Expert Systems With Applications 208 (2022) 118079

Contents lists available at ScienceDirect

Expert Systems With Applications

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

HyperNet: A conditional k-anonymous and censorship resistant decentralized hypermedia architecture

Carlos Núñez-Gómez [a], Victor Garcia-Font [b,c,d,*]

[a] High-Performance Networks and Architectures Group (RAAP), Albacete Research Institute of Informatics (I3A), University of Castilla-La Mancha, 02071 Albacete, Spain
[b] IT, Multimedia and Telecommunications Department, Universitat Oberta de Catalunya (UOC), Rambla del Poblenou 156, 08018 Barcelona, Spain
[c] Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili (URV), Av. Països Catalans 26, 43007 Tarragona, Spain
[d] CYBERCAT - Center for Cybersecurity Research of Catalonia, Spain

ARTICLE INFO

Keywords:
Revocable privacy
Distributed systems
Hypermedia distribution
K-anonymity
Ring signatures
Blockchain

ABSTRACT

Nowadays, the vast majority of Internet services used to distribute hypermedia content follow a centralized model, which is highly dependent on servers and raises several quality and security concerns. Among other issues, this centralized model creates single points of failure, requires trust on providers to avoid censorship and personal data misuse, and results in a scenario where digital content tends to disappear or be inaccessible over time, for example, when a content creator stops maintaining a site or when the content is moved to another location. To improve this, it is necessary to replicate data and follow more distributed models. Nevertheless, current platforms to distribute content in this way, either do not offer an effective mechanism to maintain the privacy of their users or they offer full-anonymity, which contributes to the dissemination of content that goes beyond the law and moral standards of many users.

This paper proposes a novel distributed architecture that enables hypermedia resource distribution ensuring censorship resistance and conditional k-anonymity. In the proposed system, users form groups to share hypermedia content where the anonymity of the publisher is preserved only if the publication follows a set of rules defined by the group. To this end, the proposed system uses threshold discernible ring signatures to enable conditional k-anonymity, the Ethereum blockchain platform to manage groups and user identities, and the InterPlanetary File System to store and share hypermedia resources in a distributed way. This document provides the design for the proposed architecture and protocols, it evaluates system risks and its security properties, and it discusses the proposal in general terms.

1. Introduction

Privacy and accountability are two sides of the same coin. These two properties are normally desirable in most of the systems on the Internet. Nevertheless, in general, systems that enhance one tend to neglect the other one. Furthermore, depending on the architecture of each system, these properties are managed in a very different way. Focussing on systems to distribute hypermedia content (e.g. web pages, images, audio, video) on the Internet, it is easy to distinguish two types of architectures: centralized and distributed systems.

Centralized architectures, like most of the client–server systems taking part in the World Wide Web, have been designed to delegate to service providers (i.e. web hosting services) the responsibility to administer both privacy and accountability. In this case, in a scenario where users publish certain content that goes against the service agreements or that breaks the law, the service provider can easily block the published content and reveal the identity of the users to the authorities to hold them accountable. However, in a globalized and borderless context like the Internet, it is frequent that users use services from different countries and, therefore, sometimes it becomes difficult to bring users or services to court in case of infringement. Regarding privacy, in this type of architecture, users have to trust that the providers will securely store their data. In addition, regarding availability, since servers are generally single points of failure, users have to trust that providers will manage the services proficiently. This can foment skepticism, especially in regions where the law is not stern with the service providers or in places with authoritarian governments that can utilize their power to control and censor the published content.

* Corresponding author at: IT, Multimedia and Telecommunications Department, Universitat Oberta de Catalunya (UOC), Rambla del Poblenou 156, 08018 Barcelona, Spain.
E-mail addresses: carlos.nunez@uclm.es (C. Núñez-Gómez), vgarciafo@uoc.edu (V. Garcia-Font).

https://doi.org/10.1016/j.eswa.2022.118079
Received 29 September 2020; Received in revised form 24 March 2022; Accepted 3 July 2022
Available online 8 July 2022

Conversely, distributed systems like BitTorrent (Cohen, 2001) use peer-to-peer (P2P) technology to enable censorship-resistant platforms. Furthermore, other communication systems such as Tor (Dingledine, Mathewson, & Syverson, 2004), Freenet (Clarke, Sandberg, Wiley, & Hong, 2001), or I2P (I2P, 2003) are designed to preserve privacy implicitly, using anonymity techniques to conceal users in a multitude. In this way, the users are voluntarily or involuntarily covering the illegal activities of other users. With these applications, people cannot only connect to websites and other Internet services without being tracked but they can also indirectly establish a connection with services that might be banned in their countries and that are being blocked by their Internet Service Provider (ISP). The anonymity properties offered by these technologies enable journalists, whistleblowers, and, in general, the population of countries with authoritarian governments to communicate freely and to access forbidden Internet sites. This has also contributed to the influx of user consumption of the dark web, a segment of cyberspace that is only accessible through systems such as Tor, where it has been estimated that more than 50% of the activity through this is for illegal ends (Moore & Rid, 2016).

Hence, some distributed architectures offer stronger mechanisms to ensure privacy. Furthermore, these also eliminate single points of failure and, therefore, the availability of published content increases and creates an architecture that is more resilient to computer failures and more resistant to governmental censorship. Nonetheless, the fact that there are no controls to avoid the distribution of certain content, makes these systems inadequate for many people. In this regard, this paper proposes a novel system that tackles the following open issues: (1) users of communication systems providing full-anonymity contribute to providing anonymity for users that use the system to publish content that can be considered inadequate. Users should be able to decide only to cover users publishing content aligned to their moral standards. (2) Once some content is published, it is difficult to prevent its dissemination even if most of the users are against its distribution. (3) Publishers of illegal content remain hidden among other users of the system and cannot be exposed and brought to justice.

Our work in this paper tackles these open issues proposing a novel decentralized hypermedia content distribution architecture that enables users to distribute content anonymously, concealing their identity within a group, providing they follow a protocol defined by the group with a set of rules regarding the type of content that can be disseminated. The infringement of the protocol will carry certain actions against misbehaving users, such as excluding users from the group, revealing their identity and/or law enforcement. The proposed architecture is based on some existing technologies such as threshold cryptography (cryptographic techniques that require the cooperation of several parties to encrypt, decrypt or sign a message), the Ethereum blockchain and its smart contracts (Buterin, 2014), and the InterPlanetary File System or IPFS (Benet, 2014). The former will be used to protect users' privacy unless a certain of group members decide otherwise due to protocol infringements. The Ethereum platform will be the basic mechanism to record users' identities in a decentralized and confidential accord. Finally, the IPFS will be used to distribute content among users. Note that these technologies are in different stages regarding their deployment. Ethereum and IPFS are widely used mature projects to enable many different types of decentralized applications. In contrast, as far as we know, the threshold cryptography algorithms used in this proposal are still a specification and, currently, there is no reliable implementation that has passed a rigorous validation process. In this paper, we present a novel architecture based on these key components and we define the required protocols that enable a hypermedia distribution system with revocable k-anonymity. Thus, in this context, k-anonymity (Samarati & Sweeney, 1998) can be defined as the property of a hypermedia publication whose publisher cannot be distinguished from other k−1 identities. We identify this property as revocable k-anonymity (as opposed to full-anonymity) because a mechanism is provided to identify the publisher under certain circumstances.

The remainder of the paper is organized as follows: Section 2 presents the state of the art on platforms in this context. Section 3 describes the building blocks and the technologies used in this proposal. Section 4 provides a system description, providing details on the participating entities, the architecture, and the main protocols. Section 5 evaluates the security properties and risks of the protocols and discusses the proposed system. Finally, Section 6 concludes the paper.

2. State of the art

Presently, there are several alternatives to communicate using platforms that preserve the anonymity and the privacy of the users. Moreover, P2P networks and other decentralized technologies, such as blockchain, are also useful to safeguard the availability and integrity of exchanged data. This section includes a state-of-the-art review about projects treating the security properties disclosed above.

Today, the most popular anonymous and private platform is undoubtedly Tor (Dingledine et al., 2004), which is based on applying Onion routing (Goldschlag, Reed, & Syverson, 1999) to the communications to hide the content of the messages and the identity of the source computer. Essentially, with Onion routing, a sender applies several overlays of encryption to a message and dispatches it to the destination through several hops. When receiving a message, each hop removes one of the encryption layers and re-sends the message to the next hop. Once the last hop removes the last layer of encryption, then the message can be sent to the destination. Furthermore, encrypted messages from several users are received and mixed by each hop, which prevents external parties from monitoring the hops and tracking users from hop to hop. In such manner, people can avoid being tracked by the sites and they may use Tor to access websites that could potentially be blocked by the ISPs of their country of residency. Moreover, this technology can also be used to provide anonymity to the content publishers using what is called Onion services, which are Internet services that can only be accessed through Tor. The Tor network is constituted by hundreds of volunteers that offer their computers to act as a relay and enable Onion routing as explained. However, some of the nodes in the network take special roles, which are not so replicated. For instance, the directory authorities are a set of only a few nodes that periodically publish a list of active relays.

Many other projects adopt P2P architectures to avoid typical problems of centralization, like denial of service (DoS) attacks. For example, Freenet (Clarke et al., 2001) is a P2P censorship-resistant distributed system that has strong mechanisms to protect the anonymity of users publishing and accessing content only within the network. Unlike Tor, this system is not meant to be used as a proxy to access content from the conventional Internet. Moreover, Freenet is designed to allow users to create private networks only accessible by a trusted group of users.

Invisible Internet Project or I2P (I2P, 2003) is another P2P network designed to establish anonymous and censorship-resistant communication between two parties. The nodes that constitute the network act as routers that are applied to establish multiple hops in unidirectional temporal channels. Similar to Tor, to allow the source and the destination to communicate privately through the multiple intermediaries, the two endpoints establish a tunnel with cryptographic layers that are removed by each of the nodes in the channel. This routing system creates a mix network that breaks the link between the source and the destination of the communication. Finally, it is worth noting that I2P also allows users to activate certain plugins to offer conventional Internet services within the I2P network, for example, a web server.

Another prominent initiative to offer ways to distribute content in a free and censorship-resistant manner is Zeronet (Kocsis, 2015). This is a P2P platform using the BitTorrent (Cohen, 2001) network to share static hypermedia content, like websites, and Namecoin (Durham, 2011) as a decentralized domain registry. In this platform, each of the

websites is published together with a JSON file with metadata about all the files of the site, and also a cryptographic signature by the owner. In Zeronet, due to the P2P architecture, users are, at the same time, site visitors and site providers, which creates a highly available network without single points of failure. Moreover, Zeronet enables using Tor to provide anonymity.

Freeweb (Shen, Liu, Liu, & Zhao, 2015) is another P2P system to avoid censorship, which permits worldwide users to surf any website. This system enables intermediary nodes from non-censored areas to act as proxies for users from the other areas. When a user enters the URL of a website in the Freeweb navigator, the system uses the P2P network to find a node that can serve the URL using an encrypted tunnel.

