CSAM Detection

Technical Summary

August 2021

Contents

Introduction

CSAM Detection enables Apple to accurately identify and report iCloud users who store known Child Sexual Abuse Material (CSAM) in their iCloud Photos accounts. Apple servers flag accounts exceeding a threshold number of images that match a known database of CSAM image hashes so that Apple can provide relevant information to the National Center for Missing and Exploited Children (NCMEC). This process is secure, and is expressly designed to preserve user privacy.

CSAM Detection provides these privacy and security assurances:

- Apple does not learn anything about images that do not match the known CSAM database.
- Apple can't access metadata or visual derivatives for matched CSAM images until a threshold of matches is exceeded for an iCloud Photos account.
- The risk of the system incorrectly flagging an account is extremely low. In addition, Apple manually reviews all reports made to NCMEC to ensure reporting accuracy.
- Users can't access or view the database of known CSAM images.
- Users can't identify which images were flagged as CSAM by the system.

For detailed information about the cryptographic protocol and security proofs that the CSAM Detection process uses, see The Apple PSI System.

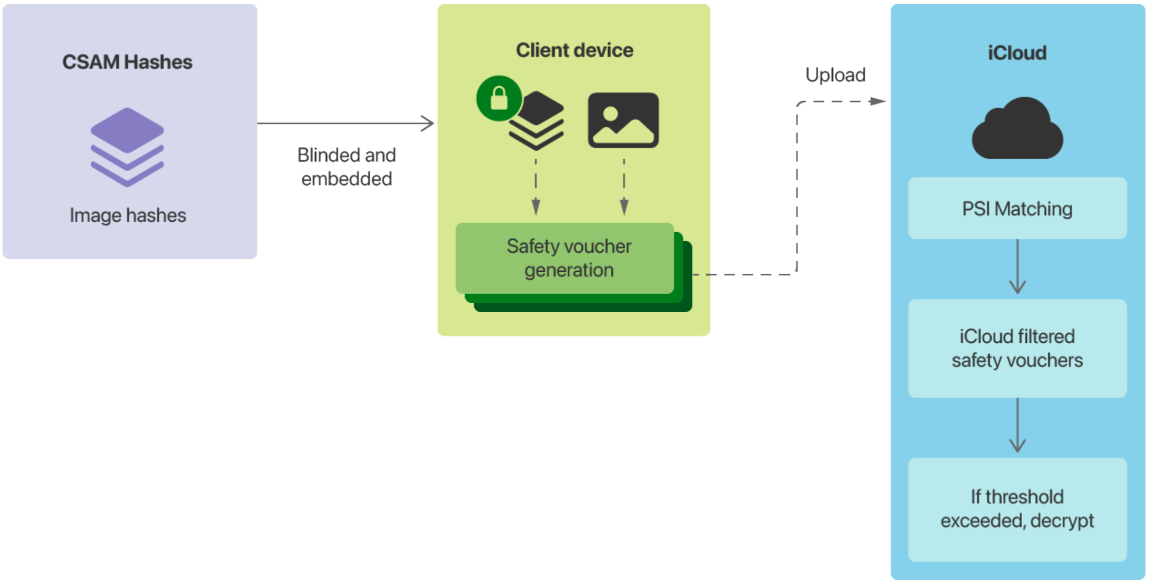
System Overview

Apple's method of detecting known CSAM is designed with user privacy in mind. Instead of scanning images in the cloud, the system performs on-device matching using a database of known CSAM image hashes provided by NCMEC and other child-safety organizations. Apple further transforms this database into an unreadable set of hashes, which is securely stored on users' devices.

The hashing technology, called NeuralHash, analyzes an image and converts it to a unique number specific to that image. Only another image that appears nearly identical can produce the same number; for example, images that differ in size or transcoded quality will still have the same NeuralHash value.

Before an image is stored in iCloud Photos, an on-device matching process is performed for that image against the database of known CSAM hashes. This matching process is powered by a cryptographic technology called private set intersection, which determines whether there is a match without revealing the result. The device creates a cryptographic safety voucher that encodes the match result. It also encrypts the image's NeuralHash and a visual derivative. This voucher is uploaded to iCloud Photos along with the image.

