White Paper

Cisco Webex Teams Security White Paper

Version 2.1 (August 2018)

Cisco Webex is a cloud collaboration platform that provides messaging, calling and meeting features. The Cisco Webex Teams application is a client app that connects to this platform, and provides a comprehensive tool for teamwork. Users can send messages, share files, and meet with different teams, all in one place. This White Paper provides an overview of the security features of Cisco Webex Teams.*

* Some of the Cisco products, services, and features described in this document are still under development or planned for future. After being described, a planned feature will be marked with a "🔨" icon. Cisco will have no liability for delay in the delivery or failure to deliver the products, services or features marked with this icon.

Cisco Webex Teams Addresses Security and Privacy Challenges for Cloud Collaboration

One of the key benefits to enterprises consuming cloud services is the ability to leverage value-added features and functionality as quickly as the cloud service provider can deploy them. But for many cloud providers, “adding value” often means having full access to user data and content. For collaboration applications, most cloud providers directly access message, call, and meeting content in order to offer features like message search, content transcoding, or integrations with third-party applications. Conversely, modern consumer collaboration services tend to be geared toward protecting consumer privacy by offering end-to-end encryption at the expense of value add features.

Cisco Webex Teams is the best of both worlds: an end-to-end encrypted cloud collaboration platform that offers enterprises the ability to choose what, if any, value added integrations are provided by Cisco and third-parties. Cisco Webex Teams makes use of an open architecture for the management of encryption keys, allowing our customers to gain exclusive control over their encryption keys and the confidentiality of their data. This means that content is encrypted on the user’s client and stays encrypted until it reaches the recipient, with no intermediaries having access to decryption keys for content unless the enterprise explicitly chooses to grant such access.

While additional features can be obtained by granting explicit access to keys, end-to-end encryption has been built into the fabric of Webex Teams from the very beginning, which means many of our value-added features and functionality operate on encrypted data. Using innovative message indexing, permissions models, authentication flows, encryption, and deployment models, Webex Teams supports features like global search of encrypted content without ever decrypting it in the cloud.

Most cloud service providers claim to be secure because they encrypt data “in transit” between users’ devices and their servers, or between their own data centers, and “at rest”, while stored on their servers. Some providers even let customers provide the keys for this encryption. But even with encryption in transit and at rest, servers in the cloud can still access customer content – so customers are still vulnerable to breaches in the cloud provider itself. The end-to-end architecture built into Webex Teams means that servers are only trusted with customer content when the customer chooses to trust them.

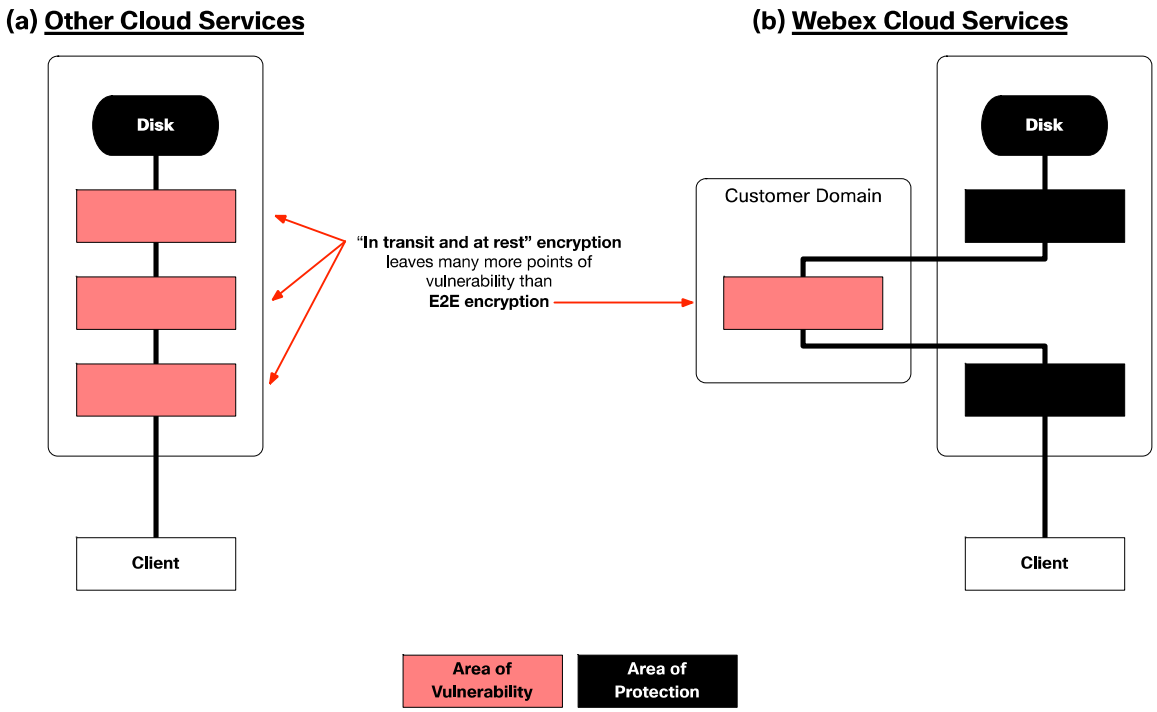

Figure 1. Security architecture of other cloud applications (A) vs. Webex Teams (B)

Our commitment to providing a trusted service offering is not limited to protection of user content. In Webex Teams, all data about users and usage is protected using a combination of privacy tools and features that includes obfuscated identity, choice, and transparency. As with end-to-end encryption, these protections were built into the service from the ground up.

