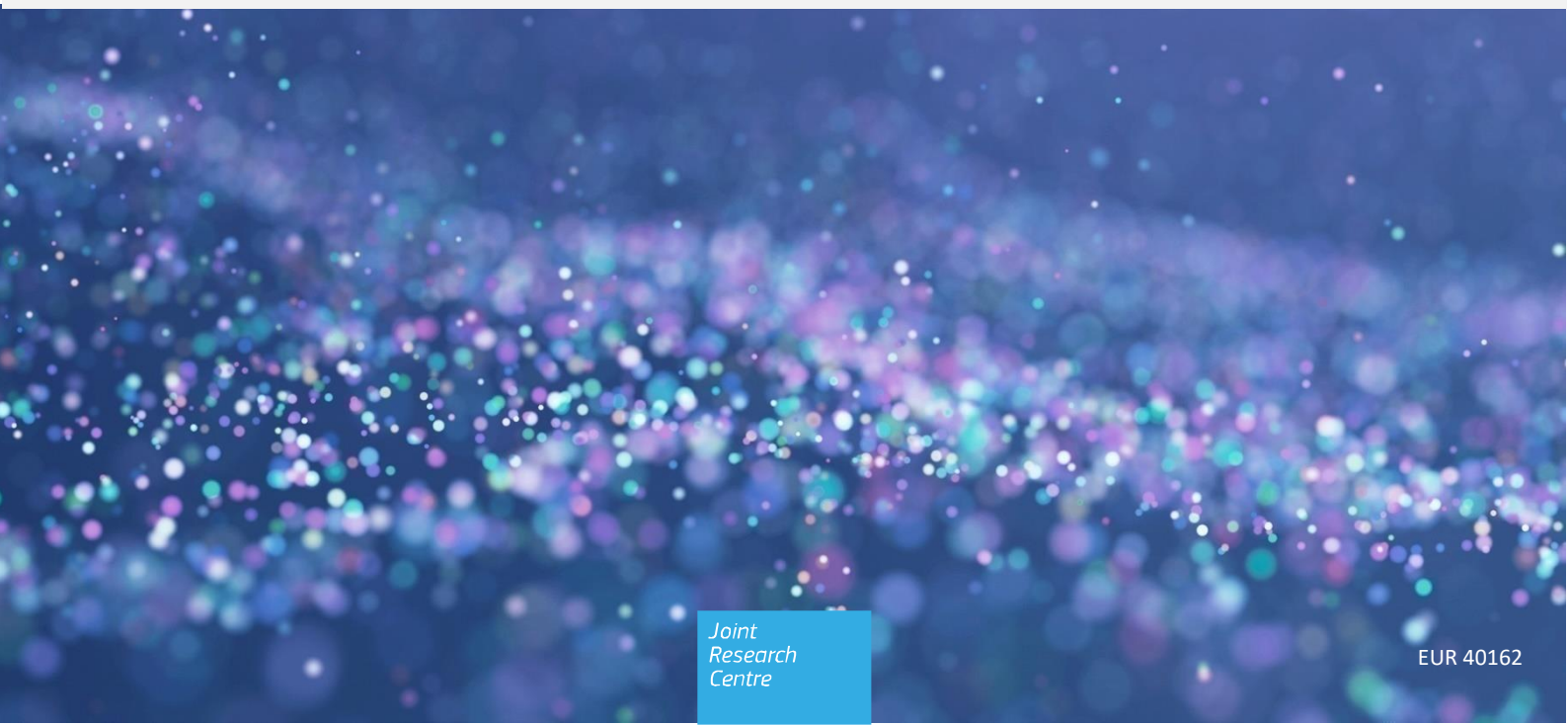
ISSN 1831-9424

Perceptual Hashing on Images Encrypted with a Homomorphic Scheme

Study of the feasibility of detecting child sexual abuse material shared via end-to-end encrypted messages

Chenu, M., Beslay, L.

2025



Joint
Research
Centre

EUR 40162

Contact information
Name: Mathilde Chenu
Email: mathilde.chenu@ec.europa.eu

EU Science Hub
https://joint-research-centre.ec.europa.eu

JRC139994

EUR 40162

PDF ISBN 978-92-68-23028-2 ISSN 1831-9424 doi:10.2760/2947924 KJ-01-24-216-EN-N

Luxembourg: Publications Office of the European Union, 2025

How to cite this report: European Commission: Joint Research Centre, Chenu, M. and Beslay, L., *Perceptual Hashing on Images Encrypted with a Homomorphic Scheme,* Publications Office of the European Union, Luxembourg, 2025, https://data.europa.eu/doi/10.2760/2947924, JRC139994.

Contents

Abstract

In 2022, the European Commission proposed "*an obligation for providers to detect, report, block and remove child sexual abuse material from their services*" in its proposal for a regulation laying down rules to prevent and combat child sexual abuse (European Commission, 2022b). The European Parliament, the European Data Protection Supervisor and the European Data Protection Board highlighted in their comments related to the proposed Regulation the importance of preserving end-to-end encryption. In a staff working document impact assessment report (European Commission, 2022a), the European Commission suggested that using homomorphic encryption would allow for both end-to-end encryption and content scanning.

In this work, we study the feasibility of using perceptual hash functions on images encrypted with a homomorphic scheme, in the context of the detection of child sexual abuse material shared via social media and messaging applications. We study both the encryption phase by the sender, and the detection phase performed by the server. Our work shows that, on top of time and memory limitations, the main issue is the effort needed to perform comparisons between ciphertexts. This last limitation is independent from the perceptual hash function and homomorphic encryption scheme used.

1 Introduction

The European Union has been making relevant efforts towards more effectively preventing and fighting child sexual abuse, exploitation and child pornography. These efforts include the detection of child sexual abuse material shared via social media and messaging applications.

Context Several initiatives have supported the issue at policy, societal and technical level, such as the Directive on combating the sexual abuse and sexual exploitation of children and child pornography (European Parliament and the Council, 2011), its proposed recast (European Commission, 2024), the EU Security Union Strategy (European Commission, 2020a), and the dedicated European Union Strategy for a more effective fight against child sexual abuse (European Commission, 2020b). It is also considered in the Horizon 2020 Research Project Starlight (Starlight, 2021).

In addition, the European Commission published in 2022 a proposal for a Regulation laying down rules to prevent and combat child sexual abuse (European Commission, 2022b). This Regulation introduces "*an obligation for providers to detect, report, block and remove child sexual abuse material from their services*".

The European Data Protection Supervisor (EDPS) and the European Data Protection Board (EDPB) published a joint opinion on this proposal (European Data Protection Supervisor, European Data Protection Board, 2022) stating that "*the structural incompatibility of some detection order with end-to-end encryption becomes in effect a strong disincentive to use end-to-end encryption*", and that "*the inability to access and use services using end-to-end encryption could have a chilling effect on freedom of expression and the legitimate private use of electronic communication services*". The Members of the Committee on Civil Liberties, Justice and Home Affairs (LIBE) from the European Parliament adopted a report on the proposed Regulation (Committee on Civil Liberties and Affairs, 2023), in which the rapporteur states that "*end-to-end encryption is an important tool to guarantee the security and confidentiality of the communications of users, including those of children. Consequently, nothing in this Regulation should be interpreted as prohibiting or weakening end-to-end encryption, while the Regulation remains open, where applicable, to existing and future technological developments*". Independently, the European Union Court of Justice ruled in the case of Podchasov v. Russia in February 2024 (European Court of Human Rights, 2024) that "*the Internet Communication Organiser's statutory obligation to decrypt end-to-end encrypted communications risks amounting to a requirement that providers of such services weaken the encryption mechanism for all users; it is accordingly not proportionate to the legitimate aims pursued*".