Using another technology called threshold secret sharing, the system ensures that the contents of the safety vouchers cannot be interpreted by Apple unless the iCloud Photos account crosses a threshold of known CSAM content. Only when the threshold is exceeded does the cryptographic technology allow Apple to interpret the contents of the safety vouchers associated with the matching CSAM images.


CSAM Hashes
Image hashes
Blinded and embedded
Client device
Safety voucher generation
Upload
iCloud
PSI Matching
iCloud filtered safety vouchers
If threshold exceeded, decrypt

The threshold is selected to provide an extremely low (1 in 1 trillion) probability of incorrectly flagging a given account. This is further mitigated by a manual review process wherein Apple reviews each report to confirm there is a match, disables the user's account, and sends a report to NCMEC. If a user feels their account has been mistakenly flagged they can file an appeal to have their account reinstated.

CSAM Detection will be included in an upcoming release of iOS 15 and iPadOS 15.

The next section describes in detail the underlying technologies and how they work together in the safety voucher process.

Technology Overview

This system combines three technologies: NeuralHash, Private Set Intersection, and Threshold Secret Sharing.

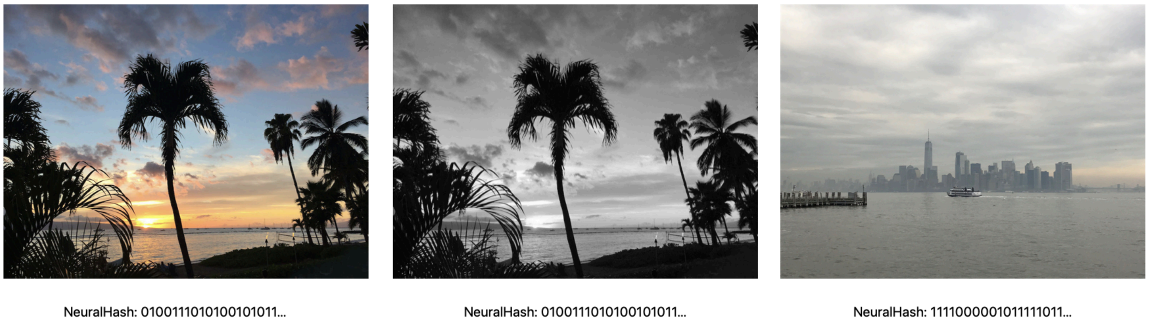
NeuralHash

NeuralHash is a perceptual hashing function that maps images to numbers. Perceptual hashing bases this number on features of the image instead of the precise values of pixels in the image. The system computes these hashes by using an embedding network to produce image descriptors and then converting those descriptors to integers using a Hyperplane LSH (Locality Sensitivity Hashing) process. This process ensures that different images produce different hashes.

The embedding network represents images as real-valued vectors and ensures that perceptually and semantically similar images have close descriptors in the sense of angular distance or cosine similarity. Perceptually and semantically different images have descriptors farther apart, which results in larger angular distances. The Hyperplane LSH process then converts descriptors to unique hash values as integers.

For all images processed by the above system, regardless of resolution and quality, each image must have a unique hash for the content of the image. This hash must be significantly smaller than the image to be sufficiently efficient when stored on disk or sent over the network.

The main purpose of the hash is to ensure that identical and visually similar images result in the same hash, and images that are different from one another result in different hashes. For example, an image that has been slightly cropped or resized should be considered identical to its original and have the same hash.


NeuralHash: 0100111010100101011...
NeuralHash: 0100111010100101011...
NeuralHash: 1111000001011111011...

Original image on the left is nearly identical to the center image, therefore both images have the same NeuralHash. The image on the right has different content and hence a different hash.

The system generates NeuralHash in two steps. First, an image is passed into a convolutional neural network to generate an N-dimensional, floating-point descriptor. Second, the descriptor is passed through a hashing scheme to convert the N floating-point numbers to M bits. Here, M is much smaller than the number of bits needed to represent the N floating-point numbers. NeuralHash achieves this level of compression and preserves sufficient information about the image so that matches and lookups on image sets are still successful, and the compression meets the storage and transmission requirements.

