Future Generation Computer Systems 145 (2023) 211–222

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A decentralised messaging system robust against the unauthorised forwarding of private content

Mirko Franco [a,*], Ombretta Gaggi [a], Barbara Guidi [b], Andrea Michienzi [b], Claudio E. Palazzi [a]

[a] Department of Mathematics, University of Padua, Padua, Italy
[b] Department of Computer Science, University of Pisa, Pisa, Italy

ARTICLE INFO

Article history:
Received 9 January 2023
Received in revised form 2 March 2023
Accepted 17 March 2023
Available online 24 March 2023

Keywords:
Blockchain
Decentralised architecture
Messaging system
NFT
Private
Sexting
Social network

ABSTRACT

The United Nations defined 17 Sustainable Development Goals (SDGs) to foster equitable, healthy, inclusive and safe communities. Clearly, they involve even social networks and, in particular, the sexuality expressed through them. For instance, consider sexting, the practice of sharing self-generated explicit content through mobile devices. Besides its popularity, this phenomenon carries several concerns, such as the possible damages caused by the spread of personal nude or semi-nude images without the owner's consent. Unfortunately, messaging applications generally used to practice sexting are not safe enough as they permit to share any received content with anyone else. Aimed at preventing sexting-related adverse consequences for the wellness of people and creating safer, gender-equal and inclusive online communities, we discuss possible technological approaches to contrast the non-consensual spread of private self-generated content and, in particular, we analyse the impact of employing decentralised architectures in this context.

1. Introduction

In 2015, the 193 members of the United Nations (UN) agreed on the 17 Sustainable Development Goals (SDGs). These are part of the broader 2030 Agenda for Sustainable Development and represent the global action plan for the next decade to end poverty and hunger, address basic social needs (e.g., health, education, social protection, job opportunities, etc.), tackle climate changes and create equitable, healthy, inclusive and safe (i.e., sustainable) communities. Some of them are strongly connected with social networks and communities: Good Health and Well-being (3), Gender Equality (5), Sustainable Cities and Communities (11) and Peace, Justice and Strong Institutions (16). Indeed, the aim of these SDGs consist of (a) promoting well-being and ensuring healthy lives for all at all ages, (b) achieving gender equality and empowering all women and girls, (c) making communities safe and sustainable and (d) promoting inclusive societies and providing access to justice for all [1].

In this context, we have to consider that the advent of social networking platforms, the spread of mobile devices and the ubiquitous availability of wireless connectivity have revolutionised our communication and fostered the creation of online communities, helping many businesses grow and providing nearly-instant access to information [2]. These platforms have thus moved part of our activities and relationships online, becoming the places where life happens for many (young) people. According to the Pew Research Center [3], in 2022, 95% of U.S. teens have access to smartphones and 46% of them state to be almost constantly online. Besides the enormous number of opportunities that social media platforms provide to adolescents (and people in general), they have also raised several new concerns about the privacy and safety of their users [4]. For instance, many messaging applications currently available in the stores allow users to send whatever they want to anyone without any limitation, thus facilitating the spreading of personal content without the owner's consent [5]. In this context, social networking platforms, if properly designed, can embody a fundamental tool in achieving SDGs.

One interesting case study is sexting — the practice of sending or receiving sexually explicit content (e.g., text, images, videos, etc.) through social media platforms (e.g., dating applications, messaging systems, etc.) [6,7]. This phenomenon has become popular among teenagers, but not only, and has gained the interest of the scientific communities of psychologists, computer scientists, sociologists and even doctors, representing an interdisciplinary topic.

Despite being considered as a normal sexual development behaviour [7], it can lead to several adverse consequences. The most famous one is the non-consensual spread of private sex-related content (i.e., revenge porn) [8], with effects that are even worse when involving images or videos. The consequences are also uneven with respect to gender, especially in the case of

* Corresponding author.
E-mail addresses: mifranco@math.unipd.it (M. Franco), gaggi@math.unipd.it (O. Gaggi), guidi@di.unipi.it (B. Guidi), andrea.michienzi@unipi.it (A. Michienzi), cpalazzi@math.unipd.it (C.E. Palazzi).

https://doi.org/10.1016/j.future.2023.03.025

revenge porn [9]. The outcome of a survey administered by Plan International in 2020 shows that 58% of girls surveyed experienced online harassment, which also includes Non-Consensual Intimate Images (NCII) abuses [10].

The frequency of the phenomenon is not fully known. Indeed, the victims may not be aware of the spread of their private content (e.g., self-generated nudes). Furthermore, reporting incidents and accessing justice systems can be difficult [11], especially if we consider that victims may be embarrassed and afraid of a second victimisation. Yet, some studies provided an estimation of the prevalence of NCII abuses. The results showed that about 10% of the population (in Australia and the United States) had experienced the diffusion of their private sex-related multimedia content without explicit consent [12,13]. This is a very high percentage if we consider that the spreading of private sex-related content has serious health consequences, even similar to sexual violence committed in person, such as anxiety, depression, post traumatic stress disorder (PTSD), self-harm and suicide [14], besides stigma and judgement. Therefore, the importance for the victims to promptly know about the spread of their sexual images is clear, thus being able to act upon, blocking their propagation and stopping the never-ending damage of not knowing who and how many may have seen their images.

In this scenario, Franco et al. [15] proposed *SafeSext*, a messaging system safer by design for sexting thanks to a forwarding control algorithm, which detects when someone is about to send a sexually-explicit image without being the owner. Their platform runs on a centralised architecture, thus presenting well-known drawbacks regarding scalability, single point of failure, privacy and trust as the most important centralised frameworks. Indeed, messaging applications such as FB Messenger, Telegram, Whatsapp, etc. offer end-to-end encryption messaging; yet, they use proprietary APIs for message storage/relaying, contacts discovery, keys management, group administration and other services. To overcome these issues, in recent years, the decentralisation of social services has been considered as a good alternative and new decentralised social platforms have been proposed; for instance, Mastodon [16], is a Twitter-like Decentralised Online Social Media (DOSMs) [17,18], whereas Steemit [19] embodies an instance of the new generation of DOSMs, called Blockchain Online Social Media (BOSMs) [20].

In this context, this paper aims to discuss two possible decentralised approaches for the prevention of unauthorised forwarding of *private content*, which is intended here as sex-related self-generated pictures. Considering this specific case study allows us to simplify the discussion while preserving its generality as other possible definitions could be possible to extend both the discussion and our proposed solutions. Consequently, our work contributes to promoting well-being, contrasting the non-consensual spread of private sex-related content, thus preventing its adverse consequences for the wellness of people of all ages and fostering safe, gender-equal and inclusive (online) communities. Furthermore, we advance the current state of the art, endowing DOSMs with a forwarding control algorithm enhancing their content moderation capabilities.

The rest of this paper is organised as follows. Section 2 presents a review of the related literature, while Section 3 provides a background on blockchain and NFTs technologies. An analysis of the properties of some commonly used messaging systems is presented in Section 4. We describe the possible approaches for the prevention of unauthorised forwarding of private sex-related images in Section 5, while Section 6 compares them. Some quantitative results are reported in Section 7. Finally, we draw our conclusions and present some future directions in Section 8.

2. Related work

Adolescents, but not only, live part of their sexual experiences online through mobile devices and social networks. Therefore, the literature regarding online sexual experiences and their risks, including sexting, has rapidly grown. Indeed, sexting can have serious consequences, such as the spread of private sex-related content [5], psychological disorders, bullying, self-harm, or even suicide [21,22]. On the other hand, it represents a new way for teenagers to explore and express their sexuality, which has become a key part of their lives [7].

In this scenario, some works discussed serious games as tools to tackle sex-related topics with adolescents. For instance, Guava et al. [23] proposed *UnderControl*, a serious game which aims to increase users' awareness about contraception and sexually transmitted infections (STI). Instead, Wood et al. [24] contributed to the field of HCI around sexuality by proposing a serious game to foster the discussion about sex, sexuality and affectivity, including sexting, among groups of teenagers. Yet, without denying its primary importance, education has poor effects in the short term. Thus, teenagers need safer solutions to practice sexting and explore their sexuality.

