# Preemptive Monitoring in End-to-end Encrypted Services

## Internet Society Technical Feasibility Evaluation

June 2024

## Summary and Recommendations

Technical measures to screen the content of messages in end-to-end encrypted (E2EE) systems introduce systemic risk for both service providers and users. They frustrate law-abiding users' intent to communicate privately and interfere with their ability to do so in practice. Systemic functions for consentless scanning lay the foundations for numerous attacks with serious and widespread impacts.

In technical terms, such measures compromise the integrity of devices and systems, increasing the risk of system-wide attacks and unauthorized access to personal data, whether accidental or malicious. This undermines the trustworthiness of the online environment, has serious economic and cybersecurity implications, and creates new opportunities for criminals to exploit. This will make it harder, not easier, for law enforcement to achieve the stated goals of the Online Safety Act (OSA) (UK 2023).

In this briefing, we outline commonly proposed approaches to client-side scanning and set out the systemic vulnerabilities[1] and risks they introduce. We identify several factors on which technical feasibility will depend but which are not considered in the legislation. The risks we identify include:

- Attacks on personal sensitive data.
- Distributed denial-of-service attacks.
- Manipulation of CSEA databases.
- Reverse engineering.
- Attacks on alerts sent for data processing.

The identified risks, systemic vulnerabilities, and other factors represent a serious obstacle to meeting the technical feasibility requirement placed on Ofcom by the OSA.

[1] We define a systemic vulnerability as one that extends beyond the targeted device or service that an individual user is using and is implemented such that any other user could be affected.

We recommend assessing candidate technologies in this area using the framework developed by the National Research Centre on Privacy, Harm Reduction, and Adversarial Influence Online (REPHRAIN), which has been comprehensively tested on the prototypes developed for the government's Safety Tech Challenge. We conclude that client-side scanning should be rejected as an approach because it is incompatible with a safe overall finding of technical feasibility.

## The Context of Monitoring Without Consent

The principle of client-side scanning (CSS) is simple: it is the monitoring of content on a user's device (the "client") by someone or something other than the user. In the context of the OSA, it has been proposed as an alternative to monitoring the user's content on the server. Variations of how CSS might be implemented include:

- Whether the user's informed consent has been sought and given.
- Whether the results of scanning are communicated to the user, third parties, or both.
- Whether or not such scanning is evident to the user.
- The nature of the information being scanned for (text, images, etc.).
- Whether the scanning is indiscriminate, applied to all content of a particular type (such as text or images), or targeted at a specific piece of content.

The OSA includes powers relating to terrorist content. However, unlike material relating to child sexual exploitation and abuse (CSEA), the Act does not create the power to require private communications to be scanned for terrorist content. Therefore, we do not address terrorist content in this document.

In the context of the OSA, CSS has been proposed to detect three kinds of content relating to child abuse: previously identified CSEA material, previously unknown CSEA material, and semantic content indicative of "grooming." Since these kinds of content give rise to a criminal offence under the Act, the presumption is that CSS will take place whether or not the user consents. CSS is consentless, preemptive surveillance of the citizen.

## Effectiveness and the Technical Feasibility Requirement

Ofcom has been tasked with deciding whether a given CSS technology is technically feasible before accrediting it and subsequently mandating its use. We believe it is important to note that feasibility depends on factors other than the technology itself, especially concerning error rates (so-called false positive and false negative results).

If a CSS mechanism generates more errors than law enforcement resources can process, we believe it should not be considered technically feasible. The error rates for automated scanning will increase as

one progresses from known CSEA to unknown CSEA to semantic analysis of potential "grooming." What may appear to be a vanishingly small error rate quickly turns unmanageable at scale (e.g., a 0.01% error rate across a billion messages means 100,000 extra alerts requiring human moderation)(Anderson 2022)..

If a technology is ineffective, it is also almost certain to fail the necessity and proportionality test required by Article 8 of the Human Rights Act 2018. First, it is illogical to argue that an ineffective technology is also necessary. Second, a partly or mostly ineffective technology is unlikely to be proportionate, particularly if its use has harmful side effects (such as violating the privacy of law-abiding users).

