The Upcoming Commission Proposal on Fighting Child Sexual Abuse Material online seeks to make use of a number of technical measures to prevent the sharing of Child Sexual Abuse Material. However, many of these methods contain significant flaws. The goal of this position paper is to explain some of the proposals the Commission has put forward and why they are problematic from a technical perspective.

Detecting Known abuse material using Client-side Scanning.

Understanding Hashing

Traditional Hashing

When detecting known abuse material, a common approach is using a method called "hashing". A hash is a fingerprint of a file: you can calculate the hash of any file, but you cannot use the hash to reconstitute the file.

The most common form of hashing is to look at the **layout of data that make up the file**, for example, using the **md5** hashing system. Here is an example of the hashes of an image file of the European Parliament. The first is the original image file, the second is a smaller version, and the third is with a watermark.

		
parlement_original.jpg `0e01a342220efc6a5a841e367b43fb92`	**parlement_small.jpg** `dd3ce41912eb2158c31910db79b74ca9`	**parliament_hi.jpg** `f7fba40ffd6522d00273827ab9a65f11`

You will notice that these hashes are completely different: because most hashing systems make a fingerprint of the **layout of data in the file**. If a file is uploaded and it matches the md5 hash of known abuse content, then we can say with almost absolute certainty that it is abuse material, because **it is almost impossible to generate a file that matches the md5 hash of another file**. However, this is not very good for detecting Child Abuse Material because **a slight change to the file completely changes the hash.**

Perceptual Hashing

To solve this, **perceptual hashing** was invented. Perceptual hashing is different, because **it looks at the structures and patterns in an image** instead of looking at the layout of data in the file. This means that **resizing an image should not change its hash**, and **adding watermarks should only make small changes to the hash.**

Let's take a look at hashes made using the **NeuralHash** hashing system, a perceptual hashing system developed by apple to detect CSAM:

		
parlement_original.jpg d019c5c81037d8b67d207687	**parlement_small.jpg** d019c5c81037d8b67d207687	**parliament_hi.jpg** d01bc5c81037d8b67d207687

Because NeuralHash looks at the content of the image, rather than the structure of the file, **a resized version of the image has exactly the same hash as the original**, and the version with a watermark has the same hash except for one letter (highlighted in red).

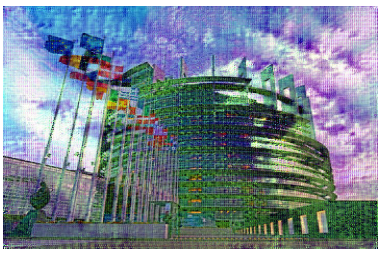
The flaws of perceptual hashing

In theory, this is great for detecting CSAM, as it allows detection even when files have been changed, but in practice **it is much more likely to create false positives**, because images with content that looks similar could have the same hash. Here are a few examples of **naturally occurring false positives**:

			
ba1d5ae53385e49505200f16	ba1d5ae53385e49505200f16	d8044c7d7058c01f9dbbce61	d8044c7d7058c01f9dbbce61

Worse still, once a malicious actor understands how the technology works, they can manipulate images to change their hash. It took hackers less than a month to reverse-engineer Neuralhash, and develop software which creates false positives (or false negatives), simply by giving the software an image, and a target hash for that image.

Below are some images that we made using that software: a picture of a puppy, a picture of the European Parliament that has the same hash as the image of the puppy, and a picture of a puppy that has the same hash as a picture of a cat.

		
`puppy.jpg` `3d8ca369f16fc0c067f53c0f`	**`parlement_puppy.jpg`** `3d8ca369f16fc0c067f53c0f`	**`cat_puppy.jpg`** `bc278683699ab24343c846a3`

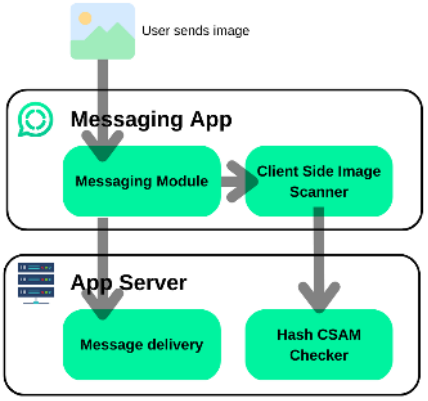
Using a more powerful computer or taking more time, images can be generated that look almost identical to the original with a totally different hash. The software we used was designed to change one image to match the hash of another. It would also be possible to develop software that just changes the hash to avoid detection while making less noticeable changes to the image.

