PHILIPS

hue personal wireless lighting

Secure

E2E Encryption
Whitepaper

Whitepaper
End-to-End Encryption (E2EE)

Release date: 01 - 09 - 2023

Table of contents

1. Introduction

Personal data protection is an integral part of the development of all Philips Hue products and their functionalities. Because successful relationships require mutual trust and shared understanding, we aim for complete transparency in the way that we deal with your personal data. We regard video data as a highly private category of data that requires dedicated protection mechanisms to limit the impact of a potential data breach. This technical document describes how video data is protected in the Hue ecosystem.

The design of the video encryption feature follows the following principles:

Secure by default: the system is designed to be secure under the default settings out of the box. This means that end-to-end encryption is enabled by default for all users when creating a new home. Specific use cases may require E2EE to be disabled, but this will always be an explicit user action by an administrator of the home and should only disable as few security mechanisms as possible.

Simplicity over complexity: encryption, key derivation, and key management tends to become very complex in various security systems. This should be avoided as much as possible while retaining a high level of security. The user should be bothered as little as possible with (re-)configuring end-to-end encryption.

User experience and features: the use of end-to-end encryption should not complicate user flows and using the system. Designing end-to-end encryption into the system from the ground up ensures that features such as multi-user access to a home are not unnecessarily complicated. In addition, end-to-end encryption should not limit the features available on the camera or in the ecosystem.

2. Video encryption technical description

2.1 Clip and snapshot encryption

The camera records audio and video clips when certain events occur. These audio and video clips are stored in the cloud so that they can be retrieved later. The connection between the camera and the cloud is protected using TLS1.2. However, this does not provide end-to-end encryption since this TLS connection terminates on our cloud server. Therefore the camera further encrypts all individual audio and video frames before uploading them to the cloud.

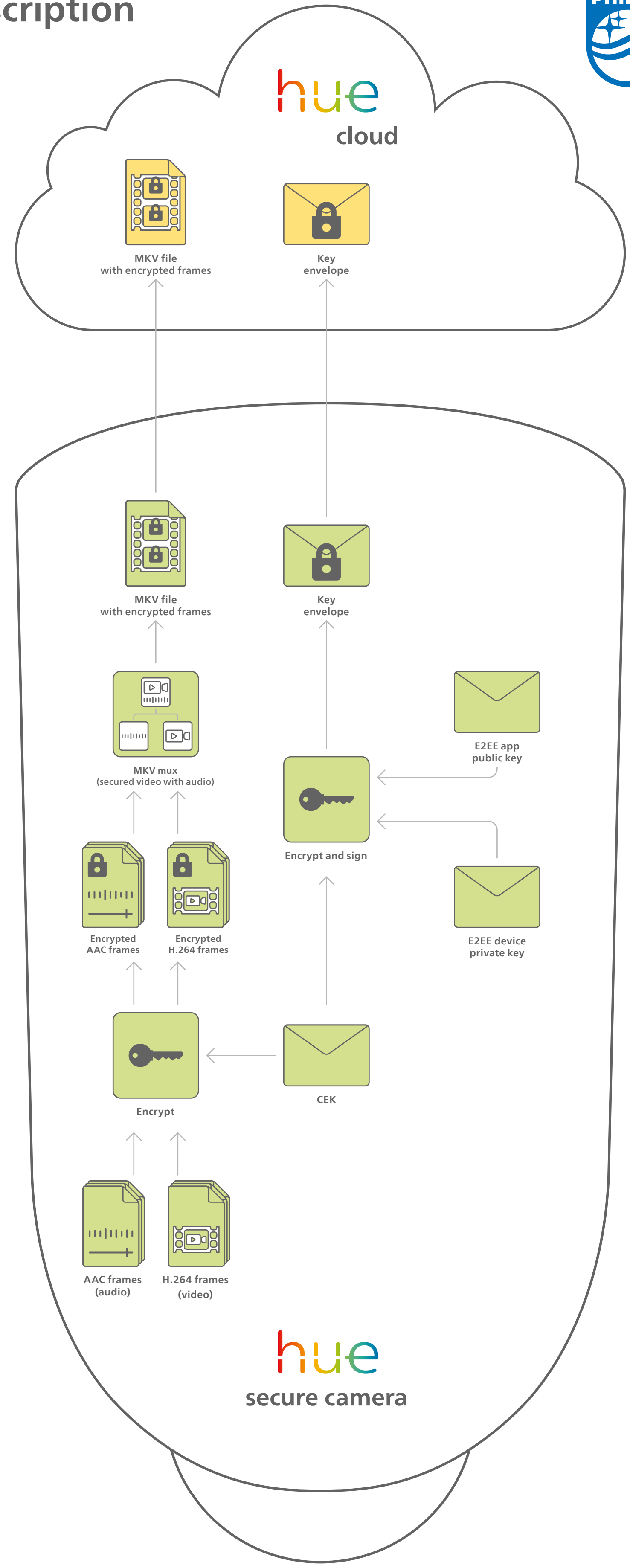
The standardized algorithm AES is used to encrypt the audio and video frames. The camera generates a fresh Content Encryption Key (CEK) for every new clip using a cryptographically secure random number generator. All audio and video frames are encrypted using the CEK so that no plain video or audio data are ever stored in the cloud.

