This document has not been adopted or endorsed by the European Commission and is intended as a basis for discussion. It may not be shared further without permission of the European Commission services.

Technical solutions to detect child sexual abuse in end-to-end encrypted communications

1. INTRODUCTION

Purpose

The purpose of this paper is to **gather expert input to inform policy** to ensure the privacy of citizens (including children) and the protection of children against sexual abuse and sexual exploitation.

Scope

This paper covers the **proactive detection**[1] **by companies** of images, videos and **text-based**[2] child sexual abuse such as grooming or sextortion.

The scope of the paper is limited to one specific type of online service, **electronic communications**, and one specific type of illegal content, **child sexual abuse (CSA).**

The focus on electronic communications is due to the fact that a large proportion of reports to the National Centre for Missing and Exploited Children (NCMEC) of instances of CSA (around 2/3 of the 16.9 million reports received in 2019, more than 800k of which concerned the EU) originate in this type of online service. These include one to one instant messaging services and email.

The focus on CSA is due to several reasons:

- the material (images and videos) identified as CSA (legally referred to as "child pornography") is context independent, unlike other types of illegal content such as terrorist material;

[1] The document focuses on detection as a first step to tackle this complex problem. The reporting of child sexual abuse after it has been detected is not covered in this document at the moment but it is of course of utmost importance to ensure that actionable and valuable information is provided to law enforcement on a timely basis.
Also, the document covers proactive detection by companies, not lawful access by law enforcement with a warrant.
The document currently does not cover either the process to develop the technical solutions (e.g. data to train and test the tools, the preparation and maintenance of the database of hashes, etc), also of key importance.
Also, the document focuses on solutions that work on real time detection, rather than detection of CSA in messages that have already been sent to the recipient.

[2] The technologies and approaches required to detect text-based threats are in general different from those required to detect images and videos. At the moment, the detection of text-based threats is more difficult and presents a higher number of false positives than image and video detection. It is therefore not easy to bundle the assessment and recommendations for text, image and video detection. The assessment of the solutions and the recommendations presented in the paper focuses mostly on image and video detection.

- it is the only illegal content whose mere possession is illegal;
- industry has been using tools to detect instances of CSA voluntarily for years, as the fight against this type of illegal content has been the least controversial;
- because of this effort, there is data available to assess the scope of the problem and the impact of foregoing detection measures.

This paper:

- **defines the problem** of the detection of CSA content in end-to-end encrypted (E2EE) communications; and
- presents a number of possible **technical solutions** that could allow the detection of CSA in E2EE communications.

A possible solution is one that allows the detection of CSA in E2EE electronic communications using **existing technologies** (e.g. hashing), as well as upcoming technologies, to the extent that these may be known today.

The paper aims to provide **a first technical assessment** to help identify possible solutions. **Substantial additional work**, beyond the scope of this paper, is likely to be needed to further evaluate, develop, and deploy the technical solutions across the companies' infrastructure.

Approach

The approach of the paper is purely **technical**. It aims to reflect in a **non-technical language** the input from top technical experts from academia, industry and public authorities from around the world, who have kindly contributed with their time and knowledge to help make progress on this matter.

The paper maps possible technical solutions and assesses them from a technical point of view across five criteria (the order does not reflect any considerations on relative importance):

1. **Effectiveness**: how well does the solution **detect and report** known and unknown CSA (images, videos and text-based threats)?[3]
2. **Feasibility**: how ready is the solution and how easily can it be implemented, in terms of **cost, time and scalability**?[4]
3. **Privacy**: how well does the solution ensure the **privacy of the communications**?[5]
4. **Security**: how vulnerable is the solution to be **misused** for other purposes than the fight against CSA, including by companies, governments or individuals?[6]

[3] This includes the ability to report to law enforcement sufficient information to enable the rescue of children from ongoing abuse and the prosecution of the offenders, as well as the ability of companies to proactively stop the abuse of their infrastructure to commit CSA related crimes. Also, a solution is more effective if it enables the detection of CSA with multiple technologies (e.g. image and video hashing, AI, etc).

[4] A solution ready to be implemented also ensures optimal user experience (e.g. no reduction of performance).

[5] This refers solely to the ability of the technical solution to ensure that neither the company, nor any actor outside the sender and the receiver has access to the content of the communication.

[6] This includes, e.g., the misuse by companies to detect other types of content; the misuse by governments for mass surveillance; the misuse by individuals to cause damage exploiting possible weaknesses that the solution may inadvertently introduce in the infrastructure; and the misuse by individuals to compromise the integrity of the solution to detect CSA and modify it so that it would not work as intended. It is important to

5. **Transparency**: to what extent can the use of the solution be documented and be **publicly reported** to facilitate **accountability** through **ongoing evaluation and oversight** by policymakers and the public?[7]

2. PROBLEM DEFINITION

The problem that this document aims to address is the following: given an E2EE electronic communication, are there any technical solutions that allow the detection of CSA content while maintaining the same or comparable benefits of encryption?

In addition to the technical aspects of the problem, which are the focus of this paper, the problem has important **policy** aspects, as it lies at the core of the privacy vs safety debate. Some voices on the safety side of the debate push for forbidding E2EE altogether or require the existence of generalised exceptional access mechanisms, whereas some voices on the privacy side would reject any solution that allows the detection of CSA in E2EE communications, as they would put the privacy of communications above anything else.

This document aims at mapping possible solutions that could ensure the privacy of electronic communications (including the privacy of children) **and** the protection of children against sexual abuse and sexual exploitation. The solutions explored are purely technical in nature, and this paper does not take a position on the related policy aspects.

3. POSSIBLE SOLUTIONS

0) Baseline solutions

These are immediate solutions that require little or no technical development. They provide reference points for comparison to the other technical solutions.

a. Non-E2EE communications

In communications that are not end-to-end encrypted (but may be encrypted with other client to server protocols), the electronic service provider (ESP) has the ability to apply various tools to detect CSA (images, videos or text) on its server. The most common ones are:

- Hashing tools[8]: they convert the image (or video) into a unique alphanumeric sequence (hash), which is compared with a database of known images and videos identified as CSA.
- Text-based tools: they detect keywords or text patterns that indicate possible CSA (e.g. grooming or sextortion).

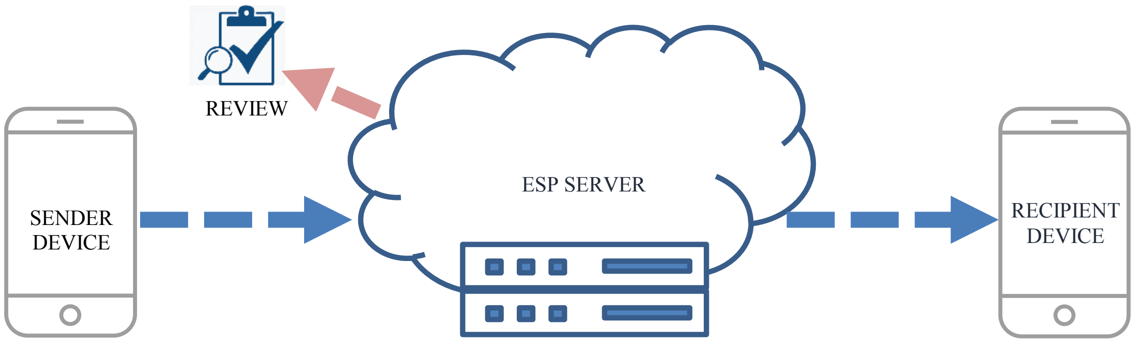
note that tech-savvy offenders (who may compromise the solution) are unlikely to use systems that allow the detection of CSA.

[7] Carnegie Endowment for International Peace, *Moving the Encryption Policy Conversation Forward,* Encryption Working Group, September 2019, p14.