Razi et al. [7] analysed some posts on a teen peer mental health forum to have some insights into the online sexual experiences of teenagers and understand how they seek support around sexting-related issues. Following this line of research, Hartikainen et al. [25] focused on how adolescents provide advice and support to the sexting-related issues posted by peers. Alsoubai et al. [26] examined the online sexual experiences of adolescents aiming at preserving sexual well-being and preventing online sexual violence, considering the different types of relationships and the level of consent, thus shedding light on these complex topics. All these three studies suggested implications for the design of social networks and messaging platforms. For instance, embedding algorithms for automatic sexual risk detection would be helpful so that social media platforms can be co-responsible for protecting adolescents [7,26]. According to their findings, besides supporting users in case of sexting-related issues and providing advice on correct and safe online behaviours, messaging systems, as well as social network platforms in general, should easily allow to obscure faces in images when nudity is detected [25]. Furthermore, another suggestion is the integration of default privacy settings and age verification systems in social media platforms [26]. Particularly noteworthy are the recommendations of endowing social platforms with functionalities that facilitate computer-mediated consent [26], even considering the General Data Protection Regulation (GDPR), and involving teenagers and sexual health experts in the design process of such systems [25].

Acknowledging the difficulty in reporting and accessing justice systems in case of sexting abuse, Falduti et al. [11] proposed a prototype of a chatbot based on decision trees for supporting victims of NCII abuses in reporting incidents and evaluated it by involving first actors in criminal justice. Instead, Franco et al. [5] proposed some guidelines to build messaging systems safer by design for sexting and described their proposed platform, named *SafeSext*, in detail in [15]. This system can detect suspicious forwarding of images (i.e., revenge porn) thanks to a forwarding control algorithm based on a perceptual hashing function. Yet, *SafeSext* runs on a centralised architecture, thus showing all the drawbacks related to this kind of platform. In particular, scalability issues may affect the user experience, whereas the overhead and the computation time should be kept low to engage users.

Services such as *SafeSext* are crucial even for people with disabilities. Indeed, contrary to what people without disabilities often think, they have sexual expectations, fantasies and experiences as well [27]. However, people with impairments, especially

Table 1
Summary of the features of the analysed applications.

Name	Architecture	E2E Encryption	Delete message	Auto deletion timer	Screenshot alert	Fwd
Badoo	Centralised	No[a]	No	No	No	Yes
Instagram	Centralised	No	Yes	Yes[b]	Yes[e]	Yes[f]
Mastodon	Federated	No[a]	Yes	Yes	No	No
Matrix	Federated	Yes	Yes	No	No	No
Session	Partially Decentral.	Yes	Yes	Yes	No	Yes
Snapchat	Centralised	No	Yes	Yes	Yes	Yes
Telegram	Centralised	Yes[c]	Yes	Yes[c]	Yes	Yes[f]
Tinder	Centralised	No[a]	No	No	No	No
Tumblr	Centralised	No[a]	No	No	No	Yes
Whatsapp	Centralised	Yes	Yes[d]	Yes[b]	No	Yes

[a]Not available.
[b]Only for photos and videos.
[c]Only for secret chats.
[d]Only within 7 min from message generation.
[e]Only for time-limited media.
[f]Unless for time-limited media.

physical ones, may have low self-esteem and difficulties having sexual interactions in real life. Therefore, sexting can help them to face these obstacles, providing an easy way to show only those parts of themselves they want, thus being the first step toward sexual experiences [28]. This is an additional confirmation of how sexting is a crucial aspect to be considered in order to reduce inequalities and guarantee individual well-being, when supported by safer-by-design platforms.

3. Background on blockchain and NFT

Formally, a blockchain is a distributed immutable digital ledger that facilitates the process of recording transactions. Each participant within the network maintains the ledger by approving and managing new entries. Logically, it can be described as a chain of blocks which contains specific information. The block is the data structure used to store information and it contains a block header, which is the metadata that helps verify the validity of a block, and block metadata which contain various information, such as the previous block hash, which connects a block to the previous one. This enhances the security of a blockchain. A transaction generally consists of a receiver address, a sender address and a value. When it occurs, it is recorded in a block. Blockchain technology is principally known thanks to Bitcoin, which was proposed in a white paper [29]. However, Ethereum has become a popular alternative to Bitcoin, thanks to the introduction of smart contracts [30]. Indeed, several Decentralised Applications (dApps) and most *non-Fungible Token (NFT)* solutions rely on smart contract-based blockchain platforms. The verification and addition of each block to the blockchain are achieved by reaching an agreement among all the nodes in the network. This agreement is obtained using consensus algorithms encompassing a set of rules for validating a block.

An important concept related to blockchain technology is the token. Tokens are assets that are implemented as smart contracts and are stored in wallets. According to their properties and use cases, there are several types of tokens including governance, utility, security, transactional and platform tokens.[1] Instead, when we consider their features, tokens can be fungible and non-fungible. In a blockchain, fungible tokens are cryptocurrencies like Bitcoin (BTC), while NFT tokens are data units representing a unique digital asset stored and verified on the blockchain. An NFT [31] is a digital asset that uniquely represents real-world objects. Each NFT is different from another NFT of the same type. They cannot be exchanged for one another without losing value because each token is unique. Thanks to their unique properties, NFTs can be used in several scenarios, such as virtual gaming, cultural heritage, digital identity, social environment, etc.

[1] https://blog.makerdao.com/the-different-types-of-cryptocurrency-tokens-explained/

4. Analysis of existing social media

We have analysed the most popular mobile applications and social media platforms that provide some messaging functionality to understand whether they include features aimed at providing a safer user experience, in particular considering sexting. For this reason, we have considered their architecture, we investigated if the application uses end-to-end encryption or not, if it allows both the sender and the receiver to delete messages, if it is possible to set an auto deletion timer for messages, if users are notified when someone takes a screenshot or starts a screen recording of their conversation and if the content generated by a user can be forwarded by the recipient to third parties.

Users can send messages through many social media platforms, not only pure messaging applications like Whatsapp or Telegram. For this reason, we have included in our analysis even popular dating and social network apps that allow to send messages: in alphabetic order, we analysed Badoo, Instagram, Mastodon, Matrix, Session, Snapchat, Telegram, Tinder, Tumblr and WhatsApp. This chosen set can be considered a good representative of the vast plethora of social media platforms that provide some messaging functionalities since it covers the heterogeneity of different features: e.g., Tinder does not allow to send media (i.e., images, videos, etc.) at all, while all the other platforms permit to send different kinds of media.

Moreover, the analysed social media represent different underlying philosophies: while most of them have a centralised architecture, Mastodon is a federated Twitter-like social network made up of independent servers organised around specific topics, Matrix is an open network for secure, decentralised communication and Session implements a partially decentralised network of servers to route messages. Table 1 shows a summary of the analysed functionalities of the considered applications.

Matrix, Session and Whatsapp have introduced end-to-end encryption, making the contents readable only to the sender and the receiver, whilst Instagram has planned to introduce this additional warranty for privacy and security in 2023. Session protects not only the messages, but also the identities of the people thanks to a decentralised onion routing network. Telegram provides encryption only for secret chats. On the other hand, Tumblr serves all over HTTPS by default. Hence, all the content is encrypted only between client and server, without any further level of security. Many applications lack transparency on their policies, especially

for privacy and encryption, making it difficult for users to acquire information [32,33]: Mastodon, Badoo, Snapchat and Tinder do not provide this information in the official description.

Users can delete already sent messages for both sides of a conversation, providing a helpful tool in case of regret on Instagram, Mastodon, Matrix, Session and Telegram. Whatsapp permits such operation only within 7 min from the message generation. Snapchat automatically deletes all the messages already read when the user closes the chat but, if the receiver saves a message, its deletion requires an explicit action (i.e., a tap on a button) to unsave and hence delete the message. Badoo, Tinder and Tumblr do not provide any deletion features.

Snapchat management of non-saved messages can be considered a sort of auto-deletion timer. This functionality is also implemented in Mastodon and Telegram secret chats, where users can set an auto-deletion timer for any message. A similar feature is provided by WhatsApp and Instagram that allow to send media (images and videos) that can be viewed only once (WhatsApp) or twice (Instagram). Instead, Session has an auto-deletion timer on its servers, i.e., that servers store a message for two weeks, known as the message's time-to-live (TTL).

Unfortunately, deletion is not sufficient to avoid the uncontrolled spread of private content since a malicious user can save a screenshot of a picture or record a conversation; thereby, control against these two actions is really important. Snapchat is the only application that shows two different alerts for screen recording and screenshot actions so users can be aware of what is happening. Instagram notifies users only when time-limited media are involved, whilst on Telegram such functionality is available only for secret chats. Other platforms do not inform the user at all.

A very simple solution is to completely avoid the forwarding[2] of any content to anyone; this would block the easiest way for a malicious user to disseminate personal content (e.g., self-generated nudes). All the considered platforms, except Tinder, Mastodon and Matrix, allow forwarding without any form of control over the ownership of the content. Our system fills this gap, introducing a forwarding control mechanism and discussing it under different architectures. This advances the current state of the art in this topic and opens a vast spectrum of new research directions.