In this whitepaper, we describe the full suite of tools that Webex Teams provides to keep customers' data safe from Cisco as well as external attackers.

- **End-to-End Security in Webex Teams**: How we ensure that Cisco can only access customer data when customers explicitly grant access
- **Securing Webex Teams Usage:** How we enable customers to ensure that usage of Webex Teams for collaboration doesn't create new security risks
- **Securing the platform:** How we assure that the Cisco Webex cloud and the Webex Teams app are safe from external attack

End-to-End Security

End-to-End security is a central security feature for Webex Teams, providing an extra layer of protection beyond standard cloud security. All customer data transmitted through the Webex cloud is encrypted before being sent, so that cloud components only ever handle customer data in a safe, encrypted form. As a result, **even if one of these cloud components is fully compromised – a situation where an "encryption at rest" and "encryption in transit" approach would fail – the attacker still can't access customer data, because it's encrypted end-to-end**. We only make exceptions to this rule with explicit customer authorization.

The separation between customer data and Cisco Webex is fundamental to the way Webex Teams is architected and operated – so much so that we think of the overall functionality of Webex Teams as being divided between two trust domains, a "Customer Domain" and a "Webex Teams Core". The Customer Domain comprises things that run directly under the customer's control: the fleet of clients and hardware endpoints used within an enterprise as well as any client-operated Webex Teams infrastructure that helps these clients communicate securely. The Webex Teams Core contains Cisco-operated Webex services that enable these clients to collaborate with each other and with the infrastructure of the Customer Domain.

The E2E Critical Path

The separation between the Customer Domain and the Webex Teams Core is ultimately enforced by cryptography – encryption of customer data protects it from access by untrusted elements in the Webex Teams Core. So the strength of the separation comes down to how well those keys for that encryption are protected. Management of access to keys involves a few key components that work together as an "end-to-end critical path":

Bootstrap	- Tells clients which KMS and Authorization Service to trust
Authorization Service	- Tells clients which Identity Provider to trust - Issues SAML requests trusted by Identity Provider - Issues tokens trusted by KMS
Identity / SSO Provider (IdP)	- Receives users' passwords - Issues SAML responses trusted by Authorization Service
Key Management Server (KMS)	- Relies on tokens from the Authorization Service - Generates and controls access to keys
Plaintext Services	- For example: Search indexing, document transcoding - Authorized by customer to access plaintext content

Figure 2. The Webex Teams end-to-end critical path

The elements of the critical path work together to ensure that only entities authorized by the customer can access the keys to decrypt that customer's content – and all unauthorized parties are locked out. To understand how the elements in the critical path together to achieve this separation, let's take a look at how a client logs into Webex Teams, illustrated in Figure 3.

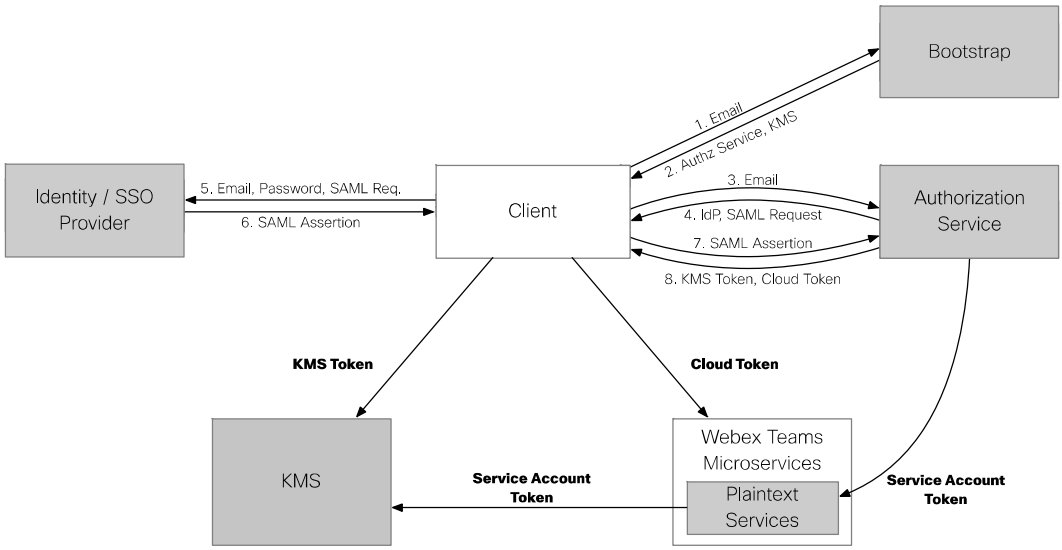

Figure 3. The Webex Teams login flow

The first thing a user does when logging into Webex Teams is enter an email address. Based on that email address, the client looks up which authorization service and Key Management Service (KMS) it should trust from a bootstrapping service provided by Webex Teams (1, 2). The client then engages in a standard SAML login with the authorization service, where the client is redirected to the proper identity provider based on its email address (3, 4), logs in with the identity provider to get a SAML assertion (5, 6), and provides the SAML assertion back to the authorization service (7). The authorization service then provides the client with two OAuth2 tokens (8): A KMS token that it will use to authenticate only to its KMS, and a cloud token that it will use to authenticate to microservices in the Webex cloud.