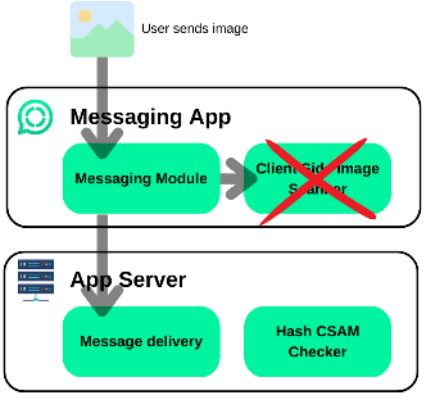
This essentially means that abusers will easily be able to trick the algorithm to hide abuse content, while innocent people could easily get in trouble for sharing a meme which had been maliciously modified to match the hash of CSAM content.

Prevention of reporting following Client-Side Scanning matches

Considering the proposal for the Commission, there are also several ways that **users with varying levels of technical knowledge could easily prevent scanning**.

The Commission's model of client side scanning would involve messaging apps including a module within their apps which would calculate hashes, either perceptual or otherwise, of image files, and send them to a server to check those hashes against a database.

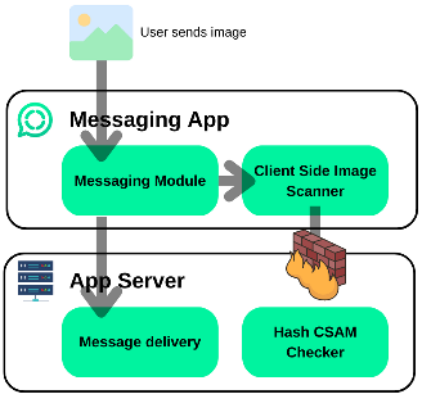




Example 1: Removal of Client-Side Scanning module

In this example, the user either removes the client-side scanning module from the app, or uses a VPNor third party service to get a version of the app that does not have the module (from outside of the EU's jurisdiction).

In some cases, the user could also build a version of the app that does not contain the module, or that sends false hashes. This would be impossible to detect.



Example 2: Blocking of Communication of hashes

In this example, the user uses a firewall, VPN, or packet interception tools to block traffic to the CSAM checker, or modifies the app so that checks are sent to a bogus location. This could be as simple as installing an app.

This means that the Client-side scanning module remains present in the app, but is not able to send hashes for checking.

Risk of abusers moving to anonymised networks

There is also a risk that users might move away from known networks to avoid scanning, and move to anonymised networks where it is almost impossible to identify them. It may make more sense to keep abusers on known networks, as it makes it easier to catch accomplices.

Known Networks (WhatsApp, Wire etc..)		Anonymised networks (Tor, I2P)	
❌	Interception not possible		Interception not possible
✅	Detection possible (though abusers could prevent detection as described above)		Detection not possible
	Identification of user possible		Identification of user not possible
✅	Identification of accomplices easy (once the abusers phone has been confiscated, a list of phone numbers is available in groups)	❌	Identification of accomplices impossible (TOR and I2P are by nature fully anonymous, users only identify each other by pseudonyms)

Conclusion

Abusers can easily remove, disable or break the scanning module, or prevent the scanning module from sending reports.

In addition to this, hashing systems in use for CSAM detection cannot reliably detect CSAM, and could be manipulated into generating significant numbers of false positives for Law Enforcement and the EU centre: for instance, a meme could be created that matches the hash of CSAM content, users sharing this meme would inadvertently trigger an avalanche of false positives: innocent citizens will likely end up suffering false positives while abusers can easily avoid detection.

Finally, these measures could also push abusers onto Anonymised networks, where catching their accomplices is impossible.

For these reasons, we do not see client-side scanning in encrypted messaging apps as an effective approach to combat CSAM.

Detecting unknown CSAM content and Grooming

Part of the Commission's proposal implies the use of Artificial Intelligence to detect unknown CSAM content and grooming, but while recent innovations in AI do show some promise for such applications, the potential risks and issues far outweigh the benefits.

Understanding Artificial Intelligence

The best way to understand how AI works is to imagine that you have somehow taught a toddler to look at Chinese symbols, and, without understanding Chinese, write other Chinese symbols that just happen to be a coherent response to the original symbols. Like the toddler, **AI has no understanding of meaning, it just looks for patterns on the basis of patterns it knows**. Again, like the toddler, **AI does not understand its responses, their meaning or their consequences**. This becomes a problem when applied to something as serious as child abuse.

Concerns with the use of AI

Using AI to guess age

Despite progress in AI technology, AI still struggles to accurately guess age. For instance, Meta recently ran into numerous issues, in particular on biases on other ethnic and sexual minorities. Correcting these issues requires modifications to the algorithm that may have unintended consequences.
Can we justify using an algorithm that might not detect CSAM of people of colour, or that may always misflag citizens from certain ethnic/sexual minorities as underage?