[8] The most widely used hashing tool is PhotoDNA, developed by Microsoft and Professor Hany Farid in 2009. See here for more information on how PhotoDNA works.

If the tools identify possible CSA, the message is flagged for manual review by content moderator or reported directly to the authorities.

Figure 1: detection of CSA in communications that are not end-to-end encrypted



Device: sends non E2EE message

ESP server: applies tools to detect CSA:

a) Detected ⇨ message forwarded for review/reporting

b) Not detected ⇨ message forwarded to recipient

Device: receives non E2EE message

Assessment:

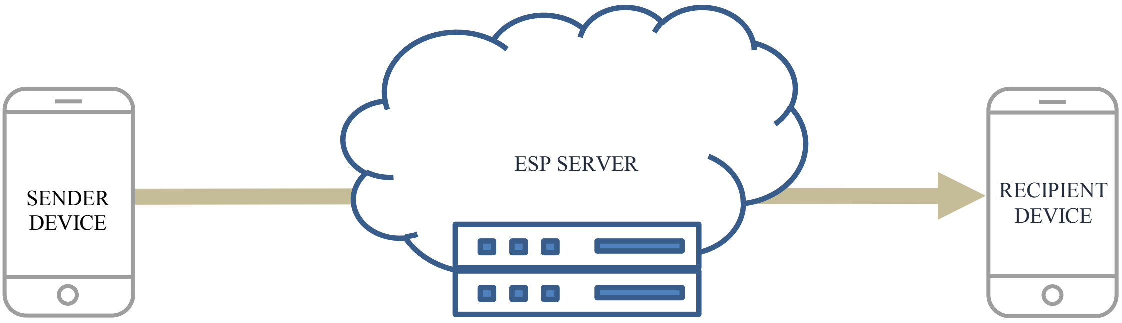
- Effectiveness:
  - High: highly effective in detecting and reporting known child sexual abuse material (CSAM) and text-based threats (i.e. as effective at detecting and reporting unknown CSAM as the current technology to detect it allows).
- Feasibility:
  - High: already in use, frequently as the default option.
- Privacy:
  - Low: the content of the communication could in principle be accessed by the ESP at any point (from a technical point of view).
- Security:
  - Medium: whereas companies can access the content of the communication for other purposes than the detection of CSA, the communication is relatively secure from unauthorised access by governments and individuals (given the use of e.g. client-server encryption).
- Transparency:
  - Medium: whereas the use of tools to detect CSA can be publicly reported (i.e. reports sent to NCMEC), it is not always clear whereas these or similar tools are used to detect other types of content, illegal or not, as oversight mechanisms not always exist.

b. End-to-end encrypted communications[9]

In end-to-end encrypted communications the sender and recipient utilize a public key protocol to agree on a secret session key, which no passive observer including the ESP can determine.

As such, the server is not able to apply the tools to detect CSA, since it does not have the private decryption key and thus no access to the content in clear.

Figure 2: detection of CSA in end-to-end encrypted communications



Device: sends E2EE message	**ESP server:** cannot apply existing tools to detect CSA at the server on E2EE messages	**Recipient:** receives and decrypts E2EE message

Assessment:

- Effectiveness:
  - None, as it is not possible to detect CSA (images, videos and text-based threats) at the server.
- Feasibility:
  - Not applicable (detection of CSA is not possible).
- Privacy:
  - High: the content of the communication can only be accessed by the sender and the recipient of the message.[10]
- Security:
  - Not applicable, since there is no solution to detect CSA that can be compromised.
- Transparency:
  - Not applicable, since the detection of CSA is not possible.

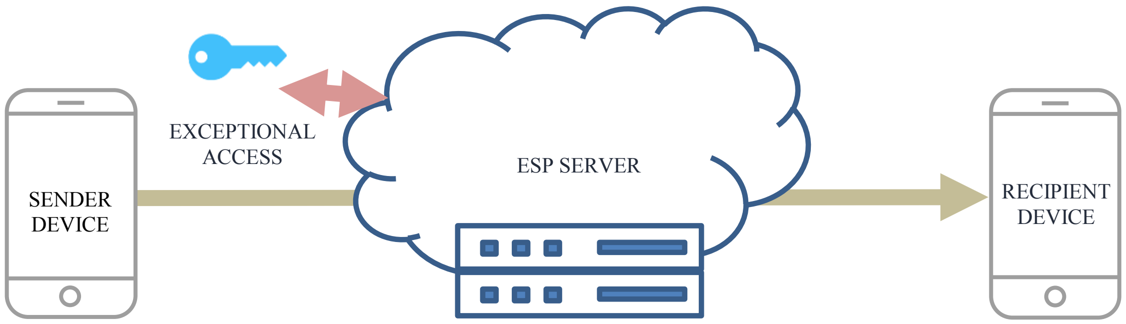
[9] This baseline solution does not include device, server and encryption related solutions, which will be analysed in the rest of the document.

[10] The only part of the communication that is not private, as in all the other solutions discussed in this document, is the fact that the sender sent a message to the recipient (metadata/traffic data).

c. End-to-end encrypted communications with exceptional access

In this type of solutions, the electronic communications system includes the possibility of exceptional access for the company and law enforcement (e.g. with a warrant), i.e. the possibility to decrypt the content of the communication:

Figure 3: detection of CSA in E2EE communications with exceptional access



Device: sends E2EE message

ESP server: cannot apply existing tools to preventively detect CSA on E2EE messages. Can only access content of a specific communication, via exceptional access

Device: receives and decrypts E2EE message

Assessment:

- Effectiveness:
  - Low: preventive detection (i.e. to reduce the proliferation of CSA and report to law enforcement for action as needed) is not possible. Detection of CSA is only possible for a specific communication, via the exceptional access.
- Feasibility:
  - Medium: the solution could be implemented but the use of exceptional access can be expensive.
- Privacy:
  - Low: all the content of the communication could in principle be accessed by the ESP at any point (from a technical point of view) using the exceptional access mechanism.[11]
- Security:
  - Medium/Medium-Low: a reasonable expectation for a standard design is to be able to prevent unauthorised access, i.e. prevent hacking the server-side implementation or cryptographically impersonating the ESP. That said, it could be difficult to decide who gets the exceptional access and who does not.
- Transparency:
  - Medium: the authorised use of the exceptional access could be reasonably documented and be publicly reported.

****

[11] This assumes that the exceptional access system is based on the ESP having the encryption keys.

There are three basic elements in an end-to-end encrypted communication: device, server and encryption type (see figure 2). These basic elements also determine the three possible types of technical solutions beyond the baseline ones: 1) **device** related, 2) **server** related, and 3) **encryption** related solutions, which the following sections will analyse.[12]

1) Device related solutions[13]

This type of solutions consists in moving to the **device** some or all of the operations done at the ESP server in communications that are not end-to-end encrypted.

The solutions where the device is involved could work both with the sender's device as well as with the recipient's device. Setting the solutions up on the sender's side helps limit the distribution of illegal material, whereas setting them up on the recipient's side helps with detecting grooming. Also, implementing detection solutions on both the sender and receiver's device might mitigate the risk of offenders modifying their apps to defeat the detection mechanisms.