In addition, we ensure that the camera does not store the CEK persistently. This prevents even the same camera from decrypting the audio and video frames stored in the cloud after the recording has been finalised.

Playing the audio and video requires the user's app to obtain the right CEK. Because the app may not be online when the clip is generated, the camera cannot reliably communicate the CEK to the app through a direct connection. Therefore, the cloud is used to store securely an encrypted version of the CEK that can only be decrypted by the user's app, ensuring the end-to-end encryption of the audio and video. This process of encrypting the CEK is called key enveloping.

To create a key envelope, the camera signs the CEK with its unique device private key using the ECDSA algorithm and encrypts the CEK with the app public key using the ECIES algorithm. The resulting cipher text is called a key envelope and can only be decrypted using the app private key. The key envelope is stored in the cloud and accessible only to those users within the home that currently have permissions to view the video clips.

The camera also generates snapshots when events occur. These snapshots may be just as privacy sensitive as the actual video footage. Therefore, the same mechanism is used to protect snapshots. The camera generates a CEK for encrypting the snapshot and generates a key envelope so that the app can decrypt the snapshot before showing it to the user.



2.2 Clip and snapshot decryption

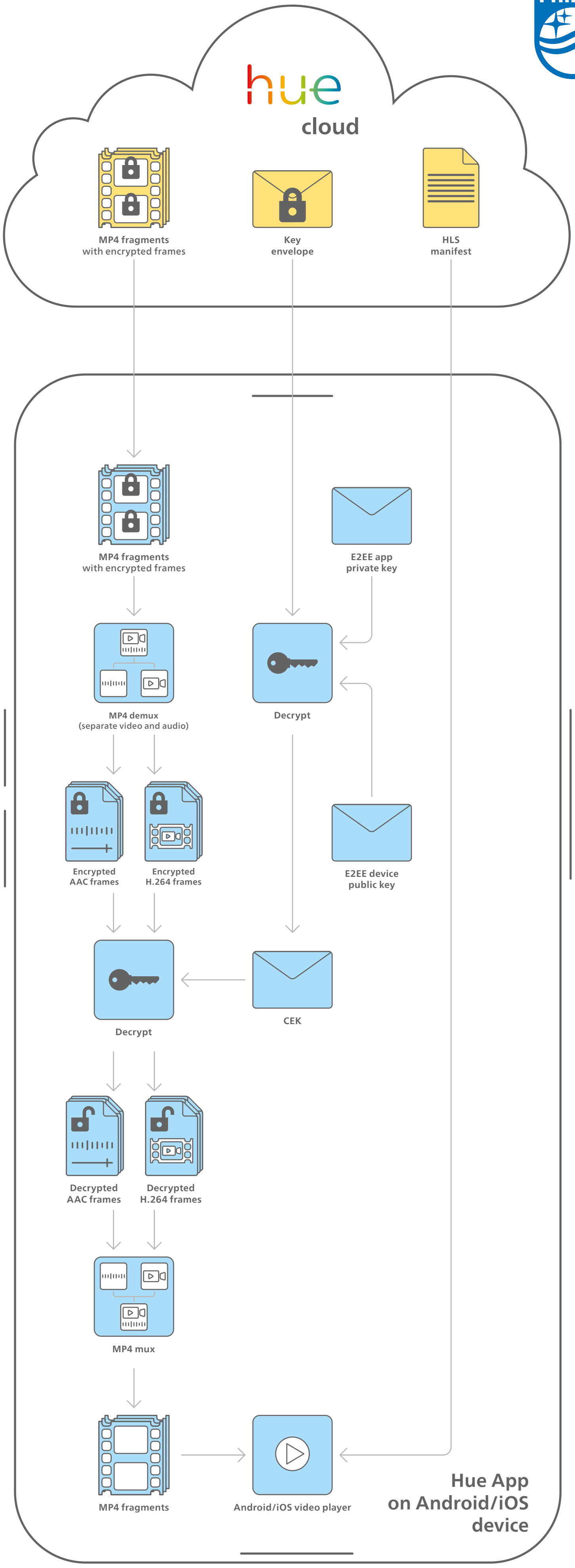
The Hue cloud stores the encrypted audio and video frames and their corresponding key envelopes. The security admin of the home (and other users of the home that have been granted permission) may request to view a video clip. The cloud then generates the necessary MP4 fragments that the app can download.

The web technology used to view the video is HTTP Live Streaming (HLS) and relies on an HLS manifest that informs the video player as to which MP4 fragments need to be downloaded.

The MP4 fragments contain all the video and audio frames. These video and audio fragments are all encrypted using the CEK. This means that in case anyone gets administrative access to the Hue cloud, they can only obtain encrypted video and audio frames since decryption of the encrypted CEK is not possible based on the information available in the Hue cloud.

The app cannot play the video before it has decrypted the audio and video frames. The app first receives the key envelope. It uses the app private key to decrypt and retrieve the CEK and it uses the camera public key to verify the ECDSA signature using the device public key. The signature verification provides assurance that the key envelope was generated by a camera in the home.

The CEK can then be used to decrypt all audio and video frames. The encrypted frames must first be extracted from the MP4 fragments in a 'MP4 demuxing' step. After decryption they must be remuxed into MP4 fragments so that the video player can play back the video. All these steps occur 'on the fly' so that no unencrypted video is stored by the Hue app on the mobile device.

