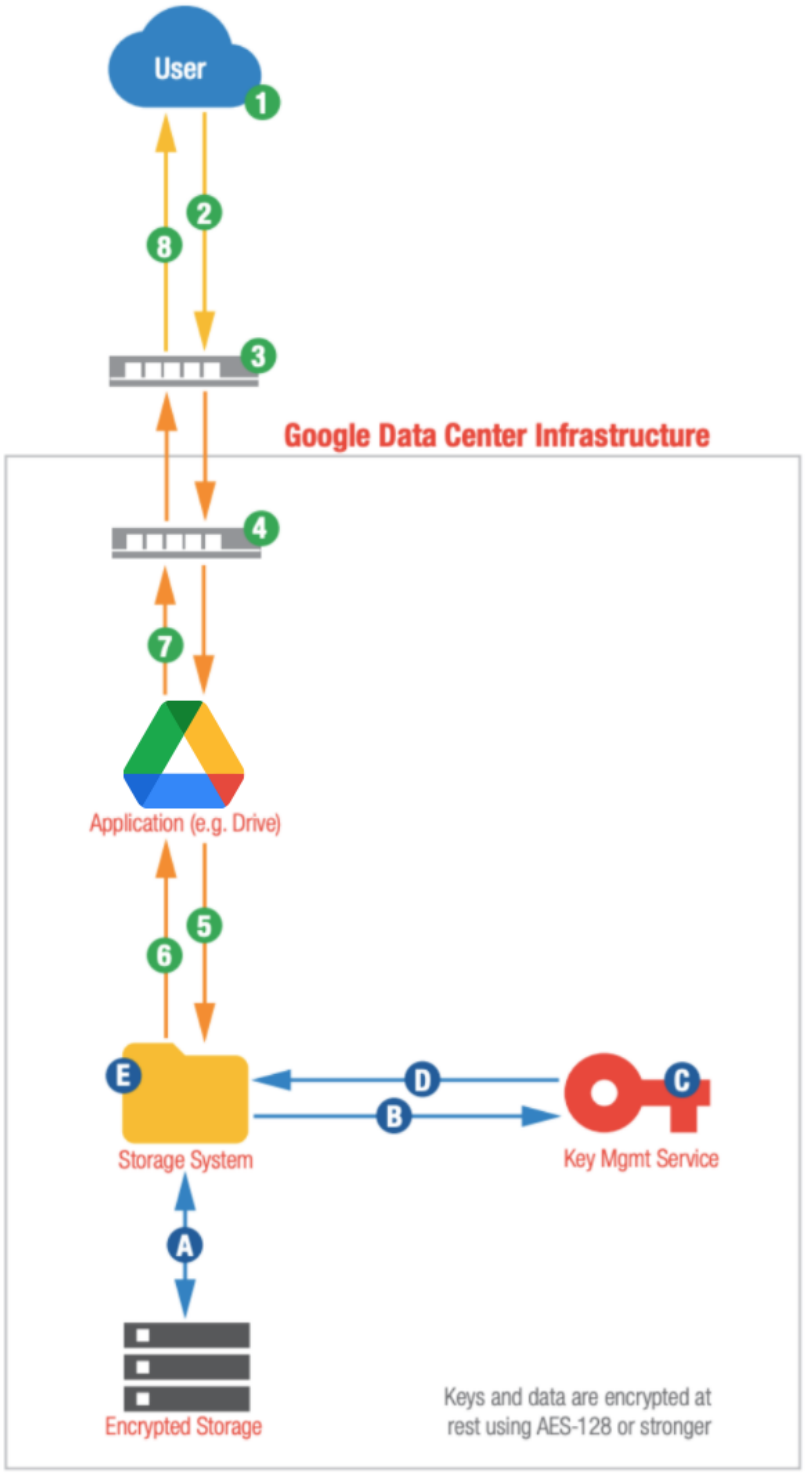
Google servers use a homogeneous environment that is maintained by proprietary software that continually monitors systems for binary modifications to ensure that only approved Google software is installed and running on Google servers. The KMS server has the same proprietary software installed on it monitoring for any unapproved modification. If a modification is found on the KMS server that differs from a standard Google image, the server is automatically returned to its official standard image state. These automated, self-healing mechanisms are designed to enable Google to monitor and remediate destabilizing events, receive notifications about incidents, and slow down potential compromise on the network or the KMS server.