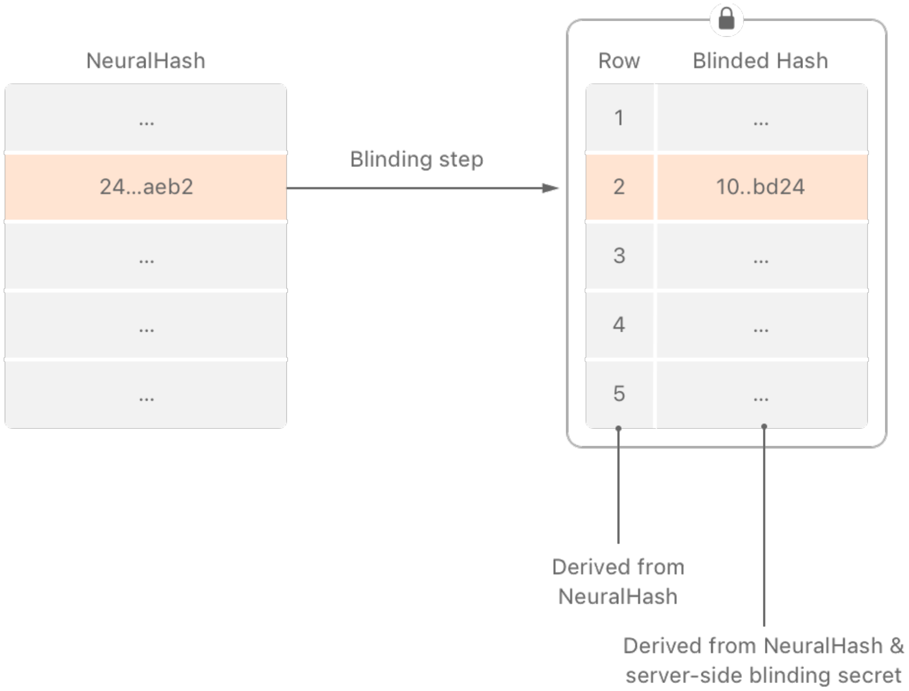
The neural network that generates the descriptor is trained through a self-supervised training scheme. Images are perturbed with transformations that keep them perceptually identical to the original, creating an original/perturbed pair. The neural network is taught to generate descriptors that are close to one another for the original/perturbed pair. Similarly, the network is also taught to generate descriptors that are farther away from one another for an original/distractor pair. A distractor is any image that is not considered identical to the original. Descriptors are considered to be close to one another if the cosine of the angle between descriptors is close to 1. The trained network's output is an N-dimensional, floating-point descriptor. These N floating-point numbers are hashed using LSH, resulting in M bits. The M-bit LSH encodes a single bit for each of M hyperplanes, based on whether the descriptor is to the left or the right of the hyperplane. These M bits constitute the NeuralHash for the image.

Private Set Intersection (PSI)

Private Set Intersection (PSI) is a cryptographic protocol that two parties use, for example Apple servers and a user's device. Before the protocol begins, Apple and the user's device have distinct sets of image hashes that each system computed using the NeuralHash algorithm. The system applies PSI in conjunction with other cryptographic techniques like Threshold Secret Sharing, described in the next section. The PSI protocol ensures that Apple learns the image hashes in the intersection of the two sets, but learns nothing about image hashes outside the intersection.