Technological landscape A perceptual hash function is a fingerprinting algorithm taking a picture or a video as input, and returning an output called hash. Perceptual hash functions, unlike cryptographic hash functions, are designed to produce the same, or very close, output for similar images. For example, two versions of the same picture but with slightly different colors, resolution or frame, should have the same hash.

This property of perceptual hash functions makes them particularly suitable to detect child sexual abuse material online. When a media is exchanged online, its perceptual hash can be computed and then compared to a databases of hashes from known problematic content. An example of such a database is the one maintained by the American National Center for Missing and Exploited Children (NCMEC), a non-profit organization established by the United-States' Congress in 1984.

Since 2013, a sharp rise in the use of end-to-end encryption has been observed. This trend naturally extended to instant messaging applications such as Signal, WhatsApp, Telegram or Messenger which also started offering end-to-end encryption communication. In 2024, WhatsApp has more than two billions users (WhatsApp, 2024), while Messenger has more than one billion users (Messenger, 2024). Both are now end-to-end encrypted by default.

Problem The growing quantity of child sexual abuse material exchanged online is particularly difficult to detect when shared via end-to-end encrypted messaging applications. The scanning process using perceptual hashing described above is ineffective on data encrypted by standard algorithms.

Experiment A specific type of encryption schemes, called homomorphic encryption, allows to perform arithmetic operations on a set of encrypted content. Once deciphered, the result is the same as

the result that would have been obtained if the operations had been performed on a non-encrypted set of data. Such schemes have been primarily designed to encrypt and analysis sensitive data, such as health data, stored on third party clouds.

A staff working document impact assessment report from the European Commission (European Commission, 2022a) explored and analysed several options suggesting as possible solution (not of its top 3) "*on-device homomorphic encryption with server-side hashing and matching*". The report concluded on a medium effectiveness and low feasibility for this option acknowledging the need for more study on the approach. The reasoning is that using end-to-end homomorphic encryption paired with perceptual hash functions would preserve the confidentiality of communications while allowing for some Child Sexual Abuse content detection.

JRC contribution The aim of this experiment is to evaluate the feasibility of using perceptual hashing on data encrypted with a homomorphic scheme to detect when Child Sexual Abuse Material is shared on messaging applications.

2 Perceptual hashing

2.1 Principle

Perceptual hashing refers to a fingerprinting algorithm applied on a multimedia such as pictures or videos. When the features of two different inputs are similar, like the same pictures but with slightly different colors, resolution or frame, the perceptual hash function produces the same, or very close, output called the hash or the digest. This property make perceptual hashing useful for copyright infringement, forensics, disinformation detection or censorship (McKeown and Buchanan, 2023). In particular it is used to detect the distribution of child sexual abuse material online.

Perceptual hash functions should not be confused with cryptographic hash functions that are specifically designed to be collision-resistant and detect the smallest change between two similar inputs.

The most commonly used perceptual hash functions are summarized in Table 1. Most of the perceptual hash functions are not open-source, though some of them have been reverse engineered. Some companies let law enforcement and tool providers use their tools for free, such as Microsoft's PhotoDNA (Microsoft, 2009) for the NCMEC.

Table 1: Main perceptual hash functions

Name	Provider	Open source
PhotoDNA	Microsoft and Dartmouth college	no
CSAI Match	Youtube	no
NeuralHash	Apple	no
PDQ	Meta	yes

Source: (Microsoft, 2009), (Youtube, 2014), (Apple, 2021) and (Meta, 2019)

The algorithm used by the PQD hash function, which is open-source, is detailed in Section 4.2.

2.2 Use against child sexual abuse material

The properties of perceptual hash functions makes them particularly suitable to detect child sexual abuse material online. When a problematic media is first reported, the perceptual hash is used as an identifier, which is stored in a database. This identifier can then be compared to other contents online to scan for this media, even if the picture is rotated, cropped or slightly altered.

Let us explain this idea with an example, as illustrated on Table 1: when two people, say Alice and Bob, want to exchange a media m via a non-encrypted messaging application, the perceptual hash of the media $\mathcal{H}(m)$ is computed by the server of the messaging application. The server has a database $\mathcal{D}$ of problematic content hashes, and checks if $\mathcal{H}(m)$ belongs to this database. If yes, it can report the content to competent authorities. Note that this method only detects known content: newly created content is not detected before being reported and integrated to the database.

Figure 1: Unencrypted message exchange between Alice and Bob

Alice		Server		Bob
m	$\longrightarrow$	m	$\longrightarrow$	m
		$\mathcal{H}(m) \in \mathcal{D}?$		

Source: Own production

An example of such a database is the one maintained by the American National Center for Missing and Exploited Children (NCMEC), a non-profit organization established by the United States' Congress in 1984. The NCMEC started their database in 1998 and have become the main clearing house worldwide, with over 10 million hashes inserted as of 2023 (National Center for Missing and Exploited Children, 2024). Their collection of hashes is notably used by Meta, among others, to detect Child Sexual Abuse

Material (CSAM) on their website. Meta is also the NCMEC main corporate contributor (National Center for Missing and Exploited Children, 2023). Other organizations also maintain databases such as Thorn (29 million hashes) (Thorn, 2023), or the Project Arachnid (Arachnid, 2023).

Note that this database matching process faces two main limitations, as developed in (Bursztein et al., 2019).

- First, a human intervention is needed after a match is found between the hash of an image exchanged and a hash in the database, to confirm the abuse, identify the victims and file a report. With the exponential growth of the number of reports in the past years, this reviewing task requires a large number of analysts. Hence there is an emerging need for automation of this reviewing process, in particular de-duplicating and aggregating reports, identifying the material most relevant for prosecution, and preventing the emotional toll for analysts. See (Bursztein et al., 2019) section 4.1.

- Second, 84% of images and 91% of videos reported to the NCMEC are reported only once. This means that new CSAM is constantly created and the vast majority of it is short lived. For the remaining 16% and 9%, the median lifetime is respectively 257 and 210 days for images and videos, with a very limited subset resurfacing thousand of time. See (Bursztein et al., 2019) section 4.5.