Google has built a system to manage key rotation. Chunk encryption keys and wrapping keys are rotated or replaced regularly.

The following diagram shows the flow for Google Drive, as an example of the related encryption mechanisms. The process begins with the user requesting access to some of their Google Drive data (step 1). The connection between the user and Google is encrypted (step 2). Google routes this request internally (steps 3, 4, 5). When the storage system needs access to an encrypted chunk, Google begins the decryption process (steps A–E) to decrypt the data the user has requested and make it available to Google servers only in memory (i.e., not stored at rest in plaintext). Finally, it returns the data to the user (steps 6, 7, 8), again in an encrypted session. The data flow diagrams are similar for other Google Workspace products as well as for encrypting data when users create data.

Encryption at Rest flow

An example of encryption in Google Drive



USER DATA FLOW

1. **Initiate Request**
   User authenticates to Google Workspace and requests Drive data.
2. **Encrypted Tunnel**
   TLS-based encryption dependent on user's browser capabilities.
3. **Google Front End**
   Directs traffic to AFEs.
4. **Application Front End (AFE)**
   Directs traffic to Application servers.
5. **Requests User Data**
   User's Drive data request goes from the Application to storage.
6. **Return Decrypted Data**
   Send user data to Application.
7. **Return User Data**
   Return user data to user.
8. **Return User Data in Encrypted Tunnel**
   Return user data to user.

DATA DECRYPTION

- **A Retrieve Data**
  Gets Encrypted Chunk and Wrapped Key.
- **B Request Key Unwrap**
  Wrapped key is sent to KMS.
- **C ACL Check**
  Is the requester (e.g. Storage System) authorized to have key unwrapped?
- **D Send Unwrapped Key**
  KMS unwraps the encryption key data, which Storage System will use to decrypt chunk.
- **E Decrypt Data**
  Storage System decrypts chunk.

Auditing and Access Control for keys data

We complement encryption with rigorous procedures for assigning and removing access to the keys, and logging employee access to the keys and data. We regularly review these procedures and logs to ensure that they are operating in a secure manner and that only the people and applications requiring access are granted it. This process is also audited every year by an independent third party.

Google authorizes only trusted individuals to have legitimate access to systems and data repositories containing customer data, including the KMS. This strict authorization extends to job duties including debugging and maintenance activities that might expose decrypted customer data to a trusted employee. Access to these systems is under the umbrella of strict policies that are clearly displayed for employees to read and also in the tools they use. Access to customer data is only allowed for a legitimate business purpose. As part of Google's long-term commitment to security and transparency, you can use Access Transparency to review logs of actions taken by Google staff when accessing user content. User-generated content consists of data such as uploaded files and user-entered text entered into Gmail, Google Docs, Google Sheets, Google Slides, and other apps.

We complement encryption with rigorous procedures for assigning and removing access to the keys, and logging employee access to the keys and data.

To help ensure that only this limited set of trusted employees uses their given access as approved by Google, we use a combination of automated tools and manual reviews to examine employee access to customer data and detect any suspicious events. We strictly enforce our policies for customer data access. We have established an incident response team to investigate violations of misappropriation of customer data. We have established a disciplinary process for noncompliance with internal processes which can result in immediate termination from Google, lawsuits and criminal prosecution.

Data on backup media

Google also encrypts all data stored on backup media. Backup media, as noted, are used as a recovery mechanism if there is a failure or corruption of the disk data and data needs to be restored. This means that backup media are accessed much less frequently than disks. Each medium contains one or more files, and each medium is protected from tampering with its own unique 256-bit secret. At backup time, a random seed is created for the medium, and the KMS is asked to encrypt the per-medium secret with a key known only to the KMS. The resulting per-medium secret is unique, and is only stored in encrypted form. This secret is used to prevent any modification of data in backups.