The KMS safeguards the keys used to encrypt a customer's data, and ensures that only entities authorized by the customer can access those keys. Clients download keys on behalf of the organizations' users. Requests by clients are authenticated with the KMS tokens discussed above. The separation between KMS tokens and cloud tokens means that the client will only ever send the KMS token to the KMS – **Webex Teams components can't impersonate users**. When the customer has authorized a Webex Teams microservice to access some content (a "plaintext service", discussed in more detail below), that service gets a special token from the Authorization Service that proves to the KMS that the service is of an authorized type (a "Service Account Token"). **Customers can opt in to specific plaintext services without giving the whole Webex cloud access to their data .**

The Webex Teams E2E methodology lets us separate out this critical path for extra protection. Since E2E encryption largely eliminates the risk of compromise for other components, we can focus on protecting the critical path. For some elements of the critical path, such as the KMS and the search indexer (a plaintext service), customers can operate their instances directly, on-premises, through our Hybrid Data Security architecture. When these components are hosted by Cisco, we keep them under separate access controls and operations 🔨 from the rest of Webex Teams.

Ultimately, this produces a dramatic reduction in the information that an attacker can get by breaching the cloud provider. This difference is visualized in Figure 1. On the left is a typical cloud service provider, using encryption in transit and at rest. Only the encrypted storage and the connections between cloud

components are protected – a breach in any cloud service results in a compromise of customer information. In contrast, the E2E encryption approach taken by Webex Teams means that customer information is encrypted by default and only decrypted when needed. With typical cloud services that handle customer information in plaintext, the more services the cloud provides, the more risk that customer information will be breached. E2E encryption allows Webex Teams to provide rich cloud services while maintaining a small attack surface.

End-to-End Content Encryption

In Webex Teams, client applications use end-to-end encryption so that they can use Cisco Webex to deliver content without that content being accessible to the cloud. The keys for this encryption are managed by a Key Management Service (KMS). The KMS for a customer is effectively the customer's agent for controlling who can access content encryption keys (and thus who can access that customer's content). We think of the components running under the customer's control, e.g., the customers KMS and clients, as comprising the Customer Domain. Even when some of these components are operated by Cisco, they are kept separate from other Webex Teams components.

The actual encryption of content, though, is performed by clients, who have to get keys from the KMS – effectively "checking the encryption keys out" from the customer responsible for the client. To do this safely, each client establishes an encrypted tunnel through Cisco Webex to the KMS for its organization. This tunnel works much the same way as the TLS and IPsec protocols used for things like HTTPS and VPNs, using an authenticated ephemeral Elliptic Curve Diffie-Hellman (ECDH) exchange. The client first receives a certificate that associates an RSA public key to the KMS's domain name (1). The client generates an EC key pair, encrypts the public component of that pair using the KMS's RSA public key, and sends the result to the KMS over Webex Teams (2). The KMS decrypts this message, generates its own ephemeral EC key pair, signs the public component with the RSA private key corresponding to the certificate, and sends the result in a message back to the client (3). At this point, the client and server have a shared secret that they can use to derive an encryption key for further messages, and the KMS has proven its identity to the client. To prove the client's identity, the client will send the KMS its KMS token on future requests (4).

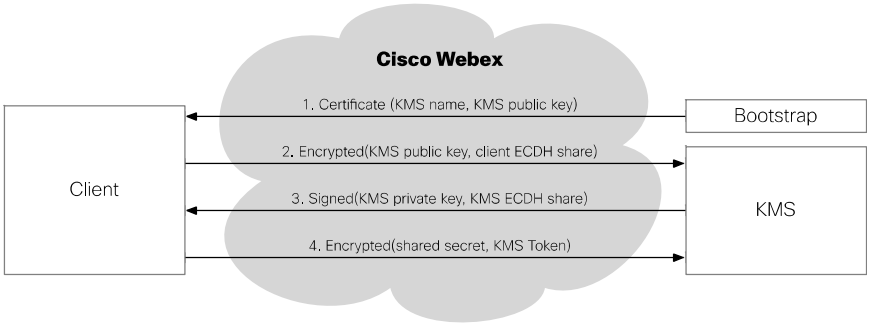

Figure 4. Secure communication with the KMS

Each encrypted item in Webex Teams, like a message or file, is tagged with a key URL that indicates what key can be used to decrypt it. When a client needs a key, it requests it from its KMS. If the key's URL indicates that it is stored on another KMS, the requesting client's KMS fetches it from that KMS on behalf of the client. Each key has an associated access control list (ACL) that identifies the users that are allowed to access the key. Before granting access to a key, the KMS storing the key verifies that the requesting user is

on the ACL. If it receives a request from another KMS, it also verifies that the requesting KMS is authorized to act on behalf of the user.