Using AI to detect abuse material

As explained previously, AI does not have an understanding of meaning and can only detect patterns: for instance, photos sent to a child's doctor, consentual nudes between teengaers, or holiday photos could be misidentified as CSAM material. This is not a hypothetical: a man was reported to have had his google account permanently and irrevocably banned because of a picture of his son he sent to his family doctor.

Furthermore, AI cannot tell the difference between consentual nudes between teenage couples and abuse material, meaning police will be flooded with false positives and teenagers privacy will be severely impacted.

Other Technical issues

In addition to these issues, not all mobile devices are powerful enough to do AI-based scanning client-side without significantly delaying uploading. Moreover, **they are vulnerable to all the same issues as Client-side scanning using perceptual hashes**: it is possible to modify an image in a way that makes the AI ineffective, or to simply remove, block or modify the detection module, as explained above.

Conclusion

Using AI to scan for unknown CSAM material, would exhibit the same issues as scanning for known content, along with significant accuracy issues. We would also be responsible for false positives and negatives disproportionately affecting sexual and ethnic minorities, and their consequences.

Verification of flagged content

In both of these systems, it is unclear if content should automatically be sent to the police if it is flagged. It is also unclear if such a use of citizens data would meet the requirements for automated processing in Article 22 GDPR.

In this case, **the police would likely be buried in an avalanche of false positives**. Given that the police are already struggling to investigate all flagged cases today, **making the haystack bigger won't help us identify more CSAM**. Finally, it is important to ask ourselves: is it really ethnical to forward nudes sent between two underage individuals to the police?

If the content is not sent to the police automatically, then what is the threshold for reporting? Given the high rate of false positives, would this not force the police to investigate large numbers of innocent people?

URL Blocking (Art 16)

The Commission's proposal also includes a provision to "block URLs". This is not technically feasible and indicates that the proposal was drafted without sufficient technical expertise.

The easiest way to understand this is **the letter analogy**: you want a specific piece of information, so you send a letter requesting that information, and receive a letter containing the information.

The Anatomy of a URL

A URL is how we represent a particular piece of content on a website. For instance, you might want to request a photo called picture.jpg from website.com. In this case, the URL could be: **https://website.com/picture.jpg**.

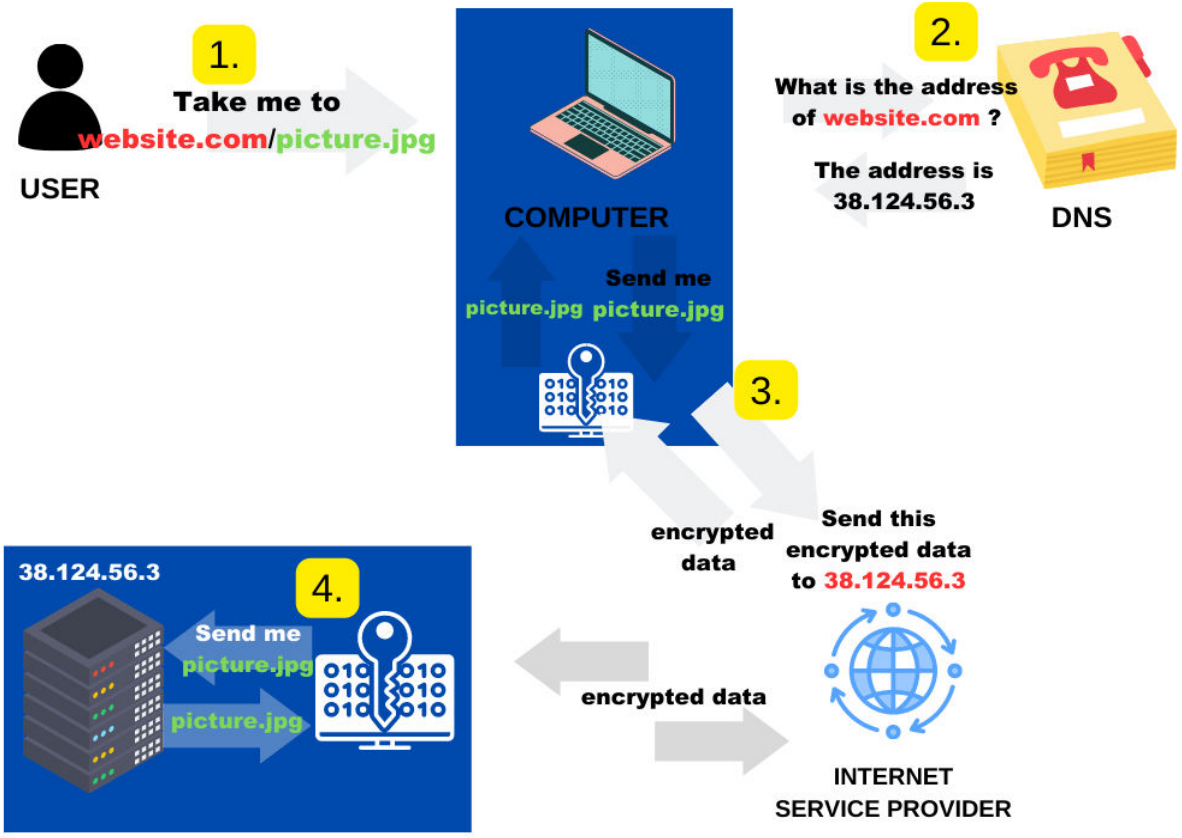
The first part (blue) is the scheme: this dictates **how the information is sent**. **Almost all websites today use https: like a letter, the content can only be seen when opened by the recipient at the end**.

