FAQ: Chat Control Client-side Scanning

In this FAQ we summarize the key points in the letter signed by more than 700 expert scientists around the globe on the issues with the Danish Presidency's compromise proposal for the Regulation to prevent and combat child sexual abuse (Child Sexual Abuse Regulation) from July 2025.

Prof. Anja Lehmann (HPI and University of Potsdam, Germany), Dr. Anne Canteaut (Inria, France), Prof. Aurélien Francillon (EURECOM, France), Prof. Bart Preneel (KU Leuven, Belgium), Prof. Carmela Troncoso (Max Planck Institute for Security and Privacy, Germany), Prof. Cas Cremers (CISPA Helmholtz Center for Information Security*, Germany), Prof. Christian Makenz (Technical University of Denmark, Denmark), Dr. Gaëtan Leurent (Inria, France), Prof. Olivier Pereira (UCLouvain), Prof. Thorsten Holz (Max Planck Institute for Security and Privacy, Germany),*

What is Client-side scanning in the context of Chat Control?

Client-side scanning is a technique that monitors messages (text, images, audio, links) on user's phones and laptops and checks whether they are considered harmful according to a policy. In the case of the Chat Control regulation, this harmful content is material related to Child Sexual Abuse. If the material is considered harmful, service providers are obliged to report violative content.

When messages are sent via an encrypted messenger (such as WhatsApp, iCloud or Signal), client-side scanning analyzes the messages *before* they are encrypted. If the content of the message is considered harmful, the message is sent to law enforcement. If not, the message is encrypted and sent to the recipient.

Through time, the scope of the regulation has varied in covering different types of content, with the latest version from July 2025 focusing solely on links and images. This FAQ applies to all types of content.

How do CSAM detectors work? Are they reliable?

Child Sexual Abuse Material (CSAM) detectors are implemented in one of three ways, depending on what material they are targeting.

Exact detection of known versions of known CSAM. Detectors can identify exact copies of images that have been seen before and were confirmed to be CSAM. For example, an image that is shared without modifications, or a link to CSAM material that is exchanged verbatim.

To perform this identification, the detector would use so-called cryptographic hash functions. A hash function is a function that takes an input (e.g., an image or a link) and outputs a short string. The hash function ensures that the mapping is almost one to one (i.e., no two images will map to the same short string), and that given the short string one cannot recover the original input.

To detect copies of known CSAM based on hash values, detectors would have a list of hash values corresponding to the known CSAM. Then, the detector would take a message, compute its hash value, and check whether this hash value is in the list. If it is, the detector can be certain that the message is a copy of the known CSAM. Revealing this hash value to law enforcement then means a third party knows the content of the message, defeating end-to-end encryption.

This detector however, can only detect exact copies. Any modifications to the image, e.g., cropping, rotating, resizing, addition of a line, changing the color of a pixel, or the compression used by all commercial messengers to save bandwidth, would make the detection fail. Thus, it is a very unreliable mechanism in the context of Chat control.

Detection of modified versions of known CSAM. These are detectors that would aim to identify variations of CSAM that have been seen before. Their goal is to overcome the limitations of detectors based on cryptographic hash functions regarding modification of the images.

These detectors use a technology called *perceptual hashing*, which also maps inputs to a short string in such a way that if two inputs are visually similar (such as a rotated image, or an image with few pixels deleted), the resulting short strings are identical or similar. These detectors have a list of perceptual hash values corresponding to the known CSAM. To decide whether an input is in the list of perceptual hash values, the detector computes the perceptual hash values of the input and checks whether this is sufficiently close to a perceptual hash value in the list. If yes, it is flagged as CSAM and reported to law enforcement as in the previous case.

Depending on how “sufficiently close” is defined, this detector might fail to detect many CSAM inputs (if close is very restrictive), detect them but also flag many non-CSAM inputs as CSAM (if close is very loose). Moreover, research has identified several major flaws in all the designs of perceptual hash functions: it is easy to avoid detection by adding modifications invisible to the human eye, it is possible to generate innocent pictures that would be classified as CSAM to arbitrarily increase the workload of law enforcement, and their false positive rate (messages that are not CSAM marked as CSAM) is so high that given the amount of material they have to analyze they would result in millions of pictures being incorrectly identified as CSAM.