In general, all aforementioned systems are based on establishing secure tunnels through multiple nodes that blend the messages from different users to avoid a third party blocking all possible paths and also prevent any party analyzing the communication to establish a direct connection between the source, the destination and the sent or published content. These systems use the crowd to forward packets and hide the real identity of the users (or the pseudo-identities like the IP address). Increasing anonymity and privacy features has resulted in the publication of a plethora of content that fails to abide the law or, at least, going beyond the moral limits of many users, which, at the same time, discourages many other users from supporting this type of system. Thus, some systems have proposed mechanisms to mitigate the impact of unwanted content. For example, to avoid hosting inadequate content, Zeronet uses a blacklist mechanism with which each user can block the content that, in his/her opinion, is not appropriate.

Recently, blockchain technology has proven to be an enabler of decentralized, secure and traceable systems. However, anonymity and privacy are also desired properties on blockchain platforms like Ethereum or distributed hypermedia transfer protocols like IPFS. There are several recent research papers specifically focused on these topics. For example, in Wang, Zhao, and Wang (2020), the authors explore and compare existing issues regarding blockchain privacy. Specifically, the paper focuses on the protection of user identities and transactions in blockchain platforms. The authors review a modicum of privacy protection mechanisms, including coin mixing, ring signatures, zero knowledge proofs and homomorphic encryption.

In Pandey and Kulkarni (2017), the authors expose some weaknesses in the public key infrastructure (PKI) model and focus on studying identity-based ring signatures. To this end, the authors implement and compare identity-based ring signatures using two different algorithms, hash-based message authentication codes (HMAC), which is a symmetric-key mechanism to verify data integrity and authenticity of a message, and the Advanced Encryption Standard (AES), which is a symmetric-key specification for the encryption of electronic data. The comparison focuses on the execution time of the signature generation and verification processes depending on the number of participants in the ring signatures. The paper also presents two potential scenarios for this type of signature: access to data in cloud computing environments and whistleblowing.

Healthcare environments are another potential scenario in which privacy is critical. Although blockchain technology is considered to be highly secure, additional mechanisms are needed to ensure the privacy of its users, not only for the patients, but also for the medical personnel, healthcare facilities, insurance companies, etc. Su, Zhang, Xue, and Li (2020) proposes an attribute-based signature scheme with attribute revocation to protect user identities in blockchain-based healthcare systems that manage and store electronic medical records (EMR). In this proposal, users are assigned different attribute keys (i.e. master-key and update-key) generated by attribute authorities such as hospitals. These attribute keys allow to protect the real identity of a user whilst facilitating the verification of his/her identity. In the proposed signature scheme, users combine the attribute keys to generate the attribute signing key and sign an EMR before storing it on the blockchain. Another research work in this area is Kumar, Marchang, and Tripathi (2020), in which the authors propose an off-chain storage system for medical records using the IPFS protocol and a consortium blockchain. The proposal ensures data integrity and patients privacy by only storing hashed data on the blockchain, while giving access to off-chain medical records to authorized entities.

AttriChain (Shao et al., 2020) is a blockchain-based framework that allows users to send transactions anonymously. However, the system offers traceability if a transaction is considered problematic by a number of participants greater than a threshold. In order to achieve this, the authors propose an attribute-based signature scheme that employs a threshold/distributed tag-based encryption mechanism. In this work, users are managed and certified by AttriChain attribute authorities. The authors base their proposal on a non-open permissioned blockchain and present a prototype built entirely on a private instance of the Ethereum platform. The permissioned blockchain sets up a single group of entities for whom there are different roles, for example, the chain owner (who acts as network admin), attribute authorities or special nodes that take the tracing responsibility.

As observed so far, anonymity and privacy are key security properties to consider when designing computer systems. While some recent blockchain research focuses on achieving full anonymity and privacy for users, contrary works propose conditional anonymity schemes that allow to revoke users' anonymity under certain circumstances. The conditional anonymity property enables scenarios in which users misbehavior should be penalized. A notable example of this is MOOCsChain (Li et al., 2022). In this paper, the authors present a blockchain and IPFS-based storage and sharing scheme of Electronic Learning Records (ELR) in Massive Open Online Courses (MOOCs) environments. To prevent users' real identities from being leaked or tracked by attackers, the authors propose a centralized Registration Authority (RA) responsible for linking users' real identities with anonymous identities in a secure manner. Consequently, only the RA entity is enabled to trace and de-anonymize users in case of misconduct.

In Lin, He, Huang, Khan, and Choo (2020), a Decentralized Conditional Anonymous Payment (DCAP) scheme based on blockchain and smart contracts is described. DCAP guarantees transaction privacy whilst allowing to trace users' real identities, enabling the regulation process necessary to minimize abuse or criminal exploitation. To achieve this, the authors designed a Condition Anonymous Payment (CAP) scheme based on signature of knowledge. This scheme allows signing cryptocurrency transactions using users' anonymous identities generated from their public–private key pairs and other system public parameters. In the event of a suspicious transaction, only one type of trusted authority (Manager) is enabled to revert the sender's anonymous identity and obtain his/her long-term public address. Malicious users de-anonymized by Managers will be registered in a smart contract acting as a blacklist, preventing these users from sending transactions in the future.

Another recent blockchain paper researching the conditional anonymity concept is Zhang and Ye (2022). In this work, a novel privacy protection method to ensure users' privacy on permissioned blockchains is proposed. The protection method is based on Conditionally Anonymous Ring Signatures (CARS). The authors designed a permissioned blockchain network composed of multiple blockchain nodes playing different roles (execution, consensus and validation nodes), and a centralized Certificate Authority (CA) responsible for issuing identity certificates and managing users' public keys. Therefore, to sign a transaction it is imperative for the user to request to the CA the public parameters of the other users participating in the ring. Although users' identities remain anonymous to other users, to prevent invalid or illegal transactions, authorized validation nodes can retrieve from a signature the user who signed the transaction.

Furthermore, beyond these academic proposals, there are also implementations of decentralized online social media, which handle decentralization, censorship-resistance and privacy from different angles.

Steemit (Steemit, 2021) was one of the first online social media platforms to use blockchain and cryptocurrencies to decentralize the service and reward users for publishing and curating content. Diaspora* (Diaspora Foundation, 2021) is a similar platform with no identity control. Therefore, users can remain completely anonymous when publishing content, which can be public or remain private to a selected group of users. Sapien network (Sapien Network, Inc., 2021) is a blockchain-based system that aims to minimize the spread of fake news (Guidi, 2020). The blockchain is used to identify users, which can connect to this network showing their real identities or anonymously. To evaluate the reliability of the publications and the sources, a reputation score is computed for each identity. SocialX (SocialX Pte. Ltd, 2019) also seeks to create a decentralized system that minimizes misinformation. Although this is a censorship-resistant system, users can flag content as inappropriate. This is a community driven mechanism to blacklist certain types of unwanted resources. Peepeth (Peepeth, 2021) claims to be a social network using blockchain technology to provide immutability to the published messages. However, to avoid distribution of inappropriate content, Peepeth relies on moderating the messages that are published in the main feed of its non censorship-resistant frontend. In this platform, users can identify themselves by linking their profiles with a profile from a centralized network, such as Twitter or GitHub. Fediverse (Fediverse, 2021) uses open standards to enable a federation of content publishing systems. One of the most popular of these systems is Mastodon (Mastodon GmbH, 2021), which is a tool to deploy nodes offering microblogging services. Although this creates a federation of decentralized systems, they cannot be considered as censorship-resistant, since each node relies on its administrators to enforce custom content moderation policies.

The projects presented above generally have as one of their main goals to guarantee the full anonymity and privacy of users' personal data. Some of the aforementioned systems leave it up to users how much personal data they make public. Other systems can provide anonymity and, at the same time, enable some kind of mechanism to blacklist or moderate published content. However, these mechanisms are either too centralized or not enough to: avoid users downloading and visualizing unwanted content the first time they access a site (1), prevent the dissemination of illegal content (2), and offer ways of conditional anonymity to expose criminals (3). This paper focuses on these three open issues and proposes a novel architecture to revoke anonymity to users that publish content that does not follow the rules of the community that, in principle, hide the real identity of the publisher.

3. Building blocks

This section presents the technologies on which the system proposed in this paper is based. First, Section 3.1 defines how blockchain technology works and describes the Ethereum platform. Secondly, Section 3.2 introduces the IPFS protocol and its naming system called InterPlanetary Name System (IPNS). Thirdly, Section 3.3 focuses on self-sovereign identities. Finally, Section 3.4 describes threshold discernible ring signatures, a protocol that enables conditional k-anonymity in digital signatures.

3.1. Blockchain

Blockchain is a technology capable of persisting data in a distributed manner ensuring data integrity and availability. It employs an underlying P2P network to connect its users or nodes without the need for third parties. A blockchain is made up of a complex structure that stores data within transactions. For instance, in cryptocurrencies, the transactions represent the monetary transfers between users. These transactions are distributed by the users of the system throughout the P2P network. A set of special network nodes called miners is in charge of collecting transactions and packing them into a block. Regularly, the blockchain is appended with a new block. Like this, the blockchain becomes an append-only data structure, where information is organized into blocks that are linked and ordered by time, forming a large database or distributed ledger. Formally, a blockchain represents a global state machine where transactions are transitions between states. In general, blockchains like the one used in this paper have the following characteristics:

- **Permissionless**. Any user can participate in the generation and validation of new blocks.
- **Transparency**. The data are distributed and shared by all nodes connected to the network. Depending on the blockchain type, different techniques are used to safeguard users' privacy. The first blockchains, like Bitcoin or Ethereum, are considered pseudo-anonymous, because users' identities are represented by public keys (or addresses) and, therefore, they are hiding behind pseudonyms. More advanced cryptographic techniques are used in other blockchain systems to increase anonymity and completely hide users' activity, such as ring signatures in Monero (Van Saberhagen, 2014) or zk-SNARKs in Z-cash (Wilcox, 2016).
- **Open and censorship-resistant**. Any user can send transactions to the blockchain. The nodes that create and validate new blocks cannot censor the information generated by other users.
- **Immutability**. Data stored in a blockchain cannot be erased or modified. The use of asymmetric cryptography and hash functions ensures data integrity and non-repudiation.
- **Availability**. Blockchains are highly distributed systems where single points of failure are minimized and stored data are highly available to users.

Since a blockchain is created among multiple nodes participating in the network, a consensus algorithm is necessary to decentralize the control of the system and ensure that participants comply with the protocol rules, that dictate which transactions have to be considered valid and how blocks have to be constructed. Basically, a consensus algorithm achieves that all participants share a common system state by establishing a procedure to decide the node of the network that will create the next block in a way that is accepted by the others. One of the most prominent consensus algorithms is Proof-of-Work (PoW). In this algorithm, the block miners are competing to resolve a computationally expensive cryptographic puzzle. The first node to find a solution to that puzzle is the node responsible for the creation and appending of a new block to the blockchain. Alternative consensus algorithms are: Proof-of-Stake (PoS), Delegated Proof-of-Stake (DPoS), Simplified Byzantine Fault Tolerance (SBFT), or Proof-of-Elapsed-Time (PoET). More information about consensus mechanisms can be found in Cachin and Vukolić (2017) and Gramoli (2020).