Other limitations arise from perceptual hash functions themselves, which can lack robustness. Several papers identified weaknesses in such hash functions, in two directions:

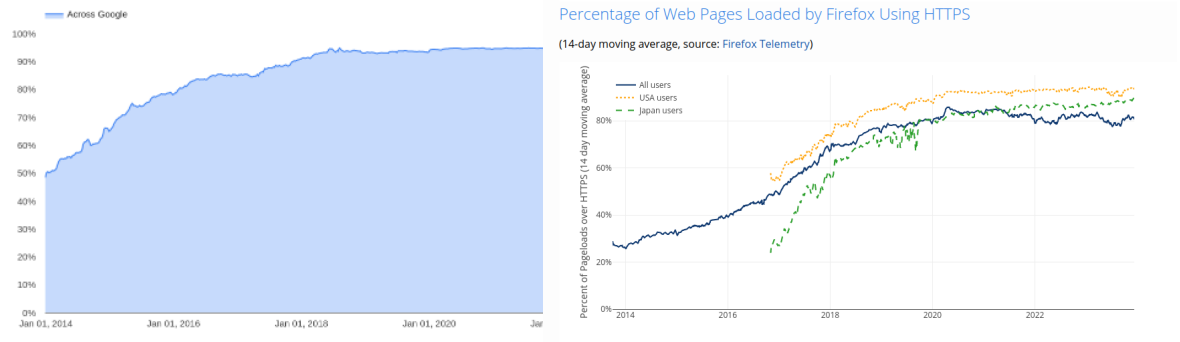
- avoiding collisions, by applying minimal changes to the original image so that it produces a very different hash, preventing it to be recognized (Hao et al., 2021) (Jain et al., 2022).

- creating collisions, by applying minimal changes to an image to create a hash collision with a different second image (Struppek et al., 2022) (Dolhansky and Canton-Ferrer, 2020).

2.3 Inefficiency on encrypted data

Since 2013 a sharp rise in the use of encryption has been observed. According to Google statistics its Chrome browser encrypted pages increase from 50% to 95% (Google, 2023), while Firefox amount of pages loaded with https increased from 20% in 2013 to 80% (LetsEncrypt, 2023). See Figure 2.

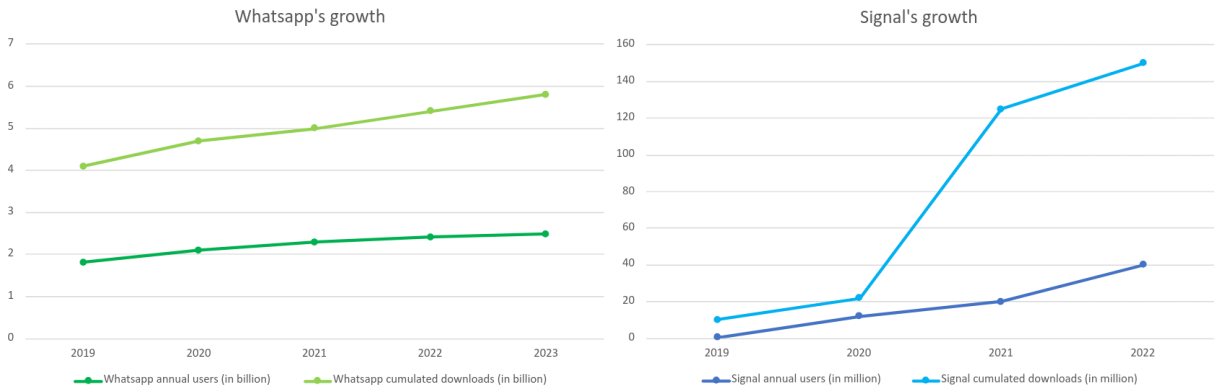
Figure 2: Percentage of encrypted pages loaded through Chrome (left) and Firefox (right) browsers



Source: (Google, 2023) and (LetsEncrypt, 2023)

This trend is also verified for instant messaging with a number of applications starting to offer end-to-end encrypted communications since 2013: Telegram in 2013, Signal in 2014, WhatsApp and Messenger in 2016. These applications became quite successful and widely adopted around the globe with the number of annual users and the number of downloads steadily growing, see Figure 3. In 2024, Signal has more than 140 million user (Business of the App, 2024), WhatsApp more than 2 billion users (WhatsApp, 2024), and Messenger more than 1 billion (Messenger, 2024).

Figure 3: Number of annual users and cumulated download for Whatsapp (left) and Signal (right)



Source: (Business of Apps, 2024)

The scanning of images by perceptual hash functions described in Figure 1 is rendered obsolete when communications are end-to-end encrypted, as described in Figure 4. When the message transiting through the server is encrypted, the server cannot extract meaningful information from the encrypted media c, and hence cannot know if the original image was in the database of known problematic content.

Figure 4: Encrypted message exchange between Alice and Bob

Alice		Server		Bob
				(pk_B, sk_B)
m				
$c = Enc_{PKB}(m)$				
c	$\longrightarrow$	c	$\longrightarrow$	c
				$Dec_{SKB}(c) = m$

Source: Own production

The inefficiency of perceptual hashing on encrypted data is the reason why the decision in 2021 by Facebook, now Meta, to encrypt by default Messenger communications created an outcry among children protection agencies. At the time 76% of the reports made to NCMEC originated from Facebook (National Center for Missing and Exploited Children, 2021), though that encapsulates more than Messenger. Meta then tasked the consultant company Business Socially Responsible (BSR) with a human rights impact assessment. The report published in 2022 states that the benefits of end-to-end encryption outrun the drawbacks for children protection.(BSR, 2022)

3 Full homomorphic encryption

3.1 Principle

Homomorphic encryption is a form of encryption that allows to perform certain type of operations on encrypted data without having to decipher it first. Note that the output of the computation is also encrypted. Once decrypted, the result of this operation is the same as the result that would have been obtained if the same operation been computed on the clear data (Marcolla et al., 2022).

Homomorphic encryption has first been considered for privacy-preserving outsourced-storage and outsourced-computation (Marcolla et al., 2022). A homomorphic encryption scheme would allow to perform computations by a third-party server on sensitive data such as electronic voting or predictive analytic on health data. It has recently been considered to scan homomorphic encrypted data for potential harmful content (European Commission, 2022a).

Although the first idea of encryption schemes that would have this potential for encryption came shortly after the publication of RSA (RSA is homomorphic with respect to the multiplication) (Rivest et al., 1978), more than thirty years have been necessary to obtain fully homomorphic schemes (Gentry, 2009). Several steppingstone-notions that have been defined toward that goal in the literature.

- Partially homomorphic encryption: scheme that supports only one type of operation (e.g., RSA with respect to multiplication)
- Somewhat homomorphic encryption: scheme that supports two operations but with constraints on the combination of these operations (e.g., unlimited number of additions only one multiplication is supported).
- Fully homomorphic encryption: scheme that supports any operation without any constraint on the number or combination of these operations. A fully homomorphic scheme can be obtained from a somewhat homomorphic scheme if a bootstrapping operation is available to reduce the accumulated noise when needed.

The main difficulty to obtain a fully homomorphic scheme is the noise propagation: each operation performed on the ciphertexts creates some error or "noise". When these errors reach a certain threshold the ciphertext cannot be decrypted properly anymore. (See (Marcolla et al., 2022) section IV). This problem has been solved by Craig Gentry in 2009 by using "bootstrapping": when the threshold is about to be reached, a refresh operation is executed which reduces the amount of noise present in the ciphertext without decrypting it. However, this bootstrap operation tends to be extremely costly (several seconds usually, depending on the schemes and the parameters).

For this reason the original constructs from Craig Gantry are fully homomorphic but unfeasible in practice (Gentry, 2009). Subsequent schemes tried to reduce the bootstrapping time, or to delay the moment when it needs to be performed.

