

Webex App Security

Cloud Collaboration Security Technical Paper

November 2022

Contents

1. Webex App Introduction	3
2. Architecture	5
3. Webex App Login	11
4. Encryption of Data in Transit	16
5. Data at Rest Protection	19
6. Webex Advanced Collaboration Features and Your Privacy	21
7. Networking	22
8. Privacy	24
9. Pro Pack and Extended Security Pack	24
10. Webex Control Hub – Security Features for the Webex App	25
11. Services	27
12. Conclusion	31
13. How to Buy	31
14. For More Information	31
15. Appendix	31



Webex is a cloud collaboration platform that provides messaging, calling, and meeting features. The Webex App is a client application that connects to this platform and provides a comprehensive tool for teamwork. Users can send messages, share files, make calls, and meet with different teams, all in one place. This security technical paper provides an overview of the security features of Webex App running on Windows, Mac, iOS, Android, Linux, ChromeOS, and Web.

1. Webex App Introduction

Many cloud collaboration providers refer to a plethora of security features when it comes to securing the cloud and protecting customer data. Vendors often refer to architectural features such as “encryption of data in-transit” and “encryption of data at rest” as underpinning security mechanisms for their service. But what do terms such as these really mean when it comes to securing cloud collaboration services and the devices that use them?

This paper provides an in-depth description of how the Webex App desktop, mobile and web applications are secured. Many aspects of security for the Webex App are covered including:

- Secure application onboarding
- Secure upgrades
- Authenticating Webex cloud services
- User authentication
- Secure media
- Secure data storage
- Securely traversing the Enterprise network edge

- Media transmission and application behavior
- Administrative security and compliance controls

This paper does not cover the following topics in detail:

- End-to-end content encryption
- Encryption keys and the Key Management Service
- Webex platform and service security

For information on the above topics, see the [Webex Messaging Security](#) paper.

Webex App security

The Webex App is a downloadable software image that provides voice, video, and messaging services to its users. These capabilities are delivered with the following workloads:

- Webex Meetings – Webex Meetings offers secure, integrated audio, video, and content sharing from the Webex App. Webex Meetings enable many people to gather from anywhere to collaborate using intelligent features such as noise removal, Webex Assistant, real-time translation, and People Insights.
- Webex Messaging – Webex Messaging provides secure persistent messaging and file sharing with the Webex App. Webex Messaging enables real-time collaboration through messages, files, reactions, emojis, and images with easy escalation to 1:1 calls or group meetings.
- Webex Calling – Webex Calling is a cloud calling solution that delivers secure enterprise-grade calling to both devices and the Webex App. The Webex App can leverage native Webex calling or enterprise Directory Number (DN) calling via Webex Calling or Unified Communications Manager (Unified CM) on-premises calling for hybrid integrations.

To secure the Webex App, Cisco uses several best practices and methodologies, including:

- Authenticated image files
- Encryption of data at rest
- Encryption of data in transit
- Secure software development using Cisco's Secure Development Lifecycle (CSDL)
- Controlled security feature implementation using Cisco's Product Security Baselines (PSB)
- User authentication using Identity Providers (IdPs) that support Single Sign On (SSO) using version 2 of the Security Assertion Mark-up Language (SAML) protocol.
- User authorization using OAuth2
- Security and compliance features configured in Webex Control Hub, the Webex App administrative portal

For more information on CSDL and PSB refer to the Processes section within the Cisco Trust Center [Trustworthy Solutions](#) area.

The remainder of this document discusses many of these topics in-depth and provides details on how Webex App security features are implemented.

Supported Oses and platforms

The Webex App uses HTTP as HTTP over TLS (HTTPS) and Secure Web Sockets (WSS) over TLS for REST based signaling, and SRTP (transported over UDP/TCP/TLS) for media. The Webex App is available for the following platforms:

- Windows
- Mac
- iOS
- Android
- Linux
- ChromeOS
- Web browser (HTML5 and WebRTC)

The Webex App can operate as a hybrid application when integrated with Unified CM calling or Webex Calling. In this case, in addition to HTTPS signaling to Webex, SIP is also used for signaling to Webex Calling or Unified CM for voice and video communications.

This document focuses on the implementation of security in the Webex App, although much of the discussion on security can also be applied to the Webex App operating as a hybrid application.

2. Architecture

Webex Solution Architecture

As shown in Figure 1, the Webex solution is based on a cloud-hosted microservice architecture. Each microservice is responsible for a specific functionality in the Webex App. For example, the directory search service in the Webex App is delivered via a microservice. Microservices are hosted in a mixture of Cisco and Public cloud service datacenters. Webex organizations can be hosted in location specific datacenters depending on the customer needs. The Webex App can also connect to 3rd party services for specific functions such as Enterprise Content Management (ECM) or Embedded Apps. If enabled by a Control Hub administrator, the Webex App can also connect to on-premises hosted services such as Unified CM or Video Mesh. Figure 1 provides an example of how the Webex App connects to multiple microservices as well as on-premises hosted services. Note: The figure does not reflect the actual specific breakdown of Webex microservices.

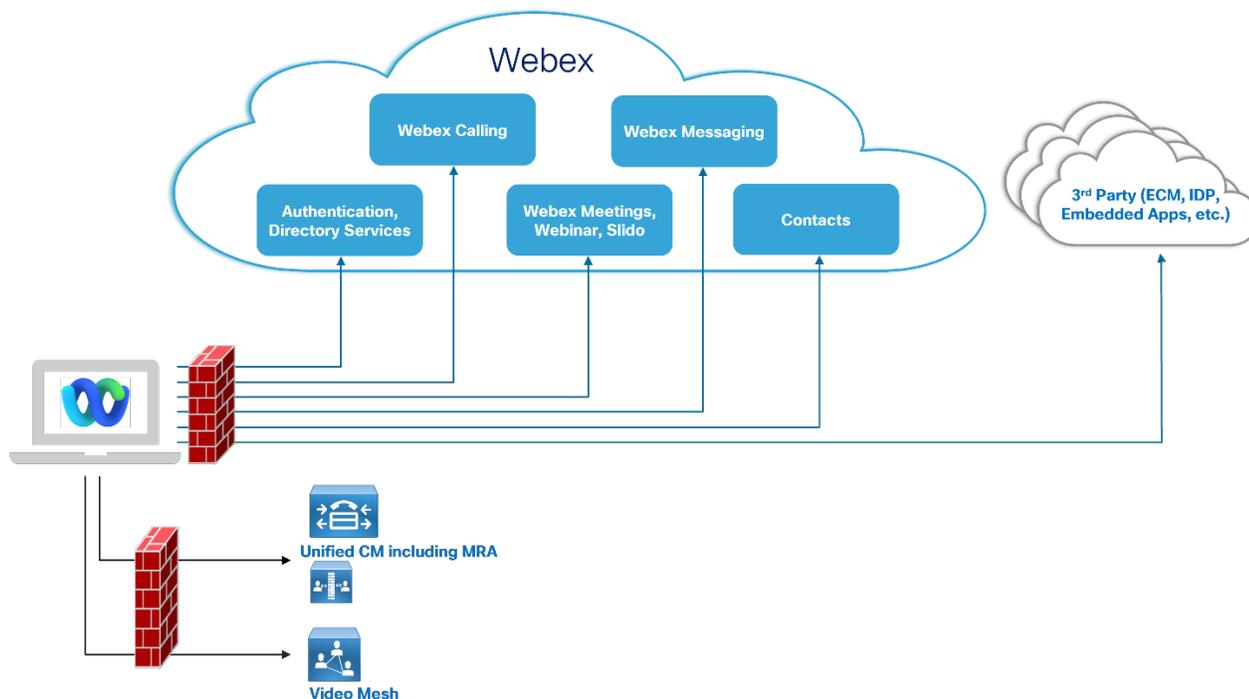


Figure 1. Webex Solution Architecture

Webex App

The Webex App is a modular application that can be deployed on various desktop and mobile operating systems. The application is built in a layered approach. Essentially there are three layers to the application as shown in Figure 2:

- Network / Platform Layer
- Unified Client Framework (UCF) Layer
- User Interface Layer

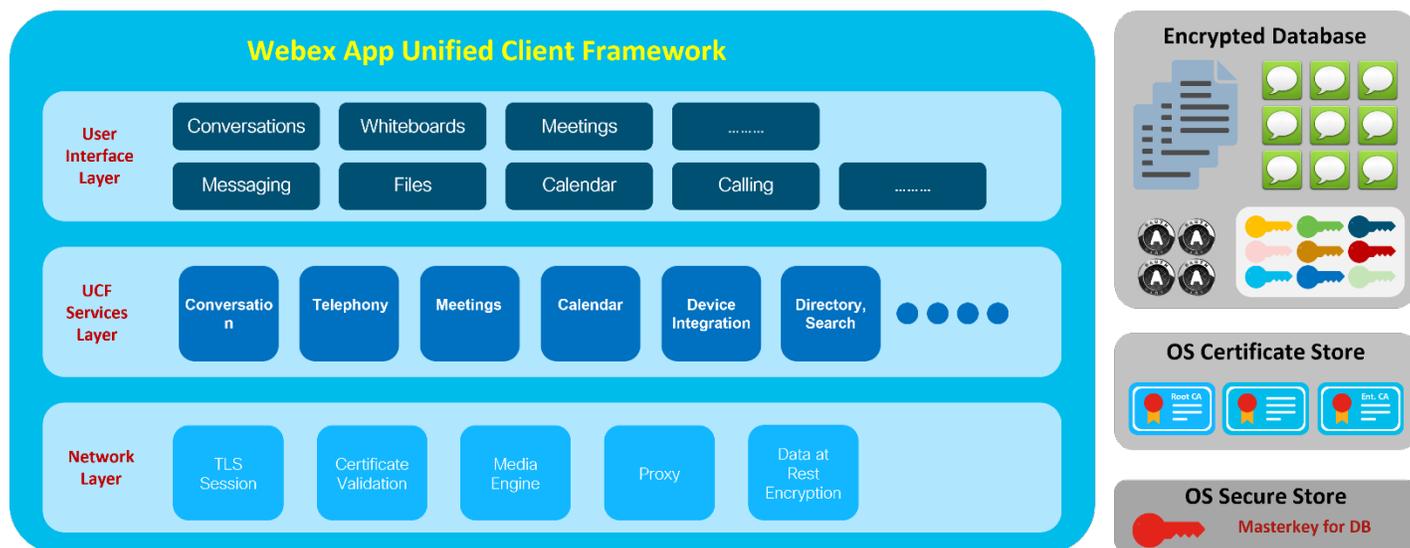


Figure 2. Webex App Layered Architecture

Network / Platform Layer

The Network / Platform Layer is built specifically for each supported Webex App platform (for example, Windows or Mac). This layer provides a range of network and platform services to the Webex App including:

- TLS Session establishment – HTTPS, WSS, SIP over TLS, Secure Media (SRTP)
- Certificate validation – The Webex App utilizes the platform certificate store to check if presented certificates are trusted.
- Platform encryption service – The Webex App utilizes the OS encryption service to encrypt any files that will be stored on the device
- Media engine – UDP/TCP/TLS media transport, media encryption/decryption
- TLS/HTTP proxy services, for example, proxy server address acquisition, proxy authentication

Unified Client Framework Layer

The Unified Client Framework (UCF) Layer is a layer common to each supported Webex App platform. This layer provides access to any available Webex services/components. For example, SIP engine used for Webex Calling is built into the UCF layer. UCF services include

- Conversation Service – service which provides access to Webex Messaging service
- Telephony Service – Webex Calling, Call On Webex, and Unified CM based calling service functionality
- Meetings Service – service which provides access to Webex Meetings service
- Device Integration – service which allows the Webex App to wirelessly pair and connect to a Cisco Video endpoint
- Directory and Search – service which allows a user initiate search from the User Interface Layer, as well as providing access to Webex directory services

User Interface Layer

The user interface layer provides the visual front end for the Webex App. This layer is responsible for rendering dialogues and orchestrating end user workflows such as creating new spaces, switching between spaces, pop out video windows, images, hover over effects, etc. The user interface layer also implements accessibility requirements, localization, multiple appearance themes including dark and light mode, and operating system or platform-specific functionality. The implementation of the user interface layer varies for each supported Webex App platform.