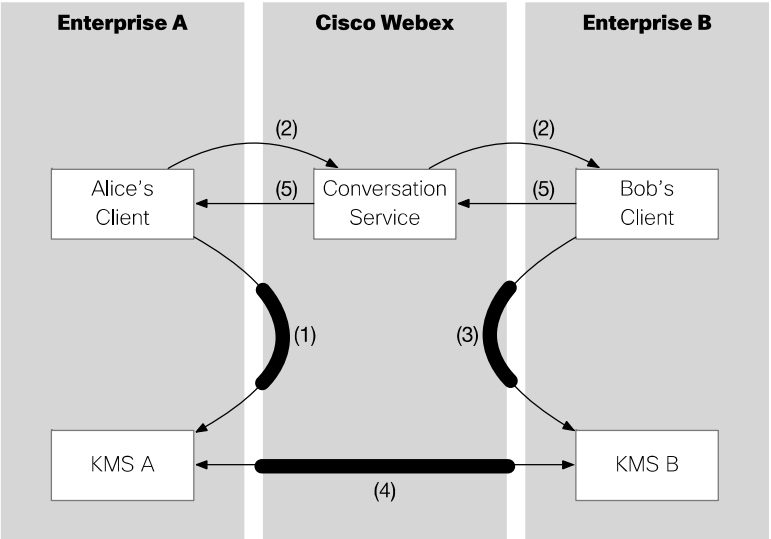

Figure 5. Secure messaging in Webex Teams

To see how this all fits together, let's look at how two users, Alice and Bob, in different organizations can send messages to each other. When Alice creates a conversation with Bob, her client gets a key for it from KMS A (via an ECDH tunnel through the cloud) and tells the KMS A that Bob is authorized (1). When she tells Webex Teams to create the conversation, she also tells it the key URL for the conversation, which the conversation service relays to Bob (2). When Bob's client joins the conversation, it requests the key from the KMS for Bob's enterprise (KMS B) (3). Bob's KMS sees that the key is stored on KMS A, and forwards the request (4). KMS A checks that Bob is authorized to receive the requested key and that KMS B is authorized to represent Bob. If these checks pass, it provides the key to KMS B, who provides it to Bob. Bob can then use the key to encrypt a message for Alice and safely send it to the conversation service in Cisco Webex (4), which will then store it and forward it to Alice when she comes online (and likewise for any other participants in the space). Since Alice's client has the same key, it can decrypt the message and show it to Alice.

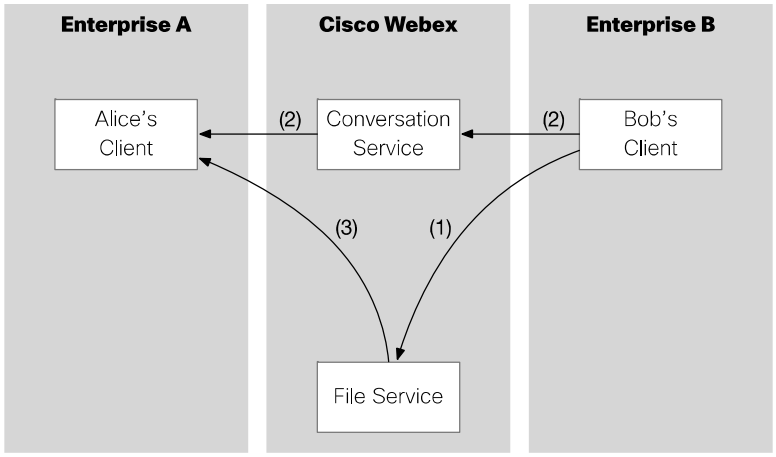

Figure 6. Secure file sharing in Webex Teams

Files in Webex Teams are protected in a similar way. When Bob wants to upload a file to a space, his client generates a new key and uses it to encrypt the file. It then sends the encrypted file to a file storage service within Cisco Webex (1). To enable other clients in the space to download the file, Bob's client constructs a message containing the key used to encrypt the file and a URL for the encrypted file. This message is then sent to the other users in the space, encrypted with the same key as other messages in the space (2). When Alice's client receives this special message, it can retrieve the encrypted file using the URL, decrypt it using the key, and show it to Alice.

The full protocol that clients and cloud components use to interact with the KMS has been published as an IETF Internet-Draft.

Controlling Access to Keys

In order to make sure that unauthorized parties can't get access to the keys used for end-to-end encryption, an organization's KMS keeps track of who is allowed to have each key. When a space is created, the KMS provisions a KMS Resource Object (KRO) that it will use to track the keys for the space and the people authorized to receive them.

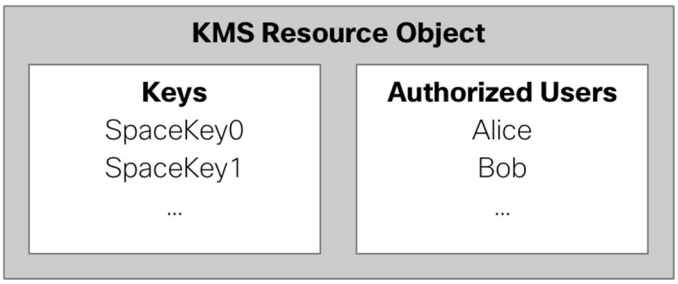

Figure 7. Structure of a KMS Resource Object

Each space has one key at any given time, which all participants in the space use to encrypt messages. When a user adds a new participant to a space, they also add the new participant to the corresponding KRO, so that the new participant can fetch the key. When someone leaves a space (or is removed), they are removed from the KRO. Then the key for the space is rotated by getting a fresh key from the KMS and

"binding" it to the space's KRO so that remaining participants can download it – but not the participant who just left 🔨.

Files are handled in a similar way. As discussed above, each file is encrypted with a separate key, which is sent alongside the URL to the file itself in an encrypted message. Since files are shared using messages encrypted for a space, a client can only decrypt a file if it can access the keys for the space.

As a result of these mechanisms, the current participants in a space can download and decrypt any of the messages or files that were sent in the space, including ones sent before a given participant joined. Once a participant leaves the space, however, they won't be able to access the keys used for anything sent after they left. In addition, because they've been removed from the KRO, they won't even be able to download any keys that were in use while they were in the space.