## Technical Feasibility and "UK-linked" Content

The Act stipulates that only "UK-linked" content must be reported to the National Crime Agency (NCA) but includes more than one set of criteria for determining whether the content is "UK-linked." Some of those criteria are subjective and vaguely defined. For example, one criterion for "UK-linked" content is that it is "likely to be of interest to a significant number of UK users." How can a service provider assess whether content is "likely to be of interest" and what constitutes "a significant number" of users? Is it a number (500, 5,000, 5 million), or is it a percentage of the service's users (e.g., 500,000 users may represent less than 2% of a platform's users)?

Other criteria include the nationality/location of the alleged offender, the location of the supposed victim, and "the place where content was generated." For most of these criteria, the scanned content is an unreliable source of evidence. For example, if an image depicts the interior of a room, there may be no indication of where that room is. Similarly, it is not reliable to assume that because an image was sent from, say, London, it was generated in London.

This is relevant to the question of technical feasibility because the more criteria are applied and the more vague or subjective the evidence, the more likely the system is to generate false positives. The more false positives, the greater the burden on law enforcement resources to assess and correct errors, and the greater the risk that people may be falsely accused of serious offenses.

The "UK-linked" stipulation may, paradoxically, also further undermine claims of the OSA's proportionality because of the extent of personal data it requires the service provider to collect, process, and retain about law-abiding users *just in case* it is subsequently needed to establish that a communication is "UK-linked." Very few online services collect data that would accurately establish a user's nationality, so the feasibility of meeting the "UK-linked" criterion would depend on a significant expansion of the personal data collected by every regulated site or service: that, too, is likely to cause the OSA to fail the proportionality test.

# Data Processing on the Device

The policy push for CSS has been a response to the increased use of end-to-end encryption (E2EE) because the purpose of E2EE is to prevent eavesdropping on users' communications by intermediaries (such as Internet service providers, mobile carriers, or messaging servers). CSS seeks to overcome this constraint by inspecting users' data on the user's device before it is encrypted. An inspection might be on the basis of so-called "perceptual hashes," detection of skin tones, or semantic analysis using machine learning tools such as "large language models," depending on the type of content targeted. This has allowed policymakers to claim that they are leaving the encryption untouched, but this claim is misleading: users' confidentiality is violated nonetheless.

## On-device scanning gives attackers a target they control

When CSS is proposed as a systemic way to detect illegal content, the assumption is that it runs whether or not the user consents since a detection mechanism that the user is free to turn off is unlikely to be effective for law enforcement purposes. This means that a mandatory scanning mechanism is present on every affected consumer device. We must also assume that criminals and hostile foreign governments are aware of this fact and have the means and opportunity to exploit the resulting "attack surface." This could include actions such as reverse engineering detection mechanisms, content-matching databases, and reporting mechanisms.

Hostile exploitation of the attack surface has two principal consequences: first, it may render CSS ineffective as a detection tool (for instance, if an attacker can manipulate the local hash-matching process); second, it puts a systemic vulnerability on a whole class of devices, exposing millions of people to bugging by unauthorized entities, including criminals and foreign states. Thus, CSS may fail to meet the stated policy objectives while at the same time increasing the likelihood of cybersecurity attacks, censorship, and repression.

## Processing limitations when using multiple databases

CSS needs access to databases against which to compare scanned content. Multiple databases to address different content categories increase the complexity of deployment. The task of keeping multiple such databases up to date on millions of "UK" devices (however that criterion is decided) has serious implications for network traffic, cost, and extra processing on the device. These create issues of scalability, testing, device and data integrity, and governance, potentially eroding the effectiveness of CSS as an enforcement mechanism.

# Data Processing Off-device

An alternative proposal is to minimize the processing performed on the client device. For example, when a user attaches an image to a private message, a hash of the image is generated before encryption, and the hash is sent to a server-side database for matching. This decreases the amount of data that must be distributed to users' devices but introduces new systemic risks and vulnerabilities.

First, this is vulnerable to any attack that can suppress or modify the alerts sent to the server. Suppression or modification of alerts could be used to circumvent detection, while modification could be used to create fake alerts, generating added workload to deal with the resulting false positives.

Second, it introduces a new disclosure (of the hash to be matched), which did not happen in the "all on the device" architecture. This means a third-party server now collects sensitive personal data about an individual's device usage, creating new vulnerabilities and exposing law-abiding users to new attacks.