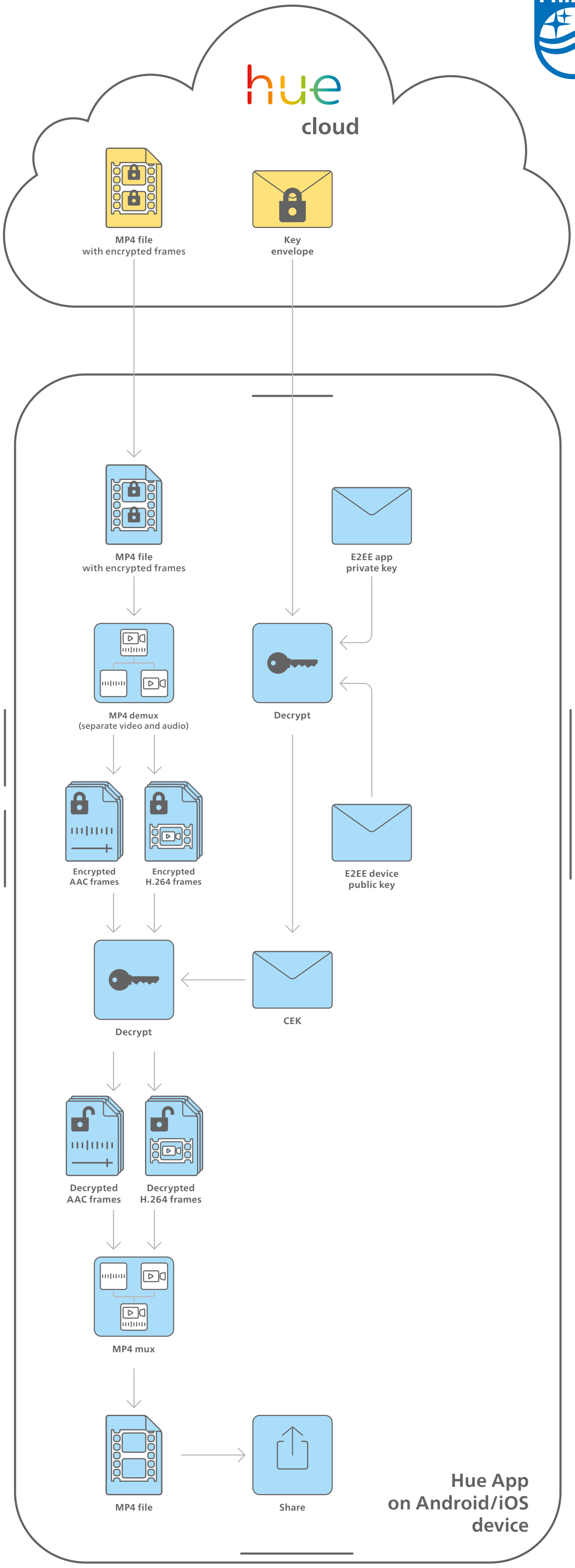


2.3 Video clip download

A user may want to share the audio and video clip with somebody outside the home (potentially non-Hue users), or backup the audio and video clip on a storage device. To do this, the app can create an unencrypted version of the audio and video clip. Although the technical details are different, the overall process is very similar to playing an audio and video clip in the app.

The app downloads and decrypts the corresponding key envelope to obtain the CEK. The app also downloads the MP4 file with encrypted frames and creates a decrypted MP4 file involving the necessary muxing and decryption steps.

At this point, the app holds an unencrypted video clip in memory that can be shared using the standard Android/iOS sharing mechanisms. It then becomes the user's responsibility to ensure that the unencrypted audio and video clip is only shared with those people that are allowed to have access to the clip, the clip is protected during transmission, and is securely stored at rest.