Hence, in PoW, the nodes compete to publish new blocks containing the latest transactions sent in the network, a process known as mining. In this competition, based on solving a cryptographic puzzle, the node that has generated a new block receives cryptocurrencies as a reward and also as fees for each transaction included in the block. These rewards and the fact that generating a block has high computational and energy costs, create an incentive system that attracts participants to the network looking for rewards and ensures that the blocks are generated complying with the cryptocurrency protocol. If the protocol is not followed, the new block would be discarded by the other nodes and the miner would lose the cost of the used electricity.

Nowadays, one of the most popular blockchains is Ethereum (Buterin, 2014). Its main goal is to provide a global computing infrastructure where its cryptocurrency, called ether, is used to pay for this service. Ethereum uses a PoW consensus algorithm called Ethash. In this protocol, the state transitions are performed with the execution of Turing complete smart contracts in the Ethereum Virtual Machine (EVM). A smart contract can be considered as a deterministic computer program that runs automatically according to the conditions defined beforehand and the received inputs. The EVM and smart contracts

aim at creating a new environment called Web3 of decentralized applications or DApps (Antonopoulos & Wood, 2018). These DApps, as in a centralized architecture, are based on a backend containing the application logic, on a frontend with the user interface, and on other services such as data storage. Ideally, in a DApp these parts should work in a fully distributed way. However, at present, some parts tend to still be centralized. It should be pointed out that thanks to being deployed on a blockchain, DApps can provide transparency, censorship resistance and high availability.

Furthermore, Ethereum also provides an off-chain messaging protocol called Whisper (Ethereum, 2014). This technology uses Ethereum's P2P network to exchange small messages between DApps, smart contracts, or nodes without storing the message content on the blockchain. In this protocol, messages are sent to all network nodes using broadcast and a time to live (TTL). Each node that receives a Whisper message forwards it to its peer nodes and stores it locally for a limited time. Then, each node receives and only gets the message if the node is interested in it or is one of the intended recipients. To safeguard data privacy, messages can be sent encrypted using symmetric or asymmetric cryptography. Moreover, in Whisper, a sender can send anonymous messages if his/her address is not specified within the message. To avoid spam or denial of service (DoS) attacks, all nodes are forced to compute a simplified PoW for each sent Whisper message. The cryptographic puzzle computation costs will depend on the message length and the selected TTL.

3.2. InterPlanetary File System (IPFS)

Blockchain technology is designed to store information of high value and small size. Fees for executing transactions and storing data generally depend on data size and processing costs for network nodes. This makes it inefficient and expensive to store large amounts of data in a blockchain. To overcome this issue, blockchain applications can use decentralized off-chain storage services. The InterPlanetary File System or IPFS (Benet, 2014) is a distributed storage protocol that aims to connect all devices under the same file system. In this system, data are distributed through an underlying P2P network and are content-addressed. The latter means that the protocol uses a hash function to summarize a data set or a file and use the resulting hash value to identify it throughout the network. IPFS aims to become an alternative to the HTTP protocol and its centralized resource delivery scheme. Unlike the current client–server model where users request resources to specific servers using URL identifiers, in IPFS a resource is distributed across many network nodes and, therefore, any request can be satisfied by any of the nodes indexing the resource and the latency can be minimized because requesters can use multiple sources at the same time choosing among the nodes that provide the best performance (e.g. nodes located close to the requester or with large bandwidth).

The most notable IPFS properties are forgetfulness and censorship resistance, and web optimization. Resistance to forgetfulness and content deletion are achieved through data replication, minimizing single points of failure and increasing data availability, and through the static content-addressed system, which avoids data being lost when its location changes. Furthermore, the high decentralization of IPFS makes the system censorship-resistant. In the event of an attacker attempting to censor certain content from some nodes, data can be replicated and made available through other network nodes. Note that the forgetfulness and censorship resistance of IPFS may conflict with the right to be forgotten. Politou, Alepis, Patsakis, Casino, and Alazab (2020) proposes a protocol for delegated content erasure requests in IPFS that prevents censorship and allows people to exercise such right. Regarding web optimization, IPFS allows resources to be fragmented into several parts and downloaded in parallel from multiple nodes.

Finally, it is worth noting that publishing data that has to constantly be updated is not easy in IPFS. Since IPFS is a content-addressed system, any small modification in a file completely changes its hash value used to identify the file in the network. Therefore, making updates to a file means having to manage a different IPFS link for each newly published version. To tackle this problem, the InterPlanetary Name System or IPNS is employed. IPNS is an underlying IPFS system that enables a node to create a network space with a static address derived from an asymmetric key pair. In this space, nodes can update a resource that will always be accessible using the same address. It should be pointed out that nodes can create multiple network spaces at once under different key pairs.

3.3. Self-sovereign identities

Currently, service providers implement a multitude of security mechanisms designed to protect users' identities and personal data that such providers store. Despite this, theft of confidential data keeps happening because the databases of the service providers (containing thousands of registers with personal information of their clients) represent an attractive target for attackers.

Furthermore, another important threat in this regard comes from the service provider itself. With this operational model, once a user sends personal data to a service provider, such user loses control over his/her data. From that moment on, the user can no longer be completely sure where the service provider stores his/her data, how many copies exist, if his/her data are shared with third parties, etc. Although many countries have approved data protection laws and many services include privacy policies that aim at protecting citizens' privacy, it is also a fact that these have not avoided all the problems mentioned above.

Self-sovereign identities present a decentralized alternative to identify users and store their personal data that avoids some of these issues. Basically, in self-sovereign identity projects, the users are responsible for creating, maintaining and deciding when and with whom they share their personal data. Liu et al. (2020) provides a comprehensive survey of blockchain based identity management systems. Among other, some popular projects are Sovrin (Sovrin, 2018), uPort (Lundkvist, Heck, Torstensson, Mitton, & Sena, 2017) or iden3 (iden3, 2020). In general, in these types of projects, users can autonomously create an identity based on a public–private key pair that represents them uniquely. Later on, users can use these keys to receive verifiable credentials from other digital identities. The verifiable credentials are a specification defined by the W3C (Sporny, Longley, & Chadwick, 2019) with which users can obtain certifications, endorsements or any other type of badge from a third party, which they can later use to prove a certain feature to another party in a machine-verifiable manner. These verifiable credentials contain a digital signature to cryptographically prove who issued the claim and avoid tampering. To share claims with third parties (i.e. verifiers), users group claims from different credentials into verifiable presentations which contain the digital signature of the identity subject and verification proofs to avoid replay attacks. For instance, users can receive a credential signed by an educational institution representing a degree or a credential signed by the organizer of an event stating that the user has attended a workshop. Other examples of verifiable credentials can certify the user's age, address, driver's license, etc. These credentials combined with zero-knowledge proofs (ZKP) enable a mechanism where users can verifiably prove to other entities any feature related to their identity disclosing the minimum possible amount of personal data.

3.4. Threshold discernible ring signatures

In asymmetric cryptography, also called public-key cryptography, users generate a key pair containing a private and a public key, with which they can perform two basic actions: sign or encrypt/decrypt messages. To perform a digital signature, a user can use his/her private key to generate a signature of a message that anyone can verify using the signer's public key. In the case of encryption, anyone can use the public key of another user to generate a ciphertext of a message,

ensuring that only the person or computer in control of the private key associated with that public key will be able to decrypt.

These mechanisms allow users to incorporate basic security principles in their communications, such as confidentiality, integrity and non-repudiation. However, this simple use of asymmetric cryptography is not enough to achieve the desired results of this paper. In the system proposed here, the main goal is to use digital signatures to keep a user anonymous, hiding his/her identity among a group, unless several members of the group decide that the content published by the user is inappropriate. Therefore, to achieve this, advanced digital signature schemes have to be used.

Techniques such as group signatures or ring signatures have been proposed for similar scenarios. The former, group signatures, are described in Chaum and Van Heyst (1991) and allow a signing entity to remain anonymous because the signature is performed on behalf of a group administered by a manager. The group manager, in case of dispute, can revoke the anonymity of the signer using a trapdoor. This type of signature can be used, for example, to authorize the use of a printer to any member of a department, but anonymizing the employee who has printed a specific document.

Ring signatures, formalized in Rivest, Shamir, and Tauman (2001), do not require a group manager to set up the environment and provide unconditional anonymity for the signer. Hence, they do not offer any mechanism to revoke anonymity. In this type of signature, any user can create a signature with his/her private key and the public keys of other users without requiring their cooperation. Ring signatures have been used in proposals such as CryptoNote (Van Saberhagen, 2013), which defined the technological base of Monero (Van Saberhagen, 2014), a cryptocurrency that enables anonymously sending transactions. In Xiong, Chen, and Li (2012), authors use revocable ring signatures (Liu, Liu, Mu, Susilo, & Wong, 2007) to build a privacy-preserving auction protocol. In this protocol, the auctioneer and registration manager can collaborate to reveal the identity of a malicious bidder.

Additionally, in Bresson, Stern, and Szydlo (2002), the authors propose threshold ring signatures. In this mechanism, a set of t users can collaborate to sign a message hiding their identity within a larger group of n members. With this protocol, anyone can verify that the message has been signed by t users of the group without knowing the exact members that signed. The authors of the paper argue that this mechanism is useful, for instance, in contexts with multiple portable devices and mobile applications where a third party cannot be used to centralize the process of forming ad-hoc groups to exchange sensitive information.

Nonetheless, for the system proposed in this paper, none of these mechanisms is entirely suitable for one of the following three reasons: (1) group signatures require an administrator, which leads to centralization, introduces single points of failure, and requires trust in the central party. (2) Ring signatures provide unconditional anonymity and, therefore, signers publishing inappropriate content can never be de-anonymized. (3) Threshold cryptography requires the active participation of several users to generate a signature. In our case, we require participants to only act reactively when they see inappropriate published content. Otherwise, it would be a cumbersome process if users would have to actively participate in the signature process to authorize every publication beforehand.

For the system proposed in this paper we use threshold discernible ring signatures (TDS), a mechanism formalized in Kumar, Agrawal, Venkatesan, Lokam, and Rangan (2010). In Qureshi, Garcia-Font, Rifà-Pous, and Megías (2020), we use TDS to create an emergency reporting system which, in case of reporting a true emergency, it enables the reporter to remain anonymous and get rewarded in cryptocurrency and, in case of a false emergency, the reporter can be de-anonymized and punished.