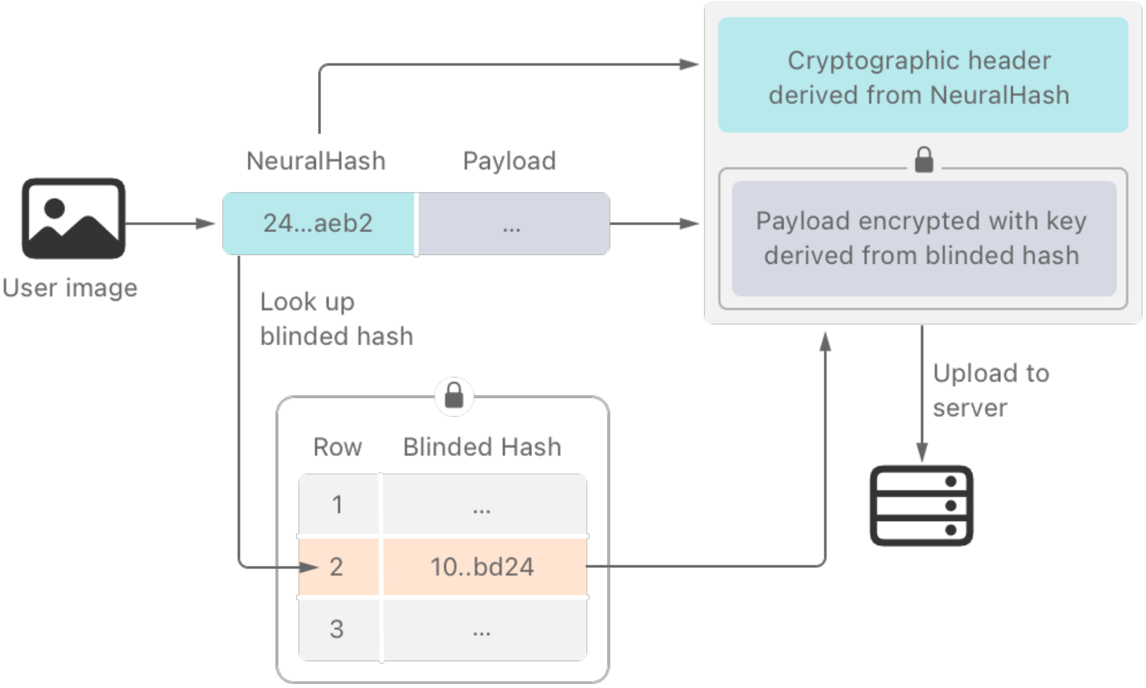
The system used for CSAM Detection extends this basic PSI mechanism to support the client including additional payload data associated with each image hash, and guarantees that this additional payload is only accessible for image hashes in the intersection of the two sets. The next section describes the main steps in the PSI mechanism: the matching database setup, the on-device PSI protocol, and the server-side PSI protocol.

Matching-Database Setup. The system begins by setting up the matching database using the known CSAM image hashes provided by NCMEC and other child-safety organizations. First, Apple receives the NeuralHashes corresponding to known CSAM from the above child-safety organizations. Next, these NeuralHashes go through a series of transformations that includes a final blinding step, powered by elliptic curve cryptography. The blinding is done using a server-side blinding secret, known only to Apple. The blinded CSAM hashes are placed in a hash table, where the position in the hash table is purely a function of the NeuralHash of the CSAM image. This blinded database is securely stored on users' devices. The properties of elliptic curve cryptography ensure that no device can infer anything about the underlying CSAM image hashes from the blinded database.



On-Device PSI Protocol. Given a user image, the general idea in PSI is to apply the same set of transformations on the image NeuralHash as in the database setup above and do a simple lookup against the blinded known CSAM database. However, the blinding step using the server-side secret is not possible on device because it is unknown to the device. The goal is to run the final step on the server and finish the process on server. This ensures the device doesn't know the result of the match, but it can encode the result of the on-device match process before uploading to the server.

Before an image is stored in iCloud Photos, the following on-device matching process is performed for that image against the blinded hash table database. The device computes the image NeuralHash and looks up the entry in the blinded hash table at the position pointed by the NeuralHash. The device uses the computed NeuralHash to compute a cryptographic header. It also uses the blinded hash that the system looked up to obtain a derived encryption key. This encryption key is then used to encrypt the associated payload data.



Next, the client creates a cryptographic safety voucher that has the following properties:

- If the user image hash matches the entry in the known CSAM hash list, then the NeuralHash of the user image exactly transforms to the blinded hash if it went through the series of transformations done at database setup time. Based on this property, the server will be able to use the cryptographic header (derived from the NeuralHash) and using the server-side secret, can compute the derived encryption key and successfully decrypt the associated payload data.
- If the user image doesn't match, the above step will not lead to the correct derived encryption key, and the server will be unable to decrypt the associated payload data. The server thus learns nothing about non-matching images.
- The device doesn't learn about the result of the match because that requires knowledge of the server-side blinding secret.