3.2 Schemes

Fully homomorphic schemes are often divided into generation depending on their timeline, their mathematical foundations, and their efficiency. A generation generally encapsulate several similar schemes and all their improvements. We refer to the appendix and to the survey from (Marcolla et al., 2022) for more details. A summary of the main fully homomorphic schemes is provided in Table 2.

3.3 Librairies

Some of the open-source libraries available are listed in Table 3. These libraries are written in `C++`, except Lattigo which is written in `Go`. In particular the OpenFHE library stands-out by gathering members of Palisade, HElib, and other libraries to build a centralized, maintained and up-to-date library for full homomorphic encryption (FHE) schemes. Palisade has now been merged with OpenFHE.

Starting in 2017, at a time where the number of libraries offering fully-homomorphic encryption is on the rise, academia and industry experts have gathered their efforts to start a standardization

Table 2: Fully homomorphic schemes summary

Names	Generation	Fully homomorphic	Target
Gentry (Gentry, 2009)	1st generation	Yes	bits
BGV (Brakerski et al., 2014)	2nd generation	Yes	integers
BFV (Brakerski, 2012)(Fan and Vercauteren, 2012)	2nd generation	Yes	integers
FHEW (Ducas and Micciancio, 2015)	3rd generation	Yes	integers
TFHE (Chillotti et al., 2020)	3rd generation	Yes	integers
CKKS (Cheon et al., 2017)	4th generation	Yes	floats

Source: Own production

Table 3: List of the main open-source libraries for fully-homomorphic encryption schemes. The CKKS* column refers to the CKKS scheme without bootstrapping procedure.

Library	BGV	BFV	FHEW	TFHE	CKKS*	CKKS
OpenFHE	yes	yes	yes	yes	yes	yes
Palisade	yes	yes	yes	yes	yes	
Lattigo	yes	yes			yes	yes
Microsoft SEAL	yes	yes			yes	
IBM HElib	yes			yes		

Source: Own production

process. They aim at explaining in simple terms the mathematical and security concepts behind homomorphic encryption, as well as clarifying the landscape of available options and parameters. Their draft standard is updated after each of their meetings and is accessible on their webpage. (Albrecht et al., 2018).

4 Experiment

The aim of this experiment is to study the feasibility of using perceptual hashing on data encrypted with an homomorphic scheme, in the context of the detection of CSAM exchanged via encrypted messages.

4.1 Description of the experiment

Alice encrypts her media m with a fully homomorphic encryption scheme $henc$ with Bob's public key pk_B. With his private key sk_B, Bob can decrypt the message and obtain the media m. When the encrypted media c transits through the server, the server computes $\mathcal{H}(c) = \mathcal{H}(henc_{pk_B}(m))$. Due to homomorphic properties this is equivalent to computing $henc_{pk_B}(\mathcal{H}(m))$. The server can then compare this results to the items in the database $\mathcal{D}$, but in order to perform the comparison the server needs to encrypt each elements of the database with Bob's public key.

These steps are represented in Figure 5 where, for simplicity, the operations for the comparison are omitted. These operations are detailed in Section 5.3, Figure 6.

Figure 5: Exchange of a message encrypted with a homomorphic scheme between Alice and Bob with analysis by server

Alice		Server		Bob
				(pk_B, sk_B)
m				
$c = henc_{pk_B}(m)$				
c	$\longrightarrow$	c	$\longrightarrow$	c
		$\mathcal{H}(c)$		$hdec_{sk_B}(c) = m$
		$= \mathcal{H}(henc_{pk_B}(m))$		
		$= henc_{pk_B}(\mathcal{H}(m))$		
		$\in henc_{pk_B}(\mathcal{D})?$		

Source: Own production

Note that, in order to allow for homomorphic computations, the encryption is performed with an asymmetric scheme, that is with a public key and a private key, while most messaging applications such as Signal and WhatsApp use symmetric protocols.

4.2 Choice and description of the hash function

We chose the PDQ hash function from Meta for our experiment as it is open source and available on Github[1] in a variety of programming languages, hence easing the compatibility with the OpenFHE library.

Successive steps of the PDQ algorithm

1. Resize the image to 512×512 resolution.
2. Conversion of the image from RGB to luminance.
3. Using a Jarosz two-pass filter, compute a 64×64 down-scaling of the 512×512 image.
4. On the 64×64 down-scaled image, compute the quality metric by computing the sum of absolute values of horizontal and vertical gradients.
5. On the 64×64 down-scaled image for the 1 to 16 slots in X and Y direction, compute a two-dimensional discrete cosine transform (DCT) 16×16 matrix.
6. For the 16×16 DCT matrix, compute the median value.

[1] https://github.com/facebook/ThreatExchange/tree/main/pdq

7. Compute the hash by considering each element of the 16×16 matrix. If the element is greater than the median append a 1 to the hash, append 0 otherwise. This gives a 256-bit hash output with Hamming weight 128.

8. Return the hash and the quality metric.

4.3 Choice of the encryption scheme and choice of parameters

The PDQ perceptual hash function uses matrices of floats. As other OpenFHE schemes can only operate on bits and integers, the CKKS scheme is the straight forward choice to use.

The CKKS scheme requires several setting options such as

- the number of plaintext values within each ciphertext.
- the multiplicative depth, i.e. the highest number of successive multiplication performed for each term.
- the desired computation precision accuracy.

Increasing the multiplicative depth and the precision lead to slower computations. That is why we aim at choosing the smallest valid parameters for our experiment. The rational behind our decisions for these three parameters is as follows:

Number of plaintext values per ciphertext To compute the perceptual hash function the algorithm performs operations on the value of each pixel. To perform the same operations on an encrypted version of the image we hence need to encrypt each pixel independently. For this reason we have set the number of plaintext values within each ciphertext to one.

Multiplicative depth Comparisons between two numbers in CKKS are expensive in term of multiplications as they require to switch to another homomorphic scheme (TFHE in this case), perform the comparison and then switch back the result in CKKS. Counting the number of successive multiplications in the source code shows that the comparison alone requires a multiplicative depth of 17. The Discrete Cosine Transform matrix computation has multiplicative depth 2, while the conversion from RGB to luminance has multiplicative depth 1. Following this analysis we have set the multiplicative depth at 20 for our experiment.

Precision The precision decreases with every operation performed. However since our final hash results from comparisons we do not need a very high precision for our intermediate computations. For this reason we kept the default initial precision value to 50.

4.4 Design of the experiment

We start by identifying the bottlenecks of the computation by describing the theoretical cost of each step of the PDQ algorithm in terms of number of additions A, multiplications M and comparisons C. Note that for CKKS with the parameters chosen, an addition is cheaper than a multiplication, which is cheaper than a comparison as described in Table 4.

Table 4: Empiric comparative costs of an addition, a multiplication and a comparison for the CKKS scheme with parameters as described above. Test realised on an 11th Gen Intel(R) Core(TM) i7-1165G7 @2.80GHz laptop for 100 operations.

Operation	Timing	Ratio over addition
Addition	0.6 ms	1
Multiplication	29 ms	48
Comparison	1260 ms	2100

Source: Own production

Conversion to luminance: The luminance is computed from an RGB image as follows: $l = a * R + b * G + c * B, a, b, c \in [0, 1]$. For the sake of simplicity we approximate the cost of the multiplication by a scalar with cost of the multiplication by a ciphertext, even though for CKKS the former is typically cheaper than the latter. Hence each pixel require $3M + 2A$. It follows that an $m \times m$ image requires $m^2 \times (3M + 2A) = 3m^2M + 2m^2A$.