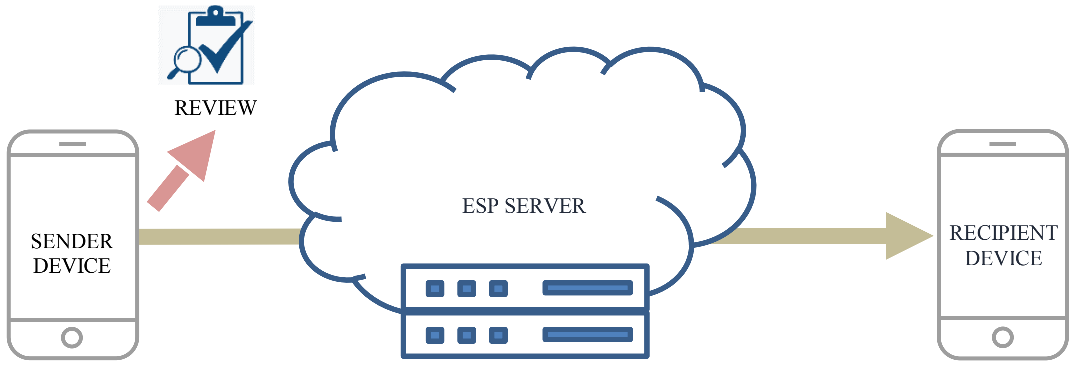
a. All detection done on-device

In this solution, all the main operations done at the server, i.e. hashing and matching for images and videos, and matching for text, are moved to the device, and applied on the message before it is encrypted. If the tools detect child sexual abuse, the message is sent for manual review (or reporting). If they do not, the message is end-to-end encrypted and sent to the recipient:

[12] Some of these solutions refer to the use of hashes. Hashes can be cryptographic (a small change in the image generates a new hash) or perceptual/robust (a small change in the image does not change the hash). Perceptual hashing has higher effectiveness but somewhat lower feasibility, as the hash set size is larger and more space is needed for the matching process. Cryptographic hashes would reduce effectiveness but be more feasible. The assessment assumes perceptual hashing unless stated otherwise.

[13] The detection tools could in principle be incorporated either at the app or the operating system level (although in the latter it could be more technically complex). It might be easier for the ESP to check against manipulation of the detection tools before allowing the operation if they are incorporated at the app level but incorporating the solutions in the operating system may be more effective and efficient to implement.

Figure 4: all detection done on-device



Device: applies tools to detect CSA (e.g. hashing of images and matching with a database of known CSAM). If CSA is:

a) not detected ⇨ encrypts message end-to-end and sends to recipient

b) detected ⇨ sends message for review and/or reporting

ESP server: cannot apply existing tools to detect child sexual abuse on end-to-end encrypted messages

Device: receives and decrypts message

Assessment:

- Effectiveness:
  - Medium-high: it would allow the detection of known CSAM. Depending on the type of device, the list of hashes may need to be limited to work properly.[14] Updating the hashset with new hashes is slower and thus less effective than a model where the hashset is in the ESPs cloud.
- Feasibility:
  - Medium-low: it could be implemented relatively easily but it would require significant storage space in the device with the current technology[15]. Updating regularly the database would also use computational capacity.
- Privacy:
  - Medium: user data is not exposed to the ESP. The possible security issues (compromise and manipulation of detection tools) may introduce vulnerabilities that could decrease the privacy of the communication.
- Security[16]:
  - Low: the solution could be easily subverted and compromised/reverse engineered to not detect or report child sexual abuse (in particular in devices without trusted

[14] That said, in the case of PhotoDNA, the additional time needed to compare hash databases of increasing size scales logarithmically, not linear. In other words, doubling the size of the database requires one extra comparison, not twice as many.

[15] For example, PhotoDNA hashes could be between 1 to 4 million, which could take around 30MB. Adding video hashes would take even more storage space. Feasibility may be increased by limiting the hash database to include only hashes of the most commonly encountered content or manage the dataset on a device/operating system level.

[16] The security of all solutions that make use of a hashing algorithm could be increased if that algorithm is updated/modified periodically, to reduce the risk of reverse engineering. Ideally, an open-source hashing algorithm very difficult to hack would be best, but it remains to be developed.

execution environments). It could also be manipulated to introduce false positives to inundate the reporting systems (e.g. NCMEC) with them. The possible leak of detection tools (e.g. hashing algorithm, hash list, keywords list), could reduce the effectiveness of similar detection tools elsewhere.

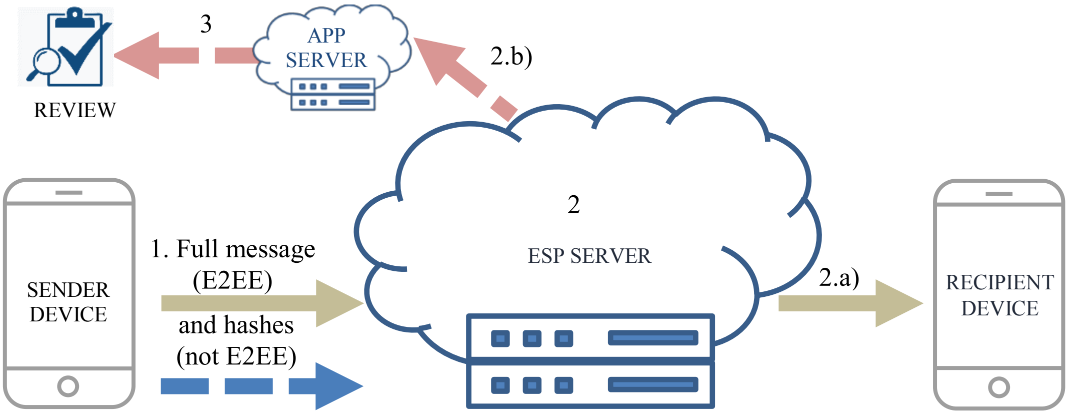
- Transparency:
  - Medium-low: the possible security issues could limit the reliability of public reporting on the use of the solution and therefore the accountability.

b. On-device full hashing with matching at server

In this solution, the device converts the images and videos in the message into hashes, encrypts the message and sends the (client to server encrypted) hashes and the full message encrypted to the server. The server compares these hashes with those in the database of hashes of confirmed child sexual abuse (matching).

If there is a hit at the server, it instructs the app server to send the full image (or video) for manual review (or reporting). If there is no hit, the server forwards the E2EE message to the recipient.

Figure 5: on-device hashing with matching at server



1. **Device**: converts the images and videos into hashes before the message is encrypted, encrypts the full message and sends the hashes (client to server encrypted) and the E2EE full message to the server.

3. **App server:** sends full image/video for review if there is a match in the server and/or reporting

2. **ESP server:** compares hashes received from the device with those in the database of hashes of confirmed CSA (matching)

a) No match ⇨ forwards E2EE message to recipient

b) Match ⇨ asks app server to send image/video to review and/or reporting

Device: receives and decrypts E2EE message

Assessment:

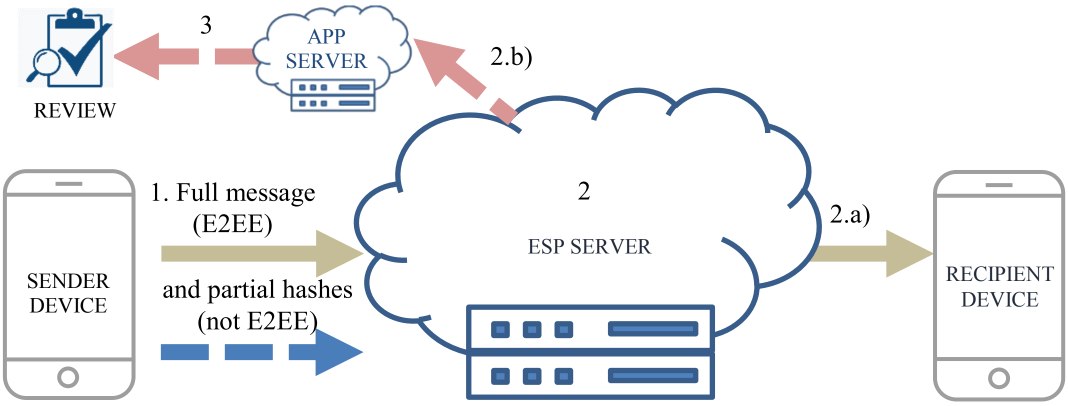
- Effectiveness:
  - Medium-high: it would allow the detection of known CSAM only. It would not be applicable to text-based threats (not possible to detect with hashing). No need to limit the hash list, as it will be located at the server.
- Feasibility:
  - High: it could be implemented relatively easily. An open-source version of the solution could be created to be used by smaller companies.
- Privacy:
  - Medium: user data (hashes) are visible to the ESP. The possible security issues (compromise and manipulation of detection tools) may introduce vulnerabilities that could decrease the privacy of the communication.
- Security:
  - Medium-low: the hashing algorithm in the device could be subverted and compromised/reverse engineered to not detect or report child sexual abuse (in particular in devices without trusted execution environments). It could also be manipulated to introduce false positives to inundate the reporting systems (e.g. NCMEC) with them. Also, the hash database in the ESP server could be manipulated to introduce non-CSAM hashes. The possible leak of detection tools (e.g. hashing algorithm), could reduce the effectiveness of similar detection tools elsewhere. Also to consider is the possibility that tech-savvy offenders (who may compromise the solution) would not use any system that allows the detection of CSA. These solutions are more likely to be used by non tech-savvy offenders (as is the case of most CSA detected and reported today).
- Transparency:
  - Medium: the possible security issues could limit the reliability of public reporting on the use of the solution and therefore the accountability.