Finally, the client uploads the image to the server along with the voucher that contains the encrypted payload data and the cryptographic header.

Server-Side PSI Protocol

The server, upon receiving the voucher, will attempt to compute the decryption key for the associated payload ciphertext by using the cryptographic header in the voucher. If the user image is a match, the server will be able to combine the cryptographic header along with the server-side blinding secret to successfully decrypt the associated data ciphertext. If not, the decryption attempt will fail. In particular, the server learns the associated payload data for matching images, but learns nothing for non-matching images.

Threshold Secret Sharing

Threshold Secret Sharing is a cryptographic technique that enables a secret to be split into distinct shares so the secret can then only be reconstructed from a predefined number of shares (the threshold). For example, if a secret is split into one-thousand shares, and the threshold is ten, the secret can be reconstructed from any eleven of the one-thousand shares. However, if only ten shares are available, then nothing is revealed about the secret.

The CSAM Detection system uses Threshold Secret Sharing to protect information about images stored in iCloud Photos when the number of matching images has not crossed a certain threshold. Only once the number of matches exceeds the threshold will the secret-sharing reconstruction algorithm enable the system to learn the additional data the client included with each of the matching images. Nothing is ever revealed about non-matching images during any step of the CSAM Detection process.

Combining Private Set Intersection and Threshold Secret Sharing

Private Set Intersection (PSI) and Threshold Secret Sharing are used in combination to ensure Apple's systems cannot access the additional data the client included with the image until the specific criteria are met.

The PSI protocol described so far does not introduce the concept of a threshold and would allow decryption of a payload whenever there was a match. In order to provide the threshold guarantees stated here, the system combines PSI with Threshold Secret Sharing as follows.

As part of setup, the device generates an encryption key for the user account, unknown to Apple. For each image, it encrypts the relevant image information (the NeuralHash and visual derivative) using this key. Next, it generates a secret share of the key on demand for that image. Note that the secret share is independent of the actual image. This secret share and the encrypted image information are included in the associated payload of the PSI protocol. This combined packet with the relevant image information protected by two layers of encryption is referred to as the safety voucher for the image.

Before a threshold is crossed, the server learns the associated payload for matches: a secret share for the user account's encryption key, and the encrypted image information. In other words, the outer layer of encryption is decrypted using the PSI protocol for CSAM matches. Because of the property of Threshold Secret Sharing, at this stage Apple does not have any knowledge about the underlying user encryption key, due to lack of sufficient secret shares. Once more than a threshold number of matches has occurred, Apple has enough shares that the server can combine the shares it has retrieved, and reconstruct the decryption key for the ciphertexts it has collected, thereby revealing the NeuralHash and visual derivative for the known CSAM matches. In other words, the inner layer of encryption is decrypted using Threshold Secret Sharing. It is important to point out that for accounts that cross the threshold, recovery of this encryption key does not lead to any information about non-matching images because the PSI protocol protects the outer layer of encryption for such matches.

Synthetic Match Vouchers

In the process described above, Apple could learn the possible count of CSAM images without the ability to decrypt the vouchers to confirm the content. Because Apple can't verify the contents of matching images until the threshold is crossed, it is important to obfuscate the CSAM count from Apple until then.