Third, sending an alert from a user's device to a third-party server increases network traffic (and, therefore, the cost and economics of mobile service provision). If tampered with, it represents a vector for distributed denial-of-service (DDoS) attacks[2], such as overwhelming the third-party server (or intermediate network components) with spurious traffic.

## Perceptual Hashing

Perceptual hashing creates an approximated fingerprint of images uploaded and checks them against a database of perceptual hashes of images classified as illegal.

### Perceptual hashing and cryptographic hashing are very different

It is important not to confuse perceptual and cryptographic hashing, particularly when the viability of a given technical solution is based on claims of the robustness and reliability of the hashing mechanisms it uses. The strength of a cryptographic hash is *not* an indicator of the reliability of perceptual hash-matching or the hash-matching system as a whole.

Cryptographic hashing is designed to protect data against undetected modification. Changing a single bit in the source data should create the "avalanche effect"—that is, it should result in massive change to the resulting cryptographic hash. This means that strong cryptographic hashes are extremely resistant to "collisions." It is very unlikely that two inputs can be found which result in the same cryptographic hash.

[2] Denial-of-service attacks seek to make a resource unavailable to its intended users by overwhelming the targeted system with apparently valid requests.. Distributed denial-of-service attacks are similar, except that the incoming traffic flooding the targeted system comes from not one but a variety of different sources and is therefore more difficult to stop.

By contrast, perceptual hashing is designed to be resilient against changes to the source data: that is, a perceptual hash should identify two images as being the same, even if one of them has been modified (for instance, in its color palette or aspect ratio). This means it is *desirable* for more than one input to match the resulting hash.

However, CSEA detection tools must be assumed to use both kinds of hash: perceptual hashes to achieve image matching and cryptographic hashes to protect the perceptual hashes against undetected modification (since, if the perceptual hashes can be modified without detection, the system is fatally compromised).

This means we have a series of elements as follows:

1. The source images used as input to:
2. Perceptual hashes (which are *designed* so that multiple inputs can produce the same hash value), used as input to:
3. Cryptographic hashes (which are designed to minimize collisions).

These steps are illustrated in the diagram below:

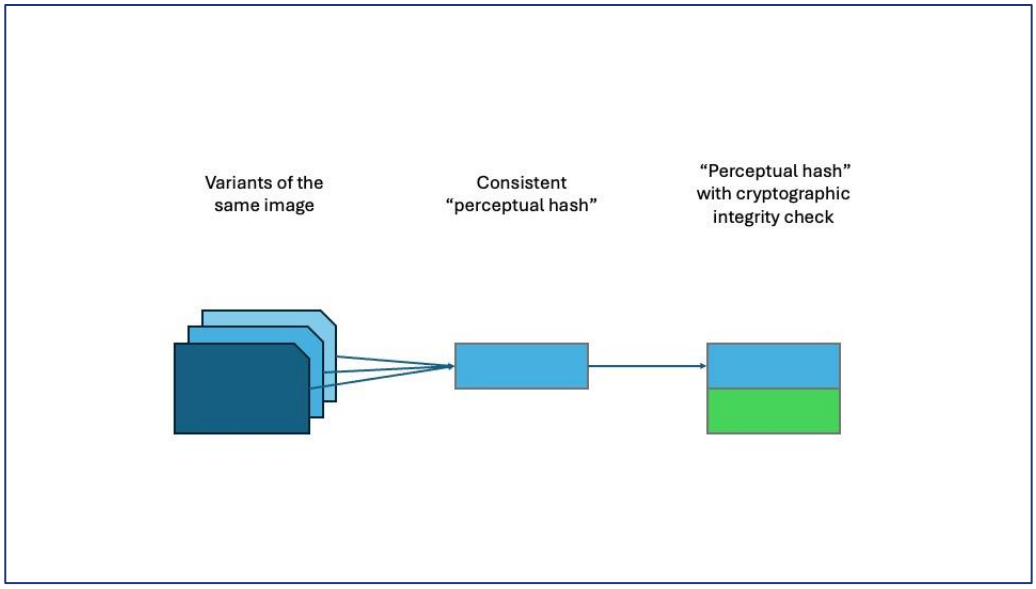


*Figure 1: Steps in the hashing process*

The nature of the input data to each element matters. If strong cryptographic hashes were being computed on the source images themselves, one would expect an extremely low collision rate. If they are being calculated on perceptual hashes, which themselves have a high collision rate, we must expect the rate of false positives in the CSEA detection system to be significantly higher than the collision rate of the cryptographic hashing algorithm used. *Therefore, the collision rate of the cryptographic hashing algorithm is not a reliable indicator of the system's overall accuracy.*