The decryption key for the per-medium secret is known only to the KMS and never leaves it. In addition, only the backup service has permission to ask the KMS to decrypt a per-medium secret. This provides a double layer of access control: (1) only authorized personnel and services may read seeds from the backup system's database, and (2) a further authorization check is required to use such a seed to ask the KMS to decrypt a per-medium secret. This provides a further protection against modification of data on a backup medium.

In addition, the backup media ciphertext contains no identifiable information about what is on that medium: all such information is contained in the encrypted files. An individual who steals a medium with the intent of determining what data is stored on it will be unable to do so.

Finally, the backup system can also back up encrypted files for which it cannot read the plaintext. For such files, it backs up the ciphertext and the wrapped key. At restore time, both are restored, again without the backup system ever seeing the plaintext.

Encryption of data in transit

As we've shown, Google Workspace encrypts customer data stored at rest on both disks and backup media. But we also want to protect your information while it's en route from one machine to another data center, ensuring these data transmissions would still be protected should they be intercepted. Data in transit may be traveling over the Internet between the customer and Google or moving within Google as it shifts from one data center to another.

Data traveling over the Internet

When you use a Google service, your information travels over the Internet between your browser, Google's servers, and, sometimes, non-Google users you are communicating with. In these scenarios, encryption helps prevent attackers eavesdropping on internet connections from accessing sensitive content such as your credentials, emails and other personal data.

Between you and Google

To protect your information, the first step is having a secure browser that supports the latest encryption and security updates. When you're a Google Workspace customer, we encrypt traffic between your browser and our data centers — whether you're using public WiFi, logging in at the office, or working from home on your computer, phone or tablet. Google websites and properties use robust public key technologies: 2048-bit RSA or P-256 ECDSA TLS certificates issued by a trusted authority (currently the Google Trust Service CA 1O1).

How this encryption works depends on each customer's client configuration. Google servers support ephemeral elliptic curve Diffie-Hellman cryptographic key exchange signed with RSA and ECDSA ("ECDHE_RSA" and "ECDHE_ECDSA"). These so-called forward secrecy methods help protect traffic between customers and Google servers from being intercepted and decrypted by a man-in-the-middle (MitM) attack.

Forward secrecy technology helps ensure that information encrypted today is less vulnerable to new methods of breaking encryption in the future

Forward secrecy by default helps ensure that information encrypted today is less vulnerable to new methods of breaking encryption in the future. With forward secrecy, keys are rotated at least every other day. This limits the impact of a compromised encryption key to information a customer exchanged over a two-day period (instead of what could be several months of data). Without forward secrecy, in contrast, an adversary could record encrypted traffic and store it with the hope of compromising the HTTPS private key at a later date. If they succeeded, they would then be able to decrypt the data.

Google servers generate a new Diffie-Hellman public key for each session, sign the public key, and use Diffie-Hellman to generate mutual private keys with the customer's browser. This helps prevent eavesdropping because every session between a customer and Google is encrypted with different public keys. An attacker would have to do two things: capture encrypted traffic *and* compromise the temporary private key before it's destroyed.

Forward secrecy also prevents a connection's private keys from being kept in persistent storage. Combined with key rotation, this feature stops adversaries who successfully compromise a single key from retrospectively decrypting data more than two days old.

Between you and non-Google users

We've now reviewed how traffic between a Google Workspace customer and Google servers is encrypted, but what happens when that customer has business beyond Google?

Google has led the industry in using Transport Layer Security (TLS) for email routing, which allows Google and non-Google servers to communicate in an encrypted manner. When you send email from

Google to a non-Google server that supports TLS, the traffic will be encrypted, preventing passive eavesdropping. We believe increased adoption of TLS is so important for the industry that we report TLS progress in our Email Encryption Transparency Report. We also improved email security in transit by developing and supporting the MTA-STS standard allowing receiving domains to require transport confidentiality and integrity protection for emails. Google Workspace customers also have the extra ability to only permit email to be transmitted to specific domains and email addresses if those domains and addresses are covered by TLS. This can be managed through the TLS compliance setting.

Data moving between data centers

We operate a highly secure and resilient private network that encircles the globe and connects our data centers to each other—ensuring that your data stays safe. Trust is built on transparency and we publish the locations of all our data centers.