2.4 Live view protection

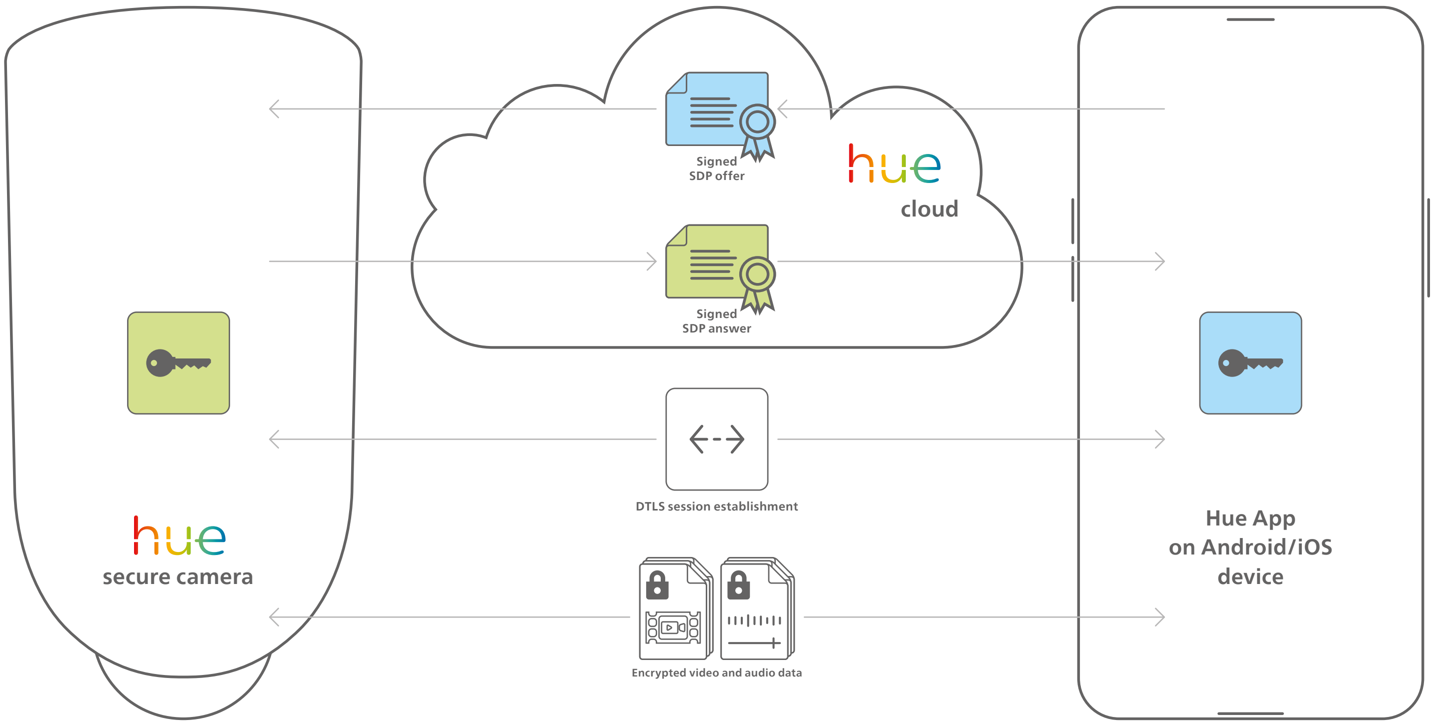
Live view uses a different underlying technology from stored audio and video clips. It uses the open-source WebRTC technology to transfer audio and video data between the camera and the app. WebRTC establishes a peer-to-peer bidirectional channel between the camera and the app. This means that typically data between the app and camera is exchanged directly. The camera transfers data via the Hue cloud only if there is no direct networking path between the app and the camera. In all cases, WebRTC uses DTLS to encrypt audio and video data.

DTLS certificates are generated each by the camera and the app at the start of a new WebRTC session. The use of DTLS ensures that the session encryption key is only known to the camera and app that generated the private keys corresponding to the DTLS certificates that were used to start the WebRTC session. So technically, WebRTC provides end-to-end encryption by default, although with one caveat.

This caveat lies in the assumption that the signaling channel is not compromised. To establish the WebRTC connection, the camera and app use a signaling server hosted in the Hue cloud. The SDP protocol is used to exchange parameters needed for the WebRTC session. The app initiates the connection by sending an SDP offer and the camera responds with an SDP answer. The Hue cloud performs access control to ensure that SDP offers and SDP answers are only exchanged between a user and the camera they own (and not with someone else’s camera).

In the event that the Hue cloud is compromised by a malicious entity, then the WebRTC connection between the camera and the app cannot be trusted. For instance, the malicious entity can establish a live view session with the camera. Researchers have shown that such active man-in-the-middle attacks are feasible for WebRTC (see Active MITM[1]). The E2EE keys established for a home allow us to additionally protect the signaling phase of WebRTC so that these active man-in-the-middle attacks become infeasible even in the event that the Hue cloud is compromised.

To protect the signaling phase, the SDP offer and answer are signed using the E2EE keys. The app signs its SDP offer with its app private key using the ECDSA algorithm and attaches the signature to its SDP offer. The camera verifies the signature using the app public key and only proceeds with establishing the WebRTC channel if this signature verifies correctly. The camera signs the SDP answer with the device private key and attaches the signature to the SDP answer. The app similarly verifies the signature using the device public key and only proceeds if this signatures verifies correctly.