The proliferation of databases is itself a source of concern even to law enforcement officers, who privately acknowledge that managing consistency and integrity across multiple distributed databases is an obstacle to efficacy. There is currently no recognized national or international governance or accreditation regime for existing CSEA databases, with current databases developing and implementing their own cybersecurity measures. We have also been made aware that existing CSEA databases do not all use current, strong cryptographic hashing algorithms, and as a result, some databases are more likely than others to produce false positives, even among cryptographic hashes.

A growing body of research points to safety (Hooda et al. 2024) and reliability (Struppek 2022) issues with both perceptual hashing and client-side scanning.

### Flooding CSEA databases with false positives

A weakness of the defense against false positives in CSEA detection is that it is based on security through obscurity. It depends on maintaining the secrecy of a large number of proliferating databases of valid hashes, which is not a strong defense against attacks on such databases. A breach of an entire database of valid hashes would clearly be catastrophic for law enforcement, but even a breach of small numbers of valid hashes can cause major disruption. They allow criminals to mount a distributed denial of service (DDoS) attack by flooding the system with false positives.

Again, this is an example of how the technical feasibility of a given CSS mechanism cannot be judged in isolation: apparently, workable mechanisms can be infeasible in practice because of vulnerabilities in other elements of the CSEA detection and enforcement process.

### Generation of false negatives

By definition, hash-matching only works for already-known CSEA. Even relatively simple forms of image manipulation can defeat the “perceptual hash” technique, allowing previously unknown CSEA to be shared and to avoid being added to the “known CSEA” database.

## CSS as a Vehicle for Other Attacks

Client-side scanning elements on a device need privileged access to system functions (for instance, to ensure that the user cannot prevent CSS from working correctly). CSS elements must also be remotely maintained and updated. This combination of requirements creates an attractive target for malicious intervention. The SolarWinds cyber-attack is an example of how remote maintenance services can be exploited (Oladimejo 2023) and how damaging and far-reaching the consequences can be.

CSS creates opportunities for other kinds of attack, such as inserting unauthorized material into the hash-matching database for scanning. Researchers in the UK have illustrated how this could be used

surreptitiously to implement facial recognition systems, far exceeding what was envisaged for the OSA (Jain 2023). The same approach can be used to re-purpose the database to scan for other forms of content.

Deploying CSS creates a system-wide toolkit for malicious activity and repressive policies of surveillance and censorship.

## Conclusions and Recommendations

As of April 2024, then-government ministers continued to make claims about the effectiveness of CSS:

*"The UK Government has already funded the development of tools to detect child sexual abuse in end-to-end encrypted environments. The experts are clear: it is perfectly possible to take the fight to predators while ensuring privacy for the rest of us". – Tom Tugendhat, former Minister of State for Security* (Hymas 2024)

As this briefing has shown, this is by no means the expert consensus, either technically or practically. Our first recommendation is to advise caution for policymakers making statements that portray unrealistic or incorrect expectations as political facts.

It is, however, useful that Mr. Tugendhat mentions government-funded attempts to develop tools in this area. In its independent review of those attempts, the National Research Centre on Privacy, Harm Reduction and Adversarial Influence Online (REPHRAIN) recognized the tensions between protecting vulnerable users on the one hand and safeguarding privacy and security on the other. REPHRAIN concluded that an assessment based on the usual current criteria (classification accuracy, false positive rates, and usability) cannot safely resolve those tensions.

REPHRAIN used a multi-stakeholder process to define a set of nine criteria and tested them in its evaluation of the government-funded development projects (Peersman et al. 2023). At the time of the report (Feb 2023), none of the proof-of-concept offerings satisfied those criteria, which is a legitimate reason to question the claims made by Mr. Tugendhat, as cited above..