In TDS, a member U_j of a ring containing n users ($U_1, \ldots, U_n$) can sign a message m on behalf of the entire ring without requiring the intervention of the other members, as in a conventional ring signature. To do this, first it is required that each member U_i publishes his/her public key (y_i) and a parameter α, being $y_i = g^{x_i}$; x_i the private key of U_i; g a generator of a subgroup of order q of Z_p^*; p and q large primes ($p, q \gg n$); and α a pseudo-random integer. However, compared to conventional ring signatures, which offer unconditional anonymity, in TDS, if t ring members cooperate, then they can revoke the anonymization of the signer. TDS consists of three procedures:

- **Signing Procedure**. $S_{TDS}(g, x_j, y_1, \ldots, y_n, \alpha_1, \ldots, \alpha_n, t, m)$ returns the discernible threshold ring signature σ of the message m. In the system proposed in this paper, this procedure is executed by the user who wishes to make a publication. More details in Section 4.3.2.
- **Verification Procedure**. $V_{TDS}(m, \sigma)$ returns 1 or 0 depending on whether the discernible threshold ring signature σ has been made on the message m using one of the private keys associated with a public key of any of the members of the ring. In the system proposed in this paper, this procedure is executed by all users when they download any content published by another member of their group. More details in Section 4.3.2.
- **Threshold Distinguisher Procedure**. $T_{TDS}(m, \sigma)$ returns an i index that indicates, from the set of public keys of the members of the ring, the one corresponding to the signer of the message. The collaboration of at least t ring members is required to be able to carry out this procedure. Briefly, these members must decrypt part of σ using their private keys and share the result of the decryption. Once t users have shared their respective decrypted shares, then it is possible by anyone to combine the public data (e.g. α), data in σ and all the shares to obtain the index i pointing at the public key of the actual signer that created σ. In the system proposed in this paper, this procedure can be executed by the users of the group when they believe that a certain publication does not follow the rules defined by the group and, therefore, they wish to de-anonymize the publisher. More details in Section 4.3.3.

These three procedures involve using cryptographic mechanisms such as equality signatures (Klonowski, Krzywiecki, Kutyłowski, & Lauks, 2008), knowledge signatures (Camenisch, 1997), and Shamir's secret sharing scheme (Rivest et al., 2001). The details of TDS go beyond the scope of this paper. More information on the specific algorithms involved in each procedure can be found in Kumar et al. (2010).

4. System description

The main goal of the proposed system is to enable the distribution of hypermedia publications in a way that publishers can remain anonymous if their publications follow certain rules agreed beforehand by a group of distributors. Distributors can become members of groups that follow their interests and their moral standards and contribute to disseminate publications and avoid censorship. For example, users of the proposed system can join a group that vetos copyright-protected content or certain offensive publications.

To reach this goal, users can create groups of k members in an autonomous and ad-hoc manner. The members of the group have to previously agree on the basic rules that will regulate their publications. Once a group has been set, then its members can publish content in a k-anonymous way (i.e. the identity of the publisher is hidden among the identities of the k members) and contribute re-distributing publications of the other members of the group. Nevertheless, to enforce users to only publish content according to the predefined rules, the system offers a mechanism to revoke the publishers' anonymity if enough members of the group cooperate.

The following sections describe in detail the proposed system. Firstly, Section 4.1 describes the key entities in the system and the basic actions that can be performed by such entities. Secondly, Section 4.2

Table 1
Key entities and basic actions.

Entity	Roles	Basic actions
Users/Members	Admin	Setup groups (create groups, accept/deny/expel users)
	Publisher	Create, sign and publish content
	Delator	Denounce content
	All	Create a DUID Register in groups Share group addresses Download content Validate content Unsubscribe from groups
Identity authorities	Trusted Identity Authority (TIA)	Register user identities Certify DUIDs Re-identify users and law enforcement
	Credential Issuing Authority (CIA)	Issue verifiable credentials

shows some potential scenarios for our proposal. Section 4.3 presents a global overview of the proposed architecture and gives details on the main protocols of the system, and, finally, Section 4.4 describes important metadata used to enable these protocols.

4.1. Key entities and basic actions

This section defines the key entities that participate in the proposed system and the actions that can be performed by each entity (see Table 1). Firstly, it is important to distinguish between two types of key entities: users who participate in the system assuming different roles, and authorities that validate user identities and issue credentials.

For this proposal, a user can be considered as any natural person who wants to participate in the system. Each user may belong to one or more groups restricted by different predefined rules. For instance, a user may want to participate in a serious political discussion group that aims to prevent the spread of fake news and, at the same time, participate in another group with completely different requirements to share humorous political content. Once a user creates or joins a specific group, he/she becomes a group member that can assume different roles throughout the lifetime of the group. The possible roles are Admin, Publisher and Delator.

Admin is the role that a user assumes to manage membership. By default, the creator of a group becomes its Admin. This user then has the responsibility to accept new users and to expel them in the de-anonymization process. Note that any member can voluntarily unsubscribe from a group at any time to stop receiving new resources. It should also be noted that the Admin does not have any special privileges regarding publication or de-anonymization actions.

A group member takes the Publisher role when he/she starts the process to publish a resource. In this case, the user creates and signs using TDS a resource and publishes it in IPFS. Then, the Publisher shares a link to this resource with the rest of the group. When the other members download the resource, firstly, they verify that its TDS is valid. If the signature is invalid, the resource is directly discarded by the members. If the signature is valid but the resource content does not follow the group rules, the group members can act as Delator and start a process to de-anonymize the Publisher.

The proposed system also requires the participation of identity authorities as Table 1 shows. This type of entity can assume two roles: Trusted Identity Authority (TIA) or Credential Issuing Authority (CIA). Basically, identity authorities with a TIA role are responsible for signing Distributed User Identifier Documents (DUID), further details on Section 4.4.3, which are documents created by the users with their cryptographic wallets and used as proof of identity to participate in the system. Prior to joining any of the groups, a user has to get his/her identity attested by a TIA. In this process, the TIA verifies in person that the user is whom he/she claims (e.g. through an identity card or passport) and registers the user in an internal database shared by all TIAs. The purpose of this is twofold. Firstly, this allows law enforcement on users in the process of being de-anonymized because of illegal publications. Secondly, this avoids users registering more than once in the system with different identities and prevents sybil attacks, as explained in Section 5. The system used by the TIAs to register users is not the focus of this proposal and, therefore, its specification is out of the scope of this paper. Census offices, local governments and other public entities of this kind are good candidates to assume a TIA role.

Once a user has a DUID signed by a TIA, then he/she can obtain verifiable credentials from the CIAs. Unlike a TIA, any type of entity can act as CIA. In this proposal, groups may require users to hold a certain credential to become a member. The credentials issued by the different CIAs will be accepted or dismissed depending on the requirements of each group. For example, a group may require users to hold a verifiable credential that certifies that they are members of a sports club, and another group may require more official certifications, such as a specific university degree.

Finally, groups are made up of verified members (i.e. users with a valid DUID and the required verifiable credentials) who share hypermedia resources signed with a TDS and can participate in a de-anonymization process if they believe that a publication goes beyond the limits set by a group. To create a group, its first member, who will also act as Admin, needs to generate and publish in IPFS a manifest file containing the group information and rules (more details in Section 4.4.1) and deploy a smart contract in Ethereum that links the manifest with a member list. This list is used to store the IPFS links to the DUID of each member. Any group member can access the group smart contract to get links and download from IPFS the DUIDs of the other members and obtain the public parameters needed to carry out the TDS.

4.2. Potential scenarios

This section describes three potential scenarios where the proposed architecture could be implemented. The scenarios have been chosen to clarify the proposal and we describe the expected outcome in three paradigmatic situations.

These scenarios are based on a group to distribute content about environmental protection. The group exclusively accepts graduates in the field of environmental sciences. When the Admin created the group, he/she defined a manifest containing a constraint stating that group members must hold a credential issued by a university (acting as a CIA) to demonstrate that they hold an environmental science degree. Other relevant constraints included in the manifest specified the topics that can be covered in the publications, the rights on the published content, and the number of members required to revoke publisher's anonymity.

Once the group has enough members, the users can start publishing articles, denouncing governmental policies, organizing protests, etc.

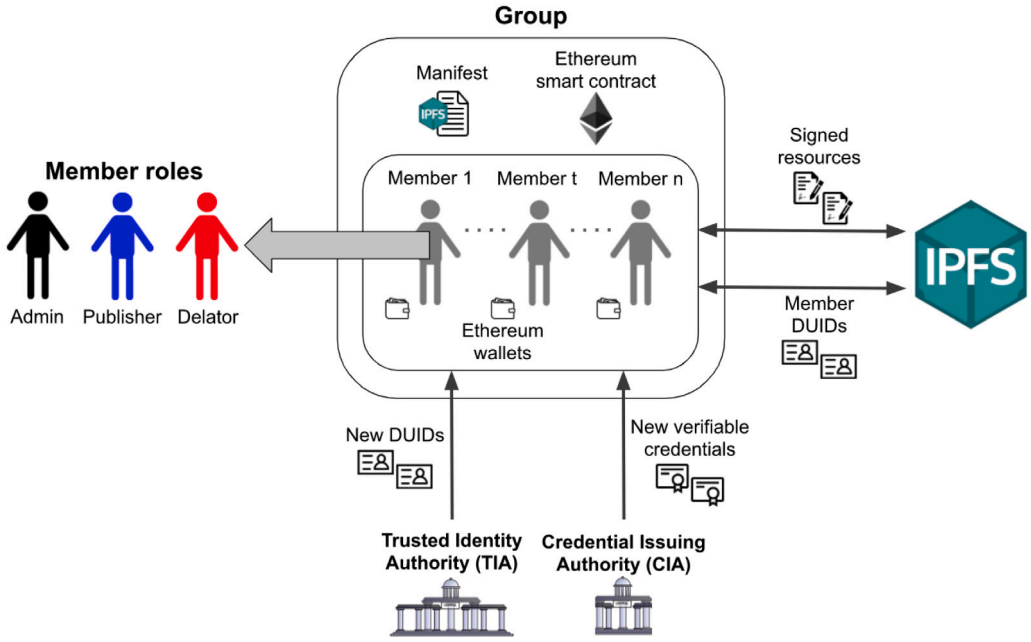

Fig. 1. Architecture overview.

When the publications follow the manifest, group members provide k-anonymity to the publishers. The following scenarios reflect what happens otherwise.

Scenario 1

A group member, acting as Publisher, begins to distribute political content not related to environmental protection. If enough members consider the publications as inappropriate, then they can cooperate, acting as Delator, to de-anonymize the Publisher. In this case, the Publisher is excluded from the group and his/her publications are deleted and no longer distributed.

Scenario 2

In this scenario, instead of distributing political content, the Publisher distributes copyrighted material. The procedure is similar as in Scenario 1. However, in this case, someone (a group member or anybody else) might report the Publisher to the authorities for breaking intellectual property laws.

Scenario 3

Considering the potential legal penalties in Scenario 2, some group members decide to create another group in which the manifest allows the distribution of copyrighted content. By doing this, the second group is totally independent from the first and, therefore, members of the first group who do not join the second do not contribute to provide k-anonymity for the publications made in the second group. When the second group begins to distribute copyrighted content, even without any Delator, it is possible to identify that all the group members are voluntarily cooperating to distribute copyrighted material. Hence, this should disincentivize group creation to carry out criminal activities. Users with this in mind probably prefer platforms that provide full-anonymity.

4.3. Architecture main protocols

The schema in Fig. 1 shows an overview of the proposed architecture and the relationships between the key entities and the basic actions described in Section 4.1. As can be seen in the figure, a group is made up of a manifest, an Ethereum smart contract (hereafter SC), and a member set. As previously stated, the members can adopt different roles throughout the lifetime of the group. To interact with the system and create, store, and manage their DUIDs and credentials, members use an Ethereum wallet. In this regard, this schema also shows the interaction between users and identity authorities in the attestation and issuance process of DUIDs and verifiable credentials. Additionally, the schema represents the communication between users and IPFS used to store and share DUIDs and hypermedia resources. To sum up, the relationships between the previously described key entities and the basic actions translate into the definition of three main protocols that determine the system behavior. These protocols are the Group Setup Protocol, the Content Publication Protocol and the Content Denouncement Protocol, which are described in detail below.