5. Prevention of the unauthorised forwarding of images

In this section, we overview the centralised approach presented in [15] and propose two decentralised approaches to prevent unauthorised forwarding of private content. We conclude discussing the possible integration of NFTs into the decentralised approaches to take into account potential malicious users.

According to the scenario considered in [15], we assume that users can employ our messaging system to exchange any type of content, including (but not limited to) what is intended to remain private. Furthermore, users can send media by forwarding them from another conversation, taking them with the camera on the fly, or selecting them from those in their gallery application. Although these assumptions require more complex approaches with respect to a closed application (i.e., an application that does not make the content available to the outside), we address the more challenging yet realistic case, aiming to find a reasonable balance between protecting private content and a satisfactory user experience.

We also anticipate that, from the perspective of our algorithms, there is no difference between sending and forwarding. Indeed, as mentioned above, forwarding can happen either thanks to an ad-hoc functionality (e.g., the forward button we are used to) or by selecting content from the smartphone (e.g., from the gallery application). Consequently, in the latter case, we cannot know in advance whether the sender is the owner of the content or someone has already sent such content.

[2] By forwarding, we intend either the presence of a forwarding feature or the possibility to save and send content later (e.g., images, videos, etc.).

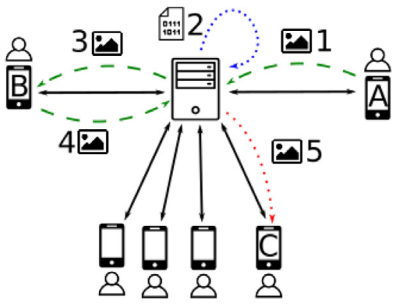


Fig. 1. The centralised approach.

Algorithm 1 Pseudocode of the centralised approach

```
1:  procedure ForwardedPicture(message)
2:      picture ← message.getPicture()
3:      if isRelevant(picture) then
4:          pictureHash ← computeHash(picture)
5:          for i ← 1, hashValuesList.length do
6:              distance ← d(hashValuesList[i], pictureHash)
7:              if distance < threshold then
8:                  owner ← hashValuesList[i].getOwner()
9:                  sender ← message.getSender()
10:                 if owner ≠ sender then
11:                     return STOP
12:                 end if
13:             end if
14:         end for
15:         saveHashValueWithOwner(pictureHash, sender)
16:     end if
17:     send(message)
18: end procedure
```

5.1. Centralised approach

In Fig. 1 we depict the centralised solution to prevent unauthorised private picture forwarding. Each user has a personal device to connect via an application to a server through which they would like to perform sexting safely. In a centralised approach, a single entity controls a server, which implements the logic of the application and controls message forwarding. The server is expected to have a complete view of the ownership of each personal image forwarded through the application, therefore it will contain large data structures and will perform the majority of the computation. Clients, on the other hand, are lightweight, maintain only minimal information and are used as interfaces to access the service offered by the server.

Algorithm 1 defines the pseudocode of the protocol executed by the server whenever a user, called *sender*, wants to forward a picture to another user, called *receiver*. The sender sends the picture to the server that, after receiving the message, checks whether the picture is *relevant*. Without loss of generality, we here define as relevant a picture with some nudity as we aim at identifying and blocking sexting abuse behaviours; however, other possible definitions may be possible in a more general version of the system aimed at blocking different types of (private) contents. If the considered picture is relevant, its hash value is

computed using a perceptual hashing function (lines 3–4). The perceptual hash value is computed such that similar images are mapped to similar values. Once the hash is computed (line 4), the algorithm searches for similar pictures in a data structure named *hashValuesList* (lines 5–6), which records the hash values of the relevant pictures seen by the server and their respective owner. To check whether two pictures are similar, the Euclidean distance between the hash values is computed, and two pictures are considered similar if their distance is below a certain threshold. In this case, the algorithm checks whether the owner of the picture is the user who is forwarding the picture. In case a mismatch is detected, the forwarding is blocked (lines 10–11), otherwise, the hash is inserted in the *hashValuesList* data structure and the picture is forwarded (lines 15–17). The fingerprinting function, the associated similarity measure and the threshold are those presented in [15].

5.2. Decentralised approaches

In the following, we show how a decentralised architecture can help in preventing private pictures (e.g., self-generated nudes) from being forwarded by users that are not their rightful owners; we consider two different approaches. We assume that each device behaves according to the defined protocol in both cases and that the system adopts end-to-end encryption.

Sender-oriented approach (Senders store the hash values of their own private pictures). Fig. 2 represents the sender-oriented approach we devised to prevent unauthorised private picture forwarding. The proposed approach, as fully decentralised, is executed by each device when a user takes a picture and when a user forwards the picture. Algorithm 2 shows the pseudocode of the two procedures included in the sender-oriented approach. Each time a user takes a picture, it begins by checking if it is relevant. If it is the case, the device computes its hash and stores it in a local and private set of hash values, called *allow list* (lines 4–5). The *allow list* will contain all the hash values of the relevant pictures generated by the user. When a private picture is forwarded by the sender *A* to the receiver *B*, the sender checks if the hash of the picture is present in the local allow list and, if the check is successful, the picture is sent (lines 12–16). If *B* decides then to forward the aforementioned relevant picture, the device has to check whether the picture originated from itself too. It does so by computing the hash of the picture and checking whether the hash of the picture is present in the local allow list. In this case, the check fails as the picture is in A's allow list and the device of the *sender* prevents the picture from being forwarded.

Receiver-oriented approach (Receivers stores the hash values of the received private pictures). Fig. 3 shows the receiver-oriented approach devised to prevent the unauthorised forwarding of private pictures. The Algorithm 3 shows the pseudocode. The protocol is executed when a user receives a picture from another user and when the user forwards a picture. In detail, each time a user receives a picture, it first checks whether it is relevant or not. In the affirmative case, its hash value is computed and stored on the device in a *known list* (see Algorithm 3, `MessageReceived` procedure), which contains the set of the hash values of the pictures received by the device. Otherwise, no action is required. Before forwarding a picture, the sender checks whether the picture is relevant or not. In the latter case, the device sends out the picture, whereas, in the former case, it computes the hash value and searches it in the *known list*. If the search fails, the sender is the owner of the photo, hence the device can forward it. Otherwise, a predefined forwarding policy has to be adopted (e.g., block the forwarding) (lines 12–19).

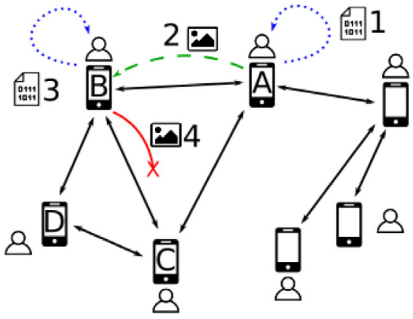


Fig. 2. The sender-oriented approach.

Algorithm 2 Pseudocode of the sender-oriented approach

```
1:  procedure TAKE PICTURE
2:      picture ← takePicture()
3:      if isRelevant(picture) then
4:          pictureHash ← computeHash(picture)
5:          allowList.add(pictureHash)
6:      end if
7:  end procedure
8:  procedure FORWARDPICTURE(pathToPicture)
9:      picture ← loadPicture(pathToPicture)
10:     if isRelevant(picture) then
11:         pictureHash ← computeHash(picture)
12:         for i ← 1, allowList.length do
13:             distance ← d(allowList[i], pictureHash)
14:             if distance < threshold then
15:                 message ← newMessage(picture)
16:                 send(message)
17:             end if
18:         end for
19:     end if
20: end procedure
```

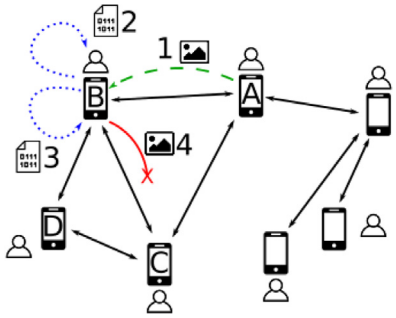


Fig. 3. The receiver-oriented approach.