Downloading the Application

The Webex App can be downloaded from various locations depending on platform.

For desktop (Windows, macOS, and Linux), the Webex App can be downloaded from <https://www.webex.com/downloads.html>

For mobile operating systems (iOS, Android and ChromeOS), the Webex App can be downloaded from the applicable App stores:

- <https://apps.apple.com/us/app/webex/id8333967564>
- https://play.google.com/store/apps/details?id=com.cisco.wx2.android&hl=en_IE&gl=US

It is also possible to be provided with the mobile app installer packages in the case where the app will be wrapped and installed using a Mobile Device Management (MDM) solution.

App Signing

The Webex App image is signed with a Public CA certificate.

As shown in step 1 of Figure 3, prior to uploading the Webex App image to webex.com or to the relevant app stores, Cisco uses a CA-signed software publishing certificate to digitally sign the software image. Cisco uses the code signing infrastructure of each platform vendor (Microsoft/Google/Apple...) to co-sign a PKCS #7 signed data object file containing the signed Webex App image, digital signature, and software publishing certificate (step 2) before pushing to webex.com or relevant app store (step 3).

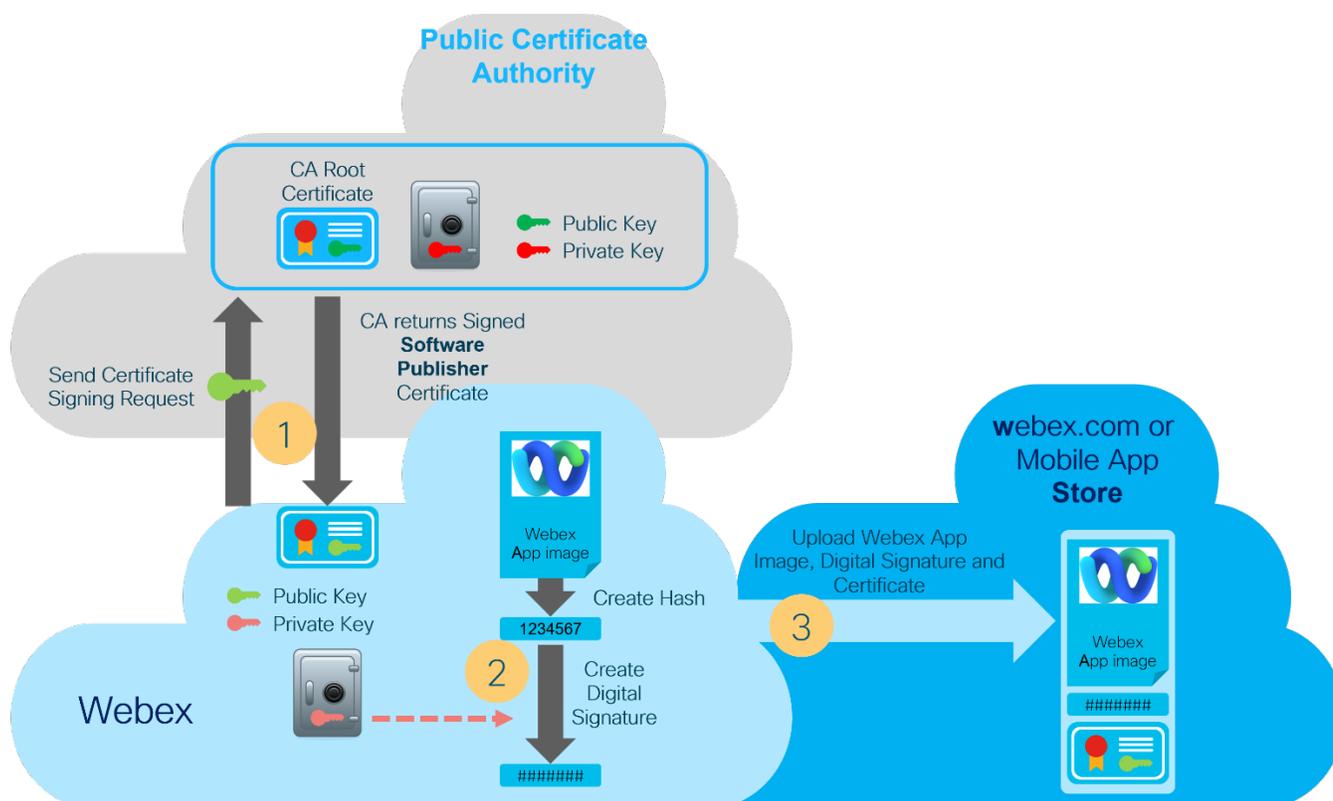


Figure 3. Webex App Image Signing

Figure 4 below details the process by which the Webex App software image integrity is verified by the local OS before installing. When the Webex App image install process begins, the platform operating system decrypts the digital signature using the CA-signed software publisher certificate's public key. The platform then creates a hash of the image locally. The platform operating system verifies the digital signature PKCS #7 signed data object file. It compares the hash in the received digital signature to the hash that was generated locally. If the two hashes match, then the integrity of the received file is verified and can be installed by the platform OS. If the two hashes do not match, the application installation fails.

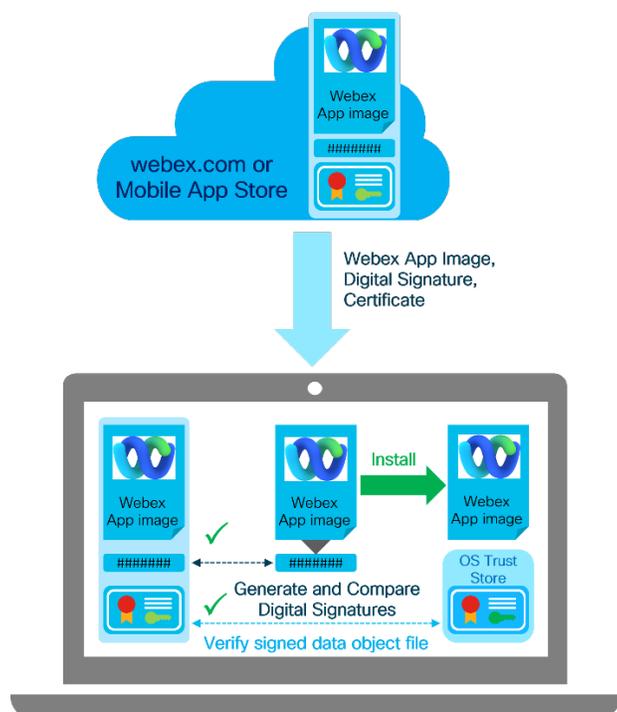


Figure 4. Webex App Image Verification by Local OS

Installation of the Webex App

The installation process of the Webex App can vary on a per platform basis. For all platforms excluding Linux, the Webex App can be installed without local machine admin rights. For large scale deployments Webex App installation can be administrator controlled.

Windows

On the Windows operating system, the Webex App can be installed in two ways.

Admin Install

This approach is typically used when the Webex App is being mass deployed to multiple devices via a deployment tool (for example, Microsoft Endpoint Manager (formerly SCCM)). Admin install can be performed by downloading the Webex.msi file from webex.com and specifying the ALLUSERS=1 installer switch at install time. This will install the app to the “C:\Program Files” directory. This requires admin rights on the local machine.

For more info on admin install, see the [Webex App Installation and automatic upgrade](#) article.

User Install

This approach is typically used when a user downloads the Webex App from webex.com. The user can download the installer file (an .exe file) from webex.com and run through the installation process. The Webex App will be installed to the user profile directory – typically the location is “C:\Users\userID\AppData\Local\CiscoSparkLauncher”. This install type does not require admin rights on the local machine.

macOS

On macOS, the Webex.dmg file can be used to install the Webex App. The dmg file can be run locally or pushed via a deployment tool. By default, the Webex App is installed to the Applications folder, although the user can choose another location. Installation of the Webex App on macOS does not require local machine admin rights.

Linux

The Webex App can be installed on machines running Ubuntu and Red Hat Linux distributions. The app can be installed using platform tools, for example, apt-get or yum. Installation of the app on these platforms requires local machine admin rights.

iOS and Android

The Webex App for iOS and Android can be downloaded and installed from the Apple App Store and the Google Play Store respectively.

Enterprise Mobile Management (EMM) software may be used to manage the Webex App on mobile devices. For more information refer to [Managing the Webex App on Mobile Devices](#).

The Webex App for ChromeOS can be downloaded from Google Play Store. Essentially, the Webex App from ChromeOS is the Webex App Android build, which can be run on ChromeOS.

Software update

The Webex App and Webex services use a continuous development model of iterative software development to deliver new features and fix issues. Typically, a new release of the app is delivered each month. The app versioning is based on the year, month and build number

Here is the application version syntax:

Webex App version **aa.bb.c.ccccc**

aa denotes the year (e.g., 42 denotes year 2022)

bb denotes the month (e.g., 9 denotes September)

c.ccccc denotes the build number (e.g., 0.23494)

For example, the September 2022 release of Webex App for Windows was version **42.9.0.23494**

The Webex App for desktop platforms (Windows, Mac, Linux) will periodically query a Webex update service to check if there is a later version of the app available. Once a later version is available, the update package will be downloaded to the local machine. The user will be presented with a restart app button. Once the app is restarted the newly downloaded update package will be loaded. Note: the update process does not perform a fresh install of the app. The update process does not require local machine admin rights, even if the initial Webex App install was performed by an admin user.

The Webex App desktop software update cadence can be controlled by the Control Hub administrator. By default, the Webex App will update to the latest version, however, the administrator can define a custom update schedule for less frequent and more predictable software updates.

The Webex App for mobile platforms (iOS, Android, ChromeOS) will be updated based on the respective App Store and the local device settings.

3. Webex App Login

Authentication

Webex App Sign-in

Overview

On initial start-up, the Webex App will establish TLS sessions with Webex and verify the authenticity of the Webex services, then the user will be prompted for their email address. The user's email address is used to determine to which Webex organization the user belongs. If the organization is using the Webex Identity service to authenticate a consumer user, for example user@gmail.com, the user is prompted for their password. If the organization is using Single Sign-On (SSO) with an Identity Provider (IdP), the SSO sign in page is presented and the user is challenged for authentication based on what is configured on the IdP.

Once authenticated, the Webex Authorization service uses the OAuth2 protocol to grant the Webex App controlled access to Webex services. The Authorization service delivers access and refresh tokens to the Webex App. The Access token contains scopes that define which Webex services the user is authorized to use. Access and refresh tokens are securely cached by the Webex App and used as a proof of authentication, so that the user does not have to sign in every time the application launches/reconnects to a service. The amount of time the access token and refresh token are valid and the ability to automatically extend the refresh tokens can be configured separately for mobile and desktop apps by the administrator through Webex Control Hub. For more details, refer to the [Token policy settings in Control Hub](#) article.

Webex App Authentication Methods

Authentication using the Webex Identity Service

Primarily used by consumer users or small to medium businesses without an IdP, the Webex Identity service offers a secure username and password-based authentication service over TLS.

Authentication using SAML based SSO with an Enterprise, or Cloud-based IdP

The Webex App supports authentication with Enterprise (on-premises or cloud) IdPs that support SSO using Security Assertion Markup Language version 2.0 (SAML 2.0). For more information on tested SSO IdPs and SSO set up details, see the [Single Sign-On Integration in Control Hub](#) article.

Using SSO with an IdP provides several benefits for users and administrators of the Webex App:

- Users can use the same credentials to sign-in to multiple applications, avoiding the requirement to remember multiple sets of credentials for multiple applications.
- If users are authenticated by their enterprise or cloud directory, for example, Active Directory (AD), Azure AD, or LDAP directories, then the enterprise can decide on the degree of password complexity, password lifetime and password re-use policies.
- SSO can be combined with Multi-Factor Authentication, where the user is required to provide more than one form of proof of authenticity in addition to a username and password, for example:
 - Something the user has, such as a security token, a smartphone App that generates a one-time passcode, a code contained in SMS text message.
 - A physical characteristic of the user for biometric security, for example, a fingerprint scan or voice or face recognition.