4.3.1. Group Setup Protocol

The Group Setup Protocol is shown in Fig. 2. The protocol specifies the steps required for a group creation (path a, i.e. 1, 2, 3, 4a, 5a) and the steps required for user registration in an existing group (path b, i.e. 1, 2, 3, 4b, 5b, 6b, 7b). The first three steps are related to DUID attestation and are common to both actions. The DUID object is described in Section 4.4.3.

The common steps of this protocol are as follows:

1 To create a group or to register in an existing one, a user must visit a TIA to attest his/her identity and get his/her DUID signed. In the case that the user wants to register in an existing group, he/she must visit one or more specific CIAs to get the verifiable credentials required by the group.
2 The TIA verifies the user's identity through a conventional personal verification process (e.g. requiring a national ID card). If the verification is successful, the TIA securely registers the user in a database shared among all TIAs, where the TIA links the user's public key y_j to the real identity of the user. The TIA signs the DUID and sends it back to the user. It should be pointed out that a user can only have one attested DUID at a time with which to be registered in multiple groups. CIAs are responsible for issuing verifiable credentials to the user. Before issuing a verifiable credential, a CIA verifies that the user's DUID is valid.
3 The user stores the DUID signed by the TIA publicly in IPFS under an IPNS address.

The steps to create a new group are (path *a*):

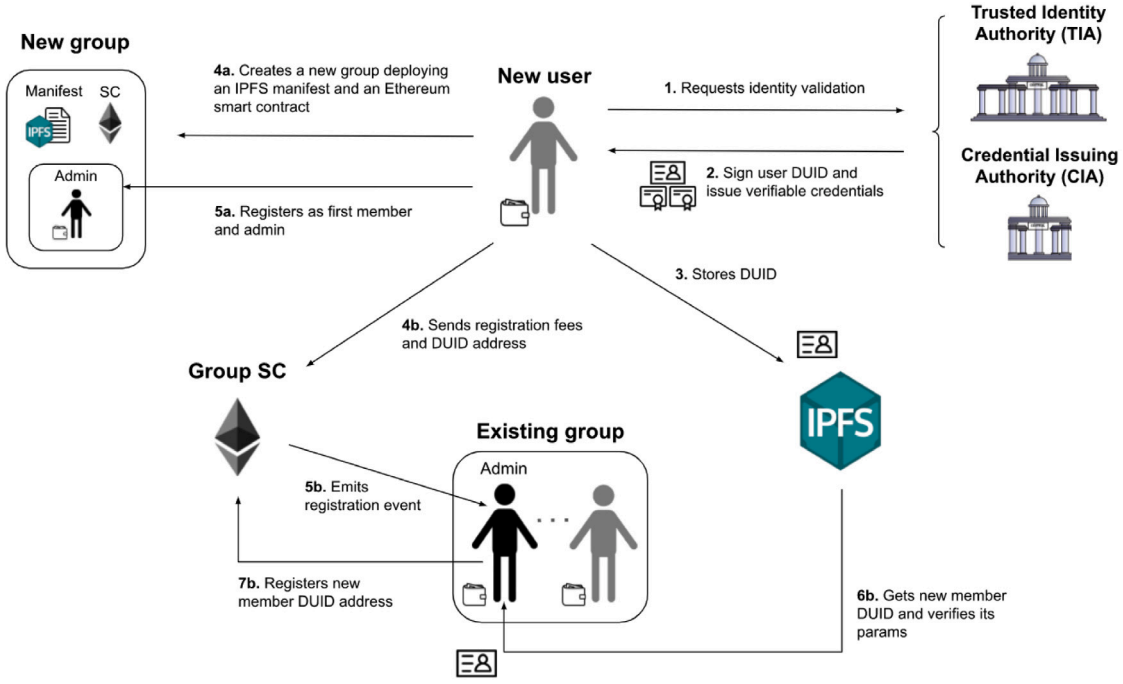

Fig. 2. Group Setup Protocol.

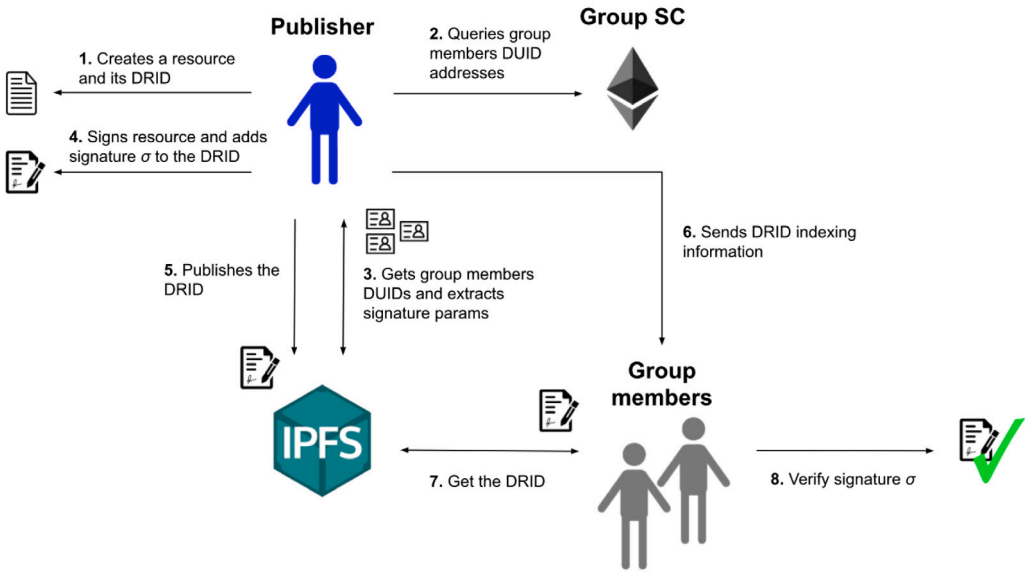

Fig. 3. Content Publication Protocol.

4a The group creator deploys a SC on Ethereum and a manifest (described in Section 4.4.1) on IPFS. The user that creates a group has to follow the manifest rules, such as owning a DUID signed by a valid TIA. Otherwise, other users will never join the group when they validate its manifest and its creator's identity.

5a The group creator registers himself/herself as a member of the group and it becomes its Admin. His/her DUID address is registered in the SC as it is done in path b for the other members.

The steps to register a user in an existing group are (path b):

4b The user sends an estimated amount of ether to the group SC to pay for the fees of the entire registration process. In the same transaction, the user also sends the IPNS address pointing to his/her DUID.

5b Once the ether for the registration process is received, the group SC emits an event to the Admin containing the IPNS address of the user's DUID.

6b The Admin downloads the user's DUID from IPFS and validates the required parameters: the public key y_j, the α_j, etc. The Admin also validates that the DUID is correctly signed by a TIA and the user owns the verifiable credentials required to access the group.

7b The Admin registers in the group SC the IPNS address of the user's DUID. It is important to note that, once a member has been accepted in a group, all members have the same role in terms of resource publication. This means that an Admin can accept/deny new group members, but he/she has no special rights regarding the published content. The remaining ether not spent in transaction fees is returned to the sender.

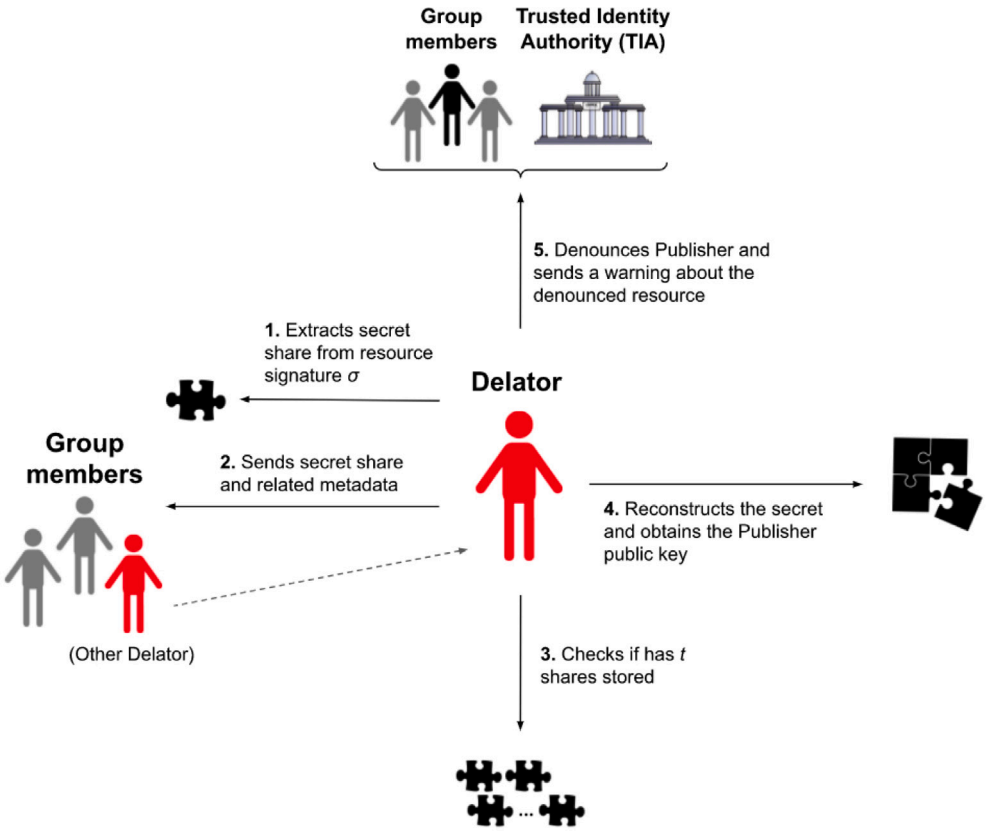

Fig. 4. Content Denouncement Protocol.

4.3.2. Content Publication Protocol

The Content Publication Protocol is shown in Fig. 3. This protocol describes the steps that a user (hereafter Publisher) that is already a member of a group has to follow to publish content in the group. The protocol is as follows:

1 The Publisher creates a resource locally and adds its content to a Distributed Resource Identifier Document (DRID). The DRID object is described in Section 4.4.4.
2 The Publisher queries the group SC and obtains the IPNS addresses of all group members DUIDs.
3 The Publisher downloads the DUIDs of the other group members from their IPNS addresses, and, from each of them, he/she extracts the public key y_j and α_j.
4 The Publisher generates a resource TDS σ using his/her private key x_i, his/her α_i and the public keys y_j and α_j of all other group members by setting a threshold t as defined in the group manifest. The generated signature is added to the resource DRID.
5 The Publisher stores in IPFS the DRID containing the resource and its signature. From here, the published resource can be downloaded by anyone who knows the resource IPFS address.
6 Once the resource DRID has been stored in IPFS, the Publisher uses the Whisper communication protocol to anonymously send to all group members the indexing information of the resource. The receivers store locally the IPFS address of the new resource together with the identifier of the group (i.e. the address of the group's SC) that has published the resource (more details in Section 4.4.5).
7 Other group members can download from IPFS a copy of the DRID containing the resource and its TDS σ.
8 A group member who downloads the DRID verifies that σ is a valid TDS generated by a user that was a member of the group at the time of signing the resource. In the case that the resource signature is not valid or the Publisher was not a group member, the group member that downloaded the new resource deletes the DRID copy stored in his/her IPFS node to avoid contributing to the resource redistribution.