5.3. Extending the decentralised approaches via NFTs

Both the proposed approaches are able to negate the unauthorised forwarding of personal pictures through a mobile application by exploiting a decentralised protocol. However, it is not reasonable to think that all users will always behave in the correct way. In particular, there are countless messaging applications available and users could exploit them to forward personal pictures. While it is not possible to block the forwarding of a picture outside of our proposed application, we can design a method to detect when this happens and then block any additional forwarding through our application. Therefore, to detect the unauthorised forwarding of private pictures, we introduce

Algorithm 3 Pseudocode of the receiver-oriented approach

```
1:  procedure MessageReceived(message)
2:      picture ← message.getPicture()
3:      if isRelevant(picture) then
4:          pictureHash ← computeHash(picture)
5:          knownList.add(pictureHash)
6:      end if
7:  end procedure
8:  procedure ForwardPicture(pathToPicture)
9:      picture ← loadPicture(pathToPicture)
10:     if isRelevant(picture) then
11:         pictureHash ← computeHash(picture)
12:         for i ← 1, knownList.length do
13:             distance ← d(knownList[i], pictureHash)
14:             if distance < threshold then
15:                 return STOP
16:             end if
17:         end for
18:         message ← newMessage(picture)
19:         send(message)
20:     end if
21: end procedure
```

Table 2
Sketch of the data contained in the NFTs.

Token number	Data
t^0	"$v_0^0, v_1^0, v_2^0 \ldots v_{49}^0$"
t^1	"$v_0^1, v_1^1, v_2^1 \ldots v_{49}^1$"
t^2	"$v_0^2, v_1^2, v_2^2 \ldots v_{49}^2$"
t^3	"$v_0^3, v_1^3, v_2^3 \ldots v_{49}^3$"
...	...

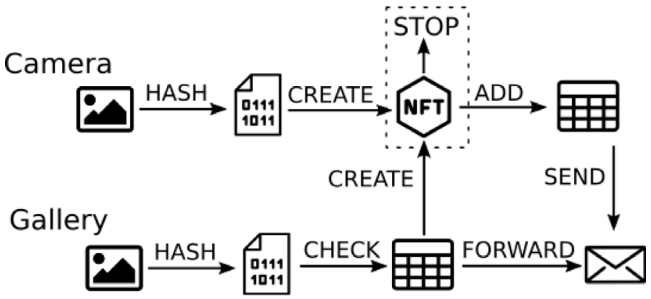


Fig. 4. Flux diagram when sending or forwarding a picture with the sender-oriented approach; steps to integrate NFTs are highlighted with a dashed box.

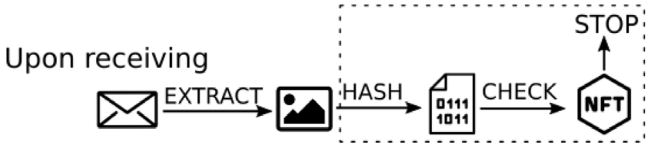


Fig. 5. Flux diagram when receiving a picture with the sender-oriented approach; steps to integrate NFTs are highlighted with a dashed box.

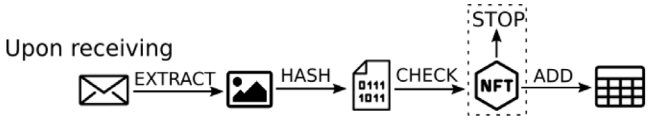


Fig. 6. Flux diagram when receiving a picture with the receiver-oriented approach; steps to integrate NFTs are highlighted with a dashed box.

the NFT technology that helps in validating the digital ownership of the pictures to our scenario, to extend the two decentralised approaches presented in Section 5.2.

We propose to associate the perceptual hash of the pictures with the respective owners through NFTs. A *Non-Transferrable NFT* (NTNFT) contract, which is an NFT where the tokens cannot be transferred, is deployed on a blockchain (e.g., Ethereum). A sketch of the data of this collection is reported in Table 2. The associated data for each token consists of an array of 50 real values, representing the perceptual hash value of a picture registered on the blockchain. Since the tokens are non-transferrable, the minter of the token always corresponds to the owner. In this way, the picture is not stored on the blockchain, having important repercussions on the amount of on-chain storage space needed, while still keeping the pictures confidential. Whenever a new NFT must be created for the hash of a picture, it is crucial for the smart contract to check all other existing NFTs to ensure that a similar picture is not already registered as an NFT. To do so, a certain threshold must be defined, so that if the distance between two hash values is below the threshold, the two pictures can be considered to contain the same subject.

Extending the sender-oriented approach. Fig. 4 shows the steps taken by the sender to forward a relevant picture. In case the picture is created via the camera, the hash of the picture is computed, and then the application tries to create a new NFT. If the NFT is created correctly, then the application can add the hash of the picture in the private *allow list* and finally send it to the receiver. We must note here that this approach requires the control of the camera, or to be notified each time a picture is taken and not all operating systems allow this feature. Moreover, not all taken pictures are relevant or are sent, so computing the hash each time the camera is used can be a waste of resources. For this reason, we consider also a generic sending or forward, i.e., when a picture from the gallery is sent, without considering how it was created or received. This permits postponing the hash computing just before sending and saving resources. Once the hash of the picture is computed, the application will check if it is present in the *allow list*. If the check is positive, the picture was already registered and can be safely forwarded. On the other hand, if the check fails, the application tries to create the NFT, following the same steps when the picture is created.

In Fig. 5 we show the steps taken by the receiver when a picture is received. The picture is extracted so its hash can be computed. Before showing the picture to the user, the application checks in the collection of NFTs if a similar image has already been added by comparing the hash value of the received picture to the hash values contained in the NFTs. If a match is found, the application checks whether the creator of the NFT is the same user who sent the message to confirm the provenance of the picture and, only in this case, display the image.

Extending the receiver-oriented approach. In Fig. 6 we show the steps performed by the receiver when a picture is received. Once the hash of the received picture is computed, the application of the receiver checks if an NFT associated with a similar hash is already present. If so, it checks whether the sender of the picture is the same who created the NFT. Before showing the picture to the user, the application checks in the collection of NFTs if a similar image has already been added by comparing the hash value of the received picture to the hash values contained in the collection of NFTs. If a match is found, the application checks if the creator of the NFT is the same user who sent the message to confirm the provenance of the picture.

In Fig. 7 we show the steps performed by the sender to forward a relevant picture. In the receiver-oriented approach,

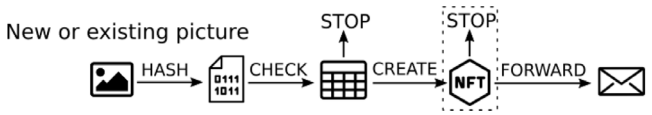


Fig. 7. Flux diagram when sending or forwarding a picture with the receiver-oriented approach; steps to integrate NFTs are highlighted with a dashed box.

when a user wants to forward a picture, the application computes its hash and, if a similar hash is found in the *known list*, the forwarding is blocked. If no similar hash is found in the *known list*, the application tries to create a new NFT. The forwarding of the picture to the receiver only happens when the token is correctly created or a token with a similar hash is found owned by the sender.

6. Discussion of the proposed approaches

In this section we compare the possible approaches for preventing unauthorised forwarding of private sex-related images.

6.1. Comparison between the two proposed approaches

If we analyse two proposed (decentralised) approaches, we can state that both are capable of detecting unauthorised forwarding, although there are some differences. In particular the sender-oriented approach require access to the camera to check whether the image is relevant and calculate its hash before saving it in the gallery, and authorisations granted by the users. However, as already discussed, it can also work without this permission. Moreover, as the sender can decide which pictures are relevant, the same system can also be used for contrasting the unauthorised forwarding of other types of private content such as health data or copyrighted items. On the other hand, the sender-oriented approach may provide a faster search operation than the receiver-oriented approach since the *allow list* is only composed of the hash values of the owner's pictures and does not depends on the number of received pictures. This is particularly useful in the case the users receive much more photos than those they send. Moreover, with the introduction of the NFTs, forwarding multiple times the same picture is particularly inexpensive because it entails only a search operation within the local *allow list*, bypassing the access to the blockchain network.

Instead, the receiver-oriented approach stores the hash values of the received images. This could introduce a small degree of redundancy of data on the whole social media because if a user forwards the same picture to multiple other users, all the receivers store the hash of the same picture. We must note that the system computes the hash value only for relevant images (i.e., sexting-related pictures), so their number should not affect the performance significantly, as the search space is limited. In general, the two approaches are dual, i.e., in the sender-oriented, the list of hash values is longer when a user sends a large number of images but receives a few of them and the opposite is in the other case.

6.2. Differences with the previous work

The main difference with the system presented in [15] is that this proposal adopts a decentralised architecture. Indeed, hash values are stored in users' devices, no more in a central server, thus bringing the well-known advantages of decentralised architectures in terms of trust and privacy. Furthermore, the search strategy iterates only over the hash values of the private photos received (or sent) by the sender, assuming that an image can be sent out only in two cases:

- the users themselves have taken the picture, so they are the owner;
- the user has previously received the image from one of their contacts.