Two-pass Jarosz filter: The two-pass Jarosz filter is computed by performing four one-dimensional filters in alternate horizontal and vertical direction. Finally the center pixel of the $\frac{m}{64} \times \frac{m}{64}$ block is selected to be the corresponding pixel of the 64×64 matrix. The $m \times m$ image is divided into $m/64 \times m/64$ blocks. For each $m/64 \times m/64$ block, the sliding-window size w for the one-dimensional filter is half of the block size, that is $w = \frac{m}{64} \times \frac{1}{2}$. From the code provided by Meta, the one-dimensional filter costs $(w-1)A + w(A+S) + w(\frac{1}{2}A + \frac{1}{2}S)$. We will make the approximation that $A = S$ for simplicity. We obtain that the cost for a one-dimensional filter is $(w-1)A + 2wA + wA = 4wA + A = (4w+1)A$.

Overall, the computation of the two-pass Jarosz filter for the entire $m \times m$ image requires $64 \times 64 \times 4 \times (4\frac{m}{64}\frac{1}{2} + 1)A = 64 \times 64 \times 4 \times (\frac{m}{32} + 1)A$

Quality metric computation: The quality indicator is an integer between 0 and 100 obtained by summing the quantized absolute value of the nearest-neighbor gradient for each pixel, with the formula:

$$\sum_{i=0}^{64} |m[i][j] - m[i+1][j]| \times \frac{100}{255} + \sum_{i=0}^{64} |m[i][j] - m[i][j+1]| \times \frac{100}{255}$$

The quality indicator is the nearest integer value of this result divided by 90, a heuristic constant, to bring it below 100. The total cost is hence $2 \times n \times (2A + M)$ for a $n \times n$ input.

Discrete Cosine Transform: For an $n \times n$ image represented by an $n \times n$ matrix A, the discrete cosine transform $r \times r$ matrix B is computed as

$$B_{i,j} = \sum_{k=0}^{n-1} D_{i,k} \sum_{l=0}^{n-1} A_{k,l} D_{j,l}$$

for i and j integers between 0 and $r-1$, with

$$D_{i,j} = \sqrt{\frac{2}{n}} \cos(\frac{2\pi}{4n} i(2j+1))$$

Note that as the $D_{i,j}$ are not image dependent they can actually be computed in clear. This greatly enhance the performances as the computation of the cosine on encrypted data would require to perform costly Fourier development, which increases further the needed multiplicative depth.

The sum $\sum_{l=0}^{n-1} A_{k,l} D_{j,l}$ costs $nM + (n-1)A$. This type of sum has to be computed n times. Then each $B_{i,j}$ costs

$$n(nM + (n-1)A) + nM + (n-1)A = n(n+1)M + (n+1)(n-1)A$$

There are $r \times r$ such $B_{i,j}$ to compute which gives a total cost of $r^2 n(n+1)M + r^2(n+1)(n-1)A$.

Median computation: The median computation of a list of size ℓ can be done with the quickselect algorithm in ℓ comparisons, that is in $\ell\, C$.

Conclusion: For the costs above, we typically have $m = 512$, $n = 64$ and $r = 16$, following the description of the PDQ algorithm.

We summarize the theoretical costs calculated in Table 5 and the cost of additions, multiplications and comparisons for the selected parameters in CKKS in table 4. From these two tables it appears that the conversion to luminance and the Discrete Cosine Transform computations are the most costly to perform.

Table 5: Theoretical cost of successive steps in PDQ algorithm in term of additions A, multiplications M and comparisons C.

Step	Cost for a 512×512 image in term of A, M and C	Cost for a 512×512 image in term of A
Conversion to luminance	$786\,432\ M + 524\,288\ A$	$38\,273\,024\ A$
Two-pass Jarosz filter	$278\,528\ A$	$278\,528\ A$
Quality Metric Computation	$128\ M + 128\ A$	$6\,272\ A$
Discrete Cosine Transform	$1\,064\,960\ M + 1\,048\,320\ A$	$52\,166\,400\ A$
Median computation	$256\ C$	$537\,600\ A$
Hash final computation	$256\ C$	$537\,600\ A$

Source: Own production

5 Results

5.1 Memory

With the parameters chosen each encrypted ciphertext is 136 bytes. As the computation of the perceptual hash requires each pixel to be encrypted separately this scales-up when considering the encryption of large images. For the present analysis we limit ourselves to 512×512 RGB images, whose cumulative ciphertext would already fill more than 100 Megabytes.

Table 6: Size of the different encrypted data types

Data	Size in bytes		
Encrypted 512×512 RGB image	$512 \times 512 \times 3 \times 136 =$	106.9×10^6 bytes	$= 107$ MB
Encrypted 512×512 luminance image	$512 \times 512 \times 136 =$	35.6×10^6 bytes	$= 36$ MB
Encrypted Discrete Cosine Transform Matrix	$64 \times 64 \times 136 =$	557×10^3 bytes	$= 557$ kB
Encrypted Hash	$256 \times 136 =$	34816 bytes	$= 35$ kB

Source: Own production

In short, a 512×512 RGB-image of less than one Megabytes would produce an encrypted image of more than 100 Megabytes. This is likely to create an issue of memory both for bandwidth and storage: encrypted 512×512 images are sent from Alice to Bob via the server, while millions of encrypted hashes must be stored by the server. Besides, the amount of memory required for intermediate operations, both for the encryption and the perceptual hash computation, require at least 8 Gigabytes of RAM. Finally, the encryption of the server's database of known-problematic hashes with each public key also requires a large memory space, given that such databases can contain million of hashes.

For example, the NCMEC database contains more than 10 million of hashes. Encrypting this database for each user with the user's public key would require 350 Gigabytes per user, meaning that a service like Meta having 2 billion users would require 700 Exabytes (10^{18} bytes) of memory to store the encrypted databases of problematic content. Besides, user's keys change regularly, sometimes for every message received in Signal or Whatsapp, meaning that the database would have to be recomputed frequently.

5.2 Timing

As the encryption and the conversion to luminance operations are applied to each pixel independently, we measured the computations for a fraction of the image, and then extrapolate the expected timings for a full size image. Limiting the size of the input image also avoids stack overflow during the encryption step.

The Discrete Cosine Transform is systematically performed on a 64×64 down-scaled image from the input, so its timing is independent of the size of the initial input. Each coefficient of the Discrete

Cosine Transform matrix is computed with equivalent formula up to indices, so we measured the computation of one matrix coefficient (125 s) and extrapolated the time needed for the entire 16×16 Discrete Cosine Transform matrix output (8.88 h).

As the Median computation and the Hash final computation are computed on the 16×16 Discrete Cosine Transform matrix, their timing is also independent from the size of the input image.