The second part (in red) is the name of the recipient: this dictates who the information should be sent to.

The third part (in green) is the message, it is only visible to the recipient.

How users request content (see infographic)

1. The user enters a URL
2. The computer uses a digital address book called **DNS** to find the digital address (IP) of the recipient.
3. The Computer encrypts the request and asks the Internet Service Provider to forward it to the digital address. (like sealing a letter and posting it)
4. The Website's server decrypts the request (opens the letter) and sends an encrypted reply, which the user receives and decrypts.



Why can't this work?

There are two points where blocking is possible. Neither of them allow for URL blocking (blocking of individual resources on a website).

DNS Blocking

Firstly, it would be possible to **force EU DNS servers to remove certain websites from the address book.** However, **this would not block the specific image** (picture.jpg), **it would block the entire website** (website.com). This is the equivalent of removing a company from the phone book because one of its employees did something wrong.

It is also **easy to bypass**: it takes 6 clicks for users to switch to using an address book outside EU jurisdiction to bypass blocking.

IP blocking

It would also be possible to have the internet service providers block digital addresses (IPs). This would be the equivalent of preventing the delivery of letters to a building. Except, on the internet, most of the buildings are skyscrapers: one IP address can host thousands of websites. **Enforcing IP blocking would almost certainly result in tens if not hundreds of innocent unrelated websites being blocked in the EU**. It is also possible to bypass IP blocking with a VPN for a few euros a month.

Conclusion

URL blocking is not possible from a technical perspective, and the other technical approaches would result in the collateral blocking of innocent websites. Finally, these approaches **ignore the reality of how CSAM is distributed:** often it is uploaded to an innocent website and removed after a few weeks to avoid detection. By the time a blocking order has come into force the content will have been moved, and **otherwise innocent sites could find themselves "delisted" from the internet**.

Age verification on app stores

The Commission also intends to introduce age verification on app stores, however this raises numerous questions about which apps will require age verification and which app stores will have to implement it.

Web browsers?

Web browsers, such as Chrome, Firefox, or Safari, are able to access a variety of online services, including ones where children may be at risk. Essentially, a child could use a web browser to access services instead of using the app. This is possible for most online services. Under the Commisison's logic, using a browser should hence require age verification, however in Practice a browser is part of the core experience on a phone: all phones ship with one. Depriving anyone under 18 of access to a browser is essentially depriving them of access to most of the resources on the internet.

Alternate app stores?

The DSA will force Google and Apple to allow alternative app stores on their platform, including app stores from outside the EU's jurisdiction. Since anyone could launch such a store It will not be possible to enforce age verification on small independent app stores from outside the EU.

Furthermore, how far do you go: does this also include app stores on PCs? How about package managers (tools used on certain Operating Systems to download apps)?

Sideloading

Sideloading is when you download an app from an external source and then install it to your phone without using the app store. As it stands there are loads of websites that provide apps to download and sideload without the app store. Young people are aware of this and have been using sideloading for years. Essentially this allows them to bypass age verification in app stores with ease.

Overall Conclusion

The Commission's proposal exhibits a frankly shocking lack of technical knowledge for a regulation that is so technical. Not only will **the proposal fail to achieve its intended goals**: **it will likely make the situation worse**, both by **exposing innocent citizens to false positives, and failing to detect abusers**. With other issues such as the duplication of hotlines work, and the extended delays for content takedown making matters worse.

We need to radically rethink the technical implementation of this file, f**ocussing detection in non-encrypted fora such as Discord, Facebook Messenger, etc..**; where grooming takes place.

Also, while it is outside the domain of EU competence, the focus needs to be on **educating children to recognise grooming and abuse and giving them somewhere to report it**, as well as **increasing police and hotline ressources** to investigate and take down CSAM, and **better coordinating MS responses through an EU centre that complements, rather than duplicates the work of hotlines.**