4.3.3. Content Denouncement Protocol

The Content Denouncement Protocol is shown in Fig. 4. Once a Publisher has distributed a new resource among the group members following the process seen in the previous section (i.e. the resource DRID contains a valid TDS σ), then other group members can verify that the new resource content follows the group manifest rules. If that is not the case, the other group members can denounce the published resource. At this point, the member executing the content denouncement protocol becomes a Delator. The protocol is as follows:

1 From σ, the Delator extracts and decrypts his/her secret share s_j. Then, the Delator stores the secret share locally (more details in Section 4.4.5).
2 The Delator sends the extracted secret share and additional metadata (more details in Section 4.4.5) to the other group members using the Whisper communication protocol. At this point, any group member can also become a new Delator to help uncover a Publisher.
3 The Delator checks if he/she has t secret shares locally. If so, the Delator goes to step 4. Otherwise, the Delator waits until other Delators contribute with their secret shares.
4 As explained in Section 3.4, any Delator having t decrypted secret shares and their related α values can use these together with σ to execute the Threshold Distinguisher Procedure to de-anonymizing the Publisher. With this procedure, the Delator obtains the index i from the public key list inside σ pointing at the public key of the signer that has generated σ.

5 The Delator has demonstrated who is the resource Publisher, so he/she has lost the k-anonymity. From here, a Delator can address the group admin or a TIA to denounce the Publisher. The Delator can also notify, through the Whisper communication protocol, the other group members about the denounced resource. Hence, group members are able to remove the indexing information of the resource (see Section 4.4.5) and the resource itself if it had already been downloaded.

4.4. Metadata

This section describes the metadata structures employed by the protocols to store and share among all system participants information about users, groups and hypermedia resources. Sections 4.4.1 and 4.4.2 describe metadata of the group manifest and the group smart contract. Section 4.4.3 presents the data structure of a DUID and defines its fields. Section 4.4.4 introduces the DRID, a document to encapsulate a resource and its TDS. Finally, Section 4.4.5 defines other data structures required by this proposal that users have to store locally.

4.4.1. Group manifest

The manifest is an essential component when creating groups because it includes a set of rules that group members will enforce on publications. Manifests must include one mandatory rule and several more optional rules. The mandatory rule has to do with the threshold t that marks the number of group members that are necessary to denounce and de-anonymize a publisher. This is important for publishers when generating the TDS of a resource, and also for the other group members, which should not accept any publication that has not a verifiable signature with the required threshold. Moreover, the optional rules have to deal with the type of publications that the group will accept, for example banning any offensive or sexual content.

In this paper, we propose to create group manifests in a way similar to legal contracts, rather than smart contracts. Therefore, in the current state of the proposal, it is the responsibility of the group members to interpret the manifest rules and report members publishing any content not complying with the rules. It falls beyond the scope of this paper and remains as future work to extend the proposal with rules that can be evaluated automatically in a smart contract friendly manner using oracles or DApps. For instance, including in the manifest a rule that uses a text analysis DApp to extract information from resources and forbid publications that can be considered lexically offensive, removing the subjectivity of group members.

Technically the manifest is generated in a JSON file that the group creator defines and publishes in IPFS before creating the group SC. This JSON file specifies the threshold value t that group members must use both to sign and validate resources, a plain text with the type of publications that will be approved or banned and a set of requirements regarding the verifiable credentials that users need to be accepted in the group.

4.4.2. Group smart contract

In this proposal, the necessary data to manage the status of a group is persisted in a smart contract deployed in Ethereum by the Admin of the group. In this way, these data are stored in a public and distributed manner on the blockchain, which allows all group members to obtain up-to-date information about the group and the other members at any time. The group SC implements the procedures to register/unsubscribe users and to publish/update group information (i.e. some of the basic actions of Section 4.1).

Users have to pay the fees for the registration procedure. A prior step before executing it involves making a deposit in the group SC with the estimated cost for the registration. When the registration procedure gets executed by the Admin, the transaction costs will be paid with this deposit and the remaining funds will be returned to the user applying to join the group. To avoid locking funds for a long time, the deposit will have an expiration date. If the Admin does not execute the registration procedure before that, then the deposit is returned to the depositor. This deposit also avoids denial of service attacks of users flooding the Admin with false membership requests.

Data persisted by a group SC are of two kinds: general information of the group and its members. Regarding the former, the SC only stores the IPFS address of its group manifest. Any other metadata is directly added to the manifest file in IPFS reducing the amount of data stored in the blockchain and saving in transaction fees. A group SC also stores a list with all registered members. For each member, the SC stores his/her IPNS address, which points at the user's DUID, and two dates, when he/she registered and, if it is the case, when he/she abandoned the group either because of a voluntary unsubscription or expulsion. These two dates allow members to know which users were members of the group at the time of generating a TDS and publishing a resource. If the creation date of a resource falls between the dates when a user was a member of a group, then, this means that the public parameters (i.e. public key, α) of this member were used to generate the TDS of the resource.

4.4.3. Distributed User Identifier Document (DUID)

Decentralized Identifiers or DID (Reed et al., 2020) are used to represent the users in the system. DID identifiers provide verifiable identities in a decentralized way, being independent of any centralized registry. Since Ethereum addresses are considered unique, each one can represent an identity or DID. Users use a verifiable data registry deployed on Ethereum to record and manage their identities, allowing them to resolve and produce DID documents from a DID identifier. These documents contain data associated with a specific DID such as identifiers (i.e. Ethereum addresses), delegated entities, public keys, service endpoints and other attributes. For example, users can manage their identities using the smart contract registry specified by the Ethereum Improvement Proposal 1056 (Braendgaard & Torstensson, 2018).

A Distributed User Identifier Document or DUID unifies a user DID and the public parameters required by the protocols described in Section 4.3. If a user wants to participate in the system, he/she must generate a DUID (using a wallet or a DApp) that contains the user DID, an asymmetric key pair for TDS and other fields listed below. Then, the user must visit a TIA which registers the user's identity and signs the content of his/her DUID. Once a user gets a signed DUID, he/she can request verifiable credentials to any CIA. It is important to highlight that a DUID does not contain verifiable credentials that could be used to identify its owner. All credentials are stored and managed by a self-sovereign identity application which the user uses in conjunction with the system proposed in this paper (see Section 3.3).

DUID data structure is specified in JSON format and its fields are set out in more detail below:

- The user DID containing a pre-generated Ethereum address.
- The public key y_j from an asymmetric key pair generated by the user. This public key is used to sign new hypermedia resources.
- A public parameter α_j for TDS.
- A signature of an object containing the parameters listed above. This is used by third parties to verify that the DUID user is the key pair owner.
- The public keys of the previous versions of this DUID (if there is any). This is necessary to revoke previous DUIDs of the same user, since a single user cannot use more than one DUID simultaneously and users have to be able to change their DUID if they lose or expose their private keys.
- The public key of the signing TIA so that any third party can verify the TIA's signature. Additional information can also be added to help identify the signing TIA (e.g. the name and contact information of the TIA).
- TIA's signature of the object that contains all fields above.

4.4.4. Distributed Resource Identifier Document (DRID)

A Distributed Resource Identifier Document (DRID) is a data structure in JSON format used in this proposal to store published resources. A DRID contains the following mandatory data: resource content, creation date and the TDS of these two fields. Additionally, a DRID can also include optional fields, such as a title, a description, MIME type or any other metadata that helps identify the resource. If these fields are included, then they are also signed in the TDS. Finally, in this proposal, the DRID is published in IPFS and the resulting address is sent to all group members as specified in the Content Publication Protocol in Section 4.3.

4.4.5. Local data

To interact with the system and follow the protocols described above, users use a DApp that executes the application logic. As can be seen in Fig. 1, users have a wallet that they use to interact with the blockchain and also to manage and store certain information that is kept locally. The main data stored by the users are:

- **DUIDs**. Users have to publish their DUID on their IPFS node to allow other group members to access this file. Users also store DUIDs locally from other users to extract the necessary parameters to sign and verify TDS.
- **Hypermedia resources**. Users download and share on their IPFS nodes resources (i.e. DRIDs) from other members of their groups.
- **Hypermedia resource metadata**. When a user creates and signs a resource, its indexing information is broadcasted to all other network users using the Whisper communication protocol. Thereafter, each user filters and retrieves messages destined for his/her groups and discards the rest. The indexing information consists of the tuple ⟨resource IPFS address, group SC address⟩, where the first field is the IPFS address of the published resource DRID and the second is the SC address of the group that publishes the resource. It should be pointed out that users store a list with the SC addresses of the groups in which they are registered, allowing them to internally group their resources.
- **TDS parameters**. Users need to store the secret shares extracted from a TDS to de-anonymize a publisher. Each user keeps a local list of shares extracted by himself/herself and received by other group members through the Whisper communication protocol. This list contains tuples in the format $\langle s_j,$ resource IPFS address, $\alpha_j\rangle$. This tuples puts together a share s_j, the IPFS address of the resource from which it is extracted and the public parameter α_j of the user j that has extracted s_j (all these parameters are required in the Threshold Distinguisher Procedure, as it can be seen in Section 3.4).

5. Proposal evaluation

In Section 5.1, we evaluate security risks of the proposed protocols and we list several contingency strategies. In Section 5.2, we discuss the proposal in general terms.

5.1. Protocol evaluation

This section lists possible risks (indicated as R below) of the proposed protocols related to integrity, confidentiality, de-anonymization, impersonation, availability, and sybil attacks. Additionally, for each risk, we describe possible contingency strategies or security measures (indicated as SM below) to mitigate these risks.

Integrity

R An adversary may try to modify a resource published by another user.

SM When an existing resource is modified, the hash value of its DRID and, therefore, its IPFS address changes, which means that the attack would be creating a new resource rather than modifying an existing one. Furthermore, with any alteration of the content, the TDS signature of the resource would not be valid and the new DRID would be rejected by the other members of the group. To overcome this fact, the attacker may try to perform an impersonation attack.

R An adversary may try to modify his/her DUID or other users' DUIDs.

SM If an adversary edits his/her DUID, the signature of the TIA will no longer be valid and, therefore, other users will discard this version of the DUID. If the adversary modifies the DUID of another user, this will create a new file in IPFS, but the original document will still be accessible from the original IPFS address and the IPNS address of its owner.

R An adversary may try to modify a group manifest.

SM A group SC stores the IPFS address of its group manifest. Any change in the manifest involves generating a new IPFS address. Only the group Admin can edit the attribute in the group SC containing this address.