Instead, in [15], the forwarding control algorithm iterates over all the hash values known by the system, which can be in the order of millions (or even more). In addition, to be able to represent the pictures according to their actual content, users have to send them in such a way that the server can see them unencrypted. This imposes an important limitation concerning the adoption of end-to-end encryption and limits the confidentiality of communication among the users. Conversely, both the proposed decentralised approaches implement end-to-end encryption so that only the sender and the actual receiver can read the message (and the image, if present). This is considered one of the most effective ways to ensure the security of the content and guarantee the users' privacy. An extensive discussion about the management of personal information based on distributed ledger technologies (DLTs) is presented in [34].

On the other hand, a decentralised approach alone does not always allow the detection of unauthorised forwarding (e.g., multiple users forward a private picture through external applications) and it is vulnerable in the case of modified clients. For instance, the system will be useless if a modified client does not execute forwarding control. We propose integrating NFTs in the decentralised approaches to address these situations so that unauthorised forwarding can be prevented. Thanks to their properties, NFTs can also help address other situations that may arise in our scenario. For instance, a user may circulate a personal picture that is not originated within the application. In this case, the NFT will keep track of the user who created the token, so in case a dispute arises, it is possible for the offended person to prove that someone else initially forwarded the picture without consent. On a side note, communication in a decentralised setting can always use end-to-end encryption, because there are no intermediaries, therefore each picture is only seen by the sender and the receiver.

7. Experimental assessment

We provide here a quantitative analysis of the considered approaches in terms of performance, scalability and delay.

7.1. Simulated configurations

The centralised approach and the two proposed decentralised approaches to prevent the unauthorised forwarding of private pictures have been implemented via Peersim [35], a Java library to simulate P2P protocols. The simulation has been implemented as a hybrid simulation (cycle-driven + event-driven) because we had to simulate a recurring behaviour (when a sender decides to forward a private picture) and the management of the pictures forwarded (when a receiver gets a private picture). We run the simulations considering fixed network delays (50 ms to 250 ms, chosen according to Verizon's Monthly IP Latency Data[3]) and duration (10 days) and a reliable network (messages cannot be lost).

Unfortunately, precise data regarding the number of sexting-related images sent every day over a messaging system and the occurrence of unauthorised forwarding are not available; yet it is well known that a quite large portion of the population worldwide has been involved in them at least once in their life [5,12,13]. Therefore, in our simulations we have employed different

[3] https://www.verizon.com/business/terms/latency/

values, compatible with the aforementioned partial information available and covering a wide spectrum of possible alternatives. In particular, we have considered a varying number of nodes, i.e., 2000, 4000, 6000, 8000 and 10000, that in a specific day sent an average of 5 relevant pictures each.

In the case of the simulation of decentralised approaches, all nodes represent peers and communicate directly with each other. However, during the simulation of the centralised approach, a known node (i.e., the first node, with identifier 0) represents the server, while all other nodes represent the users of the service and cannot communicate directly, but only through the server. All the nodes representing the users have a similar behaviour: once for each hour of the simulation, they decide to forward a private picture with a certain probability, which is a parameter of the simulation. Instead, in the simulations of the centralised approach, the server is idle and can only serve other users' requests.

To create the perceptual hash values, we employed a real dataset of pictures, called *Instagram images with captions*, publicly available on Kaggle.[4] By using a real dataset, we can estimate the time spent by each node to compute the perceptual hash values. All images were treated as relevant and each picture can be forwarded by at most one user, as so to obtain a fair evaluation of the protocols.

As for the proposed analyses, we adopt a modular approach. We identify three main components to be evaluated: the protocol used by each approach and the delays related to the introduction of the blockchain and NFT creation. For what concerns the protocols, we identify three types of delays:

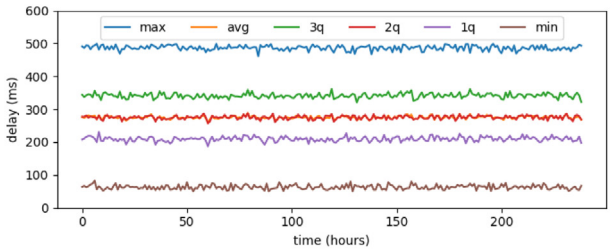
- **Network delays:** it is the time required for a message to reach its final destination (the user to which the picture is sent);
- **Data structure delays:** it is the time required to check whether a hash of a picture is already present in the collection of known hash values;
- **Hashing delays:** it is the time required to hash the pictures.

In addition, we also show a comparison of the data structure and hashing delays (their values depend on our implementation and affect the performance), as a way to understand where future optimisation could focus. On top of that, we also show the size of the private data structure used to store the hash values for the three approaches to investigate the feasibility of a decentralised scenario. Lastly, we propose an evaluation of the scalability to show what happens with each approach when the number of participating nodes increases. The plots have been created using the maximum value (i.e., the worst case) among ten runs of each simulated configuration.

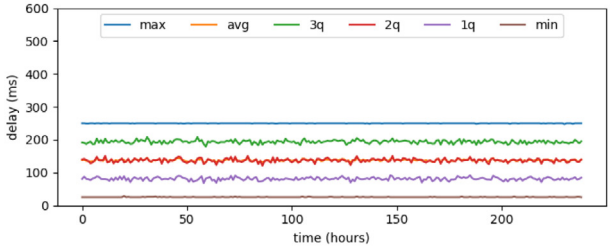
Finally, focusing on the blockchain delay, we consider the time required to execute a transaction, which can be considered to be proportional to the time required to create a new block for the blockchain. We considered a set of 4 EVM-like blockchains (i.e., Ethereum, Polygon, Avalanche, BSC) and evaluated the latest 100,000 blocks created for each blockchain so as to estimate the average delay introduced by the blockchain.

7.2. Protocol delays

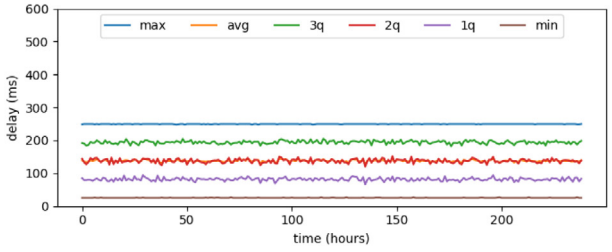
Figs. 8(a), 8(b) and 8(c) show the message network delay registered during our simulations. In all these figures, the orange line (average) overlaps with the red line (second quartile). In the case of the simulation of the centralised approach, messages carrying pictures have to travel through the server before reaching the final destination, experiencing a delay that ranges between 50 and 500 ms. On the other hand, in the two decentralised approaches, thanks to the fact that the nodes interact directly, the delays introduced by the network are halved with respect to the centralised approach, ranging from 25 to 250 ms.



(a) Network delay of the centralised approach.



(b) Network delay of the sender-oriented approach.



(c) Network delay of the receiver-oriented approach.

Fig. 8. Network delays of the simulated approaches.

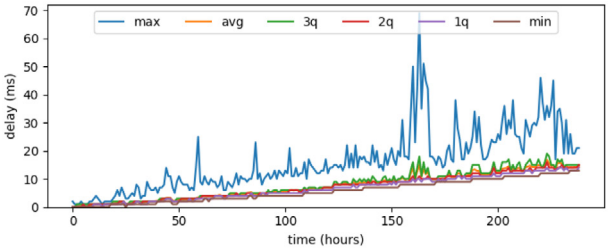
Figs. 9(a), 9(b) and 9(c) show the delay registered during our simulations for what concerns the access and management of the data structures that contain the hash values of the pictures. In the figure representing the data structure delay registered for the centralised approach, we can observe that the time to access the data structure and compute the distance between pairs of hash values increases over time. Even though the simulation has a limited duration, the time to access the hash data structure increases for each message. In a real-life scenario, the centralised architecture would cause significant delays, up to the point where most of the time is spent accessing it, making the application unusable. Regarding the simulation of the decentralised approaches, the situation is different; indeed the nodes employ a brief time (never more than 1 ms, in contrast with the tens of milliseconds of the centralised solution) to access the data structure throughout the simulation.

Figs. 10(a), 10(b) and 10(c) show the time required to hash the pictures registered during the simulation of the three approaches. In all proposed scenarios the hashing is performed in tens of milliseconds and the time required does not increase as the simulation progresses.