Feature Access to Keys

Certain Webex Teams features require a cloud service to have access to plaintext for content that would otherwise be encrypted. The current list of plaintext features is as follows:

- Voice and video calling: Media nodes need to be able to decrypt voice and video packets to provide services such as transcoding, media mixing, PSTN interoperability, and meeting recording
- Document transcoding: Creates preview images from documents uploaded to a space
- API access: Allows bots and integrations to access space content without integrating with the E2E system
- (*) Search indexing: Creates an encrypted index that can safely be stored and searched in the cloud
- (*) eDiscovery: Enables searches on encrypted messages for compliance purposes
- (*) Calendar connector: Enables meetings scheduled through Webex Teams to automatically be reflected in a customers' calendaring system. (Keys are needed to decrypt space titles.)
- IM & Presence Interoperability: Enables interoperation between Webex Teams and some other messaging systems (e.g., Cisco Jabber). Since these systems do not support E2E encryption, the component providing interoperability needs to decrypt any encrypted content.
- Webex Teams Assistant: Provides an AI-enabled smart assistant that can do things like start meetings automatically. In order to take actions for a user, the Assistant needs to access that user's Webex Teams data.

For services marked with (*), it is possible for a customer to run this component on-premises. Further discussion on encrypted search and bots/integrations can be found below.

Plaintext features are only enabled with the consent of the organization whose content is being exposed to the cloud. To enforce this consent, features have to obtain access to decryption keys from the organization's KMS, which can apply the organization's policies on which features are allowed access 🔨.

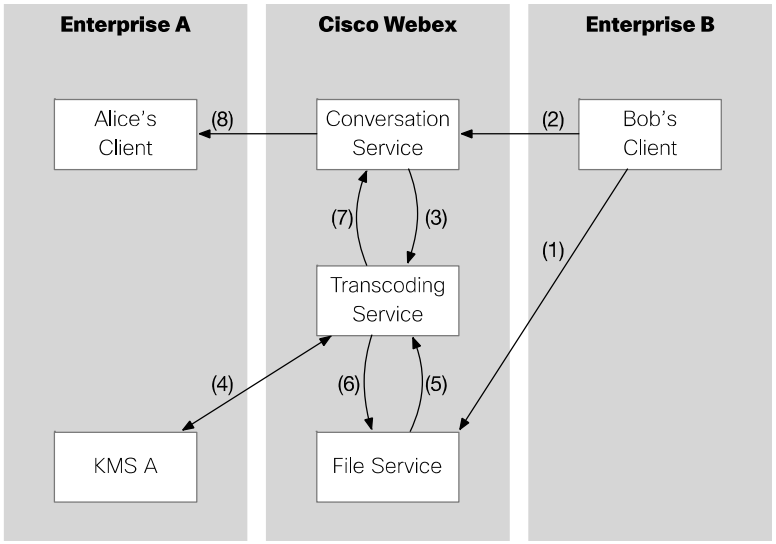

Figure 8. Key access for plaintext services

Suppose Enterprise A has enabled the file transcoding service to provide document previews in spaces owned by the enterprise. When Bob posts a file into the space (1, 2), the transcoding service will get a notification of the upload, containing the encrypted message with the file URL and key (3). It can then fetch the key to decrypt this message (4), so that it can fetch the file (5) and decrypt it. After doing the actual transcode, the transcoding service will encrypt the resulting preview images and post them to the file service (6) and send a notification to the space (7). Other clients in the space can then fetch the preview image, decrypt it, and display it as they would any other file (8). The two important things to note here are: (1) Decrypted customer content was only ever visible to the transcoding service, and (2) if Enterprise A had not allowed transcoding, the transcoding service would not have been able to decrypt the content.

Note the involvement of Enterprise A's KMS in this flow: If the transcoding service can't get the message key from KMS A, then it can't even download the encrypted file. In other words, the transcoding service (in the Webex Teams Core) can only access customer data if the customer has made the choice to allow this access by configuring the KMS to provide keys to the transcoding service.

Encrypted Search

One of the most frequently used features in any messaging system is search. Search in Cisco Webex Teams is built so that search only requires access to the plaintext of a message once, to build an encrypted index – after that, clients can do searches directly on encrypted data in the cloud. This same technology allows us to provide services like eDiscovery with strong security guarantees.

There are two major steps in the search process: creating a search index as messages are sent, and performing queries using that index. Both of these steps are assisted by a Search Indexer service. Because this service requires access to plaintext, it is part of the end-to-end critical path, so we keep it separate from the rest of the Cisco Webex. With Hybrid Data Security, enterprises can even run their own Search Indexer (just like they can run their own KMS), so that searches can be done without requiring the Cisco Webex cloud to have access to plaintext.

To build the search index, the Search Indexer service takes in a feed of every message sent within an organization. When a message is sent in a space, a copy is sent to the Search Indexer, which fetches the appropriate encryption key from its organization's KMS and decrypts the message. The Search Indexer then transforms the text of the message into a set of possible search terms, first breaking the message into individual words (tokenizing) and then reducing each word to a root form (stemming). Using a search indexing key from the KMS, the Search Indexer then uses a hash-based message authentication code (HMAC) algorithm to transform each search term into an opaque value that represents the term. The HMAC transformation is basically one-way encryption – given a particular HMAC output value, there is no way to reverse it back to the word that appeared in the original message. After all these steps, the Search Indexer has an index entry that is safe to upload to the cloud, associating a given message with a set of values.