Table 7: Timings and extrapolated timings (with *) to perform the successive steps of the PDQ algorithm on an 11th Gen Intel(R) Core(TM) i7-1165G7 @2.80GHz laptop

Step	One pixel	8×8 image	64×64 image	512×512 image
Image Encryption	52.1 ms	3279 ms	209.9^* s	3.731^* h
Conversion to luminance	4 ms	275 ms	17^* s	1126^* s
Discrete Cosine Transform	-	-	8.88^* h	8.88^* h
Median computation (On average 256 comparisons)	-	-	335 s	335 s
Hash final computation (Exactly 256 comparisons)	-	-	335 s	335 s

Source: Own production

The database of problematic hashes also needs to be encrypted with Bob's public key, with each of the 256 bits of the hashes having to be encrypted separately to allow for comparisons. This would mean an encryption time of $51.2 \times 256 = 13107$ ms, that is 13.1 s per hash encryption. The time needed to encrypt the entire database is an another limitation, since databases of problematic hashes can have million of entries.

In order to link with Meta's claim that it would take more than seven months to scan for one image, note that with 13.1 seconds per hash encryption one could (sequentially) encrypt 1.4 million hashes in seven months, leaving other computation aside, in particular the comparison. However, NCMEC has more than 10 million hashes in their database. Hence scanning with the entire database would indeed take more than seven months, even with servers more powerful than the laptop we used for our experiment.

Note that most of the operation performed can be parallelized, particularly the Discrete Cosine Transform which is performed independently for each encrypted pixel. However even if we imagine a parallelization where we have as many cores as the number of pixels, the timing for each pixel is already prohibiting when analyzing a large amount of images.

Finally, note that users usually install the messaging apps on their smartphones, which typically have less RAM and compute less operations per seconds than a laptop.

5.3 Comparison computations

The comparison between two real numbers is not an arithmetic operation, creating difficulties when implementing them through homomorphic encryption schemes. The CKKS library performs the comparison by switching to another homomorphic encryption scheme, TFHE, performing the comparison with this second scheme, then switching back to CKKS, which requires a considerable amount of time as described in Table 4.

Furthermore, the result of the comparison is also encrypted with the recipient public key, meaning that only the intended recipient of the message can know the result. Most homomorphic encryption schemes are proven semantically secure, including CKKS (Li and Micciancio, 2021), hence unless the recipient is collaborating with the server, the server will not be able to know if two ciphertexts are equal.

This is illustrated on Figure 6, where we detail the interactions needed between the server and Bob to determine if the hash of the message m appears in the database $\mathcal{D}$ of known problematic content. The recipient Bob has a private key sk_B and a public key pk_B. To send a message only readable by Bob, Alice encrypts it with an homomorphic scheme and with Bob's public key and obtains a ciphertext c. She sends c to Bob via the server, and Bob obtains m by decrypting c using his private

Figure 6: Exchange of a message encrypted with a homomorphic scheme between Alice and Bob with the comparison detailed

Alice		Server		Bob
				(pk_B, sk_B)
m				
$c = henc_{pk_B}(m)$				
c	$\longrightarrow$	c	$\longrightarrow$	c
				$hdec_{sk_B}(c) = m$
		Computes $\mathcal{H}(c)$		
		For all $h_i \in \mathcal{D}$ compute:		
		$b_i = \text{IsEqual}(\mathcal{H}(c), henc_{pk_B}(h_i))$		
		b_i	$\longrightarrow$	b_i
				$a_i = hdec_{sk_B}(b_i)$
		a_i	$\longleftarrow$	a_i
		If $a_i = 1$ then report		

Source: Own production

key sk_B. Meanwhile, the server computes the hash $\mathcal{H}(c)$ of the ciphertext. Moreover, for each hash h_i of the database $\mathcal{D}$ it also encrypts h_i with Bob's public key pk_B, and tests if $henc_{pk_B}(h_i)$ is equal to the hash $\mathcal{H}(c)$ of the ciphertext with the homomorphic IsEqual function. The IsEqual function returns $b_i = henc_{pk_B}(1)$ if they are equal, and $b_i = henc_{pk_B}(0)$ otherwise. Note that this result b is encrypted with Bob's public key, hence he is the only one able to decrypt it[2]. The server sends b_i to Bob, who decrypts it with his secret key sk_B and sends back the result a_i. If any a_i is equal to 1, it means that $\mathcal{H}(m)$ is in the database $\mathcal{D}$, and the server can report it. Note that some masking technics can be used to make sure that Bob is not lying and that he does return the correct result of $hdec_{sk_B}(b_i)$. For the sake of simplicity this is not detailed here.

This is also an issue during the computation as the quality metric, the median computation and the final hash computation require comparisons, meaning that Bob's collaboration is also needed when the server performs the hash computation. For simplicity, this necessary collaboration between the server and the recipient of the message during the computation of the hash $\mathcal{H}(c)$ is not represented on Figure 6.

While the memory requirements and timings are scheme and parameter dependent, the issue of comparisons is present with every (semantically secure) homomorphic scheme.

5.4 Reporting

If $a_i = 1$ for some i, this means that the hash of the message sent by Alice $\mathcal{H}(m)$ is equal to a hash of some problematic content h_i stored in the database $\mathcal{D}$. However, this does not necessarily mean that the contents are equal, as some collisions are possible from the hash function $\mathcal{H}$ (see for example (Dolhansky and Canton-Ferrer, 2020)). In the protocol described above, the server cannot decrypt c to check that the content is indeed problematic, and not arising from a collision.

Conversely, if $a_i = 0$, it does not necessarily mean that the content of the image is harmless, only that it is not registered (yet) in the database of known problematic content.

6 Conclusion

The use of homomorphic encryption and perceptual hashing in the context of the detection of child sexual abuse material exchanged via encrypted messages faces three severe limitations: the mem-

[2] This is inevitable for semantically secure cryptographic schemes

ory, the timing performances and the comparisons. These limitations are valid both during the hash computation performed and the comparison with the database.

Memory The images need to be encrypted pixel by pixel in order to be able to perform the operations of the perceptual hash function, meaning that images cannot be compressed and the size of the ciphertext grows linearly with the resolution of the image. This means that a large quantity of data needs to be transmitted, stored and processed.

Timing performances Independently of the size of the input, in our experiment with CKKS more than eleven hours are necessary to perform the computation of the perceptual hash on data encrypted with an homomorphic scheme, making it impossible to use in practice at a large scale.

Comparisons Finally, independently of the homomorphic scheme used, the result of a comparison between two ciphertexts is only available to the intended recipient of a message, Bob in our examples. Bob's collaboration is hence necessary for the server to compute the hash and to compare it to its database.

These limitations are even more salient when looking at the necessary time, memory and exchanges with the intended recipient of the message to encrypt and compare the targeted hash with the database. Even though using large data centers and parallelisation could improve in part the timing performances, the comparison issue would remain a hard constraint with any semantically secure cryptographic scheme.

Possible next steps on this topic could include:

- studying other perceptual hash functions, ideally open-source, to see if some of them could have better performances when paired with homomorphic encryption for the three identified bottlenecks, namely memory, timings and comparisons.
- studying other possible technics to prevent the dissemination of CSAM content, such as prevention, reporting procedures, or forums' moderation.

References

Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A. and Vaikuntanathan, V., 'Homomorphic encryption security standard', Tech. rep., Toronto, 2018.

Apple, 'Neural Hash'. https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf, 2021. [Online; accessed 10-February-2025].

Arachnid, 'A world leader in reducing the availability of child sexual abuse material'. https://projectarachnid.ca/en/#what-is-project-arachnid, 2023. [Online; accessed 11-Decembre-2023].