c. On-device partial hashing with remaining hashing and matching at server

This solution is the same as the previous one (1.b.) but in this case part of the hash is generated at the device and the rest at the server, where the matching also takes place[17]. This hybrid approach makes the process lighter and more secure:

[17] The process to create a hash has several steps: downsize the image, convert it to greyscale, etc… (see here for an illustration of the process). In this solution, the first steps to generate the hash are executed at the device and the remaining steps at the server.

Figure 6: on-device partial hashing with remaining hashing and matching at server



1. **Device**: converts the images and videos into partial hashes before the message is encrypted, encrypts the full message and sends the partial hashes (client to server encrypted) and the E2EE full message to the server.

3. **App server:** sends full image/video for review and/or reporting if there is a match in the server

2. **ESP server**: finalises the partial hashes received from the device, and compares the now full hashes with those in the database of confirmed CSA (matching)

a) No match ⇨ forwards E2EE message to recipient

b) Match ⇨ asks app server to send image/video to review and/or reporting

Device: receives and decrypts E2EE message

Assessment:

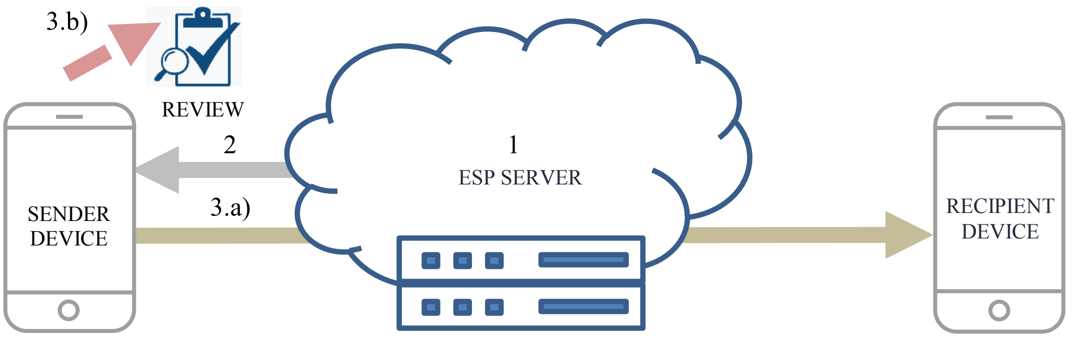
- Effectiveness:
  - Medium-high: it would allow the detection of known CSAM only. It would not be applicable to text-based threats (not possible to detect with hashing). No need to limit the hash list, as it will be located at the server.
- Feasibility:
  - Medium: proof of concept was done and it could be already in use. Depending on the size of the partial hash (which would determine the payload and upload time), this solution may be faster than 1.b. as it would lift some of the hashing burden from the device. The exact implementation details are important (e.g. to maximize performance) and remain to be defined.
- Privacy:
  - Medium: user data (hashes) are visible to the ESP and more information about the image is exposed to the ESP through the partial hash. The possible security issues (compromise and manipulation of detection tools), although improve by exposing the hashing algorithm only partially to, still may introduce vulnerabilities that could decrease the privacy of the communication.
- Security:
  - Medium: the device contains only part of the hashing algorithm, which limits the risks of reverse engineering and manipulation. This risk might be further mitigated through obfuscation techniques to scramble pixels without affecting the creation of the hash to ensure that the hash is not reversible.

- Transparency:
  - Medium: the possible security issues could limit the reliability of public reporting on the use of the solution and therefore the accountability.

d. On-device use of classifiers

In this solution, the server produces classifiers to identify child sexual abuse (images, videos and/or text) using extensive labelled data of verified child sexual abuse and non-child sexual abuse to train the machine learning system. A classifier is a set of characteristics that can determine whether the contents of a message are child sexual abuse related. The classifiers are then fed to the sender's device, which uses them to determine whether a message should be sent for review or reporting.

Figure 7: use of machine learning classifiers



3. **Device**: applies classifiers to detect child sexual abuse before the message is encrypted. If CSA is:
a) not detected ⇨ encrypts message E2E and sends to recipient
b) detected ⇨ sends message for review and/or reporting

ESP server:
1. Trains the machine learning algorithm.
2. Feeds the classifiers to the device and keeps them up to date

Device: receives and decrypts E2EE message

Assessment:

- Effectiveness:
  - Medium-low: it is basically the only solution that allows the direct detection of unknown content[18] (in addition to known content), That said, detecting child sexual abuse images and videos using machine learning is still not sufficiently developed and generates relatively high error rates (e.g. compared to hash matching). The machine learning algorithms require well-labelled data on an ongoing basis to make sure that the models are kept up-to-date. They also require constant feedback on the quality of the classification, which is particularly difficult to consistently provide in the detection of child sexual abuse in an end-to-end encrypted system. This may result in the algorithms getting outdated relatively soon if they are not updated regularly. Classifiers are more effective in detecting text-based threats like sextortion or grooming through patterns of behaviour than images or videos (although all require well-labelled data on an

[18] Hashing can also indirectly detect new content as the known images are usually found together with new ones, which are confirmed as CSA during the manual review of the detected content.

    - ongoing basis to make sure that the models are kept up-to-date, as well as feedback on the quality of the classification).
- Feasibility:
    - Medium-low: image classifiers are already in use in cloud services by companies (e.g. to recognize commonly occurring faces in photos or doing automatic grouping of images) and to some extent they start to be used to detect CSA. That said, significant development is still required, in particular for the detection of images and videos and on the possibility of running classifiers on the client side, given the size and complexity of the models and the need for frequent updates.[19] Classifiers for the detection of text-based threats (e.g. grooming) would be more feasible.
- Privacy:
    - Medium-low: the possible security issues (compromise and manipulation of classifiers) may introduce vulnerabilities that could decrease the privacy of the communication. In the particular case of behavioural classifiers, which determine possible instances of child sexual abuse based on metadata from the user, the privacy intrusion is higher than other tools such as hashing. In addition, a possibly higher rate of false positives could result in user data (not child sexual abuse) being reported / processed / reviewed. Also the classifiers could be misused to identify a range of non-CSA activities.
- Security:
    - Medium-low: the classifiers in the device could be compromised and manipulated to avoid detection (i.e. introduce false negatives), introduce false positives to inundate the reporting systems (e.g. NCMEC) (or even be used by offenders to crawl the web to search for CSA). This kind of attack could be based on sophisticated adversarial machine learning techniques that could defeat any classifier. Being able to detect new child sexual abuse threats exposes the system to be more vulnerable to adversarial attack.
- Transparency:
    - Medium: the use of the solution could be documented and be publicly reported to facilitate accountability, but how the solutions works would be more difficult to document than e.g. 1.c.