Our network is designed to minimize latency and maximize availability, helping to ensure uninterrupted access to your information with no scheduled downtime. In order to achieve this level of performance and conduct upgrades or maintenance, we often move data from one data center to another.

These shifts of data centers are imperceptible to our customers and carried out in a secure manner. Namely, data is always encrypted when it moves between data centers. Connections between internal Google servers are cryptographically authenticated between machines. Certain connections (including those to and from the KMS) are encrypted with a TLS-like proprietary transport protocol that uses AES 128-bit or higher.

Encryption protocols used by Google Workspace[1]:

- TLS 1.3
- TLS 1.2
- TLS 1.1
- TLS 1.0[2]
- QUIC

[1] This list of protocols is subject to change at any time.
[2] Google is working to deprecate old protocols and primitives as quickly as users allow. For example, TLS 1.0 and 1.1 have been deprecated in Chrome 72 and removed in Chrome 84. Google Chrome and servers support TLS_FALLBACK_SCSV to prevent attackers from inducing browsers to use lesser protocol versions. Further information is available on the Google Online Security Blog and Google Chromium Group.

Google Workspace Client-side encryption

Google Workspace uses the latest cryptographic standards to encrypt all data at rest and in transit between our facilities by default. In the default mode, Google manages cryptographic keys on behalf of its customers. With Google Workspace Client-side encryption, Google is **giving customers direct control of encryption keys, thereby making customer data indecipherable to Google.** Client-side encryption is targeted to help customers meet the needs of data sovereignty and compliance, while minimizing the impact on end-user experience.

Product Principles: What is Client-side encryption?

Google Workspace Client-side encryption (CSE) allows you to secure Meet, Drive, Docs, Sheets, and Slides data with an external encryption key that Google servers cannot access. The product is built around the following principles:

- No Google access to plain-text content: File content is encrypted in the client before being sent to Google servers for storage. Google cannot unilaterally access content, even for support cases.
- Customer sovereignty of encryption keys: To use CSE, customers need to independently set up their encryption key access service by using one of the partners that have built their services to CSE specifications.
- Preserve user experience: End users can interact with web-based experiences for editing documents or video calls. They can also share files externally or access them on mobile devices.

It's important to note that CSE targets file content for Drive and media content (video and audio streams) for Meet. Most of the metadata, including file names, labels, and the access control list, is not encrypted client-side and is used by Google for running the service.

Using Client-side encryption is optional for eligible Google Workspace customers, who can deploy CSE to their entire organization or a select set of users within their organization.

Technical Architecture

Google Workspace CSE is designed to work for browsers and mobile apps by encrypting and decrypting content on the end user's devices. The Google Workspace client calls into the Key Access Service (KACLS) that the customer has configured and deployed and the client performs cryptographic operations to seal and unseal Google Workspace content.

Key Access Service & Authentication:

KACLS is a service providing fine-grained access control to protect cryptographic key materials. In order to maintain separation of duties, KACLS must be run by an entity other than Google. Authentication is based on cryptographically-signed JSON Web Tokens (JWT) to prove user identity. Client's UI (Drive or Meet) gathers the user's identity client-side, encoded in a JWT called an ID token, and sends it to authenticate all KACLS requests. Two authentication tokens are used to isolate the Google back-ends from the third-party Key Management Service:

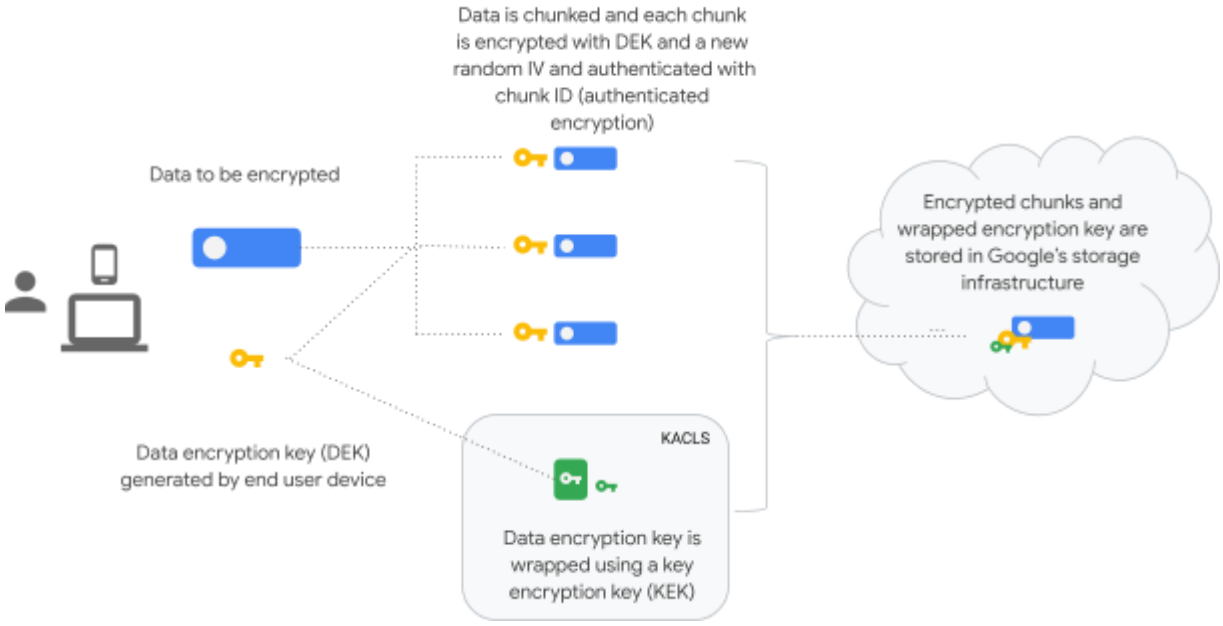
- JWT token with Google identity and resource authorization
- JWT token with third-party IdP authentication assertion

Every new Google Workspace resource created (for instance, each file revision in Google Drive) receives its own new random Data Encryption Key (DEK). After verifying the user identity and the resource identifying, KACLS will wrap (encrypt) or unwrap (decrypt) DEK to protect the content.

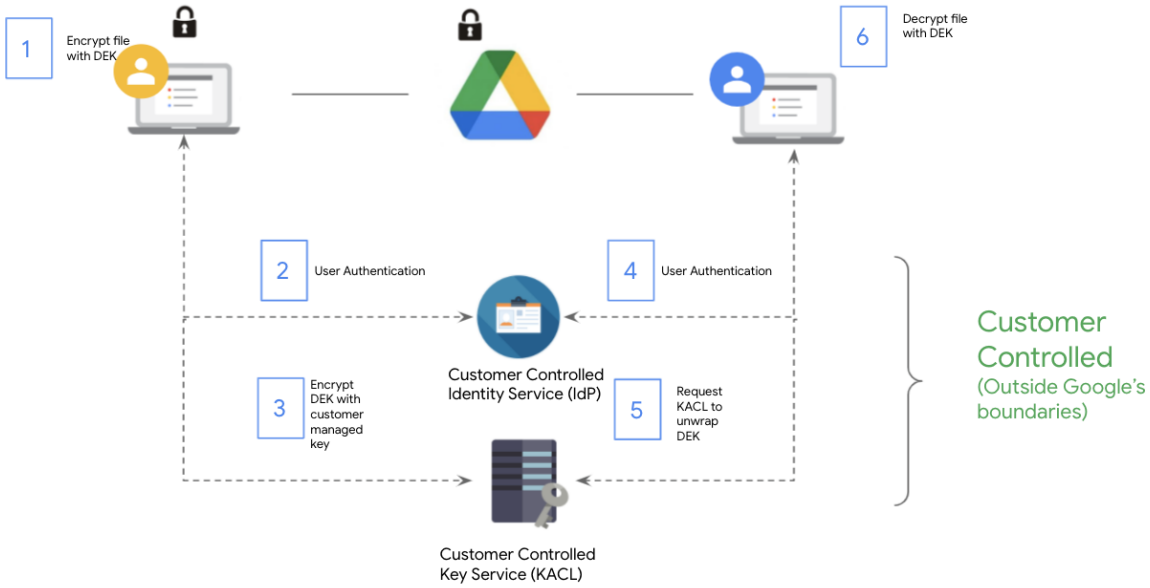
Wrapping and unwrapping by KACLS is transparent to Google, so the KACLS cryptographic key used for those operations, the Key Encryption Key (KEK), can be rotated periodically and will apply to new objects that are created after the rotation.