- The location of the user, for example, are they physically connected to the enterprise network or using a VPN connection?

Authorization

Webex App Authorization

The Webex App uses the Open Authorization protocol version 2.0 (OAuth 2.0) to provide an authenticated user's application with access to authorized Webex services without the need to share the user's credentials with the service. The Webex App also uses OAuth to grant third-party application integrations with authorized access to Webex services.

OAuth Tokens

Once the user has authenticated, the Webex App is provided with an access token and refresh token by the Webex authorization service.

Access and refresh tokens use the JSON Web Token (JWT) format with JSON Web Signature (JWS) and JSON Web Encryption (JWE). For more information, see the following:

- JWT: [RFC 7519](#)
- JWS: [RFC 7515](#)
- JWE: [RFC 7516](#)

The Access Token

The encrypted and signed access token uses the JSON Web Token (JWT) format to encapsulate information about the organization. For example, the organization ID (Org ID), user (UUID), device (Client ID) and the scope of Webex services to which the user has been granted access. Webex service access tokens have a lifetime of twelve hours by default and can be changed by the administrator via Webex Control Hub.

OAuth Access Token Scopes

The scopes in a Webex App access token define what the application is authorized to do on behalf of the user, for example:

- Read messages
- Write messages
- Read space memberships
- Write space memberships

The combination of user scopes, the user's organization, user ID, and device ID in the access token are used to authorize a user's access to Webex resources.

The Refresh Token

The refresh token also uses the JWT format and encapsulates the information required to request a new access token from the Webex authorization service, without prompting the user to re-authenticate. Access tokens are typically refreshed when they reach 75% of their lifetime. Using the default TTL of 12 hours, this means the Webex App will attempt to refresh an access token 9 hours after it was generated. A new refresh token is delivered with the access token if auto-extend of the refresh token is enabled in Control Hub. The Webex App refresh tokens have a default lifetime of 60 days. If the Webex App doesn't refresh its access token before the

refresh token expires, for example, when the Webex App is offline for a long period, then the user must re-authenticate to regain access to the Webex service.

Note: By default, the Webex App refresh token is not automatically renewed every time its access token is renewed, so the user would have to authenticate again when the refresh token expires. This can be modified by the administrator via Webex Control Hub for the Webex App running on mobile devices and on desktop devices by enabling auto-extend of the refresh token.

OAuth Authorization Code Grant and SAML based SSO Authentication

Before using any Webex service, the application must present its access token to the service, which decrypts the token and verifies that the scopes that it contains match some or all of the scopes provided by the service. If the access token's scopes do not match that of the service, access is denied. If the Webex App attempts to communicate with a Webex service without an access token, the service redirects the application to Webex authorization service, where the application can request an access token with the scopes that it requires and a refresh token.

OAuth 2.0 defines several methods by which an application can request and obtain tokens. The primary and considered to be most secure method used by the Webex App and Webex services is the Authorization Code Grant Flow defined in detail in [RFC 6749](#) and summarized below. With the Webex App, the user is requesting an access token so that their Webex App is granted access to Webex services, unlike other commonly described OAuth grant flows where the user is granting a third-party application with access to a resource owned by the user.

OAuth 2.0 is an industry standard for authorization; prior to granting access and refresh tokens to an application, the application user must be authenticated. The Webex App supports authenticating users using either the Webex Identity Service or using an Enterprise IdP and Identity Store, for example, AD, Azure AD, LDAP and so on.

The flows in the diagrams below show the OAuth Authorization Code grant flow combined with SSO-based authentication using SAML with an enterprise IdP.

In the initial request (1) shown in Figure 5, the user has not authenticated and does not have an access token to present to the Webex service. The service redirects the Webex App to the authorization service (2). The Webex App makes an authorization code grant request to the authorization service (3). This authorization code grant request contains the user's email address, client ID and a list of scopes that the access token will need to access Webex services. The request from the application does not contain a refresh token, and the authorization service then redirects the application to the Webex Identity sign in service (4).



Figure 5. Webex App: Webex Service Request Prior to Authorization and Authentication

In Figure 6 the Webex App sends a sign in request with the user's e-mail address to the Webex Identity service (5). The Webex Identity service determines that the organization to which this user belongs has SSO enabled using an external IdP and redirects the application to the IdP for SAML-based authentication (6). The IdP prompts the user for credentials which the user enters for authentication (7) with an enterprise identity store such as AD, Azure AD, LDAP and so on. When the user has been authenticated, the IdP returns a SAML assertion to the Webex App and redirects the application back to Webex (8).

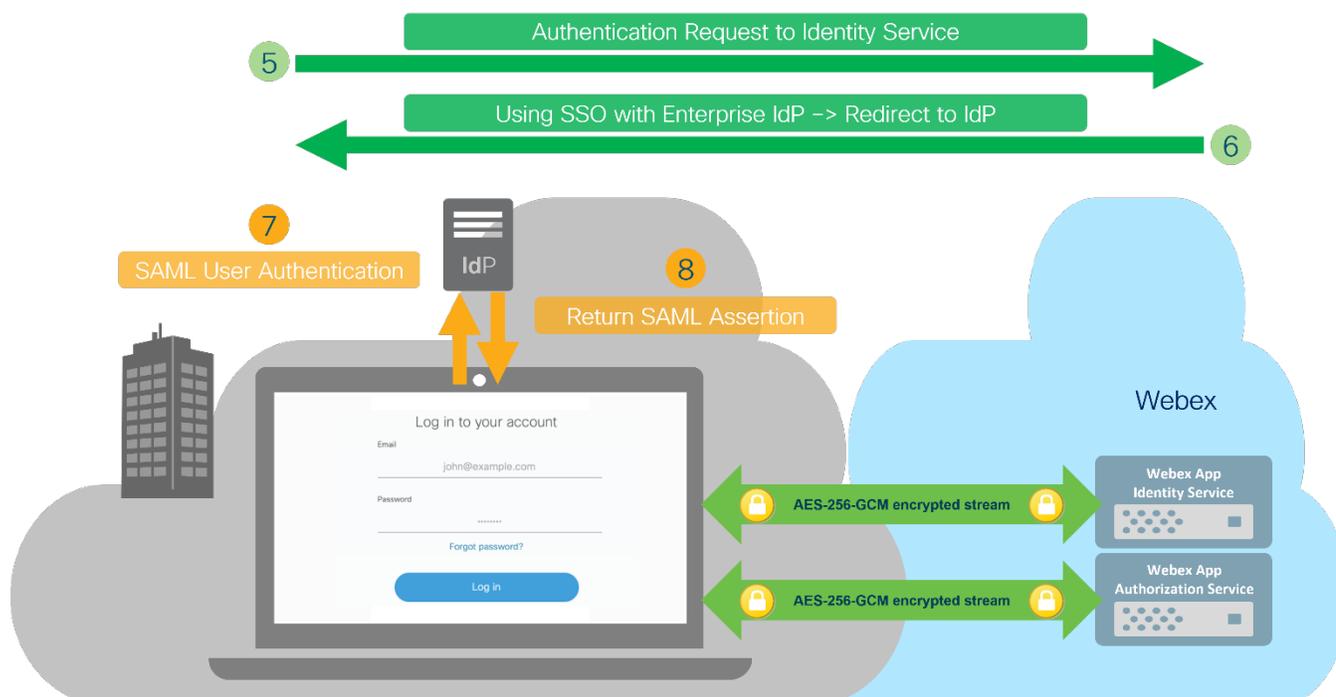


Figure 6. SSO-based User Authentication using SAML to an Enterprise IdP

In Figure 7, the Webex App returns the SAML assertion for the authenticated user to Webex Identity service (9). The Webex Identity service validates the user's SAML assertion and redirects the application to the authorization service (10). The Webex App sends the SAML assertion with User ID to the authorization service (11) which validates the SAML assertion and returns an authorization code to the Webex App (12).

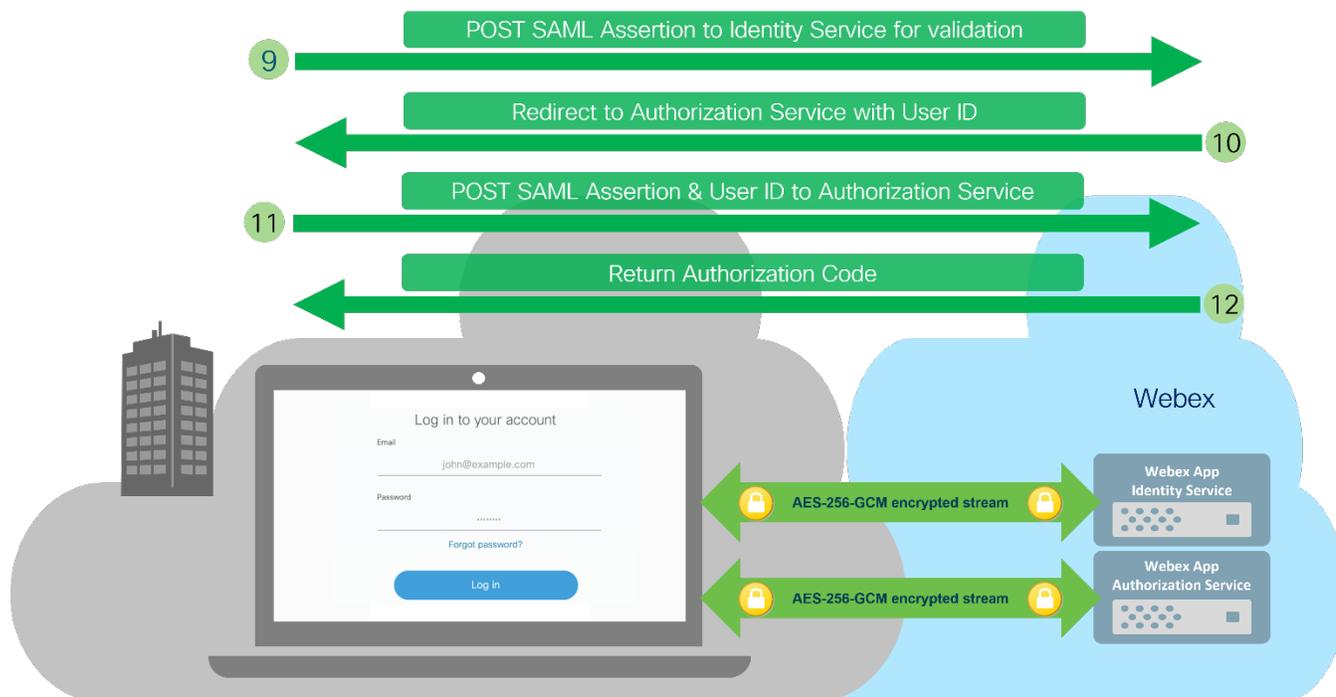


Figure 7. User Authenticated: SAML Assertion Validated by Webex Identity Service with Redirect to the Authorization Service for the Authorization Code Grant Flow

Figure 8 shows the final step in the OAuth Authorization Code Grant flow. The Webex App sends the authorization code, client ID and client secret to the authorization service (13) which validates these values and then returns an encrypted and signed access token and, if enabled in Control Hub, a refresh token (14). The Webex App can then present the access token as proof of authorization (and implicit proof of authentication) to use a Webex service (15).