Signed SDP offer
hue cloud
Signed SDP answer
DTLS session establishment
Encrypted video and audio data
hue secure camera
Hue App on Android/iOS device

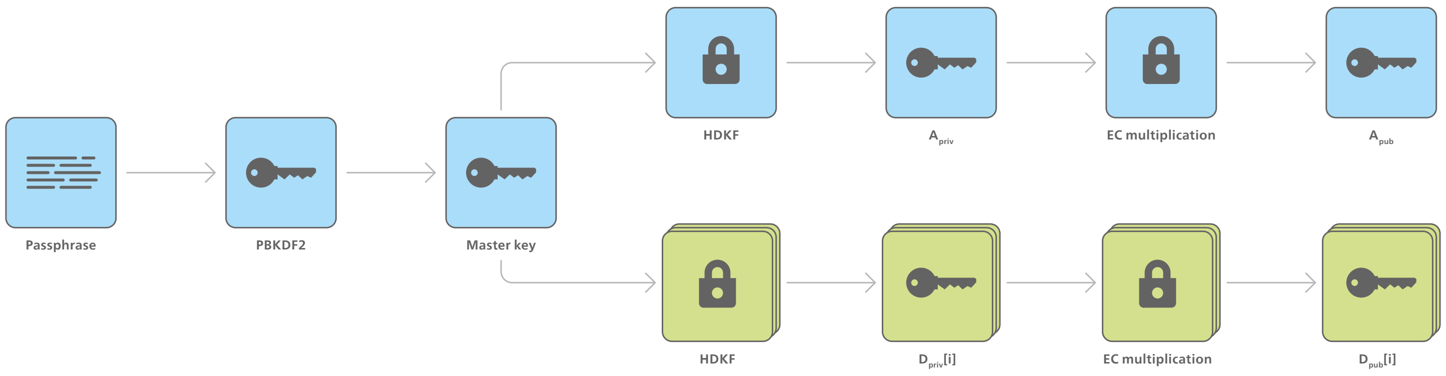
[1] https://webrtchacks.com/webrtc-and-man-in-the-middle-attacks/

3. Key management

3.1 Key management and key derivation

During enrolment of the first camera in a home, a passphrase is generated. This passphrase forms the basis for all the long-term keys used for end-to-end encryption. This passphrase is generated in the app and never shared with the Hue cloud. It can be shared directly with other members of the home allowing them to decrypt videos. Several long-term keys are derived from the passphrase for different purposes: device keys for the cameras and an app key for the apps. The following table summarizes the long-term keys used for end-to-end encryption:

Name	Type	Description
passphrase	10-word passphrase	Passphrase used as a basis for E2EE. The passphrase consists of 10 randomly selected words from a wordlist. The 10 randomly selected words are joined into a passphrase by single spaces. Example: "approval crept dislocate embellish flammable puppy scam struggle mom fox" All other keys are derived from the passphrase.
master key	256-bit symmetric key	The master key is derived from the passphrase by applying a key derivation function (PBKDF2) to the passphrase. It is then used to derive the private keys of the app and cameras.
app key pair	ECC key pair (P-256)	Key pair used by the app to establish E2EE. This key pair is shared between all apps in a home. Consists of the private key A_{priv} and the corresponding public key A_{pub}. The private key is derived from the master key by applying a key derivation function.
device key pair	ECC key pair (P-256)	Key pair used by the camera to establish E2EE. Every camera has its own key pair. Consists of the private key $D_{priv}[i]$ and the corresponding public key $D_{pub}[i]$ (where i is used to distinguish different cameras in the home based on their Device ID). The private key is derived from the master key by applying a key derivation function



During camera onboarding, a unique device key pair is generated for the camera. This device private key is shared with the camera directly from the app over Bluetooth. The app public key is also shared during the camera onboarding. The passphrase and master key are not shared with the camera to ensure that other key material cannot be extracted from a stolen or disposed camera.