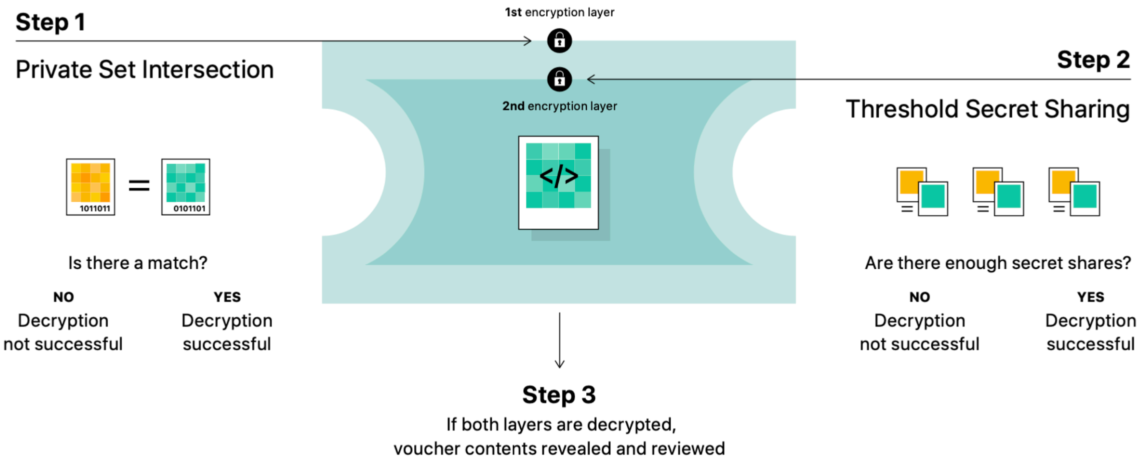
To achieve this, the device occasionally produces synthetic vouchers for images as opposed to ones corresponding to their image. Synthetic vouchers are designed to register as matches within the PSI protocol. The probability that a device uploads a synthetic voucher instead of a real voucher for an image is calibrated to ensure the total number of synthetics is of the same order of magnitude as the threshold. This provides uncertainty about the true number of matches until the threshold is exceeded.

Synthetic vouchers differ from real vouchers in the following ways: (1) the underlying image information is substituted by dummy data; (2) the secret share of the inner key is substituted by a random share that is totally independent of the inner encryption key; and (3) the cryptographic header and the outer encryption key are chosen to always result in a match on the server.

Property (1) ensures these vouchers don't have any meaningful content in them. Property (2) ensures those matches do not contribute to Apple's ability to decrypt the ciphertexts, but it is not possible to distinguish them from those matches that do until the threshold is crossed. Property (3) ensures that these vouchers always register as a match, thereby providing enough noise to the system to ensure Apple doesn't learn about potential CSAM counts with any meaningful degree of certainty until the threshold is crossed.

Safety Voucher Mechanics

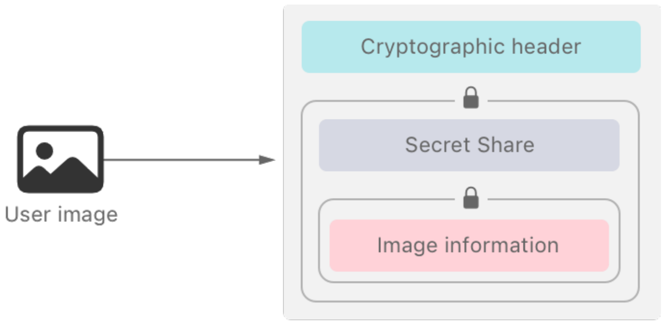
This section summarizes how all three technologies come together in the context of safety vouchers. A safety voucher is a data record associated with each image in the iCloud Photos account. It is generated on-device by the CSAM Detection process. The vouchers cryptographically encode whether an image has a match within the known database of blinded hashes of CSAM images, as well as corresponding additional data (the NeuralHash and the visual derivative). At a high level, the output of the PSI protocol reveals to the server whether there is a match. This leads to decrypting the outer layer of encryption. Once there are more than a threshold number of matches, secret sharing allows the decryption of the inner layer, thereby revealing the NeuralHash and visual derivative for matching images.



On-Device Voucher Generation

For each image to be stored in iCloud Photos, the device generates a safety voucher that contains image-related information (NeuralHash + visual derivative) protected by two layers of encryption: first, by the PSI protocol for the associated payload data, and second by the encryption scheme used in Threshold Secret Sharing, to encrypt the image information contained within. In addition, the first layer also encrypts a secret share of the encryption key used in the second layer.