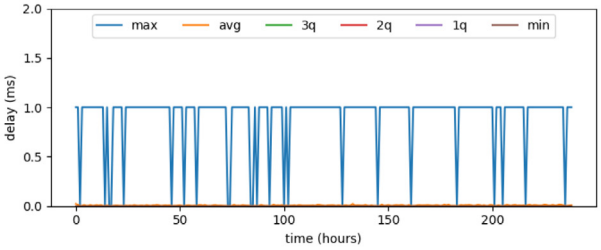
7.3. Delays comparison

Figs. 11(a), 11(b) and 11(c) compare the data structure and hashing delays registered during the simulation of the three

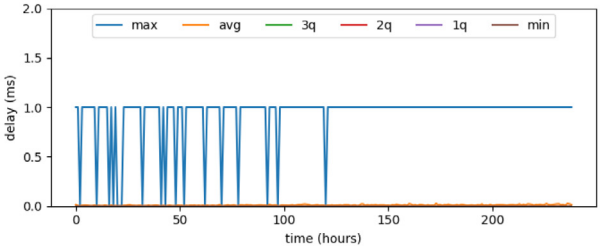
[4] https://www.kaggle.com/datasets/prithvijaunjale/instagram-images-with-captions



(a) Data structure delay of the centralised approach.

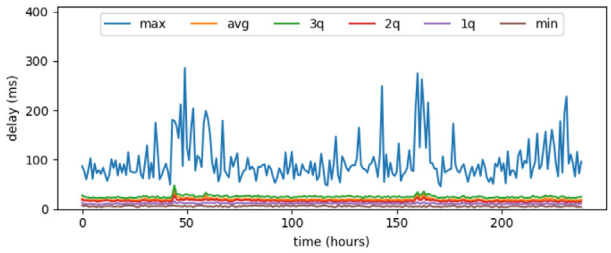


(b) Data structure delay of the sender-oriented approach.

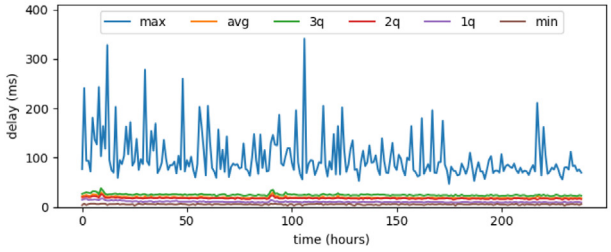


(c) Data structure delay of the receiver-oriented approach.

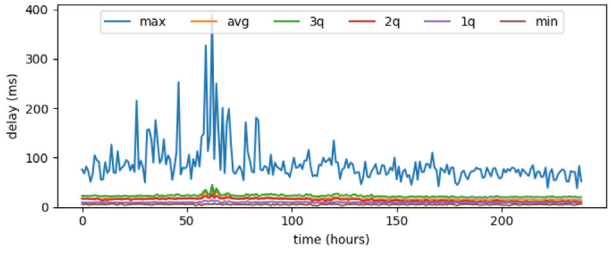
Fig. 9. Data structure delays of the simulated approaches.



(a) Hashing delay of the centralised approach.



(b) Hashing delay of the sender-oriented approach.



(c) Hashing delay of the receiver-oriented approach.

Fig. 10. Hashing delays of the simulated approaches.

approaches. In the simulation of the centralised approach, at the beginning of the simulation, most of the computation is again employed to calculate the hash of the pictures (blue bars). However, as the simulation unfolds, while the hashing time is mostly constant, the time required to access and manage the hash data structure increases (red bars), up to the point that the two delays are of comparable magnitude. It is to be expected that in a real-life scenario, the time required to access the data structure and compare the hash values of the pictures would slowly but surely surpass all the other delays, becoming the prominent source of delay. The decentralised approaches show a rather different situation. Figs. 11(b) and 11(c) show that the data structure delay of the two fully decentralised approaches (red bars) is negligible with respect to the hashing delay (blue bars) and that a few seconds are collectively spent by all nodes for the hashing operations. Indeed, it is crucial to consider that, while the computation is spread across multiple nodes in the two simulations of the decentralised approaches, all the computation in the centralised approach is concentrated in a single node, the server.

7.4. Hash data structure size

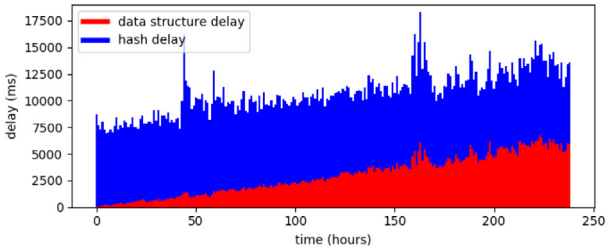
Figs. 12(a) and 12(b) show the space required to store the hash values with respect to the number of messages circulating through the simulation for the two decentralised approaches. In each figure, the black line represents the space required by the private data structure in the centralised approach, to be used as a comparison. In the two decentralised approaches, we can see that the data structures tend to increase over time and they achieve comparable memory space usage, although, on average, the memory required by the sender-oriented approach is slightly lower. Let us compare the two decentralised approaches with the centralised one. We see that the absolute amount of memory needed by the central server is much higher with respect to the peers in the decentralised approaches.

7.5. Scalability evaluation

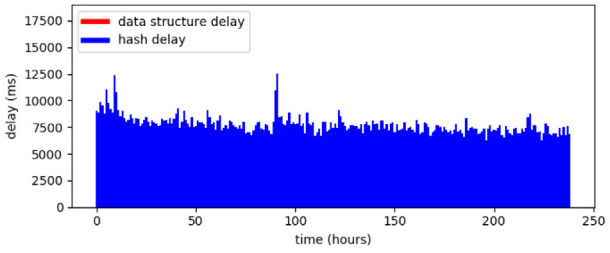
We conclude our evaluation by proposing a comparison between the two decentralised approaches and the centralised approach by increasing the number of nodes in the simulations.

In Fig. 13, we show a comparison of the average hashing delays combined with the average data structure access delay for the three simulated approaches. For all the approaches, we considered only the delays registered during the last hour of the simulation. As shown in the figure, the two decentralised approaches achieve comparable results, only requiring, on average, a few milliseconds to generate the hash of the picture and access the private hash data structure. On the other hand, for what concerns the centralised approach, we see that the server employs more time as the number of simulated nodes increases, exceeding 90 ms per picture with 10,000 nodes simulated.

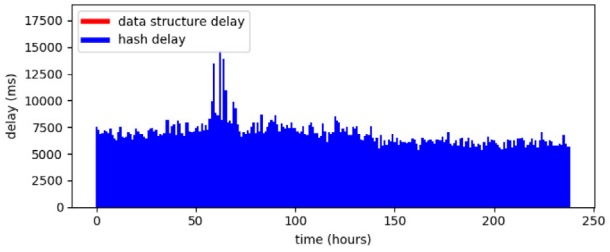
In Fig. 14, we show a comparison of the hash data structure size for the three simulated approaches. In particular, for the two decentralised ones, we show the average hash data structure size of the peers, while for the centralised approach, we show the size of the hash data structure stored by the server. For all approaches,



(a) Delay comparison for the centralised approach.



(b) Delay comparison for the sender-oriented approach.



(c) Delay comparison for the receiver-oriented approach.

Fig. 11. Delay comparison of the simulated approaches.

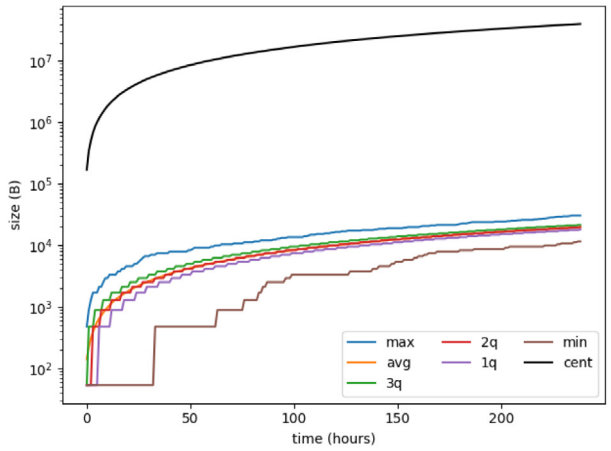
Table 3
Statistical measures concerning the block creation time (in seconds) of four EVM-like blockchains.

	blocks	min	avg	max	std dev
Ethereum	16148731 16248730	12	12.06	36	0.869
Polygon	37085419 37185418	2	2.14	65	0.697
BSC	24057821 24157820	3	3.04	15	0.373
Avalanche	23913474 24013473	0	2.04	12	0.535

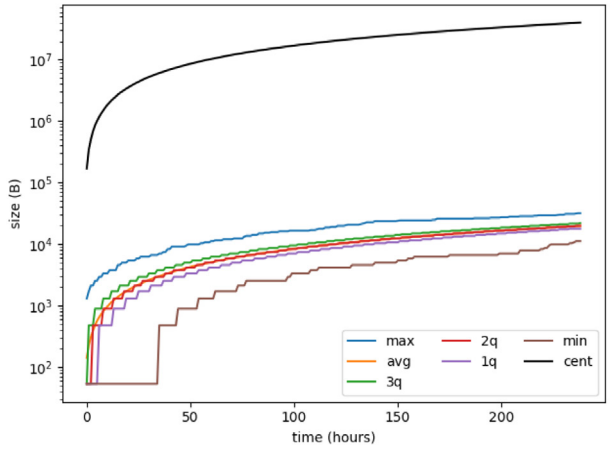
we show the situation at the end of our simulation and for an increasing number of simulated nodes. Also in this case, we see comparable results obtained by the two decentralised approaches while the size of the data structure of the centralised approach grows proportionally with the increased number of nodes simulated. The fact that the data structure increases as more users join the service is one of the major causes of the delays observed in Fig. 13; the situation becomes progressively worse the more nodes are added to the network.