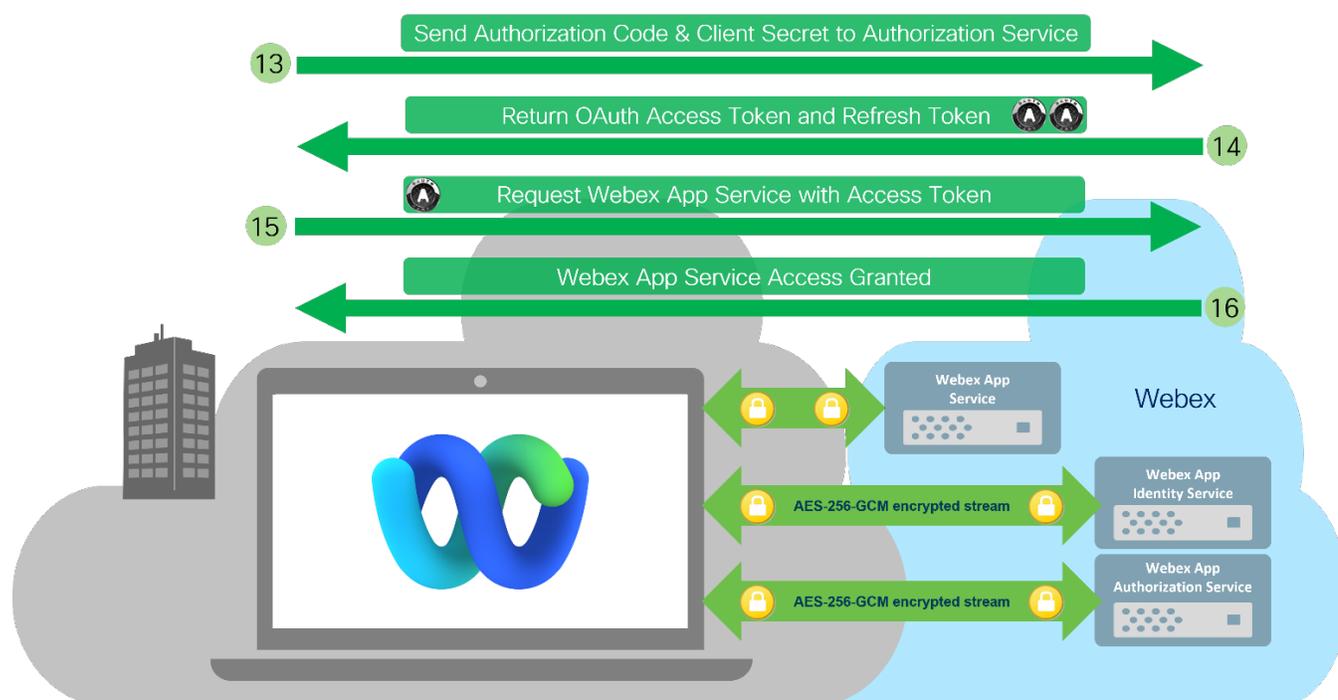


Figure 8. Authorization Code Grant Flow Completion: OAuth Tokens sent to the Webex App for Webex Service Access

When the access token's lifetime reaches 75% of its validity period (9 hours by default), the Webex App sends its refresh token to Webex authorization service to request a new access token and refresh token. The OAuth access and refresh tokens are securely stored in the application platform's OS and can be revoked by a Webex App administrator through the Reset Access option in Webex Control Hub or through the [Authorizations API](#).

4. Encryption of Data in Transit

Webex App Signaling Encryption

The Webex App makes multiple TLS/HTTPS connections to Webex. These connections are outbound only and some connections are upgraded from HTTPS to bi-directional Secure WebSocket (WSS) connections.

The use of Transport Layer Security (TLS) to provide encryption of data in transit is a common industry practice. For general information on TLS, refer to the [Appendix](#).

TLS signaling connections from the Webex App to Webex services use TLS version 1.2 or 1.3. The cipher selection is based on the Webex server TLS preference.

Using either TLS 1.2 or 1.3, Webex prefers ciphers suites using:

- ECDHE for key negotiation
- RSA-based certificates (3072-bit key size)
- SHA2 authentication (SHA384 or SHA256)
- Strong encryption ciphers using 128 or 256 bits (for example, AES_256_GCM, AES_128_GCM, and CHACHA20_POLY1305)

For example:

- TLS 1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

- TLS 1.3: TLS_AES_256_GCM_SHA384

For more information on the signaling encryption specific to Webex Messaging, Webex Meetings, or Webex Calling, refer to the [Services](#) section in this document.

For general information on TLS version and cipher suites, refer to the [Appendix](#).

Webex App Certificate Validation

As part of the TLS handshake protocol, once the Client Hello and Server Hello messages have been exchanged and a cipher suite negotiated. The server sends the Webex App the CA signed server certificate, intermediate certificate, and CA root certificate. For each certificate, the Webex App verifies and validates the digital signature, issuer, validity period, revocation status (using Online Certificate Status Protocol (OCSP) or Certificate Revocation List (CRL)), key usage extensions, and server hostname.

Webex App Media Encryption

The Webex App uses real-time media for audio, video, and content sharing streams. Typically, media from any Webex App transits from the user's location to Webex media nodes (except for Unified CM 1:1 calling with an enterprise number), where the streams are switched and distributed. This is true, for example, with Webex Calling, Call on Webex to one other person, and multiparty calls. Optionally, instead of cloud-based Webex media nodes, on-premises Webex Edge Video Mesh Nodes (VMNs) may also be deployed to mix and distribute media locally.

Cisco secures all Webex App media streams using the Secure Real-Time Transport Protocol (SRTP), described in [RFC 3711](#). Voice and Video codecs, the media encryption cipher and encryption keys are securely negotiated over HTTPS, using SDES ([RFC 4568](#)).

With Zero Trust end-to-end encrypted Webex Meetings, the media is first encrypted based on [SFrame](#) and then sent on the encrypted SRTP channel. The media encryption key for the end-to-end encrypted meeting that is used for SFrame is derived from the exchange of Message Layer Security (MLS) key packages between the participants. For more information, see the [Meeting](#) section in this document.

Strong cipher suites are negotiated between the Webex App and Webex. Typically, AEAD_AES_256_GCM cipher suite is negotiated.

Figure 9 shows the Webex App encrypted media using the AEAD-AES-256-GCM cipher suite.

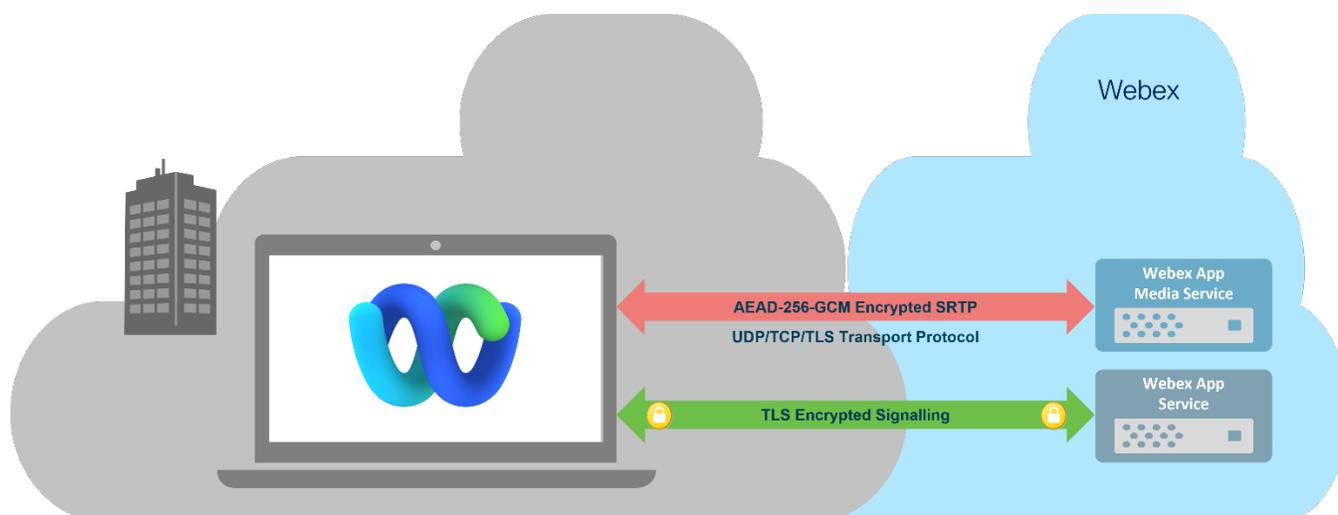


Figure 9. Encrypted Webex App Media and Signaling

In line with [RFC 3550](#) RTP – A Transport Protocol for Real -Time Applications, Cisco uses UDP with destination port 5004 as the preferred transport protocol for the Webex App voice and video media streams. Webex Messaging uses UDP with destination port 5004, Webex Meeting uses UDP with destination ports 5004 and 9000, and Webex Calling uses UDP with a range of destination ports (refer to [Table 2 in the Calling section](#)).

With messaging, the Webex App also supports TCP with destination port 5004 as a fallback media transport protocol. However, we do not recommend TCP as a transport protocol for voice and video media streams, as TCP is connection oriented and designed to reliably deliver correctly ordered data to upper layer protocols. Using TCP, the sender retransmits lost packets until they are acknowledged, and the receiver buffers the packet stream until the lost packets are recovered. For media streams, this behavior manifests itself as increased latency or jitter and in turn this affects the media quality experienced by the call's participants.

With meetings and messaging, the Webex App also supports TLS (secured TCP) as a last resort option for media transport. Using TLS with destination port 443, can also mean that this Webex App media traffic will need to pass through an enterprise's proxy server to reach Webex media servers. Since proxy servers are primarily designed to intercept and forward HTTP-based web traffic, media quality can be impacted if the proxy server reaches its performance threshold and delays or drops packets when processing large numbers of high bandwidth media streams.

The Webex App media flows in both directions using a symmetric inside-initiated, 5-tuple (source IP address, destination IP address, source port, destination port, protocol) stream outbound to Webex.

The Webex App also uses STUN ([RFC 5389](#)) for firewall traversal and media node reachability testing.

The Webex App uses a media node discovery process at start up, when network connections change, and periodically to determine the reachability of media node clusters available to their organization (cloud media nodes and on-premises Video Mesh Nodes). During call establishment, the Webex App sends its reachability report to Webex. Based on the available transport protocol (UDP/TCP/TLS), Round Trip Times (RTT), and resource availability, Webex determines the media node that will be used by each app.

As shown in Figure 10, media cascade connections are established if multiple media servers are used in a call.

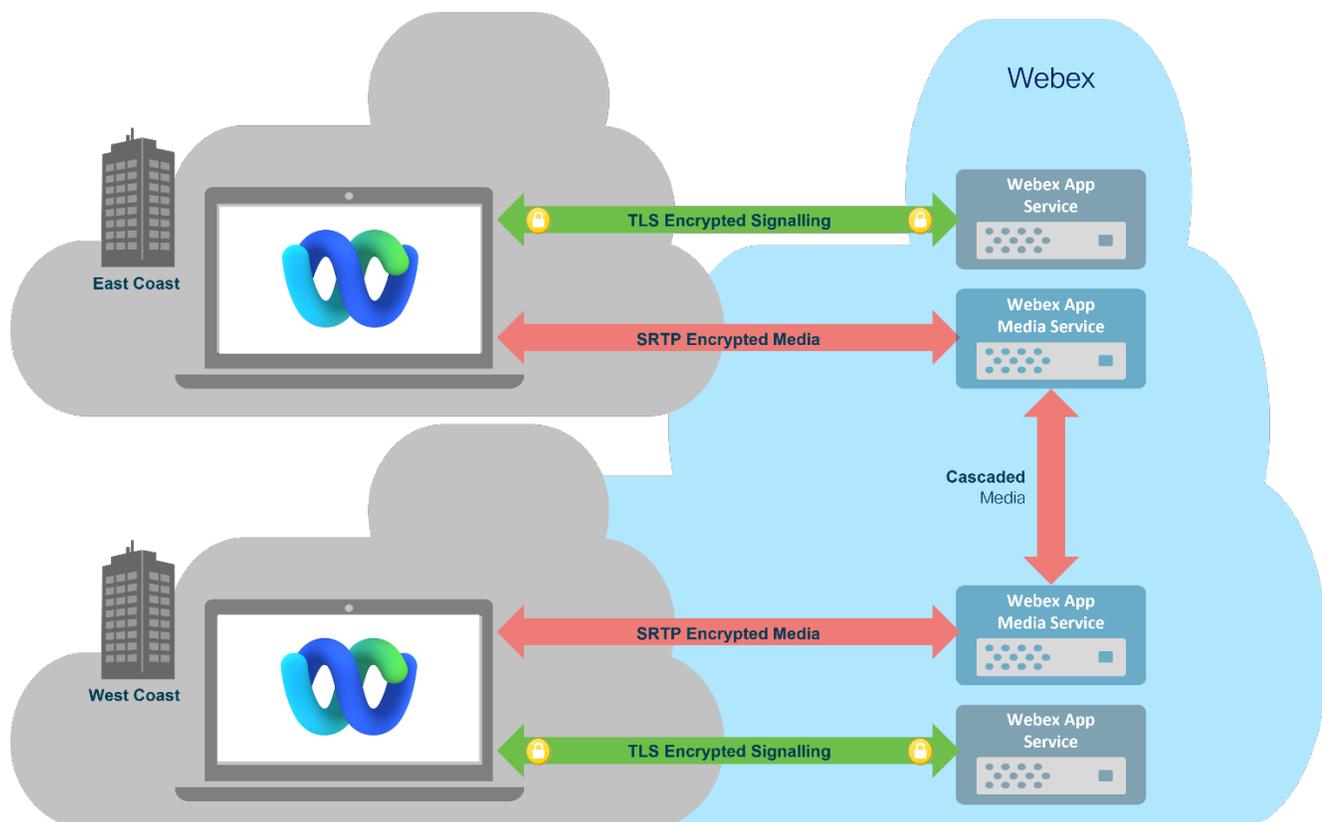


Figure 10. Media and Signaling: 1:1 Call between Webex Apps

For more information on the media encryption specific to Webex Messaging, Webex Meetings, or Webex Calling, refer to the [Services](#) section in this document.