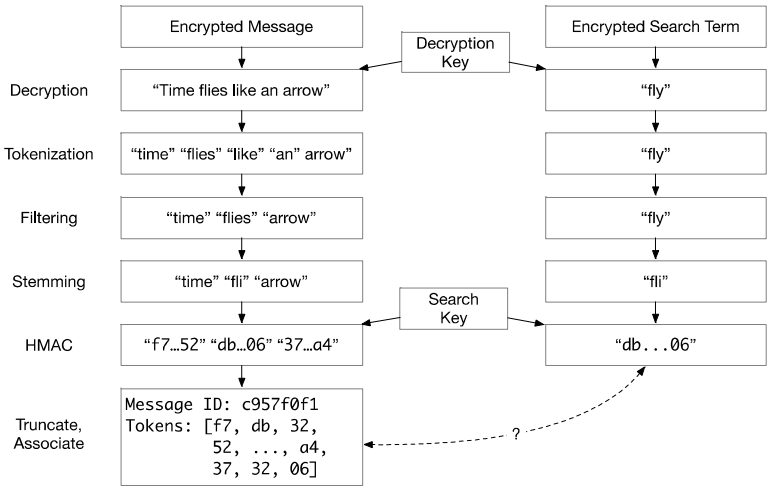

Figure 9. Encrypted search in Webex Teams

To perform a search, a client sends its query to the Search Indexer in an end-to-end encrypted message, just as if it were having a one-on-one conversation with the indexer. The Search Indexer then repeats the same process on the query that it did on the original message (tokenizing, stemming, etc.) to reduce the message to its constituent parts. However, before applying HMAC to the parts, the Search Indexer retrieves from Cisco Webex a list of rooms the user is part of. The search indexer then generates the HMAC value for each part in the query with each room's search key. So if a client in 10 rooms types in a two-word search query, the Search Indexer will produce around 20 HMAC output values. The Search Indexer passes these values off to the Search Service in the Cisco Webex cloud, which can compare them to its encrypted index and inform the client about matching messages.

We built the search capability in Webex Teams without sacrificing either security or user experience. While other cloud services need to decrypt user content in the cloud in order to provide search, our approach lets us offer the same snappy search experience without the cloud ever needing to access user content.

Real-Time Media Encryption

All real-time media in Cisco Webex Teams, such as voice, video, and desktop share, are transmitted using the Secure Real-Time Transport Protocol (SRTP). SRTP provides confidentiality, integrity, and authenticity protection for real-time media against network attackers. Currently, real-time media are not encrypted end-to-end: **Cisco Webex decrypts real-time media for mixing, distribution, and public switched telephone**

network (PSTN) interoperability purposes.

To further increase the security of SRTP in the future, Cisco is one of the driving forces between the Privacy Enhanced RTP Conferencing (PERC) working group at IETF. The goal of PERC is to enable end-to-end encrypted media while still being compatible with some functions provided by the cloud. As this new standard matures, Cisco Webex Teams will make use of it to extend the end-to-end protections provided to messages and files today so that they also cover real-time media. Note that PERC will not affect PSTN call decryption, which will require cloud decryption by the PSTN provider for the foreseeable future.

Enterprise and User Choice

Webex Teams is designed to give both users and enterprises privacy choices without presenting complicated configuration interfaces. For enterprise administrators, choices include:

- Single Sign-On (SSO): Administrators can configure Webex Teams to work with their existing SSO solutions. We support identity providers using Security Assertion Markup Language (SAML) 2.0 and Open Authorization (OAuth) 2.0.
- Directory synchronization: Administrators can have employee lifecycle changes reflected in Webex Teams in real time when using Microsoft Active Directory.
- Data sharing with Cisco partners: Enterprises can choose whether to share quality of service and engagement data with their Cisco partners to enable higher level partner support.
- Enterprise privacy controls to comply with EU GDPR (EU General Data Protection Regulation) are detailed in the Webex Service Privacy Data Sheet. These controls ensure that right to export, right to be forgotten, time-bound purges of user content and other rights related to processing personal data are covered.

For users, choices include:

- Device permissions. The Webex Teams app requests a variety of device permissions, including phone, microphone, camera, audio recording, screen sharing, calendar, contacts, files and photos, and push notifications. On most platforms (especially mobile platforms and the web) these require explicit user permission and the user can revoke permission at any time.
- Proximity features. On mobile devices, Webex Teams clients can automatically pair with Cisco voice and video endpoints by listening for ultrasonic and Wi-Fi signals when the Webex Teams app is active. Because this requires use of the device's microphone and Wi-Fi antennas, we offer the option for users to turn this feature off if they so choose.
- Profile photos. Profile photos are encouraged, but not required to use Webex Teams.
- External participant indicators. The Cisco Webex Teams app makes it clear to users, through visual indicators, when a room contains participants that are not part of their enterprise organization.
- Room moderator control. Rooms in Cisco Webex Teams can be moderated, allowing chosen room participants to be made in to moderators that have exclusive control of the room's title and participant list.
- Further privacy controls are defined in the Webex Service Privacy Data Sheet.

Details about data collection and privacy in Cisco Webex Teams can be found at the Cisco Trust Center.

Transparency

We want our users and customers to understand what their choices are and how we are managing and protecting the data they entrust to us. We use a layered model of transparency to make this happen. Short disclosures that help users make real-time decisions are provided within the Webex Teams app itself. Further information is available in our support pages, which get updated on a regular basis. And for all the details of what information we collect, how we use it, and how we protect it, we provide the Webex Service Privacy Data Sheet.

Cisco is also committed to publishing data regarding requests or demands for customer data that we receive from law enforcement and national security agencies around the world. We will publish this data twice yearly (covering a reporting period of either January-to-June or July-to-December). Like other technology companies, we will publish this data six months after the end of a given reporting period in compliance with restrictions on the timing of such reports.

More information can be found at in the transparency section of the Cisco Trust Center.