Encryption of Data (Drive):

Google Workspace CSE uses envelope encryption to protect data and it relies on web browsers for performing client-side operations. First, a data encryption key (DEK) is generated in a Google Workspace client and is used to symmetrically encrypt the document bytes. That DEK is then sent by the client to the KACLS to be encrypted symmetrically (using its KEK). The encrypted content and the encrypted DEK are then sent to Google infrastructure for storage. Crypto operations leverage Google Tink, an open-source library that provides secure and easy-to-use cryptographic APIs. For the web platform, Tink is implemented using SubtleCrypto, a WebCrypto API, supported by modern browsers.



Data Encryption using Google Workspace Client-side encryption



An example of Client-side encryption with Google Drive

Any read/write of a file requires a round trip to KACLS. The per-chunk encryptions/decryptions are performed locally in the browser using the DEK. When a file is updated with new content, a new revision is created. New revisions create a new DEK and a new request to the KACLS server for encrypting it.

Drive for Desktop:

For encryption, Drive for desktop uses the same technology described above as the other Drive clients (web). It exposes the client-side encrypted files as regular files with special icons/overlays on the user's machine. File content is decrypted on access and new file content is encrypted before it is uploaded to Google's servers.

In order to get access tokens from the third-party Identity Provider (IDP), a browser window is opened on the system's default browser. The received tokens are stored in the operating system provided credential store where passwords and other confidential information are stored.

One of the unique challenges with Drive for desktop is that while editing a file, desktop applications (for example, Microsoft Word, Adobe Photoshop) create temporary files to store intermediate states of the edited file. In order to guarantee that these temporary files (with potentially confidential file content) are not uploaded to Google servers we expose the encrypted files as shortcuts (Windows) / symlinks (MacOS) which point to the original files in a virtual folder. Any files created on that virtual folder are automatically encrypted before uploaded to the Google servers. This ensures that confidential file content is always encrypted before sending them to Google's servers.

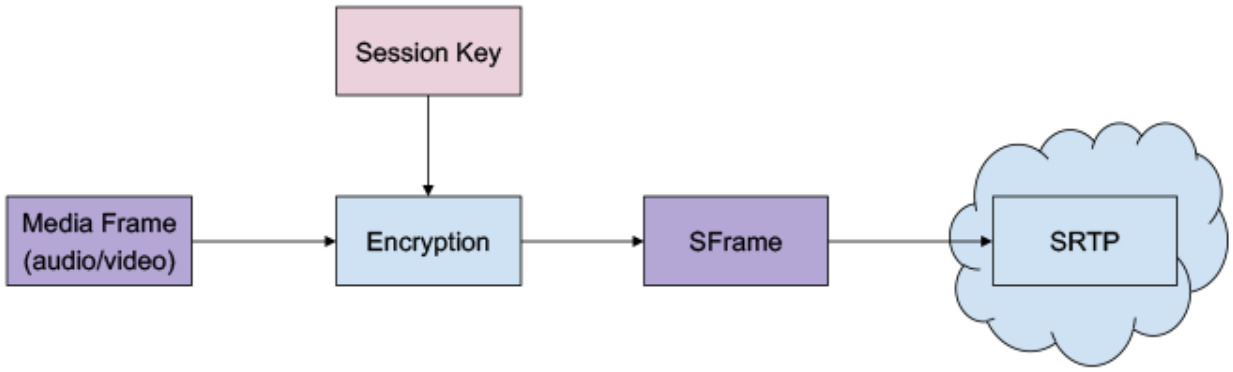
Encryption of Data (Meet):

When initially joining the CSE meeting, the Meet client for moderator/owner of the meeting generates a new random session key. This key is encrypted and only available to the participants of the CSE meeting. Users are unable to join the CSE meeting until the moderator/owner has joined first and created its key.