Boura, C., Gama, N., Georgieva, M. and Jetchev, D., 'CHIMERA: combining ring-lwe-based fully homomorphic encryption schemes', J. Math. Cryptol., Vol. 14, No 1, 2020, pp. 316–338.

Brakerski, Z., 'Fully homomorphic encryption without modulus switching from classical gapsvp', In 'Advances in Cryptology - CRYPTO 2012', Lecture Notes in Computer Science, Vol. 7417. Springer, pp. 868–886.

Brakerski, Z., Gentry, C. and Vaikuntanathan, V., '(leveled) fully homomorphic encryption without bootstrapping', ACM Trans. Comput. Theory, Vol. 6, No 3, 2014, pp. 13:1–13:36.

BSR, 'Human rights impact assessment: Meta's expansion of end-to-end encryption', Tech. rep., https://www.bsr.org/en/, April 2022.

Bursztein, E., Clarke, E., DeLaune, M., Elifff, D. M., Hsu, N., Olson, L., Shehan, J., Thakur, M., Thomas, K. and Bright, T., 'Rethinking the detection of child sexual abuse imagery on the internet', In 'The World Wide Web Conference, WWW 2019, San Francisco', ACM, pp. 2601–2607.

Business of Apps, 'Signal Revenue and Usage Statistics'. https://www.businessofapps.com/data/whatsapp-statistics/, 2024. [Online; accessed 5-Dec-2024].

Business of the App, 'Signal Revenue and Usage Statistics'. https://www.businessofapps.com/data/signal-statistics/, 2024. [Online; accessed 5-Dec-2024].

Cheon, J. H., Han, K., Kim, A., Kim, M. and Song, Y., 'Bootstrapping for approximate homomorphic encryption', In 'Advances in Cryptology - EUROCRYPT 2018', Lecture Notes in Computer Science, Vol. 10820. Springer, pp. 360–384.

Cheon, J. H., Kim, A., Kim, M. and Song, Y. S., 'Homomorphic encryption for arithmetic of approximate numbers', In 'Advances in Cryptology - ASIACRYPT 2017', Lecture Notes in Computer Science, Vol. 10624. Springer, pp. 409–437.

Chillotti, I., Gama, N., Georgieva, M. and Izabachène, M., 'TFHE: fast fully homomorphic encryption over the torus', J. Cryptol., Vol. 33, No 1, 2020, pp. 34–91.

Committee on Civil Liberties, J. and Affairs, H., 'Report on the proposal for a regulation of the european parliament and of the council laying down rules to prevent and combat child sexual abuse'. https://www.europarl.europa.eu/doceo/document/A-9-2023-0364_EN.html, 2023.

Dolhansky, B. and Canton-Ferrer, C., 'Adversarial collision attacks on image hashing functions', CoRR, 2020. URL https://arxiv.org/abs/2011.09473.

Ducas, L. and Micciancio, D., 'FHEW: bootstrapping homomorphic encryption in less than a second', In 'Advances in Cryptology - EUROCRYPT 2015', , edited by E. Oswald and M. FischlinLecture Notes in Computer Science, Vol. 9056. Springer, pp. 617–640.

European Commission, 'Eu security union strategy'. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0605&qid=1737370428390, 2020a.

European Commission, 'Eu strategy for a more effective fight against child sexual abuse'. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52020DC0607, 2020b.

European Commission, 'COMMISSION STAFF WORKING DOCUMENT IMPACT ASSESSMENT REPORT Accompanying the document Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL laying down rules to prevent and combat child sexual abuse'. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52022SC0209, 2022a.

European Commission, 'Proposal for a regulation of the european parliament and of the council laying down rules to prevent and combat child sexual abuse'. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52022PC0209, 2022b.

European Commission, 'Commission proposal to review Directive 2011/93/EU on combating the sexual abuse and sexual exploitation of children and child pornography'. https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/13073-Combating-child-sexual-abuse-review-of-EU-rules_en, 2024.

European Court of Human Rights, 'Case of podchasov v. russia'. https://hudoc.echr.coe.int/eng/?i=001-230854, 2024.

European Data Protection Supervisor, European Data Protection Board, 'Edpb-edps joint opinion 04/2022 on the proposal for a regulation of the european parliament and of the council laying down rules to prevent and combat child sexual abuse'. https://www.edpb.europa.eu/our-work-tools/our-documents/edpbedps-joint-opinion/edpb-edps-joint-opinion-042022-proposal_en, 2022.

European Parliament and the Council, '2011/92/eu, directive 2011/92/eu - combating the sexual abuse and sexual exploitation of children and child pornography'. Official Journal of the European Union, 2011.

Fan, J. and Vercauteren, F., 'Somewhat practical fully homomorphic encryption', IACR Cryptol. ePrint Arch., 2012, p. 144.

Geelen, R. and Vercauteren, F., 'Bootstrapping for BGV and BFV revisited', IACR Cryptol. ePrint Arch., 2022, p. 1363.

Gentry, C., A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University, USA, 2009.

Google, 'HTTPS encryption on the web'. https://transparencyreport.google.com/https/overview, 2023. [Online; accessed 11-Decembre-2023].

Hao, Q., Luo, L., Jan, S. T. and Wang, G., 'It's not what it looks like: Manipulating perceptual hashing based applications', In 'Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security', Association for Computing Machinery. ISBN 9781450384544, p. 69–85. .

Henley, A., 'Implementing cosine in c from scratch'. https://austinhenley.com/blog/cosine.html, 2020. [Online; accessed 1-Sept-2023].

Jain, S., Cretu, A.-M. and de Montjoye, Y.-A., 'Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning', In '31st USENIX Security Symposium', USENIX Association.

LetsEncrypt, 'Let's Encrypt Stats'. https://transparencyreport.google.com/https/overview, 2023. [Online; accessed 11-Decembre-2023].

Li, B. and Micciancio, D., 'On the security of homomorphic encryption on approximate numbers', In 'Advances in Cryptology - EUROCRYPT 2021', Lecture Notes in Computer Science. Springer.

Lu, W., Huang, Z., Hong, C., Ma, Y. and Qu, H., 'PEGASUS: bridging polynomial and non-polynomial evaluations in homomorphic encryption', In '42nd IEEE Symposium on Security and Privacy, SP 2021', IEEE, pp. 1057–1073.

Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H. P. and Aaraj, N., 'Survey on fully homomorphic encryption, theory, and applications', Proc. IEEE, Vol. 110, No 10, 2022, pp. 1572–1609.

McKeown, S. and Buchanan, W. J., 'Hamming distributions of popular perceptual hashing techniques', Forensic Sci. Int. Digit. Investig., Vol. 44, No Supplement, 2023, p. 301509.

Messenger, 'Messenger'. https://m.facebook.com/messengerfacts, 2024. [Online; accessed 4-April-2024].

Meta, 'PDQ'. https://github.com/facebook/ThreatExchange/blob/main/hashing/hashing.pdf, 2019. [Online; accessed 10-February-2025].

Microsoft, 'PhotoDNA'. https://www.microsoft.com/en-us/photodna, 2009. [Online; accessed 3-Decembre-2024].

National Center for Missing and Exploited Children, '2021 CyberTipline Reports by Electronic Service Providers (ESP)'. https://www.missingkids.org/content/dam/missingkids/pdfs/2021-reports-by-esp.pdf, 2021.