2) Server related solutions

This type of solution consists in **moving to secure enclaves in the ESP server or to third party servers** some or all of the operations done at the ESP server in communications that are not end-to-end encrypted (e.g. client to server encrypted).

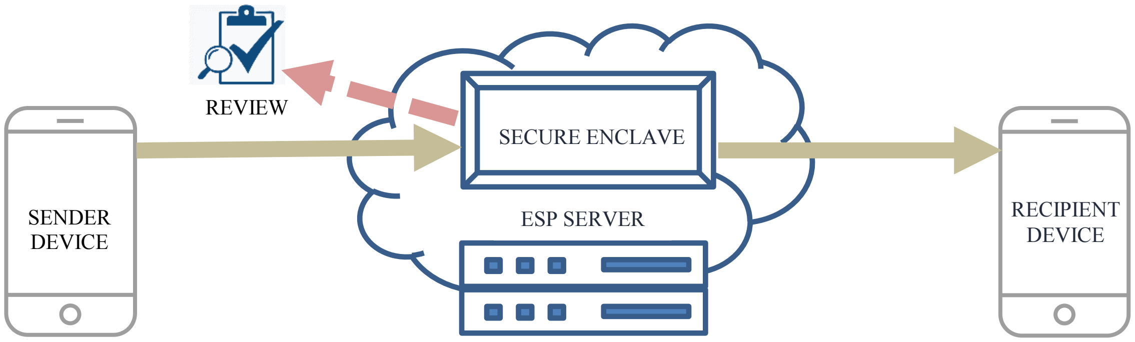
a. Secure enclaves in the ESP server

In this solution, also known as trusted execution environments or trusted platform modules (TPM), the ESP server contains a "secure enclave" that allows compute intensive operations

[19] Current image classifier models can range from 16 to 70 MB, whereas the maximum acceptable size of an app running on the device would be 4-5 MB.

to happen on the cloud, but in a closed off environment. The enclave can decrypt the user info and perform the same operations and checks as done in communications that are not end-to-end encrypted (see figure 1), while protecting the sensitive information inside the enclave:

Figure 8: secure enclaves in the ESP server



Device: sends encrypted message to the enclave in the ESP server.	**Secure enclave in the ESP server:** decrypts the message and applies tools to detect child sexual abuse. If CSA is: a) detected ⇨ forwards message for review and/or reporting b) not detected ⇨ encrypts message end-to-end and forwards it to recipient	**Device:** receives and decrypts E2EE message

Assessment:

- Effectiveness:
  - Medium-high: it could allow the detection of known and unknown CSAM. No need to limit the hash list, as it will be located at the server. It requires technology that is currently being developed. This could also open up possibilities to develop new technologies to detect child sexual abuse.
- Feasibility:
  - Medium-low: on one hand, it is a solution that simplifies the detection process and similar systems are already in use today for other applications (e.g. Intel's SGX or Software Guard Extensions, in Microsoft's Cloud[20], and other trusted execution environments). On the other hand, only a few companies have access at the moment to the hardware and software required in this solution, given its operational complexity[21] (although the technology may become more accessible in a few years in particular if it is offered as a service by the cloud providers). Also, there are compatibility issues to address in the design of the solution (i.e. the processor in the client side needs to be able to communicate with that in the enclave, and the enclaves need to be able to communicate among themselves).

[20] Microsoft has recently announced the availability of Azure virtual machines running on SGX hardware that allows the users to write their own code to run in a secure enclave to which the service provider does not have access.

[21] For example, on SGX systems there is a cost every time data is moved from the main memory into the enclave memory so it is necessary to consider the amount of data and number of times that it goes back and forth in and out of the enclave.

- Privacy:
  - Medium-high: user data (hashes or the message) are not visible to the ESP nor are the operations to detect child sexual abuse. The possible security issues (e.g. compromise of third-party server by state actors) could affect the privacy of the communication.
- Security:
  - Medium: the solution fully relies on trusting that the secure enclave works as intended and it has not been compromised (some vulnerabilities in this type of systems have already been found). The company making the enclave would be the only one having the key to the inner workings of the enclave and could become a target of bad actors. By accessing the enclave, bad actors would also have access to the decryption keys for the communications between the sender and the recipient. That said, it could be possible to attest that the code running in the enclave has not been modified from the time it was deployed and that the user has connected to the right enclave, carrying out the right processes, although this feature has been compromised in the past.[22] In addition, the check could remotely check the code but not the hashes used.
- Transparency:
  - Low: it is unclear how the use of the secure enclave could be documented and be publicly reported to facilitate accountability through ongoing evaluation and oversight by policymakers and the public.

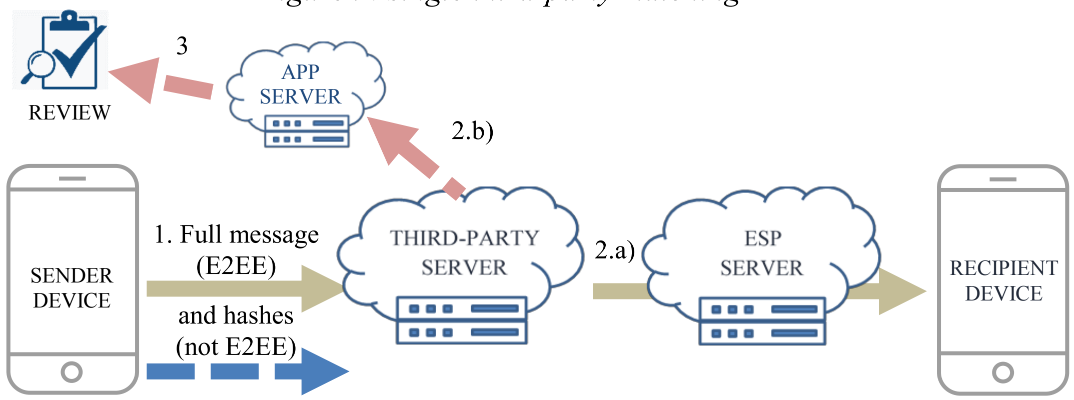
A possible way to mitigate some of the above concerns (in particular on security and transparency) could be to send to the secure enclave the hashes not E2EE for matching in the secure enclave. This would e.g. eliminate the risk of leaking the private E2EE keys if the enclave is compromised. In this case the trust in the secure enclave would be limited to protecting the hashing algorithm and its parameters.

b. Single third-party matching

This solution is the same as 1.b. (on device full hashing with matching done at server), but with the matching done at a trusted third-party server instead of at the ESP server:

[22] See here.

Figure 9: single third-party matching



1. **Device:** converts the images and videos into hashes before the message is encrypted, encrypts the full message and sends the hashes (client to server encrypted) and the E2EE full message to the third-party server.

3. **App server:** sends full image/video for review if there is a match in the third-party server.

Third-party server:
2. Compares hashes received from the device with those in the database of hashes of confirmed child sexual abuse (matching)
a) No match ⇨ forwards E2EE message to recipient
b) Match ⇨ asks app server to send image/video to review and/or reporting

Device: receives and decrypts E2EE message

Assessment:

- Effectiveness:
  - Medium-high: it could allow the detection of known CSA[23]. No need to limit the hash list, as it will be located at the third-party servers.
- Feasibility:
  - Low: scalability could be an issue, although this could be a service for a smaller companies offered on top of the cloud infrastructure of larger ESPs. It requires a combination of code running on the sender's device and (third party) server and therefore certain interdependence, which would influence e.g. the latency of message transmission.
- Privacy:
  - Medium-low: user data (hashes) are not visible to the ESP and no operations to detect CSA would occur at the ESP server. The possible security issues (e.g. compromise of third-party server by state actors) could decrease the privacy of the communication. That said, it is likely that the third party would have to work very closely with or be effectively part of the ESP that provides the communication service, which may raise privacy concerns. If the third party does not work on real time (i.e. analysing the message before it is sent) and instead analyses the message after it has been sent, the dependence on the ESP could be lower[24]. Also, the third party could be part of the client provisioning, which could reduce the privacy concerns.