For more general information on TLS version and cipher suites, refer to the [Appendix](#).

5. Data at Rest Protection

Windows, macOS, Linux, ChromeOS, iOS & Android Applications – Data Storage

Encryption of data at rest applies not only to content stored in Webex, but also to content stored by the Webex App. The following content is securely stored by the Webex App for Windows, macOS, Linux, ChromeOS, iOS and Android:

- Messages
- Preview files, files converted to Portable Network Graphics (PNG) file format
- Space encryption keys
- Profile pictures
- Space details
- Meeting details
- Whiteboard files
- Directory cache
- Config cache
- Logs. Note that some logs may not be stored encrypted, but those logs do not have sensitive information nor personal identifiable information (PII).

- Passwords, if any. Note that passwords are not stored except in a deployment where the users are configured for Unified Communications Manager (Unified CM) Calling, and where Unified CM is not SSO nor “OAuth refresh token flow” enabled. In this deployment, passwords are encrypted and stored on the local device.
- Access and refresh OAuth tokens.

The Webex App on desktop and mobile devices store this content in an SQLite database, except for the access and refresh OAuth tokens which are stored encrypted in the OS platform secure store. The SQLite database is encrypted using the AES-256-OFB algorithm. Like the access/refresh OAuth tokens, the master key for the SQLite database is encrypted and stored in the OS secure store. For example, Windows Data Protect API, macOS/iOS Secure Enclave and Keychain, and Android Keystore.

This encryption mechanism depends on the platform where the Webex App is installed.

Figure 11 depicts how data is encrypted at rest.

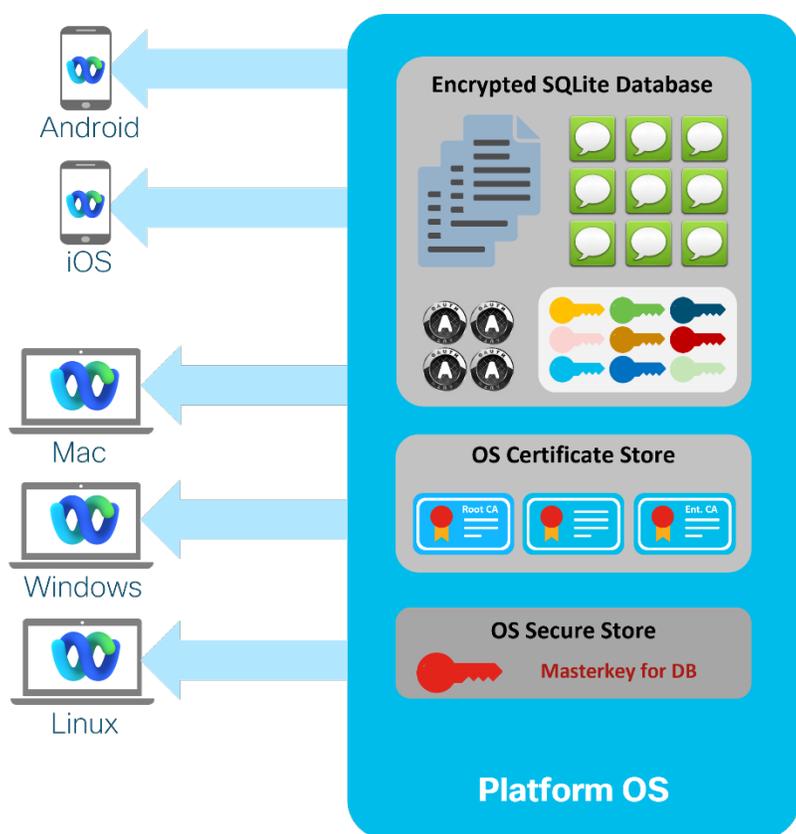


Figure 11. Webex App: Encryption of Data at Rest

Table 1 shows the encryption mechanisms for the master key and for the OAuth tokens depending on the platform where the Webex App is installed.

Table 1. Encryption of the Master Key and OAuth Tokens

PLATFORM	ENCRYPTION SOLUTION
Windows (64 and 32 bits)	Windows Data Protection API (DPAPI)

Apple macOS	Apple Mac Keychain
Apple iOS and iPadOS	Apple iOS Keychain
Android	Android Keystore
Linux	GNOME keyring or Secret Service API
ChromeOS	Android Keystore

The OAuth tokens are stored on the client. It is possible for the organization administrator to reset the access for a specific user from Webex Control Hub or via Authorization API. This revokes all OAuth access and refresh tokens owned by the user and deletes cached content. This option signs the user out of all Webex Apps and they must reauthenticate to connect again.

The end-user can also see a list of all the signed-in devices and select the devices that need to be signed out (Webex app settings/preferences > General > Recent sessions > show details). “End session” revokes the OAuth access and refresh tokens on the device and deletes cached content.

Files downloaded by the Webex App are decrypted prior to storage. The storage location of downloaded files is determined by the user, for example, the Windows Downloads folder.

Webex App for Web – Data Storage

Webex App for web (<https://web.webex.com>) does not permanently store content. Messages, files, encryption keys and tokens are deleted when the browser or browser tab is closed. One exception to this case is when the “Remember Me” option is selected by the user to bypass user authentication. In this case, the access and refresh tokens are stored and reused when the Webex App is relaunched in the browser.

6. Webex Advanced Collaboration Features and Your Privacy

Webex devices and the Webex App deliver far more than great audio and video for calls and meetings; we have developed a range of amazing [AI features](#) that can enhance and improve your experience when in a Webex call or Webex meeting:

- Background noise removal
- Optimize for my voice/all voices
- Music mode
- Gesture recognition
- Face recognition
- Language intelligence:
 - Webex Assistant
 - Closed captioning
 - Real time translation
 - Meeting transcription
- Room interpretation
- People presence detection

- Proximity Pairing

At first glance, one might assume that some of these advanced collaboration features leverage collaboration AI by silently streaming audio and video to Webex for processing. In fact, most of these advanced Webex features process audio and video locally. Webex only streams media to cloud media nodes when you are in a call or meeting.

For features and products using AI and ML, Cisco implements an overarching design principle to never retain unnecessary personally identifiable information (PII) from customer data.

For more details on how we use advanced collaboration features with Webex, refer to the [Webex Advanced Collaboration features Providing Amazing User Experiences while Protecting Privacy](#) article.

To find more information about Cisco and data privacy, see:

1. [Cisco online privacy statement](#)
2. [Cisco data protection and privacy](#)

7. Networking

Most enterprises implement multiple products and features to protect their internal networks and data, and to control external access. The Webex App supports the following enterprise network security features, protocols, and products:

- Network Access Control (802.1X)
- Firewall Traversal
- Proxy Server – Authentication, Traffic filtering and Inspection

These security features are discussed at a high level in this document. For more details on the Enterprise Network Requirements for Webex App services, including Webex App IP subnets for media and URLs for signaling to services see (and subscribe to) the [Network Requirements for Webex Services](#) article.

Firewall and Proxy Traversal

Most security conscious customers deploy both a firewall and proxy server to control access from applications and devices in their enterprise networks to the Internet and associated cloud services, such as the Webex App. Specific implementations may vary, but a common deployment forces all HTTP-based traffic through a proxy server allowing only HTTP traffic originating from the proxy server to traverse the firewall and reach the Internet. Other traffic types such as UDP or TCP-based media from the Webex App are not directed to the proxy server but traverse the firewall only except if the media falls back to port 443, then the media will go through the proxy (see Figure 12).

Note: All on-premises Webex Apps, devices, and hybrid services initiate outbound connections only to cloud-based Webex Services.

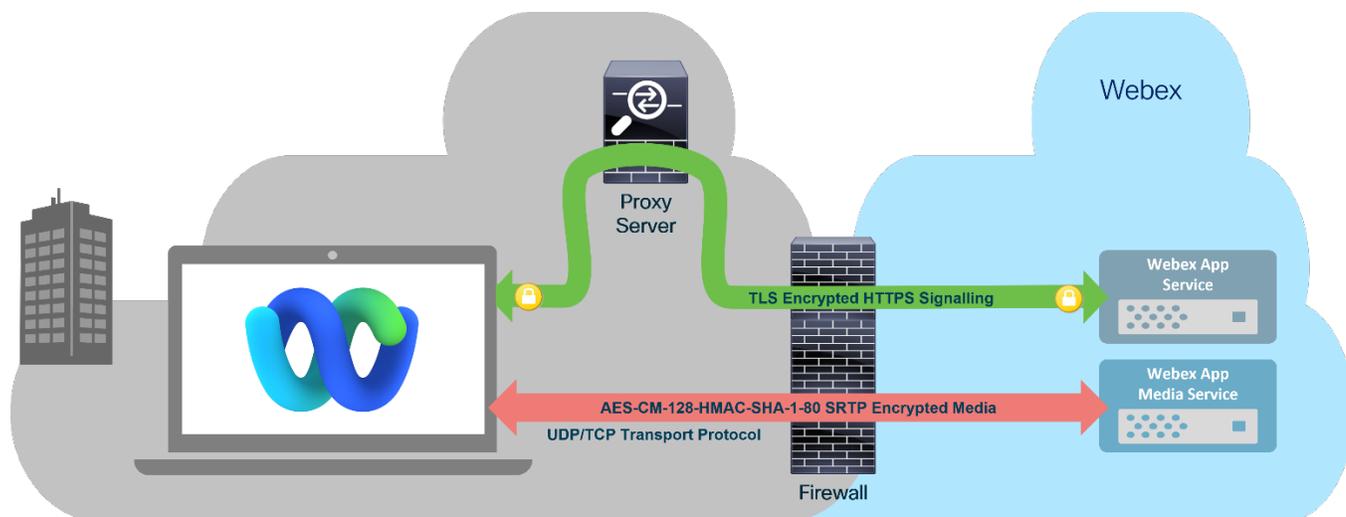


Figure 12. Typical Webex App Proxy and Firewall Traffic Flows

HTTP Proxy Traffic Inspection and Certificate Public Key Pinning

Traffic inspection by a proxy server, where the proxy decrypts, inspects and re-encrypts the TLS/HTTPS traffic traversing the server, is commonly used by security conscious customers. Traffic inspection by a proxy server has limited value for Webex App traffic, as decrypting and inspecting traffic only reveals signaling information. User generated messages and files are end-to-end encrypted by the Webex App. If you want to inspect and moderate user messages and files shared in Webex Messaging spaces, use a Data Loss Prevention (DLP) or Cloud Access Security Broker (CASB) application, such as Cisco Cloudlock.