Once authenticated the moderator/owner will create the session key and pass it to a Key Access Control List (KACL) server configured by its domain that will encrypt the session key. The encrypted session key is then stored within the ephemeral session of the meeting on Google's infrastructure.

Once the encrypted key is successfully stored in the ephemeral session, the invited users will be allowed to join the meeting. Only users invited by the moderator/creator are allowed to join, if any user tries to join the meeting through a meeting code or meeting link, the connection attempt will be rejected. Knocking (access attempt without being in meeting guest list) is not supported due to the requirement for strong authentication.

Now that each client is able to access a shared session key, we use the SFrame protection scheme (currently an IETF RFC draft) to encrypt the audio and video streams. The WebRTC Insertable Streams API is used to intercept and process full video and audio frames locally. This design allows existing client side effects and processing to continue working.



Only the audio/video media streams are encrypted by CSE, other traffic like signaling and metadata are not changed. Signaling refers to the process of setting up and negotiating the call. The metadata contains, for example, timestamps, IP addresses and bitrates, which are used for quality assurance, usage statistics and call debugging. Other features containing user data, such as chat and polls, are disabled in CSE meetings.

While most CSE meetings will be internal within a single organization CSE also supports participants from other Google Workspace organizations, consumers[3] and participants that we normally can't authenticate.

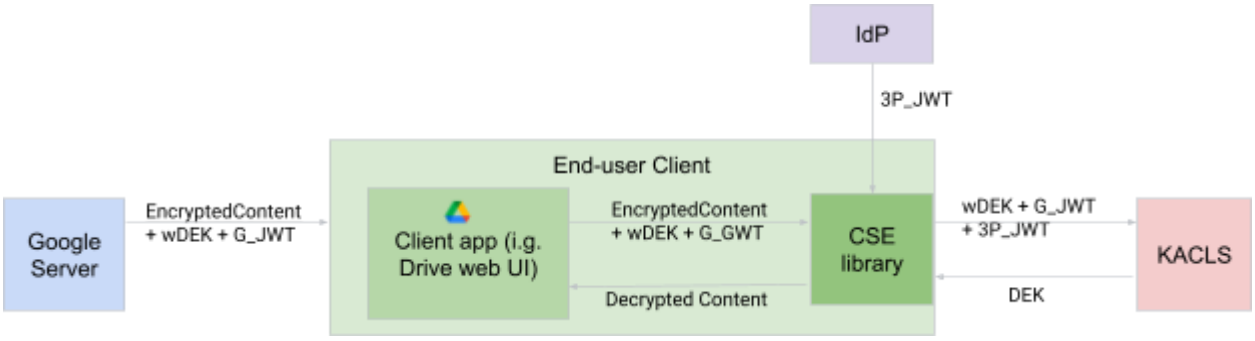
The KACL server of the meeting moderator/owner has control over the decryption of the session key and can control trusted issuers for the authentication token. This means that the administrator of the KACL server can choose which IdPs to trust and allow to authenticate users joining their CSE meetings.

Threat Model

We consider the following systems handling the data:

- IdP
- Google Servers
- End-user client (browser, mobile app)
- KACLS

[3] Google consumer accounts are created by self-service and are for instance any gmail.com email addresses.



Trust model for Google Workspace Client-side encryption

- **IdP** provides a "third-party JWT" (3P_JWT) authentication token. 3P_JWT proves the identity of the user independently of Google.
- **Google servers** provide encrypted content, wrapped DEK (wDEK) for that content and Google JWT (G_JWT) authorization token. G_JWT proves the user is authorized to access a specific resource (for example, a document).
- **End-user client** propagates wDEK, G_JWT, 3P_JWT to KACLS and gets DEK in return. DEK is used to decrypt the encrypted content.
- **KACLS** receives wDEK, G_JWT and 3P_JWT. It verifies that the JWT signatures are valid and trusted. It unwraps the wDEK and verifies the resource identifier of G_JWT matches before returning the DEK.

Note that the strict separation of duties ensures only a properly authenticated and authorized user can access decrypted content.

Threat Analysis

With Google Workspace CSE, the system is designed to prevent unilateral access by a single party.