Therefore, a detector based on perceptual hash functions is not reliable in the context of Chat Control, and there are serious doubts that it is even technically feasible to avoid all these flaws simultaneously. In addition, for some perceptual hash functions it may be possible to deduce some information on the image (e.g., the contour of a person) from the hash value, showing that these hash functions leak information on the input.

Detection of unknown potential CSAM. These detectors aim to identify CSAM that has never been seen. For this, they rely on patterns in CSAM material.

To detect patterns, these detectors rely on machine learning algorithms, which are trained on known CSAM (images, audio, grooming text) to learn patterns. To train the algorithm, it is necessary to convert the material into features that represent it. To decide whether an input is CSAM, the input is converted into features and these features are fed to the algorithm that outputs a decision. Because the notion of CSAM is complex and inherently contextual, such algorithms typically have low accuracy, i.e., their output is often incorrect. As an example, it is very difficult for a human to decide whether a picture of children next to a swimming pool, a picture taken by a parent for medical reasons, or sexting pictures between consenting teenagers should be reported as CSAM.

Similar to detectors based on perceptual hash functions, it is easy to manipulate inputs (e.g., modifying the values of pixels in images) such that when the input is converted into features, the patterns change enough for the detector to not flag the image. Therefore, the detector is not reliable in the context of Chat Control.

Why is client-side scanning different from malware scanning?

Intuitively, on-device CSAM detection through client-side scanning might seem similar to malware checks by antivirus software, but the two are fundamentally different. When antivirus scanning finds potential malware on a consumer device, it informs the user who is asked to make a decision about reporting. That is, malware scanning is voluntary, transparent, and not tied to law enforcement backdoors. This is not the case with client-side scanning, which implies *mandatory* on-device CSAM detection and *automated* reporting of any material matched by the algorithm to law enforcement. Thus, client-side scanning is inherently more invasive than any malware protection and more open to abuse (and unreliable as explained above).

Does client-side scanning break the confidentiality provided by end-to-end encryption?

Encryption is the tool we use to provide confidentiality of communications. Encryption converts messages into unintelligible scrambled data, such that only those who know the decryption key can tell what is in the messages. End-to-end encryption additionally means that no one other than the sender and intended receiver of the communication can tell what is in the messages.

The client-side scanning in the proposed regulation scans messages, and informs law enforcement of those deemed potentially harmful. This breaks the confidentiality condition of end-to-end encryption: a third party who is not the sender or receiver -- in this case law enforcement officers -- will learn what is in the message.

This is *independent* of the kind of detector used. Even when only hash values are input to the detection, all messages from all users will be read and, when a match is found, the content of the message will be known outside the sender and recipient devices, and will not be confidential anymore. This is especially relevant when there are no ways to technically constrain or audit the content for which the users' private messages are scanned (see point below).

Thus, client-side scanning that can inform a third party (e.g., law enforcement) about the content of messages completely breaks the guarantees of end-to-end encryption.

Can client-side scanning be constrained to only detect for CSAM?

Client-side detectors identify material that is either in a list (for cryptographic and perceptual hash values), or whose patterns coincide with those in the training data. The detecting algorithms are independent of what material is in the list, or was used to train the algorithm. If the list or the training dataset contains content associated to terrorism, anti-government stances, or protest organization, the detection mechanism would perform the same actions.

Thus, once client-side detectors are deployed there are no technical means of constraining their use, as this only depends on the *configuration* they are provided with, which can easily be changed via a remote update. These could be legitimate updates pushed by those deploying client-side scanning, or illegitimate updates by actors exploiting a vulnerability of the system. Given the sensitivity of CSAM, the detection technologies must not allow for reconstruction of the content they are checking for, which makes it impossible for users to check that the detectors only check for CSAM, and not more.

More information about all the issues above can be found in this paper.