As shown in Figure 13, when utilizing traffic inspection, the proxy server presents its Enterprise CA signed certificate, instead of the Webex service certificate, to the Webex App. This allows the proxy server to establish a TLS connection to the Webex App and to encrypt/decrypt and inspect the signaling. Similarly, signaling traffic from the proxy server to Webex can also be encrypted/decrypted and inspected.

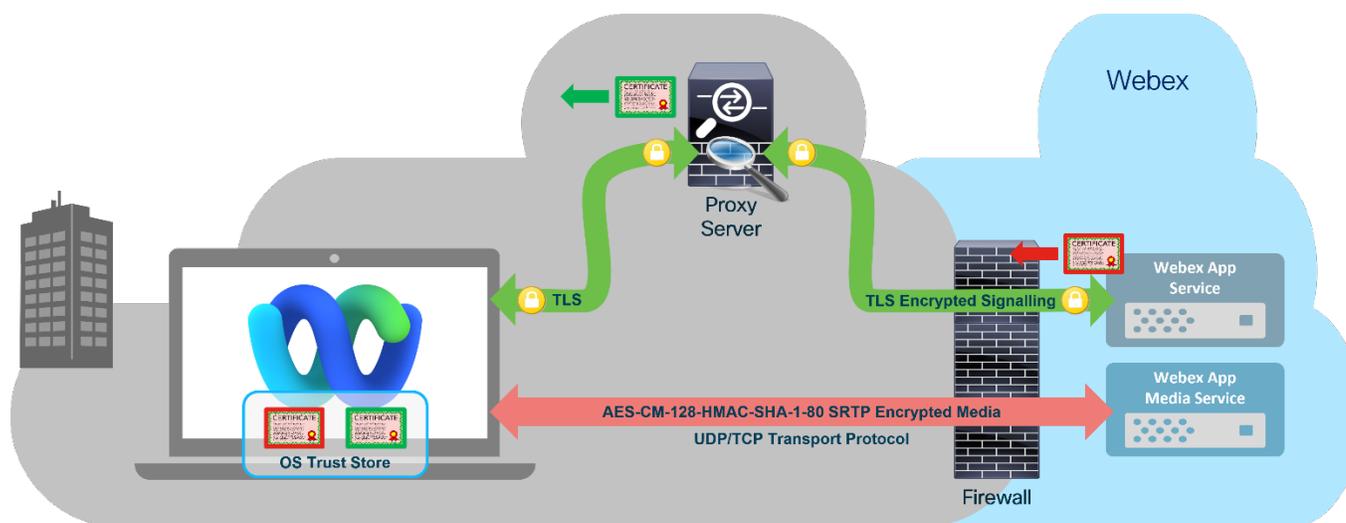


Figure 13. Proxy Server TLS Traffic Inspection

The Webex App implements certificate public key pinning by embedding a copy of the public keys of intermediate certs in the application software to dramatically decrease the risk of server impersonation. These

pinned certificate public keys are then used by the Webex App to verify that the certificates received in the TLS server certificate are the ones expected. Without certificate public key pinning, any publicly trusted CAs could issue a certificate that a bad actor could use to intercept Webex Messaging communications. With traffic inspection by the proxy server, the enterprise CA signed certificate will not match the certificate pins embedded in the application software. In this case, the Webex App searches the OS trust store for a certificate that matches the one received from the proxy server. If a matching certificate is found, the TLS connection from the Webex App to the proxy server is allowed to be established.

8. Privacy

Webex App and Privacy Data Handling

Cisco has implemented appropriate technical and organizational measures designed to secure personal and customer data from accidental loss and unauthorized access, use, alteration, and disclosure. Personal data includes both identifying information (first name, last name, user ID, and so on) and user generated content (meeting recordings, messages and files sent or received, and so on)

Additional information about collection and retention of personal and customer data by the Webex App is available in the [Webex App and Webex Messaging Privacy Data Sheet](#).

Support Logs

When a customer submits a request for support services or other troubleshooting to Cisco Technical Assistance Center (TAC), Cisco TAC may receive and process support data which could include personal data. Support data (logs, configuration and firmware files, core dumps, and so on) includes details related to a support incident, such as authentication information, information about the condition of the product or system, and registry data about software installations and hardware configurations. The [Cisco Technical Assistance \(TAC\) Service Delivery Privacy Data Sheet](#) describes Cisco's processing of such data.

Data Retention Policy

Personal, customer, and support data is retained by Cisco for varying periods of time based on the type of information. Support data is retained for the longest period of time in order to comply with Cisco business record audit policies. Customer and personal data are generally retained for shorter periods of time, but in the case of user generated content the administrator can set longer retention periods. For additional information about Webex App data retention, refer to the [Webex App and Webex Messaging Privacy Data Sheet](#).

9. Pro Pack and Extended Security Pack

Advanced security capabilities and integrations are available to administrators of a Webex organization via the following premium offers and service add-ons:

- Pro Pack for Control Hub

Pro Pack for Control Hub is a premium offer for customers that require more advanced security, compliance, and analytics capabilities. Advanced capabilities include Hybrid Data Security with on-premises key management, mobile device security controls, and flexible long-term retention and historical reporting.

For more information on the Pro Pack for Control Hub offer, refer to the [Control Hub \(Data Security and Privacy\) Data Sheet](#).

- Extended Security Pack for Control Hub

Extended Security Pack is an add-on flex collaboration offer for Control Hub which bundles capabilities such as data loss prevention, anti-malware, and Duo MFA in one tightly integrated solution. This add-on ensures administrators and compliance officers have the tools they need to protect data within their organization.

For more information on the Extended Security Pack, refer to the [Control Hub Extended Security Pack Data Sheet](#).

10. Webex Control Hub – Security Features for the Webex App

This document has discussed the protocols and ciphers that are used to secure the Webex App. Administrators of an organization of Webex App users also need to have tools and administrative features that provide them with control over how the Webex App functions. For example, administrators may wish to implement file sharing controls, or to restrict communication with users outside of their organization.

The following section highlights some of the security features available in Webex Control Hub that provide security related controls. The list of features discussed below is not exhaustive, and security can also be implemented in adjunct applications that are integrated with the Webex App (such as Data Loss Prevention applications and Enterprise Content Management applications).

Token Revocation

Figure 14 shows the Reset Access option in Webex Control Hub, which allows an administrator to revoke all OAuth access and refresh tokens owned by the user. This option signs the user out of all instances of the Webex App and requires the user to reauthenticate.

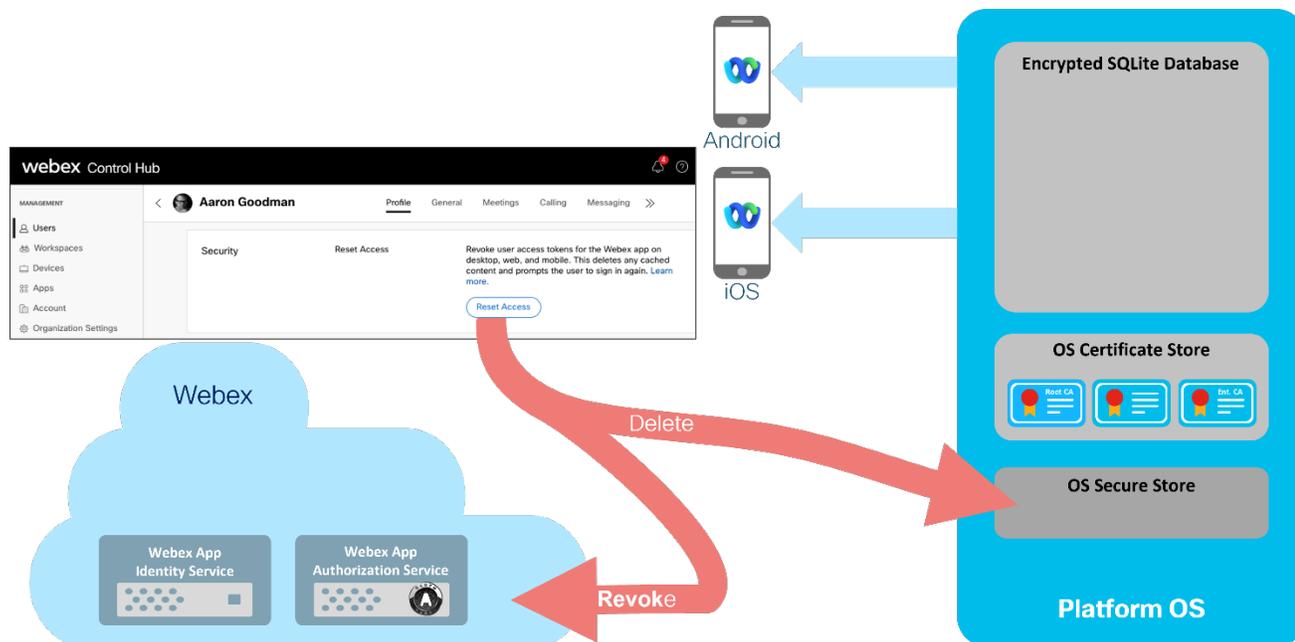


Figure 14. Reset Access: Revoke User Access and Refresh Tokens

Web Client Inactivity Timeout

The Webex App web client can be signed out automatically after a configurable inactivity time period, based on whether the application is connecting to Webex services on the enterprise network or remotely (see Figure 15).

Webex Teams Web Client Idle Timeout

Automatically log users out of an idle session. You can change the amount of time the client will remain idle until the user is logged out of their account.

To check whether users are connected to your organization's network, provide the URL of internal site that allows cross-origin-resource to share CORS with teams.webex.com.

Off network:

In network:

Figure 15. Webex Control Hub Webex Web App Idle Timeout

Managing the Webex App on Mobile Devices

Webex Control Hub has several options that allow an administrator to control how the Webex App is used on mobile devices, such as file upload and download controls, access control with data wipe function for mobile devices, and lock screen PIN enforcement.

For even more control, an Enterprise Mobile Management (EMM) solution can be deployed. There are two types of EMMs:

- Mobile Device Management (MDM)

It controls the mobile device by enforcing security policies at the device or profile level.

- Mobile Application Management (MAM)

It controls the apps on mobile device by enforcing the security policies at the application level.

The Webex App for mobile devices are certified with Cisco Meraki® Systems Manager and VMware AirWatch MDM applications, but support is not limited to just these providers. An organization using an MDM can include the Webex App in their Enterprise App store and use the MDM application to provide basic controls such as preventing copying data and screen captures, enabling backups, and setting remote wipe and pin lock requirements on devices. MDM applications are typically used on corporate owned mobile devices.

Mobile Application Management (MAM) applications can be used to control applications on employee-owned mobile devices. Using MAM you can manage the application rather than control the device. Wrapped versions of the Webex App that allow them to be managed by a MAM, for example, IPA files for iOS and AAB files (containing the APK files) for Android, are available as a Customer Connection Program offering. Using these wrapped apps, a Mobile Application Management (MAM) administrator can apply additional security policies to the Webex app.

There are two methods to deploy an EMM:

- Microsoft Intune

A separate app, Webex for Intune (SDK), is available from the Apple App Store and Google Play Store.

For more information, refer to the [Webex App installation with Microsoft Intune](#) article.

- AppConfig

AppConfig is a community that defines basic operating system (OS) level application control capabilities for iOS and Android.

For more information about securing the Webex App on mobile devices, refer to the [Webex App Secure mobile devices](#) article.

11. Services

Messaging

Service Overview

Webex Messaging allows the exchange of various types of information. This mainly includes messages, of course, but it also includes file upload/preview/download, screen sharing, and whiteboarding.

Signaling and Messaging

Signaling and messages are sent using the WSS (WebSocket Secure) protocol on outgoing port TLS 443. That channel is encrypted since it is sent through TLS. In addition to the transport being encrypted, the messages

themselves are also encrypted with an encrypting key that is unique to each space. Encryption keys of the Webex spaces are managed by the Webex Key Management Service (KMS).

The TLS version that is used is 1.2.

For more information on TLS and the cipher suites, refer to the [Webex App signaling encryption section](#).