National Center for Missing and Exploited Children, 'Contributors'. https://www.missingkids.org/footer/about/annual-report, 2023. [Online; accessed 20-July-2023].

National Center for Missing and Exploited Children, 'Impact'. https://www.missingkids.org/ourwork/impact, 2024. [Online; accessed 11-July-2024].

Rivest, R. L., Adleman, L. and Dertouzos, M. L., 'On data banks and privacy homomorphisms', .

Starlight, 'Enhancing the eu's strategic autonomy in the field of artificial intelligence (ai) for law enforcement agencies (leas)'. https://www.starlight-h2020.eu/sites/default/files/2023-03/starlight_leaflet.pdf, 2021.

Struppek, L., Hintersdorf, D., Neider, D. and Kersting, K., 'Learning to break deep perceptual hashing: The use case neuralhash', In 'Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency', FAccT '22. Association for Computing Machinery, p. 58–69.

Thorn, 'How Hashing and Matching Can Help Prevent Revictimization'. https://www.thorn.org/blog/hashing-detect-child-sex-abuse-imagery/, 2023. [Online; accessed 23-October-2023].

WhatsApp, 'WhatsApp'. https://www.whatsapp.com/about, 2024. [Online; accessed 4-April-2024].

Youtube, 'CSAI Match'. https://protectingchildren.google/tools-for-partners/#learn-about-our-tools, 2014. [Online; accessed 10-February-2025].

List of abbreviations and definitions

BFV Brakerski, Fan, Vercauteren (Brakerski, 2012, Fan and Vercauteren, 2012)

BGV Brakerski, Gentry, Vaikuntanathan (Brakerski et al., 2014)

CKKS Cheon, Kim, Kim, Song (Cheon et al., 2017)

CSAM Child Sexual Abuse Material

CSAI Child Sexual Abuse Imagery

DCT Discrete Cosine Transform

FHE Full Homomorphic Encryption

NCMEC National Center for Missing and Exploited Children

PDQ Perceptual hasher using Discrete cosine transform with Quality metric

RAM Random Access Memory

RGB Red Green Blue

RSA Rivest, Shamir, Adleman

List of figures

List of tables

Annexes

Annex 1. Mathematical foundations

The homomorphic adjective refers to a mathematical property: a homomorphism is a structure-preserving map between two algebraic structures of the same type. For example, for a given $m \in \mathbb{R}$, the multiplication by m,

$$f : (\mathbb{R}, +) \longrightarrow (\mathbb{R}, +),$$
$$x \mapsto mx$$

is a homomorphism for the addition. Indeed, we have $f(a + b) = m(a + b) = ma + mb = f(a) + f(b)$.

Note that it does not have to be the same operation on the domain and codomain. For example, the exponentiation $g : (\mathbb{R}, +) \longrightarrow (\mathbb{R}, \times), x \mapsto e^x$ is also an homomorphism. We have $g(x + y) = e^{x+y} = e^x \times e^y = g(y) \times g(y)$.

In the case of encryption, an encryption algorithm Enc is homomorphic for an operation written $*$ if $Dec(Enc(a) * Enc(b)) = Dec(Enc(a * b))$.

Annex 2. Fully homomorphic encryption schemes

Several generations of schemes are usually considered.

First generation The first generation gathers schemes based on ideal lattices and on integers.

The initial idea from Gentry in 2009 (Gentry, 2009) used ideal lattices. its security relies on the shortest vector problem. It can homomorphically encrypt single bits. Despite being the first fully homomorphic encryption scheme it is impracticable for efficiency reasons.

In 2010 van Dijk, Gentry, Halevi and Vaikuntanathan introduced a new scheme using integers and the Approximate Greatest Common Divisor problem. It can homomorphically encrypt integers. However its high computational complexity and large key size maintains it in the unpracticable realm.

Second generation

From 2009, research has been focused on reducing the cost of this bootstrapping operation. The schemes of the second generation get closer to this goal by introducing a new operation after each operation performed on the ciphertext: the relinearization and modulus switching. This allows to delay or to avoid altogether the need for costly bootstrapping operations, leading to fully homomorphic somewhat feasible encryption schemes (Brakerski et al., 2014) (Brakerski, 2012, Fan and Vercauteren, 2012).

BGV and BFV are based on the LWE and RLWE problem. Both schemes can be used as fully homomorphic encryption using bootstrapping or as leveled fully homomorphic encryption for which a certain number of computations can be performed without bootstrapping. The two schemes BGV and BFV improved the state-of-the-art by augmenting the threshold of non-return, so that ideally all computations can be performed without reaching this threshold. However the bootstrapping operation remains costly. The second generation also includes the schemes derived from NTRU, developed by Stehlé and Steinfeld.

Overall second generation schemes are good for leveled schemes and computations over large arrays of number in a finite field, but non optimal when bootstrapping is involved.

Third generation

The third generation gathers two LWE and RLWE-based schemes. They stand from the second generation by improved bootstrapping procedure.

The first one has been developed by Gentry, Sahai, Waters. Their scheme is referred to as GSW for the original version and as FHEW when all the improvements are taken into account. FHEW gives better performances than BGV for bootstrapping. It also replaces key and modulus switching by the approximate eigenvector method which improves the relinearisation process compared to the second generation.

A variant of FHEW, referred to as TFHE, has been developed by using the LWE problem over a torus instead of a ring. It provides better bootstrapping procedure and smaller key size.

Fourth generation

In 2017 Cheon, Kim, Kim and Song developed a leveled homomorphic scheme HEAAN with floating point arithmetic and approximate computations which make the scheme faster (Cheon et al., 2017). It is often referred to as CKKS for the initials of its authors. In 2018 a fully homomorphic version of this scheme was released in (Cheon et al., 2018).

In the following years several improvements have been made to CKKS to decrease the noise during computation, and make the bootstrapping more efficient.

Note that it is possible build a scheme allowing to switch between third-generation TFHE and second-generation FV, as well as between from fourth-generation CKKS to second-generation FV, providing flexibility between the schemes (Boura et al., 2020) (Lu et al., 2021).

Getting in touch with the EU

In person

All over the European Union there are hundreds of Europe Direct centres. You can find the address of the centre nearest you online (european-union.europa.eu/contact-eu/meet-us_en).

On the phone or in writing

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

— by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),

— at the following standard number: +32 22999696,

— via the following form: european-union.europa.eu/contact-eu/write-us_en.

Finding information about the EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website (european-union.europa.eu).

EU publications

You can view or order EU publications at op.europa.eu/en/publications. Multiple copies of free publications can be obtained by contacting Europe Direct or your local documentation centre (european-union.europa.eu/contact-eu/meet-us_en).

EU law and related documents

For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex (eur-lex.europa.eu).

EU open data

The portal data.europa.eu provides access to open datasets from the EU institutions, bodies and agencies. These can be downloaded and reused for free, for both commercial and non-commercial purposes. The portal also provides access to a wealth of datasets from European countries.

Science for policy

The Joint Research Centre (JRC) provides independent, evidence-based knowledge and science, supporting EU policies to positively impact society

EU Science Hub
Joint-research-centre.ec.europa.eu

Publications Office
of the European Union