Confidentiality

In the scenario of this proposal, confidentiality does not represent a risk for the publications because, so far, we have assumed that users use the groups to distribute open content. Publishing encrypted resources is also possible with the described architecture. However, the security of this is beyond the scope of this paper, since it depends on the specific encryption and key distribution protocol used by the interested parties.

De-anonymization

R An adversary may try to de-anonymize users tracing back the users' activities by exploring the transaction graph in Ethereum using as a starting point the addresses associated with the digital identities.

SM Best practice to avoid this type of attack in a blockchain is to not re-use addresses and create new addresses for each interaction with the blockchain. Users can as well use mixers to ensure that previous transactions are unliked to future transactions required by this proposal.

R Several TIAs may collude to de-anonymize a user.

SM TIAs may know if a user is a member of a group. However, TIAs cannot know which member of the group published each resource since this is protected by the TDS schema as explained in Kumar et al. (2010). Collusion does not give any additional information to the TIAs.

R An adversary may try to de-anonymize a Publisher using the TDS signature of a publication.

SM As explained in Kumar et al. (2010), obtaining the signer of a TDS signature is not feasible in polynomial time without the cooperation of at least t members sharing their secret shares.

R A group Admin may try to identify users when registering in a group.

SM In the registration process, users send a transaction to a specific group SC containing the IPNS address of their DUIDs. Since a user DUID only contains public parameters and none of these identifies the natural person behind the DUID, it is not possible to know users' real identities. Furthermore, the Ethereum address used to send the transaction can be different for each group registration, which avoids transaction tracking.

R An adversary may try to de-anonymize a Publisher that has published content in several groups or in a group and also in a conventional online social media.

SM The protocols presented in this paper do not protect publishers in case of inference attacks. Therefore, adversaries may use text analysis techniques to link several publications from the same author posted in different sites. However, it should be noted that a Publisher does not need to be the author of the text. Furthermore, the proposed protocol does not protect the Publisher regarding potential leaks in the text or in the published files' metadata.

Impersonation

R An adversary may try to impersonate a natural person and register as a member of a group.

SM To impersonate someone, an adversary has to hold a DUID signed by a valid TIA. For that, an adversary has two options: try to bypass the physical security measures of TIAs to confirm people's identities (for example, by using counterfeit identity cards or driver's licenses), or try to impersonate a TIA and sign himself/herself a DUID. The second option implies an attack against a digital signature algorithm, for example, RSA. The security of RSA is based on the difficulty of the factorization of the product of two large primes. Breaking this cryptosystem, and in general other standard digital signature mechanisms, is considered infeasible.

R An adversary may try to impersonate a group member to create transactions to remove the victim from the group.

SM Impersonating a user to create a valid transaction implies generating valid ECDSA digital signatures in Ethereum. Without knowing the secret keys of the victim, the attacker would have to solve the elliptic curve discrete logarithm problem, which is considered infeasible.

R An adversary may try to impersonate a group member and publish content in his/her name.

SM To publish valid content accepted by the other members of the group, the adversary has to create a TDS signature. Without knowing the secret key of the victim, the adversary cannot forge a TDS signature in polynomial time, as it is explained in Kumar et al. (2010).

R An adversary may try to impersonate a member of a group to start a process to denounce some publication as improper and de-anonymize the Publisher.

SM As explained in Kumar et al. (2010), the secret shares for each user are verifiably encrypted in N shares. Without the private key of the victim, it is not feasible to reveal his/her secret share, because the adversary would have to solve the discrete logarithm problem, which is considered infeasible.

Availability

R An adversary may try to block another user joining a group.

SM To avoid a user joining a group, an adversary may try to prevent that an Admin confirms a user as a member of the group, or the adversary may try to prevent the candidate from sending a request to the Admin. In both cases, this would imply preventing users from sending transactions to Ethereum. This attack can be considered infeasible in this scenario because Ethereum and, in general, blockchain systems, are considered as highly distributed systems. An analysis of attacks against the blockchain, like the Eclypse attack, falls out of the scope of this paper.

R An adversary may try to selectively block publications of other users, or he/she may try to block the reception of publications to specific victims.

SM Users publish content in IPFS, which is a highly distributed system considered as censorship-resistant. When some content is published, it gets distributed to several peers, which then can distribute the content to more peers, and so on. Therefore, blocking certain URLs or certain peers would not completely block the distribution or the reception of the content, because a victim could always find other peers to obtain the data or use other peers to distribute the content. In any case, this proposal is based on the resilience and distribution capabilities of IPFS and finding possible problems or attacks against this system falls out of the scope of this paper.

R An adversary member of a group may try to prevent messages forwarding to other group members, for example, messages containing the decrypted secret shares to de-anonymize a Publisher.

SM Forwarding messages to a recipient does not depend only on a single member and, therefore, it would be necessary for several members of a group to collude to isolate another member. Furthermore, to mitigate this issue, users may add more members of the group as their peers.

R An admin may remove a user from a group even if the user has never published inappropriate content.

SM The transaction where an admin removes a user from a group includes a field that contains the proof (i.e. the list of decrypted secret shares of the TDS signature) that indicate that the required number of group members has cooperated to de-anonymize the user.

Sybil attack

R An adversary may try to register many times in one group creating different digital identities. Then, the adversary may try to use these identities to outnumber the other members and publish inappropriate resources that cannot be de-anonymized or de-anonymize valid resources.

SM TIAs register real identities of users in a shared database before signing their DUIDs. If a user tries to register several DUIDs with the same or with different TIAs, the TIA involved in the process will reject the request of the user and will not sign a second DUID. With a single DUID, a user cannot register more than once in a group.

5.2. Discussion

This section evaluates and discusses in a general way the proposal of this paper. Firstly, we will evaluate the proposed system in terms of scalability and robustness. Secondly, we will discuss how our proposal tackles the open issues listed in Section 1.

Regarding scalability and robustness, the system is prepared to handle as many users, transactions and publications per second as the underlying technologies can support. This is so because the proposal divides the different publishing groups into completely different applications. Hence, there is no single point of the architecture common for all the groups. Conversely, the groups segment the users. In this way, neither the number of total users, publications or groups limits the proposal, but the number of users in a single group can indeed represent a problem, which is discussed below.

Nevertheless, the underlying technologies that enable the proposed architecture may represent problems for its scalability and viability. It is widely known that blockchain technology, and in particular Ethereum, has problems with scalability. Sometimes, transactions take too long to confirm and become costly at certain times of network congestion. However, it must be taken into account that for the proposal of this paper, Ethereum is only used to index the identities of the users in a decentralized way. Therefore, the users only need to send transactions to the blockchain to create groups and to register/remove members from them. These are sporadic actions that are not time-constrained and will not require users to spend much in fees. Regarding the publication of hypermedia content, IPFS becomes a possible bottleneck, since this system often takes minutes to publish resources. Shen, Li, Zhou, and Wang (2019) contains more information on the performance of IPFS. Anyways, this proposal is designed to guarantee the integrity and availability of hypermedia publications over time, so low latency when making a publication is not an important requirement at this time.

The previous paragraph has analyzed the proposal capacity to globally support users, transactions, and publications. However, it is also

necessary to analyze the scalability of the groups independently. In this sense, for the architecture presented in this paper, it remains as future work to carry out a study on the adequate number of members that each group should have. This study has to contemplate several dimensions. It must be taken into account that the number of members used to create TDS signatures and the threshold established to de-anonymize publishers directly influence the computing time and the size of the signatures. Besides, the minimum amount of users in a group to preserve k-anonymity at all times may vary depending on the unique characteristics of its members or their type of publications. Furthermore, in this paper, we have taken the underlying technologies as black-boxes. The security analysis of these technologies also remains as future work. The Ethereum ecosystem can be considered secure since it is being used by thousands of people and applications everyday and there are many security analysis of the protocol and its implementations. Bugs are eventually found and corrected. However, the TDS signature protocol must be evaluated in detail and, currently, it cannot be trusted. A formal security evaluation of TDS falls out of the scope of this paper and it is one of the first steps of the remaining future work. Subsequently, we plan to implement a proof of concept (PoC) of the proposed architecture and, then, comprehensively and methodically analyze the incentives of the actors introduced in this paper, and other potential players, to follow (or not) the proposed protocols. For this analysis, we plan to use a game-theoretical approach following the co-privacy (co-utility) theory (Domingo-Ferrer, 2010), as we did in Qureshi et al. (2020).

This paper has the objective of presenting a theoretical vision of the protocols for the proposed solution. However, with the implementation in the PoC, we will be able to empirically evaluate the performance and the scalability of the solution. For the purpose of this paper, we validate the performance of the key components separately with the following existing work:

- In Garcia-Font (2020), we implemented a PoC of an architecture for decentralized user-centric data management applications for communications in smart cities. The PoC allowed us to demonstrate the feasibility of using Ethereum and IPFS to create an online social networking platform, where data are stored and managed in a decentralized way. In a smart city scenario, this can reduce the dependency on service providers, and give the control of the personal data to the citizens.
- In Qureshi et al. (2020), we presented an emergency reporting system that uses TDS to create anonymous reports where the reporter gets rewarded in cryptocurrency if an incident management authority confirms that the reported emergency is true. Otherwise, the anonymity of the reporter can be revoked and he/she is punished in cryptocurrency. In this paper, we conducted a performance evaluation concerning the TDS scheme executed in three different mobile devices ((1) Samsung Galaxy S10, Android 9.0 OS, octa-core (4 1.78) GHz processor, 8 GB of RAM and 128 GB of internal storage; (2) Samsung A5, Android 8.0 OS, octa-core 1.9 GHz processor, 3 GB of RAM and 32 GB of internal storage; and (3) Samsung Galaxy Tab A7.0, Android 5.1.1, Quad-core 1.3 GHz core processor, 1.5 GB of RAM and 8 GB of internal storage). Among other issues, in this study, as shown in Fig. 5, we measured the required time to generate the TDS considering groups of 3, 8 and 10 members. As the figure shows, in the worst case scenario, one of these mobile devices takes below 25 s to generate the signature in a group of 10 members, which was considered as good enough for the emergency reporting system. Furthermore, it must be borne in mind that, unlike the emergency reporting system, the protocols proposed in this paper do not have time constraints and users do not necessarily need to use mobile devices.

Concerning censorship resistance, which is one of the main desired objectives of the proposed architecture, it should be noted that it is achieved thanks to the combination of the underlying technologies (i.e. Ethereum and IPFS). Both systems are highly distributed and are themselves considered censorship-resistant. In the proposed protocols, the entities that could potentially ban, block or, somehow, censor the users are the group admins and the TIAs. However, it is worth noting that the former do not have different rights than the rest of the members of a group concerning the possibility of censoring publications or de-anonymizing users. Although an Admin can remove a user from a group at any point, as discussed in Section 5.1, doing so would be immediately visible to all users in the group, who would lose trust on the Admin and would not have incentives to participate in the group in the future. For the latter, the TIAs, users may choose any TIA to validate their DUID. Hence, if a TIA does not properly perform its functions, it can easily be replaced by another TIA.