[23] The use of classifiers is in principle possible with single third parties but it would be part of a different solution.

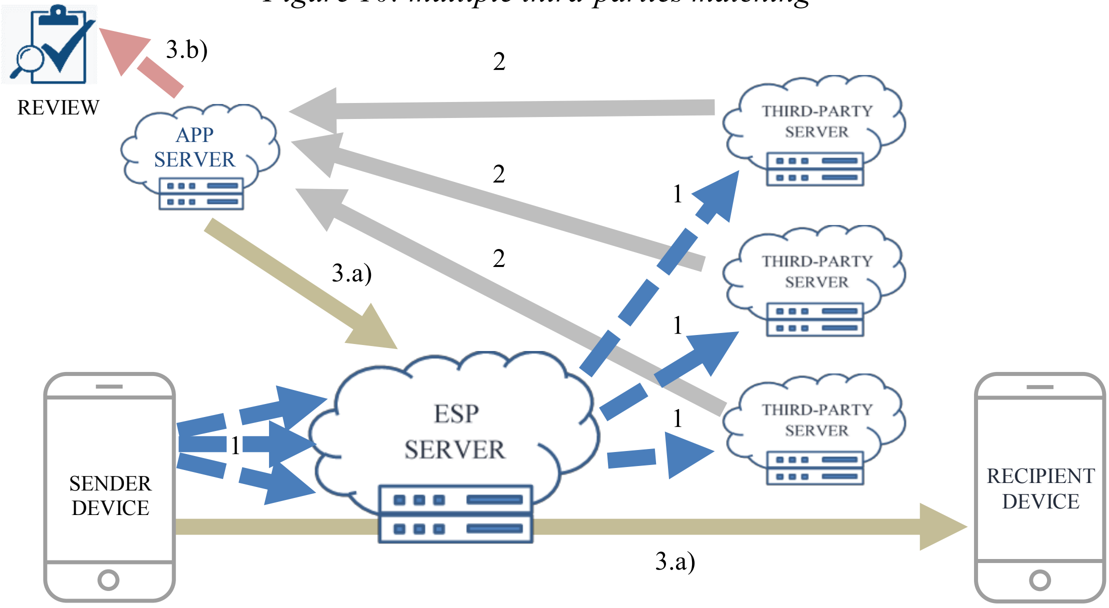
[24] The processing of messages after they have been sent to the recipient (i.e. batch processing with some timescale) could be applied to other solutions as well (see footnote 1 on the scope of the solutions).

- Security:
  - Medium-low: in addition to the security concerns of 1.b) (on-device full hashing with matching at the server), e.g. risk of manipulation of the hashing algorithm, the third-party server could be compromised by state or individual actors.
- Transparency:
  - Medium-low: the possible security issues could limit the reliability of public reporting on the use of the solution and therefore the accountability.

c. Multiple third-parties matching

In this solution, based on **multi-party computation** (MPC), the device converts the image (or video) into a hash, breaks it into parts, encrypts them with the third party keys and sends these parts to multiple third-parties for partial matching through the ESP server (which does not have access to the encrypted partial hashes). The app server compiles the responses from the third-parties and determines whether a match has occurred. If there is a match, the app server sends the full image (or video) for review/reporting. If there is no match, the ESP server forwards the E2EE message to the recipient:

Figure 10: multiple third-parties matching



1. **Device:** converts the images and videos into hashes before the message is encrypted, breaks them into parts, encrypts them with the third-party keys and sends them through the ESP server to multiple third-parties for partial matching and sends the E2EE full message to the third-party server.

3. **App server:** compiles the responses from the third-parties and determines whether a match has occurred:

a) No match ⇨ asks server to forward E2EE message to recipient

b) Match ⇨ sends message for review and/or reporting.

Third party servers:
2. Do partial matching of the multiple hash parts and sends info back to device.

ESP server:
No action beyond routing the hashes to the third parties.

Device: receives and decrypts E2EE message

Assessment:

- Effectiveness:
  - Medium-high: it could allow the detection of known CSA[25]. No need to limit the hash list, as it will be located at the third-party servers.
- Feasibility:
  - Low/medium-low: the multiple round-trip requests between the device and the servers before the message can be sent could slow performance, in particular with slow internet connections. It requires a combination of code running on the sender's device and (third party) server. A similar technology is already in use by Google and online merchants[26] but further research would be required to see how it could be applied in this situation (in particular on scalability) and what would be the costs, including computational overhead.
- Privacy:
  - Medium: user data (hashes) are not visible to the ESP and no operations to detect child sexual abuse would occur at the ESP server. The possible security issues (e.g. compromise of third-party server by state actors) could decrease the privacy of the communication. That said, the solution could offer better privacy than solution 2.b) (single third party matching): if at least one of the parties is trustworthy the hash will remain private. On the other hand, it is possible that the larger companies, which also offer electronic communication services, turn themselves into the third parties of this solution for the smaller companies, which may generate some privacy issues.
- Security:
  - Medium: in addition to the security concerns of 1.b) (on-device full hashing with matching at the server), e.g. risk of manipulation of the hashing algorithm, the third-party servers could be compromised by state or individual actors. That said, compared to solution 2.b) (single third-party matching), the risk will be lower as bad actors would need to compromise multiple servers instead of one.
- Transparency:
  - Medium: the possible security issues could limit the reliability of public reporting on the use of the solution and therefore the accountability.

****

Another possible server related solution would be to use **classifiers running on the server**, feeding on **metadata**. This seems to be the approach taken by **Facebook**[27] as it plans to switch to E2EE by default in its Messenger service[28] but the technical details remain unclear.

[25] The use of classifiers is in principle possible with single third parties but it would be part of a different solution.

[26] See here and here. The technology allows Google and online merchants to compute certain profile information on internet users (.g. the average age of buyers of a certain watch) without sharing all the data they have about those users.

[27] As indicated here.

[28] As announced in March 2019.

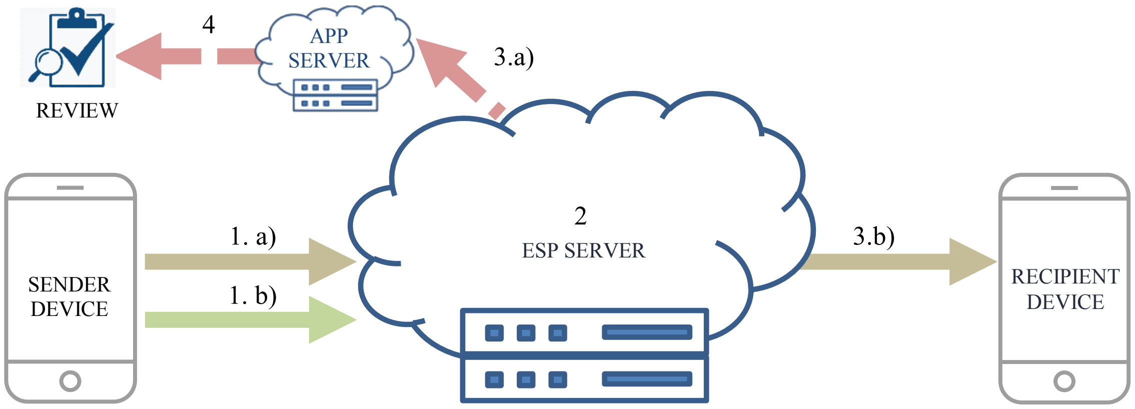
3) Encryption related solutions

This type of solutions consists in using **encryption protocols** that allow the detection of CSA in encrypted electronic communications.

a. On-device homomorphic encryption with server-side hashing and matching

In this solution, images are encrypted using a carefully chosen partially homomorphic encryption scheme (this enables an encrypted version of the hash to be computed from the encrypted image). The encrypted images are sent to the ESP server for hashing and matching against an encrypted version of the hash list[29] (the server does not have the homomorphic decryption keys):

Figure 11: on-device homomorphic encryption with server-side hashing and matching



1. **Device:** sends to the ESP server
   a) full message, E2EE, and
   b) images/video homomorphically encrypted

4. **App server**: sends full image/video for review if there is a match in the ESP server.