Screen sharing

Screen sharing outside of a call, or a meeting is sent over the media path using the Secure Real-Time Transport Protocol (SRTP), described in RFC 3711. Media is sent by default on UDP destination port 5004, in line with RFC 3550 RTP – A Transport Protocol for Real-Time Application, as the preferred transport protocol. If that port is blocked, the Webex App falls back to TCP port 5004, and if this one is blocked, it will be sent on TLS port 443, possibly going through a web proxy if a web proxy is configured.

The media encryption cipher and encryption keys are securely negotiated over HTTPS, using SDES (RFC 4568). In general, the Webex App negotiates the cipher suite AEAD-AES-256-GCM but could revert to the AES_CM_128_HMAC_SHA1_80 for interoperability.

File Sharing

Files can be shared through Webex Messaging. Files are stored in the Webex by default and many controls on the files are available to the administrator via Control Hub. For example, file sharing can be disabled if the user is outside of a specific IP network, file sharing can be blocked for external users in internal group spaces, or vice-versa, can be blocked for internal users in externally owned spaces. A separate Enterprise Content Management (ECM) solution can also be used for file sharing and the file sharing control would be managed by the ECM. The Webex App would also connect directly to the ECM to access the files without having to go through Webex. Note that whiteboards and annotations are always stored in Webex, even if an ECM is provisioned for file sharing. Files sharing runs over the signaling channel on TLS port 443.

For more information on Webex Messaging security, refer to the [Webex Messaging security technical paper](#).

Meeting

Service Overview

Webex Meetings is part of the Webex collaboration solution that enables meetings hosted in Webex and are accessible anywhere around the globe and across different devices. Meetings service enables global employees and virtual teams to collaborate in real time with voice, video, and content share as though they were working in the same room.

Signaling

All communications between the Webex App, cloud registered Webex devices and Webex services are encrypted and transmitted via HTTPS over TLS on port 443. Webex uses TLS protocol version 1.2 or later with high strength cipher suites for signal. When selecting ciphers, Webex services will select the strongest possible cipher based on a preference order available in the customer's environment. For the list of supported ciphers, refer to the Webex Meetings Security technical paper referenced below.

In addition, Webex Meetings support standard SIP or H.323 devices connecting through Cisco Expressway or Session Border Controller (SBC) at the enterprise network edge to Webex over TCP or TLS or MTLS for signal. However, Cisco strongly recommends using an encrypted connection to cloud services so no unencrypted traffic will traverse through the internet.

Media

After establishing the signal channel, the Webex App and services use SRTP to encrypt the media streams (VoIP audio, video, screen, or document share) on a hop-by-hop basis. For each SRTP call leg, Webex media servers need access to the media encryption keys to decrypt the media streams.

Encrypted media can be transported over UDP, TCP or TLS. Cisco prefers and strongly recommends using UDP as the transport protocol for voice and video media streams. This is because TCP and TLS are connection-oriented transport protocols designed to reliably deliver correctly ordered data to upper-layer protocols. Using TCP or TLS, the sender will retransmit lost packets until they are acknowledged, and the receiver will buffer the packet stream until the lost packets are recovered. For media streams over TCP or TLS, this behavior manifests itself as increased latency or jitter, which affects the media quality experienced by the call's participants. In terms of media port prioritization, UDP ports 5004 and 9000 are preferred for SRTP. If they are blocked, TCP 5004 will be tried, followed by TLS port 443. For details on network ports requirements, refer to the [Network Requirements for Webex Services](#) article.

Media packets are encrypted using either AES-256 or AES-128 based ciphers. The Webex App and Webex Room Devices use AES-256-GCM to encrypt media, and the media encryption keys are exchanged over TLS-secured signal channels.

For more information on Webex Meetings security, refer to the [Webex Meetings Security technical paper](#).

With Zero Trust End-to-End Encrypted Meetings, Webex does not have access to the encryption keys and hence, cannot decrypt the media streams. Webex Zero Trust end-to-end encryption uses standard track protocols to generate a shared meeting encryption key (Messaging Layer Security (MLS)) used to encrypt meeting content (Secure Frame). With MLS, the meeting encryption key is derived at each participant's Webex App or device using its private key (never shared) and every other participant's MLS key package which contains several items including the public key of the participant. The meeting encryption key never traverses the cloud and is rotated when new participants join the meeting.

For more details on Zero Trust Security based end-to-end encryption, refer to the [Zero Trust Security for Webex security paper](#).

Calling

Service Overview

Calling with Webex is part of the overall Webex suite enabling one-to-one and one-to-many voice and video calls leveraging the Webex platform.

As shown in Figure 16, calling with the Webex App encompasses three distinct but fully integrated types of calling:

1. Call on Webex
This calling option is called 'Call on Webex' within the client and leverages the Webex platform for calling.

When making calls in the Webex App, this is the default calling type.

2. Webex Calling

Uses the Webex App to call from the enterprise DN through the Webex Calling multi-tenant platform. This calling option involves selecting the Enterprise or Work number for making a call and provides PSTN calling capabilities.

3. Calling in the Webex App (Unified CM)

Uses the Webex App to call from the enterprise DN through on-premises Unified CM calling or Dedicated Instance for Webex Calling platform. This calling option also involves selecting the Enterprise or Work number for making a call and provides PSTN calling capabilities as well.

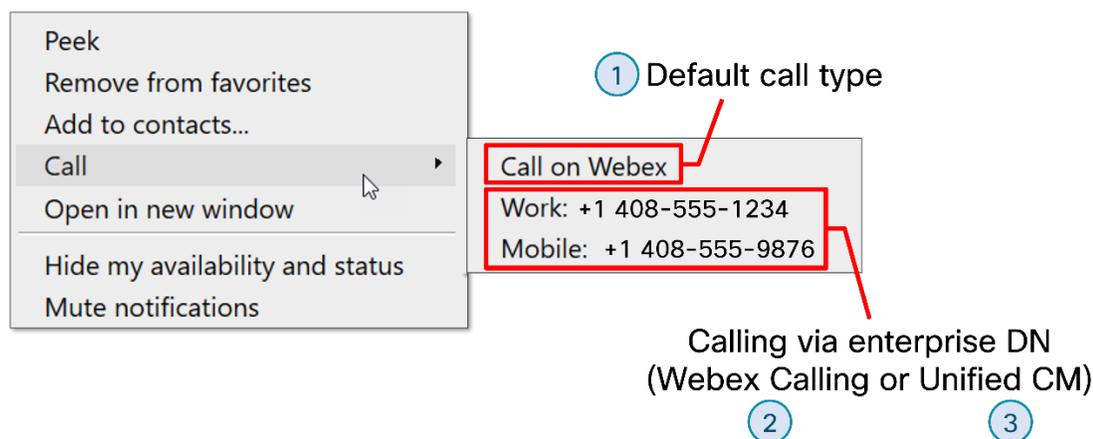


Figure 16. Webex App Calling Options

Signaling and Media

As shown in Table 2 below, depending on the type of calling used, the signaling and media protocols and port numbers will vary.

Table 2. Signaling and Media for Webex App Calling

WEBEX APP CALL TYPES	CALL SIGNALING	CALL MEDIA
Call on Webex	<ul style="list-style-type: none"> • HTTPS and WSS (WebSocket Secure) protocol • Outgoing encrypted traffic port TLS 443 	<ul style="list-style-type: none"> • SRTP (Secure Real-time Transport Protocol)¹ • Port UDP² 5004 (default)
Webex Calling	<ul style="list-style-type: none"> • SIP (Session Initiation Protocol) • Outgoing encrypted traffic port TLS 8934 	<ul style="list-style-type: none"> • SRTP • Port UDP 19560-65535
Unified CM Calling (Also applies to Dedicated Instance for Webex Calling which requires Expressway MRA for Unified CM)	<ul style="list-style-type: none"> • SIP • Outgoing traffic port TCP 5060 • Optional³ outgoing encrypted traffic port 5090 (OAuth) 	<ul style="list-style-type: none"> • RTP (or optionally SRTP)³ • Port UDP 16384-32766 • MRA port UDP 36000-59999 (always encrypted with SRTP)

Connectivity)	<ul style="list-style-type: none"> • Outgoing CTI port TCP 2748⁴ • Outgoing encrypted traffic over MRA port TLS 5061 (OAuth) 	
---------------	---	--

¹ Described in RFC 3711.

² Aligning with RFC 3550. Fallback to TCP port 5004 or TLS port 443.

³ Encrypted signaling and media optional for on-premises call control.

⁴ Not supported over MRA

Note: The information in Table 2 above applies solely to call signaling and media. It does not account for traffic flows related to initial authentication/authorization or configuration of the Webex App.

For complete information on Webex Calling network requirements, refer to the [Port Reference Information for Webex Calling](#) article.

For more information on Webex Calling security including encryption cipher suites, refer to the [Webex Calling Security paper](#).

For more information on Webex Services network requirements for all workloads (meetings, messaging, and calling), refer to the [Network Requirements for Webex Services](#) article.

12. Conclusion

Be collaborative and get more done, faster, using the Webex App. Webex is a trusted industry leader in web and video conferencing, messaging, and calling. Webex offers a scalable architecture, consistent availability, and multilayer security that is validated and continuously monitored to comply with stringent internal and third-party industry standards. We connect everything more securely to make anything possible.

13. How to Buy

To view buying options and speak with a Cisco sales representative, visit [How to Buy Cisco Products](#).

14. For More Information

[Webex Messaging](#)

[Webex Meetings](#)

[Webex Calling](#)

15. Appendix

TLS Overview and TLS Version History

Transport Layer Security (TLS) was created to provide authentication, confidentiality (data encryption), and data integrity between client and server applications. Determining the identity of the server is achieved by the server sending its CA signed certificate, chain of intermediate certificates and optionally the CA root certificate to the client for verification, as shown in Figure 17. The session is secured using symmetric encryption, the encryption cipher suite, and encryption key generation method for the session being negotiated before any data is sent. Key

generation involves an exchange of values between the client and server, that allows both to generate a shared secret that is not transmitted and therefore not available to eavesdroppers. Once the cipher suite has been negotiated and the shared symmetric encryption key generated, each encrypted message is sent with a message authentication code to detect if the data in transit has been modified.

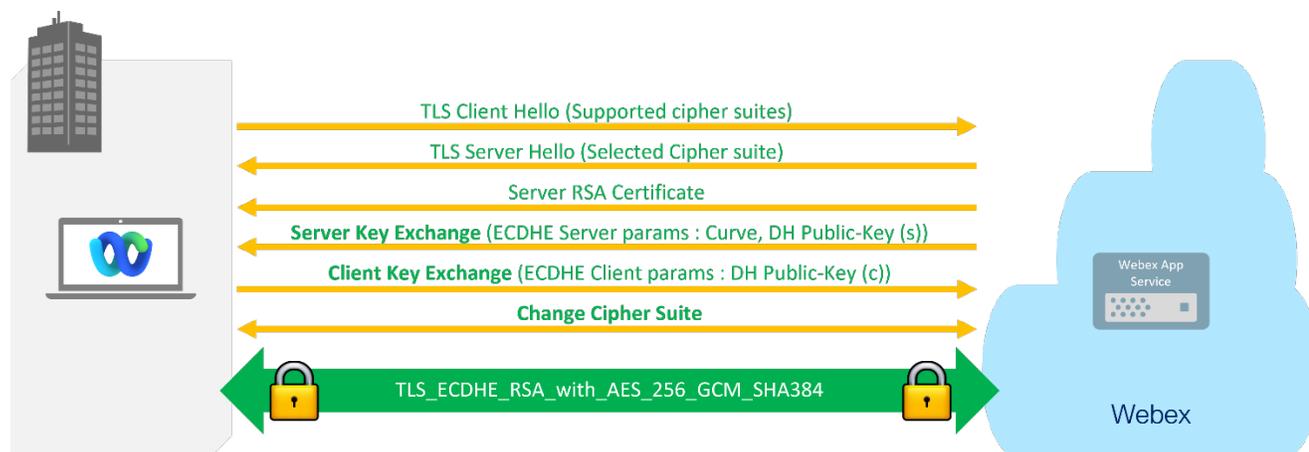


Figure 17. Webex App TLS Handshake

TLS 1.0

TLS 1.0 was first defined in 1999 and is based on the Secure Sockets Layer Protocol Version 3.0 (SSL 3.0). Evolving regulatory requirements and security vulnerabilities discovered in TLS 1.0 and SSL, have led to recommendations that they be disabled, in favor of the newer TLS versions of 1.1 and 1.2. More recently, the initiative to deprecate SSL and TLS 1.0 has been driven by the Data Security Standards defined by the Payment Card Industry (PCI). As of June 30, 2018, in order to comply with the PCI Data Security Standard (DSS), all websites must use TLS 1.1 or higher.