Cisco has also invested in a number of transfer mechanisms to enable the lawful use of data across jurisdictions, including:

- Binding Corporate Rules
- EU-US and Swiss-US Privacy Shield Frameworks
- APEC Cross Border Privacy Rules
- EU Standard Contractual Clauses

Securing Webex Teams Usage

Collaboration tools need to fit into an organization's overall operations security approach. Cisco Webex Teams provides tools for administrators to manage usage of Webex Teams to limit risk. Webex Teams also works well with other tools that enterprises use to keep themselves safe, from firewalls to content management systems.

Management of Content Shared through Webex Teams

As enabling as information-sharing can be for an organization, it also presents risks. For example, the risk that confidential information will be shared inappropriately, or the risk that necessary content will not be accessible when needed for compliance purposes. Cisco Webex Teams provides a suite of tools that allow enterprises to manage these risks, such as:

- **Archiving and Retention:** All user content in Webex Teams is stored for a defined retention interval, after which it is deleted. This retention interval can be customized by an administrator. If one is not specified, a default retention interval is applied 🔨. Webex Teams can also be connected with external archiving services.
- **Data Loss Prevention:** Webex Teams can integrate with multiple DLP providers to identify policy violations and take immediate remediation action.

- **eDiscovery:** The Webex Teams eDiscovery console allows compliance administrators to search and extract relevant messages and files, as well as contextual data such as timestamps, space IDs and participants IDs.
- **Compliance APIs:** Publicly available APIS used by a special Compliance Officer Role are available for enterprises who would like to do custom integrations for their content management.

Extending Webex Teams

Cisco Webex Teams provides open APIs that enterprises can use to automate Webex Teams and connect it with other services. We offer three different ways to extend Webex Teams:

- **Bots** provide extended functionality for an entire enterprise, such as a call recording service. A bot must either create a space or be invited to it before it has access. Even within a space, a bot only has access to messages that reference the bot explicitly (with a "mention").
- **Integrations** provide extended functionality for a single user, such as a personal assistant or document translation. Integration have the same capabilities in Webex Teams that the associated user does – access to the same spaces, messages, files, calls, etc. An integration can be thought of as a cloud or server hosted client with no user interface and with some additional intelligence.
- **Webhooks** are a way that bots or integrations can have Webex Teams "call out" to external services when certain events happen in Webex Teams. Webhooks are only provided with metadata that is already visible to Webex Teams; they do not have access to end-to-end encrypted content. For example, a user can set a webhook to be notified when there is a message in a space, and the webhook will only be informed of information about the message that is visible to Webex Teams, such as who sent the message and which space it was sent in – not the content of the message.

More information about Cisco Webex APIs can be found at https://developer.webex.com/.

In order to provide developers with APIs that are easy to learn and use, we do not require bots and integrations to explicitly integrate with the Webex Teams end-to-end encryption system. Instead, developers can use a Webex Teams SDK or the Webex Teams API server.

Using the SDK is the more secure option. When developers use the Webex Teams SDK, the SDK will handle all the work of integrating with the E2E encryption system – the SDK authenticates directly to the appropriate KMS and does all the encryption/decryption locally. Customers that use SDK-based bots and integrations need to make sure that the code for the bots/integrations runs in a secure context, but they don't need to worry about Cisco Webex having access to any keys or content.

In contexts where it's not possible to use the SDK, Webex Teams also provides an API server that can handle KMS interactions and decrypt content on behalf of the bot or integration. When a bot or integration requests access to encrypted content (such as a message or file), the API server requests the necessary encryption key from the appropriate KMS, decrypts the content, and provides it to the bot or integration. From a security perspective, the API server is a plaintext service, placing it on the end-to-end critical path. And that means that it's up to each enterprise to decide whether to provide it access to the enterprise's content.

While we believe that the security of Cisco Webex is the best in the industry, every Cisco customer has different security requirements. The key to making this hybrid model work is customer choice. Customers can choose to use only the core Webex Teams system or to extend it with bots and integrations. Cisco Webex was designed around well-documented, standards based, APIs which means that bots, integrations,

and webhooks can all be developed by customers or third parties without permission from Webex Teams. Cisco also provides a collection of bots and integrations at https://apphub.webex.com

With an open platform come concerns around how to secure the enterprise's content from 3rd party integrations. Integrations management through the Control Hub are going to allow an administrator (1) visibility into available integrations (2) usage of these integrations by their users (3) capability to set an allow/deny policy for these integrations 🔨. Similar Bot management capabilities will be provided 🔨.

Device and Browser Protection

In order to keep sensitive information shared through Webex Teams private from local attackers, it's important for the devices that Webex Teams runs on to be secure. Webex Teams offers administrators several ways to assure the safety of their organization's Webex Teams clients, for example:

- Require that mobile devices are secured with a PIN
- Remotely wipe Webex Teams content in the event that a device is lost or a user leaves the organization
- Automatically log out users of the web clients for Webex Teams and Control Hub after a period of inactivity 🔨
- Prohibit file uploads or downloads from certain types of client

Moreover, all of this function is native to Webex Teams – there's no need for a separate Mobile Device Management (MDM) or Mobile Application Management (MAM) system. Nonetheless, Cisco Webex Teams can still be managed through MDM/MAM, and has been verified to work with several MDM/MAM controls, such as:

- Preventing screen capture
- Preventing copy/paste
- Local back-ups
- Remote wipe
- Requiring PIN lock

Predictable Network Footprint

Enterprises have an increasingly challenging balance to strike: They want the flexibility of cloud-deployed applications, but they want the assurance of knowing what's going on in their networks. Cisco Webex Teams is designed to meet both of these needs by having a network traffic profile that stays within defined boundaries. Those boundaries are broad enough to enable the flexibility that a cloud-deployed product needs, but narrow enough to limit the risk that malicious traffic could fit within them.