- **IdP**: It does not have access to the content to be decrypted, therefore cannot gain access to data.
- **Google Servers**: Google servers do not get access to 3P_JWT and therefore cannot authenticate with KACLS to decrypt content. Data was encrypted using random DEK and native cryptographic functions from the client that are also used for HTTPS.
- **KACLS**: KACLS does not have access to the encrypted content, so it cannot recover the decrypted content.
- **End-user client**: End-user client uses a variety of mechanisms to prevent leaking DEK or decrypted content, such as SafeHTML and Content Security Policy to prevent cross scripting attacks (XSS).
- **Data in transit**: All communications are encrypted in transit using the HTTPS stack from the end user client. TLS certificate transparency support enables higher assurance.

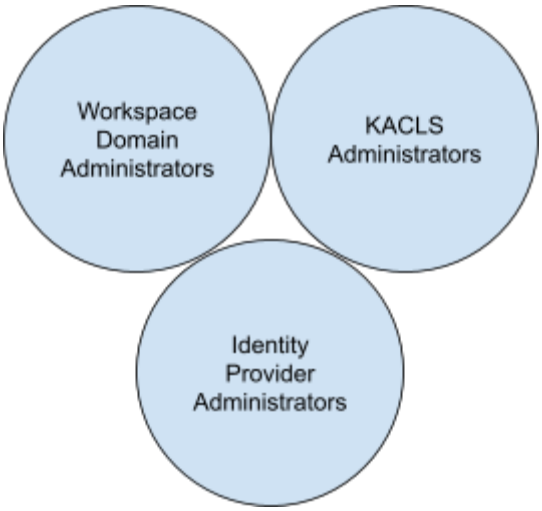
Operational Requirements

Google Workspace domains taking advantage of CSE should take the following measures to protect their data.

Confidentiality

In order to preserve confidentiality, it is critical to ensure proper separation of duties for the different components of the system. The following three groups should be distinct from each other in order to safeguard domain data from unilateral access:

- Google Workspace Domain Administrators
- KACLS Administrators
- Identity Provider Administrators



We also recommend that each domain evaluates the back-end service used by KACLS to encrypt and decrypt client keys and ensure appropriate security for the cryptographic keys, for instance using a Hardware Security Module (HSM).

Availability

Google Workspace domains should ensure the high availability of the KACLS and IdP services used for CSE. Downtime for those services will temporarily prevent access to sealed content.

In order to preserve the availability of the data, it is critical to prevent accidental cryptoshredding (destruction of encrypted content due to loss of ability to decrypt it). As such, the cryptographic keys used by KACLS should follow the appropriate best practices for backing up data.

Abuse prevention

Given that plain-text content is not available for Google servers for CSE encrypted files, anti-abuse capabilities are more limited than non-CSE encrypted content. As a consequence, Google Workspace customers should consider additional protection for their users by taking advantage of Google Workspace settings (1,2) as well as client-side scanning of content.

Client-side encryption provides new levels of confidentiality for Google Workspace customers. We've designed the service to ensure that Google does not have access to users' plain-text content. It is important for customers to evaluate the operational requirements to make sure they have full control over encryption keys, while delivering the native experience their end users expect.

Encryption is only part of our comprehensive security strategy

Google Workspace customers' data is encrypted when it's on a disk, stored on backup media, moving over the Internet or traveling between data centers. Providing cryptographic solutions that address customers' data security concerns is our commitment. But it's important to note that, while encryption is important and necessary, it's not enough, by itself, to protect your information. Instead, it has to be part of an in-depth, well-organized, and executable security and privacy strategy — like the one we have at Google, which is outlined in the Google Workspace Security page. This comprehensive data protection approach is rare and not typically present in many centralized local computing centers.

Indeed, security has always been central to our daily operations and culture. Our custom hardware and unique data storage architecture are designed with security in mind. We constantly invest in security innovation, employing many highly trained security experts and supporting their extensive and intensive research efforts. We also operate in a manner that helps us quickly respond to newly identified threats and develop better ways to align the protection of customer information with the ever-evolving risk that typifies modern computing. By design our systems restrict access to customer data to only a limited

number of individuals and specific applications that require access. For more information on our security practices, please see our Google Workspace Security page.