In more detail, the device starts by generating an encryption key for the user account, unknown to Apple. For each user image, it encrypts the relevant image information (the NeuralHash and visual derivative) using this key. This forms the inner layer encryption (as highlighted in the above figure). A secret share of the key and the encrypted image information is included in the associated payload of the PSI protocol.

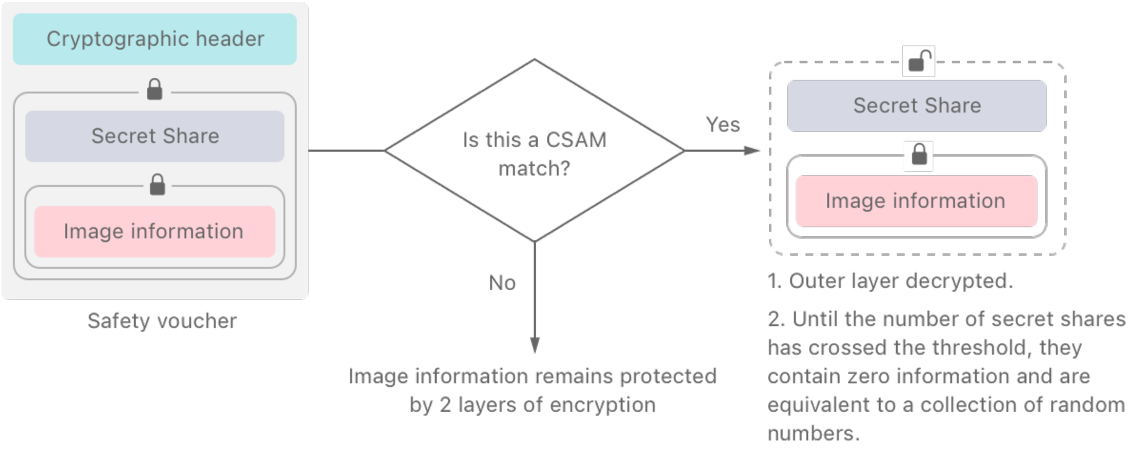


Safety voucher

Next, as part of the PSI protocol, the device computes its NeuralHash and looks up the entry in the blinded hash table at the position pointed by the NeuralHash. The device uses the computed NeuralHash and the blinded value from the hash table to compute a cryptographic header and a derived encryption key. This encryption key is then used to encrypt the associated payload data. This forms the outer layer of encryption for the safety voucher.

Outer-Layer Unwrapping of Vouchers in iCloud

The output of PSI protocol on the server reveals whether there is a match or not. For CSAM matches, the cryptographic header in the voucher combines with the server-side blinding secret (that was used to blind the known CSAM database at setup time) to successfully decrypt the outer layer of encryption. For non-matches, this decryption is not possible. The decrypted results for matches are the secret share and inner ciphertext (encrypted image information). It is important to note that until a threshold is crossed, secret shares contain absolutely no information. Because of the property of Threshold Secret Sharing, they are equivalent to a set of arbitrary, randomly chosen numbers.


Cryptographic header
Secret Share
Image information
Safety voucher
Is this a CSAM match?
Yes
No
Secret Share
Image information
Image information remains protected by 2 layers of encryption
1. Outer layer decrypted.
2. Until the number of secret shares has crossed the threshold, they contain zero information and are equivalent to a collection of random numbers.

In summary, for non-matches, the image information in the vouchers remains doubly encrypted because the outer layer cannot be decrypted. For matches, the image information remains encrypted by the inner layer.