7.6. Blockchain delays

To evaluate the potential impact of the blockchain, we report in Table 3 some statistical measures of the time required to mine a new block in four EVM-like blockchains. For each blockchain, we report the block range used for the measurements as well as



(a) Data structure size: centralised approach vs. sender-oriented approach.



(b) Data structure size: centralised approach vs. receiver-oriented approach.

Fig. 12. Data structure comparison of the simulated approaches.

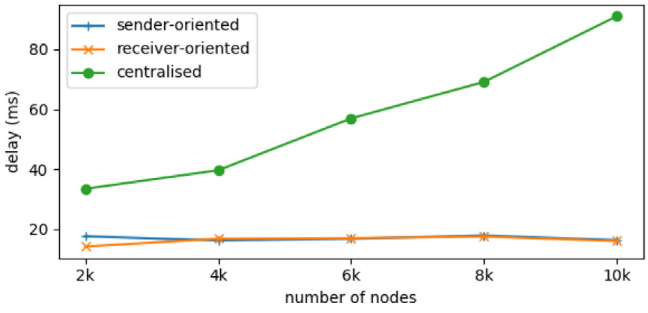


Fig. 13. Comparison of the delays registered during the last hour of the simulation for an increasing number of simulated nodes.

the minimum, maximum, average and standard deviation of the distribution of the time elapsed between two consecutive blocks. Despite the various blockchains being based on the Ethereum Virtual Machine (EVM), there are substantial differences that should be taken into account. In particular, while Ethereum is the most well-known among the considered blockchains, it is by far the slowest when considering the time required to create a new block. Binance Smart Chain (BSC) has a low block creation delay, just over 3 s and a standard deviation below four-tenths of a second. Polygon is the fastest, with only just over two seconds needed to create a new block, but only slightly faster than BSC.

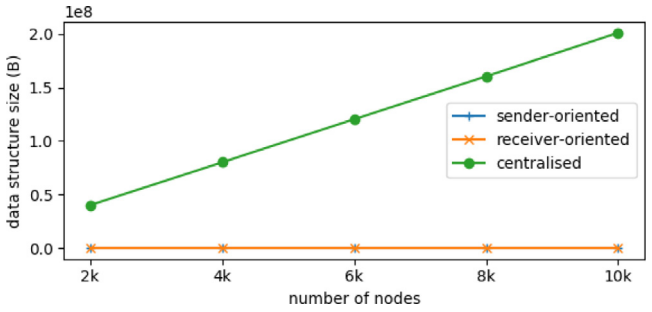


Fig. 14. Comparison of the average data structure size (the data structure size of the server in the centralised approach) registered during the last hour of the simulation, for an increasing number of simulated nodes.

In our measurements, Avalanche achieves the best (lowest) block confirmation times and the low standard deviation in the block creation time distribution hints that the network is very consistent at creating new blocks, which is a very desirable property to ensure the introduction of the blockchain does not introduce delays that make the application unusable.

8. Conclusion and future directions

Social media platforms have revolutionised our lives and communities, even becoming helpful tools in achieving SDGs if endowed with appropriate functionalities. One representative example is sexting, a widespread phenomenon among teenagers (but not only) defined as the practice of sending or receiving sex-related content through social networking sites or messaging applications, through which practitioners risk the unauthorised spread of their private content. The effects of that involve the health and well-being of people, even causing psychological issues and suicidal thoughts, as well as increasing the inequalities within our communities by predominately affecting women rather than men. Unfortunately, many messaging services allow the sending of previously received content without any limitations, thus exposing users to the aforementioned sexting-related issues.

In this paper, we have discussed two decentralised solutions that can be employed to contrast the unauthorised forwarding of sex-related self-generated images and extended them employing the disruptive blockchain and NFTs technologies to consider the presence of malicious nodes. We have also compared them to the centralised solution, showing the superiority of our approaches in terms of performance, scalability and delays. In addition, we have described how to endow a decentralised social network with a forwarding control mechanism, thus enhancing its content moderation capabilities. Furthermore, through our solutions, we contribute to preserving the well-being of people of all ages and fostering inclusive, gender-equal, and safe communities, protecting people from the adverse consequences of sexting while allowing them to safely express and explore their sexuality online.

We plan to extend our research in several directions. For instance, evaluating our solutions considering the energy constraints of mobile devices is crucial since their wide adoption also depends on low energy consumption. Besides the implications for user experience, this is also in line with some SDGs: Responsible Consumption and Production (12) and Climate Action (13). The management of screenshots and screen recording attempts needs further investigation, even considering the differences between operating systems, because of their potential impact on our solutions. Furthermore, involving adolescents, by adopting a teen-centric approach, and sexual health experts in the design and development of our solutions would be particularly useful, helping us to create a more effective system. Finally, another possible evolution of this system regards considering alternative definitions for "relevant" contents. Indeed, as discussed in Section 6.1, rather than focusing only on self-generated nude pictures, the general framework of our system can be adapted to protect communication regarding health data, patents, etc. To do so, it is sufficient to allow users to flag whatever content they want to protect (e.g., messages, pictures, etc.) and apply the rest of the discussed solution. In particular, the sender-oriented approach seems particularly suited for this generalisation.

CRediT authorship contribution statement

Mirko Franco: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Ombretta Gaggi:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Barbara Guidi:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing. **Andrea Michienzi:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Claudio E. Palazzi:** Conceptualization, Methodology, Investigation, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The employed dataset of images is available online. Instead, the simulation data are generated randomly.

References

[1] United Nations, Transforming our world: the 2030 agenda for sustainable development, 2015, https://wedocs.unep.org/20.500.11822/9814, (Accessed 11 December 2022).
[2] S. Counts, K.E. Fisher, Mobile social networking as information ground: A case study, Libr. Inf. Sci. Res. 32 (2) (2010) 98–115, http://dx.doi.org/10.1016/j.lisr.2009.10.003.
[3] E.A. Vogels, L. Rainie, H. Nolan, Teens, social media & technology 2022, 2022, https://www.pewresearch.org/internet/2022/08/10/teens-social-media-and-technology-2022/, (Accessed 11 December 2022).
[4] M. Franco, S.A. Falyoun, K.E. Fisher, O. Gaggi, Y. Ghamri-Doudane, A.J. Nashwan, C.E. Palazzi, M. Shwamra, A technology exploration towards trustable and safe use of social media for vulnerable women based on islam and arab culture, in: Proceedings of the 2022 ACM Conference on Information Technology for Social Good, GoodIT '22, 2022, pp. 138–145, http://dx.doi.org/10.1145/3524458.3547259.
[5] M. Franco, O. Gaggi, C.E. Palazzi, Improving sexting safety through media forwarding control, in: 2022 IEEE 19th Annual Consumer Communications & Networking Conference, CCNC, 2022, pp. 1–6, http://dx.doi.org/10.1109/CCNC49033.2022.9700555.
[6] Y. Barrense-Dias, A. Berchtold, J.-C. Surís, C. Akre, Sexting and the definition issue, J. Adolescent Health 61 (2017) http://dx.doi.org/10.1016/j.jadohealth.2017.05.009.
[7] A. Razi, K. Badillo-Urquiola, P.J. Wisniewski, Let's talk about sext: How adolescents seek support and advice about their online sexual experiences, in: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, 2020, pp. 1–13, http://dx.doi.org/10.1145/3313831.3376400.
[8] M. Franks, Drafting an effective 'revenge porn' law: A guide for legislators, SSRN Electron. J. (2014) http://dx.doi.org/10.2139/ssrn.2468823.
[9] M. Wood, C. Barter, N. Stanley, N. Aghtaie, C. Larkins, Images across europe: The sending and receiving of sexual images and associations with interpersonal violence in young people's relationships, Child. Youth Serv. Rev. 59 (2015) 149–160, http://dx.doi.org/10.1016/j.childyouth.2015.11.005.