Our second recommendation is that the full set of REPHRAIN criteria be used to assess any candidate technologies in this area. The criteria are fully documented in the REPHRAIN report and are summarized below in Appendix A.

Based on the evidence provided in this brief, CSS would compromise the integrity of devices and systems, leaving them open to system-wide attacks, as well as malicious and accidental breaches of personal data. We believe this means that CSS technologies, by their nature, are incompatible with an overall finding of technical feasibility. Therefore, our final and over-arching recommendation is that Ofcom not accredit CSS technologies because of the inherent systemic vulnerabilities they introduce.

# Appendix A – REPHRAIN Evaluation Criteria

Source: (Peersman 2023)

| | | |
|---|---|---|
| 1 | Human-centered, grounded in human rights, human dignity and individual autonomy | This criterion assesses the purpose, design, and implementation of the system |
| 2 | Human rights impact: privacy, protection of personal data, freedom of expression | This criterion assesses the impact of the deployment, use, and operation of the system |
| 3 | Security | This criterion considers "secure from what?" (including, for instance, DoS and DDoS attacks, hash poisoning, insider attacks), and assesses threat and vulnerability models |
| 4 | Effective performance, robustness and scalability | This is the functional assessment without which a technical feasibility appraisal is incomplete |
| 5 | Explainability, transparency, auditability and provenance | This assessment establishes whether there is effective governance in the *design* of the system |
| 6 | Disputability and accountability | This assesses whether there is effective governance in the *operation and use* of the system |
| 7 | Fairness/non-bias | This represents the correct application of principles 1, 2, 5, and 6 |
| 8 | State of the art | This criterion allows an assessment of the technical feasibility of individual elements of the system, as distinct from the operational feasibility of the system as a whole |
| 9 | Maintainability | This represents the correct application of principles 3, 4, and 8. |

These evaluation criteria are only a first step. To assess the technical feasibility of CSS, there must also be a process for assessing whether the technology, *as implemented, deployed and used*, does what it was expected to do, results in effective enforcement, and has the desired societal outcomes.

# References

Anderson, Prof. R. 2022. 'Chat Control or Child Protection?': https://arxiv.org/pdf/2210.08958, accessed 15 July 2024

Hooda A, Labunets A, Kohno T, and Fernandes E (2024) 'Experimental Analysis of the Physical Surveillance Risks in Client-Side Content Scanning', Proceedings of the Network and Distributed Systems Security Symposium 2024: https://www.ndss-symposium.org/wp-content/uploads/2024-1401-paper.pdf, accessed 15 July 2024

Hymas C (23 April 2024) 'Children as young as three "tricked into producing online sexual images"', https://www.telegraph.co.uk/news/2024/04/23/children-young-as-three-tricked-make-online-sexual-images/, *The Telegraph*, accessed 15 July 2024

Jain S, Crețu A-M, Cully A, and de Montjoye Y-A (2023) 'Deep perceptual hashing algorithms with hidden dual purpose: when client-side scanning does facial recognition', *IEEE Symposium on Security and Privacy 2023, pp. 234-252*, https://ieeexplore.ieee.org/document/10179310, accessed 15 July 2024

Oladimejo S, Kerner S 'SolarWinds hack explained: Everything you need to know': https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know, *TechTarget*, accessed 15 July 2024

Peersman C, Llanos J, May-Chahal C, McConville R, Chowdhury P, and De Cristofaro E (2023) 'REPHRAIN: Towards a Framework for Evaluating CSAM Prevention and Detection Tools in the Context of End-to-end encryption Environments: a Case Study', https://bpb-eu-w2.wpmucdn.com/blogs.bristol.ac.uk/dist/1/670/files/2023/02/Safety-Tech-Challenge-Fund-evaluation-framework-report.pdf, accessed 15 July 2024

Struppek L, Hintersdorf D, Neider D, and Kersting K (2022) 'Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash', *ACM FAccT* 2022: https://arxiv.org/abs/2111.06628, accessed 15 July 2024

UK. 2023. *Online Safety Act 2023*, Schedule 4(2)(c): https://www.legislation.gov.uk/ukpga/2023/50/schedule/4/paragraph/2, accessed 15 July 2024