The app private key is shared between all apps in a home. A single private key for all apps may seem like a security limitation, but in practice end-to-end encryption is an additional layer of security on top of the Hue cloud access controls. Even if the private key is available to a user, the home admin can revoke that user's access to the encrypted video clips and to the WebRTC signaling channel through the Hue cloud. The end-to-end encryption passphrase or any of the derived keys by themselves cannot be used to access video clips or live streaming. Security-conscious users, still have the possibility to update the passphrase through the app (though this means that it needs to be re-shared with others in the home and cameras need to be re-provisioned with a new key). This additional step if needed completely revokes the old passphrase and any keys derived from it.

3.2 Sharing the E2EE passphrase

To grant someone access to video clips and live streaming they need to be invited to join that home along with (1) receive the right permissions ('admin' or 'security advanced'), and (2) receive the E2EE passphrase of that home. They can then derive the app private key to decrypt key envelopes or start a live streaming session. There are two ways to share the E2EE passphrase:

1. Manually: an existing user looks up the passphrase in their app or in a backup and shares the passphrase with the new user. This can, for instance, be done verbally or via secure messaging.

2. Via a QR code: an existing user generates a QR code in their app that encodes the E2EE passphrase. The new user scans and stores the passphrase in their app.

There is no way to share the passphrase via the Hue cloud as this would defeat the purpose of end-to-end encryption.

3.3 Key provisioning and backup and resetting

During onboarding to a home, cameras need to be securely provisioned with their credentials: the unique device private key and the app public key of the home. These are shared over a point-to-point Bluetooth Low Energy (BLE) connection directly between the app and the camera.
The security of this connection is based on a device certificate stored on the camera as well as a QR code or hex code printed on the camera. This ensures that the user's app connects to the right device and that the credentials sent over BLE are secure against anyone impersonating the user's camera.
The user is requested to create a backup of the E2EE passphrase during the first camera setup. This is necessary because the E2EE passphrase is not stored in the Hue account. If a user logs in to his account on another phone, they need to enter the E2EE passphrase from a backup or using a QR code generated on an another instance of the app with the knowledge of the passphrase. The user may choose not to create a backup of the E2EE passphrase, but in case of a lost or damaged phone there is no way (for the user and for Philips Hue) to recover the E2EE passphrase. Similarly, if the user deletes the Hue app and all associated data, then the E2EE passphrase is deleted. In case of a lost E2EE passphrase, a new passphrase can be generated and the cameras must be reconfigured with keys derived from this new passphrase.

4. Turning off live view protection

The additional signature verification for WebRTC ensures that a malicious entity with access to the signaling server cannot establish a WebRTC live streaming session with the camera. However, partner integrations such as with Amazon and Google are also built on WebRTC to exchange audio and video between the camera and the respective partner's cloud. The SDP offer/answer for WebRTC session establishment with partner integrations must not be signed because no E2EE keys have been exchanged with the partners. Therefore, to enable partner integrations, the additional end-to-end protection must be explicitly disabled by the user. By default, partner integrations are not possible following the 'secure by default' security principle.

Live view protection can be enabled and disabled on a per-camera basis. The app instructs the camera to enable or disable live view protection using an interactive protocol that exchanges MQTT messages via a broker in the cloud. Messages are signed using the app private key and device private key to ensure that the cloud cannot forge these messages and the app can verify whether the setting has been successfully updated. Turning off live view protection disables the feature for all apps and users in the home. If a new user joins a home with live view protection disabled, then this feature is disabled by default for the new user.

Note that disabling live view protection does not disable encryption. To be precise: the camera and app no longer verify that the SDP offer and answer originate from the expected camera and app. This means that the app cannot verify that it is viewing an unmodified live feed (or a feed from the right camera) and the camera cannot verify that the WebRTC viewer is indeed the genuine Hue app or a partner. Standard access controls are still enforced by the Hue cloud (users still need to have the right permissions in the right home). Live view is also still encrypted as this is mandated by WebRTC.