[10] Plan International, Free to be online?, 2022, https://plan-international.org/publications/free-to-be-online/, (Accessed 13 December 2022).
[11] M. Falduti, S. Tessaris, On the use of chatbots to report non-consensual intimate images abuses: The legal expert perspective, in: Proceedings of the 2022 ACM Conference on Information Technology for Social Good, GoodIT '22, 2022, pp. 96–102, http://dx.doi.org/10.1145/3524458.3547247.
[12] N. Henry, A. Powell, A. Flynn, Not Just 'Revenge Pornography': Australians' Experiences of Image-Based Abuse. A Summary Report, RMIT University, 2017.
[13] Y. Ruvalcaba, A.A. Eaton, Nonconsensual pornography among U.S. adults: A sexual scripts framework on victimization, perpetration, and health correlates for women and men, Psychol. Violence 10 (1) (2019) 68–78.
[14] S. Bates, Revenge porn and mental health: Qualitative analysis of the mental health effects of revenge porn on female survivors, Feminist Criminol. (2016) 1–21.
[15] M. Franco, O. Gaggi, C.E. Palazzi, Can messaging applications prevent sexting abuse? A technology analysis, IEEE Trans. Mob. Comput. (2023) 1–14, http://dx.doi.org/10.1109/TMC.2023.3238189.
[16] M. Zignani, S. Gaito, G.P. Rossi, Follow the "Mastodon": Structure and evolution of a decentralized online social network, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 12, (1) 2018, pp. 541–550.
[17] A. Datta, S. Buchegger, L.-H. Vu, T. Strufe, K. Rzadca, Decentralized online social networks, in: B. Furht (Ed.), Handbook of Social Network Technologies and Applications, 2010, pp. 349–378, http://dx.doi.org/10.1007/978-1-4419-7142-5_17.
[18] B. Guidi, M. Conti, A. Passarella, L. Ricci, Managing social contents in decentralized online social networks: A survey, Online Soc. Netw. Media 7 (2018) 12–29.
[19] B. Guidi, A. Michienzi, L. Ricci, A graph-based socioeconomic analysis of steemit, IEEE Trans. Comput. Soc. Syst. 8 (2) (2021) 365–376, http://dx.doi.org/10.1109/TCSS.2020.3042745.
[20] B. Guidi, When blockchain meets online social networks, Pervasive Mob. Comput. 62 (2020) 101131, http://dx.doi.org/10.1016/j.pmcj.2020.101131.
[21] N. Döring, Consensual sexting among adolescents: Risk prevention through abstinence education or safer sexting? Cyberpsychol.: J. Psychosoc. Res. Cyberspace 8 (1) (2014) 9, http://dx.doi.org/10.5817/CP2014-1-9, URL https://cyberpsychology.eu/article/view/4303.
[22] P. Korenis, S. Billick, Forensic implications: Adolescent sexting and cyberbullying, Psychiatr. Q. 85 (2013) http://dx.doi.org/10.1007/s11126-013-9277-z.
[23] V. Guana, T. Xiang, H. Zhang, E. Schepens, E. Stroulia, UnderControl an educational serious-game for reproductive health, in: Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play, in: CHI PLAY '14, 2014, pp. 339–342, http://dx.doi.org/10.1145/2658537.2662983.
[24] M. Wood, G. Wood, M. Balaam, Sex talk: Designing for sexual health with adolescents, in: Proceedings of the 2017 Conference on Interaction Design and Children, IDC '17, 2017, pp. 137–147, http://dx.doi.org/10.1145/3078072.3079747.
[25] H. Hartikainen, A. Razi, P. Wisniewski, Safe sexting: The advice and support adolescents receive from peers regarding online sexual risks, Proc. ACM Hum.-Comput. Interact. 5 (CSCW1) (2021) http://dx.doi.org/10.1145/3449116.
[26] A. Alsoubai, J. Song, A. Razi, N. Naher, M. De Choudhury, P.J. Wisniewski, From 'friends with benefits' to 'sextortion:' a nuanced investigation of adolescents' online sexual risk experiences, Proc. ACM Hum.-Comput. Interact. 6 (CSCW2) (2022) http://dx.doi.org/10.1145/3555136.
[27] S. Wachs, M.F. Wright, M. Gámez-Guadix, N. Döring, How are consensual, non-consensual, and pressured sexting linked to depression and self-harm? The moderating effects of demographic variables, Int. J. Environ. Res. Public Health 18 (5) (2021) http://dx.doi.org/10.3390/ijerph18052597.
[28] How sexting helped me embrace my disabled body, 2015, https://femsplain.com/how-sexting-helped-me-embrace-my-disabled-body-bc33833f7f88, Online (Accessed 04-11-2021).
[29] S. Nakamoto, A. Bitcoin, A peer-to-peer electronic cash system, Bitcoin 4 (2008) https://bitcoin.org/bitcoin.pdf.
[30] C. Dannen, Introducing Ethereum and Solidity, Vol. 1, 2017.
[31] Q. Wang, R. Li, Q. Wang, S. Chen, Non-fungible token (NFT): Overview, evaluation, opportunities and challenges, 2021, arXiv preprint arXiv:2105.07447.
[32] M. Furini, S. Mirri, M. Montangero, C. Prandi, Privacy perception and user behavior in the mobile ecosystem, in: Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good, GoodTechs '19, 2019, pp. 177–182, http://dx.doi.org/10.1145/3342428.3342690.
[33] M. Furini, S. Mirri, M. Montangero, C. Prandi, Privacy perception when using smartphone applications, Mob. Netw. Appl. 25 (3) (2020).
[34] M. Zichichi, S. Ferretti, G. D'Angelo, V. Rodríguez-Doncel, Data governance through a multi-DLT architecture in view of the GDPR, Cluster Comput. 25 (2022) http://dx.doi.org/10.1007/s10586-022-03691-3.
[35] A. Montresor, M. Jelasity, PeerSim: A scalable P2P simulator, in: 2009 IEEE Ninth International Conference on Peer-to-Peer Computing, 2009, pp. 99–100.

Mirko Franco is a Ph.D. Student in Brain, Mind and Computer Science at the Department of Mathematics of the University of Padua, under the supervision of Prof. Claudio E. Palazzi. He previously completed his B.Sc. degree and M.Sc. degree in Computer Science at the same university, respectively, in 2019 and 2021. His current research activity mainly focuses on mobile systems and social platforms and is involved in the organisation of conferences such as IEEE NES and ACM GoodIT.

Ombretta Gaggi is an Associate Professor in Computer Science at the Department of Mathematics of the University of Padua. She received her M.S. degree in Computer Science from University of Venice in 1998 and her Ph.D. degree in Computer Science from University of Bologna in 2003. Her research interests include accessibility and web technologies, cross-platform mobile development and user interface design. She is member of several technical program committees of international conferences and of the steering committee of ACM GoodIT and associate editor of Multimedia Tools and Applications Journal.

Barbara Guidi is an Assistant Professor at the Department of Computer Science of the University of Pisa. She received her B.Sc. and M.Sc. in Computer Science from the University of Pisa, Italy, in 2007 and 2011, respectively. She received her Ph.D. degree in Computer Science from the University of Pisa, in 2015. She was a Co-Chair for the conference EAI GoodTechs 2017, and Co-Chair of several workshops. She has been involved in the TPC of several International conferences, and workshops. She is part of the Editor Board for relevant journals, such as IEEE Access and PlosOne. Her current research interests include Decentralized Online Social Networks, Blockchain technology, NFT, Metaverse, and Social Network Analysis.

Andrea Michienzi received his Ph.D. in Computer Science at the Università di Pisa in July 2021, and currently holds an assistant professor position at the same university. His main research interests include the dynamics of socio-economic networks, enabling technologies for the metaverse, and the application of the blockchain to social networking platforms. He is one of the organisers of the Open Challenges in Online Social Networks workshop.

Claudio E. Palazzi is an Associate Professor in Computer Science at the Department of Mathematics of the University of Padua. He received his M.S. degree in Computer Science from UCLA in 2005, his Ph.D. degree in Computer Science from UniBO in 2006, and his Ph.D. degree in Computer Science from UCLA in 2007. His research interests are primarily focused on the design and analysis of Internet architectures and mobile systems.He is member of the steering committee of conferences such as IEEE CCNC, ACM GoodIT and IFIP/IEEE Wireless Days and associate editor of IEEE Transactions on Multimedia and Elsevier Computer Networks.