ESP server:

2. Operates on the homomorphically encrypted images/videos to extract a hash.
3. Compares the hash with the hash database of known CSAM. If:
   a) detected ⇨ instructs the app server to send the message for review
   b) not detected ⇨ forwards to recipient E2EE message

Device: receives and decrypts E2EE message

[29] See paper by H. Farid (reference 1 of encryption related solutions in annex 2), which shows that it is possible to build perceptual hashes on encrypted images that have about the same efficacy in terms of false positives and detection rate as PhotoDNA, but taking longer time (about 10-15 seconds per image, without doing any optimization to reduce the time, versus the one thousandth of a second that PhotoDNA currently takes). This could also be a type of privacy homomorphism.

Assessment:

- Effectiveness:
  - Medium: it could allow the detection of known child sexual images[30]. It would not be applicable to videos (too slow) or text-based threats. No need to limit the hash list, as it would be located at the server.
- Feasibility:
  - Low: proof of concept for images exists but additional research and development is needed to reduce processing times (currently at around 15 seconds per image on mobile).[31] No comparable commercial applications on electronic communications exist. At the moment, the computational power required on the server would render this solution expensive.
- Privacy:
  - Medium: user data (hashes) are visible to the ESP. Similar privacy as solution 1.b.
- Security:
  - Medium: no risk of leaking of hash database, or hashing and matching algorithm on the client side, as all these calculations would take place at the server. The solution does not prevent the possibility that the database of hashes could be tampered with at the server, as the other solutions with hash lists on the server.
- Transparency:
  - Medium-high: the use of the solution could be documented and be publicly reported to facilitate accountability.

****

Another possible encryption related solution would be to use machine learning and build **classifiers** to apply on **homomorphically encrypted** data for instant classification. Microsoft has been doing research on this but the solution is still far from being functional[32].

[30] The use of classifiers is in principle possible with partial homomorphic encryption but it would be part of a different solution.

[31] See table II on execution times in Tarek Ibn Ziad, M., et al., *CryptoImg: Privacy Preserving Processing Over Encrypted Images*, University of California, Los Angeles, 2019.

[32] More information on Microsoft's work on homomorphic encryption is available here.

4. Overview

The table below summarises the above assessments and classifies the possible solutions into 3 groups: top 3 (i.e. most promising, although some research may be needed), needs research (i.e. it could be considered but substantial research is still needed), and to be discarded (i.e. currently not worth pursuing at this point if there is a need to prioritise, but could still be of interest in the future):

Type	Solution	Effectiveness	Feasibility	Privacy	Security	Transparency	Overall
0) Baseline	a. Non-E2EE communications	★★★★★	★★★★★	★	★★★	★★	N/A
	b. E2EE communications	N/A	N/A	★★★★★	N/A	N/A	N/A
	c. Encrypted communications with exceptional access	★	★★★	★	★★	★★★	N/A
1) Device related	a. All detection done on-device	★★★★	★★	★★★	★	★★	Needs research
	b. On-device full hashing with matching at server	★★★★	★★★★	★★★	★★	★★★	Top 3
	c. On-device partial hashing with remaining hashing and matching at server	★★★★	★★★	★★★	★★★	★★★	Top 3
	d. On-device use of classifiers	★★	★★	★★	★★	★★★	Needs research
2) Server related	a. Secure enclaves in ESP server	★★★★	★★	★★★★	★★★	★★	Top 3
	b. Single third-party matching	★★★★	★	★★	★★	★★	Discard
	c. Multiple third-parties matching	★★★★	★	★★★	★★★	★★★	Needs research
3) Encryption related	a. On-device homomorphic encryption with server-side hashing and matching	★★★	★	★★★	★★★	★★★★	Needs research

5. RECOMMENDATIONS

On possible solutions:

- Immediate: on-device hashing with server side matching (1b). Use a hashing algorithm other than PhotoDNA to not compromise it. If partial hashing is confirmed as not reversible, add that for improved security (1c).
- Long term:
  - Invest in research on secure enclaves in ESP server to make the technology more accessible (2a).
  - Invest in research on multiple third-parties matching, leveraging existing applications (2c) and identifying possible third parties.
  - Invest in research on classifiers to supplement hashing and matching, but not replace it (1d).
  - Invest in homomorphic encryption research with regard to image matching (3a).
- *To be further completed with the experts' input*

Other considerations:

- PhotoDNA update: PhotoDNA, the hashing technology most widely used, is more than 10 years old and it may require an update now and then periodically every few years to keep up with the latest developments (and make it less vulnerable to manipulation, including by modifying the images to avoid detection).
- Quality and integrity of hash databases: a number of solutions rely on the detection of child sexual abuse through hashing technology. The quality of this detection (and therefore the effectiveness of those solutions) depends on the quality and integrity of those databases.
- Industry standards for detection: the creation of industry standards for the detection tools (e.g. image and video hashing) could facilitate the development and deployment of coherent and interoperable solutions across industry.
- Open source tools: open source tools could also facilitate the development and deployment of solutions across industry. However, substantial research may be required to produce open source tools that cannot be manipulated to reduce their effectiveness or be misused. In particular, all solutions considered under device-related solutions are based in part on "security by obscurity", that is, it is required for the security and effectiveness of the solution that the opponent does not know the full details of the scheme.
- Open competition: an open competition with a substantial prize[33], could encourage not only the development of open source tools and industry standards, but also the development of new possible solutions to detect and report child sexual abuse in end-to-end encrypted electronic communications.

[33] For example, similar to the open competitions organized by NIST on cryptography or by the EU-funded projects NESSIE and ECRYPT (eSTREAM).

- Reporting mechanisms: when describing the solutions, the paper does not analyse in detail what happens after child sexual abuse is detected, i.e. review and reporting mechanisms. These mechanisms depend on national legal obligations. These can have an influence on the effectiveness of some solutions (e.g. training of machine learning classifiers, which rely on a stream of well-labelled material to remain effective).
- Industry standards for reporting and transparency: when using hash databases, it would be useful to know not only the total number of reports sent to relevant statutory bodies from matches, but also the matches not sent to statutory bodies but removed based on the terms of service, and matches not sent to statutory bodies nor removed.
  The effectiveness of a hash database is currently only known to the company using it. It could be useful to have a third party perform regular testing/auditing using a sample non-CSAM match similar to the EICAR test file in the anti-virus industry.
- Safety by design: the development of technical solutions that could strike a balance between ensuring the privacy of electronic communications (including the privacy of children) and the protection of children against sexual abuse and sexual exploitation is facilitated when that balance is aimed at from the start, from the design stage.