TLS 1.1

In April 2006, TLS 1.1 was published as a minor update to TLS 1.0. For more information on TLS 1.1, see [RFC 4346](#). The primary changes in TLS 1.1 provide protections against Cipher Block Chaining (CBC) attacks, by adopting explicit initialization vector selection and changing the way that padding errors are processed. Most web browser vendors plan to deprecate TLS 1.1 in 2020.

TLS 1.2

In 2008, TLS 1.2 was released and published as [RFC 5246](#). TLS 1.2 is currently the most widely used version of TLS and has several improvements in security when compared to TLS 1.1, particularly for negotiation of cryptographic algorithms.

A summary of the major differences between TLS 1.1 and TLS 1.2 described in RFC 5246 include:

- Stronger hashing algorithms (SHA-2) for MAC and pseudorandom function (PRF) computations
- Client and server ability to specify the accepted hash and signature algorithms.
- Support for cipher suites using Authenticated Encryption with Associated Data (AEAD) e.g., AES-128-GCM and AES-256 GCM, which are computationally more efficient and not only encrypt, but also authenticate data.
- AES cipher suites that support elliptical curve cryptography
- TLS extensions (see <https://tools.ietf.org/html/rfc6066>)

TLS 1.3

TLS 1.3 was defined by [RFC 8446](#) in August 2018.

A summary of the major differences between TLS 1.2 and TLS 1.3 described in RFC 8446 include:

- The list of supported symmetric encryption algorithms has been pruned of all legacy algorithms. The remaining algorithms are all Authenticated Encryption with Associated Data (AEAD) algorithms.
- A zero-roundtrip time (0-RTT) mode was added to improve connection setup times.
- Static RSA and Diffie-Hellman cipher suites have been removed. All public key based key exchange mechanisms now provide forward secrecy.
- Elliptic curve algorithms are now in the base specification and new digital signature algorithms, such as ECDSA, are included.

TLS 1.3 has several significant improvements over TLS 1.2 but has yet to be widely adopted.

The Webex App - TLS Version and Cipher Suite Negotiation

The Webex App uses TLS version 1.2 or later for signaling. Webex services (messaging, encryption, file storage and so on) only support TLS version 1.2 or later.

TLS 1.1 and 1.2 support many cipher suites, greater than 100 ciphers, for more information, see <https://www.openssl.org/docs/manmaster/man1/ciphers.html>). Many of the cipher suites listed in the above link have known vulnerabilities and should be avoided, for example, cipher suites that use DES or RC4 encryption. The Webex App can negotiate only a small subset of cipher suites with Webex Services.

Figure 18 shows that during TLS session establishment, the App sends a list of its supported cipher suites in order of preference in a Client Hello message to the TLS server. The server selects one cipher suite, based on the subset of cipher suites that both the client and server support and the server preference and returns this selected cipher suite to the App in a Server Hello message.

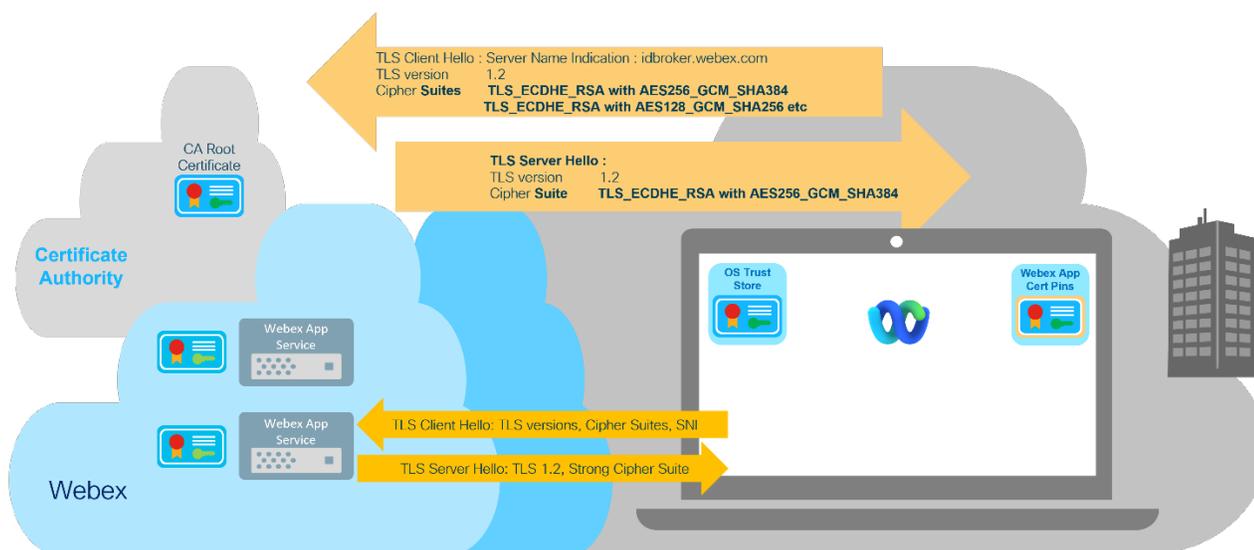


Figure 18. Webex App: Client to Server TLS Negotiation

TLS signaling connections from the Webex App to Webex services use only TLS version 1.2 or later. The TLS 1.2 ciphers that are negotiated include:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_CHACHA20_POLY1305_SHA256

These cipher suites meet the guidelines defined in the US National Institute of Standards and Technology (NIST) Special Publication 800-52 Revision 2. For more information, see [Guidelines for the Selection, Configuration, and Use of Transport 4 Layer Security \(TLS\) Implementations](#).

These ECDHE based cipher suites are a subset of those listed in Section 3.3.1.1.2 – “Cipher Suites for RSA Certificates”, the other ciphers defined in Section 3.3.1.1.2 use Diffie Hellman Ephemeral (DHE) for key generation which is less CPU efficient than ECDHE.

All the negotiable cipher suites have the following features in common:

Elliptical Curve Diffie Hellman Ephemeral (ECDHE) Key Generation

Elliptical Curve Diffie Hellman (ECDH) is an anonymous key agreement protocol that allows two parties, each having an elliptic curve public and private key pair, to establish a shared secret over an insecure channel, through the exchange of their elliptic curve public keys. The shared secret is used to derive the symmetric encryption key used by the TLS session.

With Elliptical Curve Diffie Hellman Ephemeral (ECDHE) key generation, for each TLS session: the client and the server exchange new elliptic curve public keys to generate a new shared secret and derive a new symmetric encryption key. Using ECDHE provides “forward secrecy”, whereby if the elliptic curve private key, shared secret, or symmetric key for a TLS session is compromised, previous and future TLS sessions cannot be decrypted using these compromised values.

RSA Authentication

ECDHE-RSA key generation uses the Rivest–Shamir–Adleman (RSA) public key associated with the server certificate to sign and authenticate the ephemeral elliptic curve public keys exchanged between the client and server.

- The RSA public key in the server’s RSA certificate is sent from the server to the client prior to key negotiation and is used to sign the ephemeral elliptic curve public key sent by the client to the server.
- The server’s RSA private key is used to sign the ephemeral elliptic curve public key sent by the server to the client.
- The server’s signed ephemeral elliptic curve public key has its integrity authenticated by the client, verifying its digital signature using the server’s RSA certificate public key.
- The client’s signed ephemeral elliptic curve public key has its integrity authenticated by the server using its private key.

Encryption Algorithms

Advanced Encryption Standard (AES)

The negotiable symmetric key encryption algorithms used between the Webex App and Webex services use a subset of the Advanced Encryption Standard (AES) ciphers defined by the US National Institute of Standards and Technology (NIST):

- AES-256-GCM
- AES-128-GCM
- AES-256-CBC
- AES-128-CBC

Standardized in 2001 the AES algorithm features a cipher block size of 128 bits and three key length options: 128, 192 or 256 bits.

AES symmetric encryption is widely used today, mostly with 128-bit or 256-bit key lengths. The US National Security Agency classifies AES-256 as strong enough to protect TOP SECRET data. For more information, see <https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm>.

GCM and CBC Encryption Modes

The AES encryption algorithms used by the Webex App support two encryption modes for fixed size blocks of plaintext:

- Galois Counter Mode (GCM)
- Cipher Block Chaining (CBC) mode

Galois Counter Mode (GCM) encryption was introduced with TLS1.2. GCM encryption operates on 128-bit blocks of data and can be performed in parallel, making the algorithm more efficient than other algorithms such as CBC. GCM is an Authenticated Encryption with Associated Data (AEAD) algorithm. The output of an encryption cycle for a block of plaintext is the encrypted cipher text and an authentication tag. Only AEAD based encryption algorithms are allowed with TLS 1.3. AES encryption using GCM is faster and more secure than AES CBC.

Cipher Block Chaining (CBC) mode is an older encryption algorithm and is available for use in TLS versions 1.0, 1.1 and 1.2. CBC mode encryption also uses a 128-bit data block size but cannot take advantage of parallel processing as encryption is sequential and each block of plaintext XORed with the previous ciphertext block before being encrypted. CBC is useful for interoperability with older TLS implementations and can be faster when used on older CPU hardware.

TLS 1.2 supports over one hundred negotiable cipher suites, some of which use cipher suites with known weaknesses or performance inefficiencies.

Secure Hash Algorithms: SHA-256 and SHA-384

SHA-256 and SHA-384 are part of the SHA-2 family of secure hash algorithms. Secure Hash Algorithms (SHA) are used in TLS to authenticate the data's origin and/or to validate the integrity of data. Hash algorithms such as the SHA family can be used in TLS as follows:

- To provide message authenticity when the encryption algorithm doesn't use authenticated encryption, for example: AES-CBC
- As part of a Pseudo Random Function (PRF) used to compute the shared master secret during client-server key exchange. For more information, see <https://tools.ietf.org/html/rfc5246#section-5>
- To authenticate the encrypted Finished message of the TLS Handshake protocol

The SHA-2 algorithms are part of a family of secure hash algorithms defined in NIST FIPS 180-4. This family of secure hash algorithms can be grouped into three sets:

- SHA-1: 160-bit hash output
- SHA-2: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256
- SHA-3: SHA3-224, SHA3-256, SHA3-384, SHA3-512

SHA-1

Because of known weaknesses, Certificate Authorities and web browsers deprecated using certificates with SHA-1 based digital signatures in 2017. Any certificates used for TLS connections should be using SHA-2 based digital signatures.

The use of SHA-1 with certificates should not be confused with its use in a TLS cipher suite. For example, although not preferred, SHA-1 for message authentication is still supported in TLS cipher suites such as TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA. In this context, SHA (SHA-1) is used to generate a keyed-hash message authentication code (HMAC) for each encrypted message. HMAC uses a symmetric cryptographic key, specifically generated for this MAC function during the TLS handshake, in conjunction with a hash function (SHA-1) to provide a strong form of message authentication.

SHA-2

The SHA-2 family of hashing algorithms produce hash outputs of either 224, 256, 384, or 512 bits. SHA-2 is the most commonly used set of secure hash algorithms with TLS 1.2 cipher suites and is used by Certificate Authorities for digital signatures in all new certificates. TLS 1.3, published as RFC 8446 in August 2018, specifies the use of SHA-2 hashing algorithms, specifically SHA-256 and SHA-384, removing the SHA-1 hashing algorithms.

SHA-3

Like SHA-2, the newer SHA-3 family of hashing algorithms also produces hash outputs of either 224, 256, 384, or 512 bits. They are not widely implemented today.

November 2022