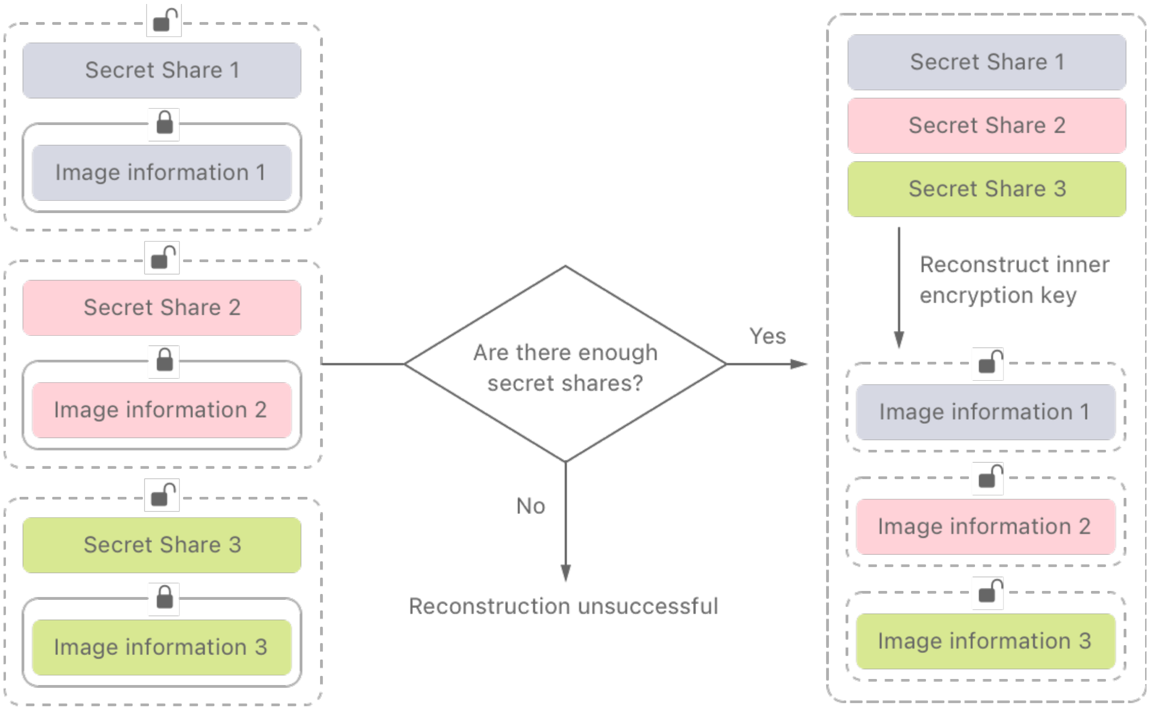
Inner-Layer Unwrapping of Vouchers in iCloud

Recall that the cryptographic technique of Threshold Secret Sharing enables a secret to be split into distinct shares so the secret can then only be reconstructed from a predefined number of secret shares (the threshold). For example, if a secret is split into one-thousand shares, and the threshold is ten, then the secret can be reconstructed from any eleven of the one-thousand shares. However, if only ten shares are available, then nothing is revealed about the secret.

In this system, the secret is a device-generated inner encryption key that is used to encrypt the image information (NeuralHash + visual derivative) for all of the user’s images. The safety voucher for an image contains this encrypted image information and a secret share of this inner encryption key embedded inside the outer layer of PSI encryption.

Every time there is a match, the PSI protocol decrypts the outer layer of encryption providing a secret share and the encrypted image information for the matched image. As highlighted before, at this stage, before a threshold is crossed, the collection of secret shares contains zero information and is equivalent to a collection of arbitrary random numbers.

Because of the property of Threshold Secret Sharing, once an iCloud Photos account accumulates more than a threshold number of vouchers corresponding to images that are CSAM matches, the server has a sufficient number of secret shares and can use it to reconstruct the inner-layer decryption key. The server then uses the decryption key to decrypt the inner encryption layer and extract the NeuralHash and visual derivatives for the CSAM matches. Only those images that have a voucher that corresponds to a true CSAM match can have their vouchers’ data decrypted.



For simplicity's sake the figures above omit the synthetic vouchers described in the previous section that are introduced by the device so that Apple does not learn the potential count of CSAM matches before exceeding the threshold.

Nothing is learned about non-matching images. Even if the device-generated inner encryption key for the account is reconstructed based on the above process, the image information inside the safety voucher for non-matches is still protected by the outer layer of encryption. Thus, with a combination of Private Set Intersection and Threshold Secret Sharing, Apple is able to learn the relevant image information only once the account has more than a threshold number of CSAM matches, and even then, only for the matching images.