Cisco Webex Teams only sends two types of traffic: HTTPS/WebSockets, and real-time media. HTTPS and WebSockets are used to communicate with Cisco Webex, and run over TCP port 443. Real time media packets use UDP on a small range of ports. Media packets are exchanged with Webex Teams media servers located within specified IP ranges. A full description of the network requirements for Webex Teams can be found in this Collaboration Help article.

Securing Cisco Webex Teams

Cisco Webex Teams incorporates a full suite of security mechanisms to ensure that it is safe from interference by outside actors. It has ubiquitous, high-grade encryption, so that data is protected in transit and at rest, in addition to the end-to-end protection discussed above. Webex Teams engineering follows Cisco's industry-leading practices to reduce the likelihood of vulnerabilities in Webex Teams, and to ensure that when vulnerabilities exist, they are found and fixed quickly.

Communications Security

All network communications in Webex Teams are protected by the Transport Layer Security (TLS) protocol, using certificates from publicly-trusted certificate authorities (CAs). This includes communication between clients and the core, communication among Webex Teams services in the core, and communication between the core and customer-hosted services in the Security Realm. This level of protection ensures that attackers in customer networks, transit networks, or cloud data centers cannot read, intercept, or modify Webex Teams communications.

Communications between Webex Teams clients and the Webex Teams Core use an additional technique known as "public key pinning". Public key pinning dramatically decreases the risk of server impersonation. Without pinning, any of the roughly 2000 publicly-trusted CA could issue a certificate that a bad actor could use to intercept Webex Teams communications. Pinning isolates this risk to a handful of CAs that have been well vetted. In addition to verifying that the CAs we allow through pinning have strong security practices themselves, we require these CAs to commit that they will not delegate their signing authority to anyone else, since this would introduce a risk that the delegate's practices would not be up to our standards. This commitment must be expressed in the issuer's Certification Practice Statement (CPS) and in the CA's certificate (by including "pathLenConstraint" set to zero).

Some customer network environments include security devices that impersonate TLS servers. These devices are sometimes known as "SSL inspection" devices / proxies. By default, these devices are incompatible with pinning, because the certificate the client sees is not from one of the approved CAs. Webex Teams desktop clients and the Webex Teams web client thus apply a more flexible pinning policy: They allow a TLS connection if the server's certificate is issued by a pinned CA, or **if it issued by a CA that an administrator has installed on the host computer**. Webex Teams room systems (including SX, DX, MX, and Room series) can also be configured to trust a customer CA. Webex Teams mobile clients and Webex Board do not have this affordance, and cannot be used in networks where SSL inspection is performed.

Encrypted Storage

In addition to encrypting data as it transits the network, Webex Teams also applies encryption "at rest" to guard against the compromise of storage devices it relies on. For most Webex Teams services, customer data is already encrypted using the end-to-end encryption techniques discussed above.

A user's Webex Teams app stores the user's credentials, messages and files a user has received, keys to decrypt those content items – all on computers and phones that are more likely to be stolen than a server in a datacenter. The Webex Teams app maintains all information in an encrypted database, then protects the database encryption keys using platform-provided APIs such as the Windows Data Protection API. (The Webex Teams web client does not currently use encrypted storage.) In addition, Webex Teams enables an administrator for an enterprise to remotely delete data cached on a client device.

An enterprise's Webex Teams KMS manages a database of keys that are used to encrypt that enterprise's content. For customers using the Cisco-operated KMS, the records in this database are encrypted with a master key that is only stored in memory. For customers operating their own KMS, the database is provided by the customer and encrypted using a master key stored separately in a secure configuration file. There will also be a capability to rotate the master key 🔨.

Platform and Service Security

Webex Teams has ISO 27001:2013 certification and SOC2 Type 2 and SOC 3 attestation. Compliance with these standards entails maintaining a high level of operational security, performing vulnerability assessments and penetration tests, undergoing annual audits by a third party auditor, and adhering to an SLA for incident response times.

Cisco Webex Teams has also conducted a HIPAA self-assessment based on the HHS Security Risk Assessment tool, and is ready for use in health care, consistent with customer needs for HIPAA compliance.

The development, deployment, and operation of Webex Teams software and services follow the Cisco Secure Development Lifecycle (CSDL).

Incident Response and Vulnerability Reporting

The Cisco Product Security Incident Response Team (PSIRT) is responsible for responding to Cisco product security incidents. Cisco PSIRT is a dedicated, global team that manages the receipt, investigation, and public reporting of security vulnerability information that is related to Cisco products and networks. The on-call Cisco PSIRT works 24 hours with Cisco customers, independent security researchers, consultants, industry organizations, and other vendors to identify possible security issues with Cisco products and networks.

Individuals or organizations that are experiencing a product security issue are strongly encouraged to contact the Cisco PSIRT. Cisco welcomes reports from independent researchers, industry organizations, vendors, customers, and other sources concerned with product or network security. Please contact the Cisco PSIRT using one of the following methods.

	Emergency Support
Phone	+1 877 228 7302 (toll-free within North America) +1 408 525 6532 (International direct-dial)
Hours	24 hours a day, 7 days a week

	Non-Emergency Support
Email	psirt@cisco.com
Hours	Support requests that are received via e-mail are typically acknowledged within 48 hours.

More information can be found at:
http://www.cisco.com/web/about/security/psirt/security_vulnerability_policy.html