- *To be further completed with the experts' input*

ANNEX 1: EXPERTS

Last name	First name	Organization	Organization type
Farid	Hany	University of California, Berkeley	Academia
Levine	Brian	University of Massachusetts Amherst	Academia
Preneel	Bart	Katholieke Universiteit Leuven	Academia
Stamos	Alex	Stanford Internet Observatory	Academia
LaMacchia	Brian	Microsoft	Industry
Benaloh	Josh	Microsoft	Industry
Barbosa	Norman	Microsoft	Industry
Davis	Sean	Microsoft	Industry
Harings	Johanna	Microsoft	Industry
Pancini	Marco	Google	Industry
Naruns	Natalie	Google	Industry
Omara	Emad	Google	Industry
Lieber	David	Google	Industry
Hall	Stephen	MEGA	Industry
Cashman Kirstein	Emily	Thorn	NGO
Rust-Smith	David	Thorn	NGO
Starr	John	Thorn	NGO
Boorse	Kristin	Thorn	NGO
Walker	Sarah	Thorn	NGO
DeLaune	Michelle	NCMEC	NGO
Sheehan	John	NCMEC	NGO
Johnston	Mark	Canadian Centre for Child Protection	NGO
Richardson	Lloyd	Canadian Centre for Child Protection	NGO

Last name	**First name**	**Organization**	**Organization type**
Boudry	Douglas	Australian Federal Police	Government
Dalins	Janis	Australian Federal Police/Monash university	Government
Coisel	Iwen	European Commission	Government
Beslay	Laurent	European Commission	Government
Carame Soto	Jose Luis	Spanish Guardia Civil	Government
de Dios Gómez Gómez	Juan	Spanish Guardia Civil	Government
Levy	Ian	NCSC	Government
Robinson	Crispin	GCHQ	Government
Steube	Jens	Europol	Government

Annex 2: references

0) General

1. Preneel, B., *The Never-Ending Crypto Wars*, presentation, imec-COSIC KU Leuven, 16/09/2019.
2. Snap Inc., Snap Inc. Response to Sen. Blackburn, 17/07/2019.
3. Weaver, N., *Encryption and Combating Child Exploitation Imagery*, Lawfare, 23/10/2019.
4. WhatsApp , *WhatsApp Encryption Overview, Technical white paper*, 4/4/2016.
5. Pfefferkorn, R., *William Barr and Winnie The Pooh*, Center for Internet and Society, 7/10/2019.
6. Stanford Internet Observatory, *Balancing Trust and Safety on End-to-End Encrypted Platforms*, 12/09/2019 (Stanford workshop).
7. Stanford Internet Observatory, *Mitigating Abuse in an End-to-End World*, 11/01/2020 (New York workshop).
8. Stanford Internet Observatory, *Mitigating Abuse in an End-to-End World*, 17/02/2020 (Brussels workshop).
9. Bursztein, E; Bright, T.; DeLaune, M.; Elifff, D.; Hsu, N.; Olson, L.; Shehan, J.; Thakur, M.; Thomas, K.; *Rethinking the Detection of Child Sexual Abuse Imagery on the Internet*, Proceedings of the 2019 World Wide Web Conference (WWW '19), 13-17 May, 2019, San Francisco, CA, USA.
10. Levy, I.; Robinson, C.; *Principles for a More Informed Exceptional Access Debate*; Lawfare, 29/11/2018.
11. Farid, H.; *Facebook's plan for end-to-end encryption sacrifices a lot of security for just a little bit of privacy*; Fox News, 16 June 2019.
12. Carnegie Endowment for International Peace; *Moving the Encryption Policy Conversation Forward*; Encryption Working Group, September 2019.
13. Millican, J.; *E2EE for Messenger: goals, plans and thinking*; Facebook; Real World Crypto 2020, January 8-10, 2020.
14. Dalins, J.; Wilson, C.; Boudry, D.; *PDQ & TMK + PDQF - A Test Drive of Facebook's Perceptual Hashing Algorithms*; Australian Federal Police and Monash University; December 2019.
15. Harold Abelson, Ross J. Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Michael A. Specter, Daniel J. Weitzner, *Keys under doormats*. Commun. ACM 58(10): 24-26 (2015).

1) Device related solutions

1. Mayer, J., *Content Moderation for End-to-End Encrypted Messaging;* Princeton University; 6 October 2019,
2. Callas, J., *Thoughts on Mitigating Abuse in an End-to-End World;* 14 January 2020,
3. Portnoy, E., *Why Adding Client-Side Scanning Breaks End-to-End Encryption*, Electronic Frontier Foundation, 1 November 2019,

4. Green, M., *Can end-to-end encrypted systems detect child sexual abuse imagery? – A Few Thoughts on Cryptographic Engineering*, 8 December, 2019.
5. Green, M., *Client-side CSAM detection: technical issues and research directions,* presentation at Stanford Internet Observatory event in New York, 11/01/2020.
6. Weaver, N., *Some Thoughts on Client Side Scanning for CSAM*, presentation at Stanford Internet Observatory event in New York, 11/01/2020.
7. Stamos, A., *Written testimony before U.S. House of Representatives Committee on Homeland Security on "Artificial Intelligence and Counterterrorism: Possibilities and Limitations"*, June 25, 2019.

2) Server related solutions

1. Makri, E., Rotaru, D., Nigel P. Smart, N.P., Vercauteren, F., *EPIC: Efficient Private Image Classification (or: Learning from the Masters)*; KU Leuven, Belgium; Saxion University of Applied Sciences, The Netherlands; University of Bristol, UK; 2017,
2. Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M., Wernsing, J.; *CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy;* Princeton University, Microsoft Research; 2016.
3. Liu, J., Lu, Y., Juuti, M., Asokan, N., *Oblivious Neural Network Predictions via MiniONN transformations;* Aalto University; 2017.
4. Juvekar, C., Vaikuntanathan, V., Chandrakasan, A., *GAZELLE: A Low Latency Framework for Secure Neural Network Inference;* MIT; 2018.
5. Riazi, M. S., Songhori, E. M., Weinert, C., Schneider, T., Tkachenko, O., Koushanfar, F., *Chameleon: A Hybrid Secure Computation Framework for Machine Learning Applications;* UC San Diego and TU Darmstadt, Germany; 2017.
6. Riazi, M. S., Samragh, M., Lauter, K., Chen, Hao., Koushanfar, F., Laine, K., *XONN: XNOR-based Oblivious Deep Neural Network Inference*; UC San Diego and Microsoft Research; 2019.
7. Portnoy, E., *Azure Confidential Computing Heralds the Next Generation of Encryption in the Cloud;* Electronic Frontier Foundation; 18 September 2017.
8. Frankle, J. et al.; *Practical Accountability of Secret Processes*; Massachusetts Institute of Technology; Proceedings of the 27th USENIX Security Symposium; August 2018.
9. Hastings, M.; *General purpose frameworks for Secure Multi-Party Computation;* University of Pennsylvania; Real World Crypto 2020, January 8-10, 2020.
10. Damgård, I, Nielsen J.B., Cramer, R., *Secure Multiparty Computation and Secret Sharing*, Cambridge University Press, 2015.

3) Encryption related solutions

1. Farid, H., Singh, P., *Robust Homomorphic Image Hashing*, Dhirubhai Ambani Institute of Information and Communication Technology, University of California, Berkeley and Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India, 2019,
2. Iliashenko, I., *Optimisations of fully homomorphic encryption*, KU Leuven, 2019,

3. Minelli, M., *Fully homomorphic encryption for machine learning*, PSL Research University, 2018.
4. European Commission, *Putting privacy at the heart of biometric systems*, 2011.
5. Yakoubov, S., *A Gentle Introduction to Yao's Garbled Circuits*, Boston University, 2017.
6. Tarek Ibn Ziad, M., et al., *CryptoImg: Privacy Preserving Processing Over Encrypted Images*, University of California, Los Angeles, 2019.
7. Gentry, C. *Fully Homomorphic Encryption Using Ideal Lattices*. In the 41st ACM Symposium on Theory of Computing (STOC), 2009.