Turning off live view protection does not affect snapshot and video clip encryption. Snapshot encryption and video clip encryption cannot be disabled.

5. Security considerations

The security of video encryption is based on various underlying assumptions. This section clarifies these assumptions and provides guidance on what is expected of the user to minimize the probability and impact of a data breach.

5.1 End-point security

One of the main assumptions underlying video encryption is that the endpoints (the camera and the mobile phone) are secure.
This is a strong but critical assumption for the confidentiality of the audio and video. The camera and the app both have access to unencrypted video data. This means that if an attacker has full access to the camera or to the mobile phone, then they have access to video data.

We allocate significant resources to prevent security vulnerabilities in our devices, apps and the Hue cloud. We are also proactive to fix and patch any identified vulnerability (see https://www.philips-hue.com/en-us/support/security-advisory for more information). The purpose is to ensure that the camera and our app are free of any known software vulnerabilities that could impact the confidentiality of the audio and videos. Fixes for vulnerabilities in Philips Hue products are released using firmware updates. These updates are signed by Philips Hue using cryptographic signatures to ensure that only legitimate software updates can be installed.

Users can maximize the security of their system by ensuring that their Hue devices and mobile OS are up to date. Users should only download the Hue app from the official iOS or Android app store and also ensure to keep the Hue app up to date. Furthermore, users should refrain from rooting or jailbreaking their phone as this may compromise the OS security and hence the security of the Hue app.

5.2 Hue accounts and passphrases

The Hue account is protected with an account password with the possibility of adding a second factor of authentication. Users are strongly encouraged to enable and use the second factor of authentication as this greatly improves account protection. It is the user's responsibility to choose a strong non-guessable password and protect that password to prevent others to access their account. The Hue account restricts access to encrypted videos generated in the home to users with the role of admin or advanced security. Even though these encrypted videos cannot be viewed without the knowledge of the E2EE passphrase, users must be careful with their authentication information.

The E2EE passphrase protects the confidentiality of audio and video data that is stored in the cloud. Therefore, it must only be shared with users that are trusted to view this data. If an attacker has access to the E2EE passphrase and encrypted videos, then the attacker can decrypt and view these videos. We advise to create a backup of the E2EE passphrase so that access to videos can be regained if the user loses his phone. This backup should be stored in a secure location that is not easily accessible to others. In case of a suspected breach of the E2EE passphrase, the user can change the passphrase and update their cameras to use keys derived with the new passphrase.

One limitation of the current key management is that there is only a single E2EE app key derived from passphrase that is shared with all users in the home. This means that E2EE keys cannot be revoked or renewed for specific users because users do not have an individualized E2EE keys. They can, however, be revoked by changing the E2EE passphrase for a home. Introducing individualized E2EE app keys would lead to other complexities in the system: users would not be able to decrypt old videos, adding a user would involve adding the new keys in every enrolled camera, and revocation of keys would involve complex interactions and asynchronous key updates throughout the system. Using a single app key instead of individualized app keys avoids this complexity. The user always has the option to update the E2EE passphrase if desired, although this involves updating all cameras in the home as well as sharing the E2EE passphrase to the other (remaining) users in the home. Finally, sharing the E2EE passphrase between users in the system serves as a backup of the E2EE passphrase making it less likely to be lost.

5.3 Exported videos

Exported videos are decrypted before they are shared by the user. They are no longer protected using the E2EE encryption keys. After sharing an exported video with another user, it can be further shared without restrictions.

5.4 Metadata

Metadata associated with videos is not end-to-end encrypted. This means that, although the video/audio data itself is encrypted using the E2EE keys, there is other associated data that is stored in the Hue cloud. Examples of such data are the events that triggered a recording, a timestamp for a recording, the length of the recording, and the size of the video itself. This data is protected using standard encryption techniques and account management, but is not additionally protected using E2EE keys.