As we have seen, the proposed architecture allows the formation of groups that provide conditional anonymity to publishers. Our proposal achieves revocable k-anonymity relying on the cooperation of a certain amount of group members, unlike other schemes involving blockchain technology seen in Section 2, such as Li et al. (2022) and Lin et al. (2020) that use centralized authorities to trace and de-anonymize users, and Zhang and Ye (2022) that uses a permissioned blockchain and a centralized CA to issue and manage the users' certificates. Moreover, our proposal tackles the three general open issues listed in Section 1 related to content distribution systems providing k-anonymity: (1) users of communication systems providing full-anonymity contribute to providing anonymity for users that use the system to publish content that can be considered inadequate. Users should be able to decide only to cover users publishing content aligned to their moral standards. (2) Once some content is published, it is difficult to prevent its dissemination even if most of the users are against its distribution. (3) Publishers of illegal content remain hidden among other users of the system and cannot be exposed and brought to justice.

The system proposed in this paper tackles these three concerns. Regarding publishers' anonymity, (1), this proposal does not provide full-anonymity to the publishers. As long as enough members of the group do not cooperate to de-anonymize a publisher, the proposed system ensures that the publisher's identity remains k-anonymous, and not even prominent entities, like the TIAs or the group Admin, can expose the publisher. In this regard, it remains as future work to explore the benefits in terms of anonymity, verifiability and other additional functionalities of using other platforms to administer decentralized and self-sovereign identities, such as uPort (Lundkvist et al., 2017), iden3 (iden3, 2020) or Sovrin (Sovrin, 2018), combined with zero-knowledge proofs of knowledge.

Regarding (2), once the Content Denouncement Protocol has been successfully executed for a publication, any Delator can warn the rest of the group members about the denounced resource. This means that for an indeterminate period of time, until not enough Delators have participated in the protocol, a publication that does not follow the group manifest can be distributed and downloaded by several members of the group. However, once the protocol is executed, the publication will eventually be removed from the potential distributors. This cooperative approach reduces the amount of unwanted content that each member of the group will receive and download. Furthermore, registering identities and obtaining credentials from the TIAs and CIAs are time-consuming steps, and registering in a group involves paying Ethereum fees. Therefore, conditional k-anonymity and the threat of exclusion from the group should disincentivize malicious publishers beforehand, even to attempt to publish resources that do not infringe any laws, but that do not follow the group manifest.

Regarding (3), the proposed system has been designed to enable a way to expose and bring publishers of certain types of content to justice. The proposed protocols do not ensure that publishing illegal content does not become feasible. As mentioned before, these protocols

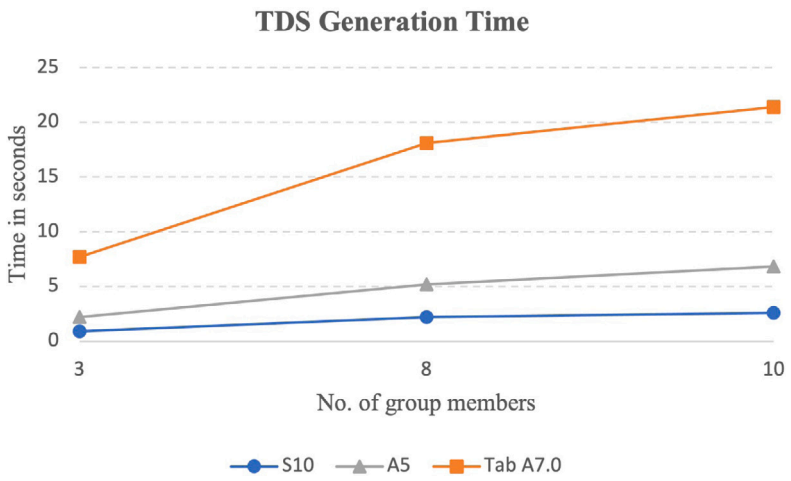

Fig. 5. TDS generation time.
Source: Qureshi et al. (2020).

disincentivize publishers to do so in a group that has a manifest forbidding such publication types. Nevertheless, as mentioned in Section 4.2, a group of users could potentially create a group with the goal of distributing offensive or illegal content. However, in this case, the rest of users of this protocol would not be contributing to hide the identities of the members of this group. Hence, the members of a group with a manifest that allows this type of activities can directly be considered as collaborators of hiding illegal or offensive actions. Therefore, these users would rather use a protocol that provides unconditional anonymity.

Finally, it is worth noting that this proposal has been designed bearing in mind to simplify as much as possible the issues related to cryptocurrencies. In the proposed protocols, the transactions with the blockchain are limited to creating groups and to registering/removing members. This fact, as mentioned above, not only narrows down the scalability and congestion problems due to current blockchain technological limitations but also reduces the fees that users have to pay. Besides, many other proposals for decentralized applications create custom tokens to deploy incentive mechanisms to ensure the proper behavior of the parties and to attract new users. The large number of tokens and their specificities in the blockchain space can be confusing for certain users and it definitely complicates the use of any new application, since this not only requires users to acquire or earn the new tokens but also that the new token gains trust among the public. For this proposal, the creation of custom tokens have been avoided, and ether is the only necessary token to interact with Ethereum.

6. Conclusions

Nowadays, there are many platforms to share hypermedia content on the Internet, which are built either following a client–server or a distributed model. The former is a model that has the typical problems of centralized architectures: single-points of failure, the possibility of data misuse, censorship, etc. In contrast, distributed models avoid some of these problems, but have to deal with certain issues related to privacy. Some of the distributed platforms, like many P2P systems, do not offer any mechanism to protect users' privacy and, therefore, it is easy to see information that can identify users (e.g. IP address) and the content that they are sharing. At the same time, some networks use privacy models that offer full-anonymity, like Tor. These allow participants to completely hide their identity among all users of the network, which makes all participants cooperators of hiding the identities of those disseminating illegal and offensive content.

In this paper, we propose an architecture for a censorship-resistant decentralized application to publish hypermedia content in a way that publishers have revocable k-anonymity. In the proposed system users join groups depending on the type of publications that they want to do or contribute to disseminate. When users create a group, they have to specify the rules that its members will enforce. Then, when a publisher follows the rules, the other members of the group contribute to disseminate his/her publication and the group is used to cover the individual identity of the publisher. Conversely, when a publisher does not follow the rules, the members can cooperate to de-anonymize the publisher. To enable this system, in this paper we propose an architecture based on Ethereum and IPFS to create decentralized digital identities and distribute content, and threshold discernible ring signatures to sign hypermedia content with conditional anonymity as explained above.

CRediT authorship contribution statement

Carlos Núñez-Gómez: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Victor Garcia-Font:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing.

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.

Acknowledgments

This work was supported by the Spanish Government under Grant RTI2018-095094-B-C22 "CONSENT" and by the Spanish State Research Agency under the project PEJ2018-003001-A.

References

Antonopoulos, A., & Wood, G. (2018). *Mastering ethereum* (1st ed.). O'Reilly Media, Inc, USA.

Benet, J. (2014). *IPFS-content addressed, versioned, P2P file system*: *Technical Report*, Retrieved from https://arxiv.org/pdf/1407.3561.pdf.

Braendgaard, P., & Torstensson, J. (2018). EIP: Ethereum lightweight identity. Retrieved from https://eips.ethereum.org/EIPS/eip-1056. (Accessed 24 June 2020).

Bresson, E., Stern, J., & Szydlo, M. (2002). Threshold ring signatures and applications to ad-hoc groups. In *Advances in cryptology — CRYPTO 2002* (pp. 465–480). Springer.

Buterin, V. (2014). *A next-generation smart contract and decentralized application platform*: *Technical Report*, Retrieved from https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_White_Paper_-_Buterin_2014.pdf.

Cachin, C., & Vukolić, M. (2017). Blockchain consensus protocols in the wild. (pp. 1–24). arXiv preprint. Retrieved from https://arxiv.org/pdf/1707.01873.pdf.

Camenisch, J. (1997). Efficient and generalized group signatures. In *International conference on the theory and applications of cryptographic techniques* (pp. 465–479). Springer.

Chaum, D., & Van Heyst, E. (1991). Group signatures, . In *Advances in cryptology - EUROCRYPT 91, volume 547* (pp. 257–265). Springer.

Clarke, I., Sandberg, O., Wiley, B., & Hong, T. W. (2001). Freenet: A distributed anonymous information storage and retrieval system. In *Designing privacy enhancing technologies* (pp. 46–66). Springer.

Cohen, B. (2001). BitTorrent. Retrieved from http://www.bittorrent.org. (Accessed 1 October 2019).

Diaspora Foundation (2021). diaspora*. Retrieved from https://diasporafoundation.org. (Accessed 21 September 2021).

Dingledine, R., Mathewson, N., & Syverson, P. (2004). Tor: The second-generation onion router. In *Proceedings of the 13th conference on USENIX security symposium - Volume 13* (p. 21). Berkeley, CA, USA: USENIX Association, Retrieved from http://dl.acm.org/citation.cfm?id=1251375.1251396.

Domingo-Ferrer, J. (2010). Coprivacy: Towards a theory of sustainable privacy. In *International conference on privacy in statistical databases* (pp. 258–268). Springer.

Durham, V. (2011). Namecoin. Retrieved from https://www.namecoin.org. (Accessed 1 October 2019).

Ethereum (2014). Whisper protocol. Retrieved from https://eth.wiki/concepts/whisper/whisper. (Accessed 17 January 2020).

Fediverse (2021). Fediverse. Retrieved from https://fediverse.party. (Accessed 21 September 2021).

Garcia-Font, V. (2020). SocialBlock: An architecture for decentralized user-centric data management applications for communications in smart cities. *Journal of Parallel and Distributed Computing*, *145*, 13–23.

Goldschlag, D., Reed, M., & Syverson, P. (1999). *Onion routing for anonymous and private internet connections*: *Technical Report*, Naval Research Lab Washington DC Center for High Assurance Computing Systems (CHACS).

Gramoli, V. (2020). From blockchain consensus back to Byzantine consensus. *Future Generation Computer Systems*, *107*, 760–769. http://dx.doi.org/10.1016/j.future.2017.09.023.

Guidi, B. (2020). When blockchain meets online social networks. *Pervasive and Mobile Computing*, *62*, Article 101131.

I2P (2003). I2P. Retrieved from https://geti2p.net. (Accessed 1 October 2019).

iden3 (2020). iden3. Retrieved from https://iden3.io. (Accessed 13 July 2020).

Klonowski, M., Krzywiecki, Ł., Kutyłowski, M., & Lauks, A. (2008). Step-out ring signatures. In *International symposium on mathematical foundations of computer science* (pp. 431–442). Springer.

Kocsis, T. (2015). ZeroNet. Retrieved from https://zeronet.io. (Accessed 1 October 2019).

Kumar, S., Agrawal, S., Venkatesan, R., Lokam, S. V., & Rangan, C. P. (2010). Threshold discernible ring signatures. In *International conference on e-business and telecommunications* (pp. 259–273). Springer.

Kumar, R., Marchang, N., & Tripathi, R. (2020). Distributed off-chain storage of patient diagnostic reports in healthcare system using IPFS and blockchain. In *2020 international conference on COMmunication Systems NETworkS* (pp. 1–5). http://dx.doi.org/10.1109/COMSNETS48256.2020.9027313.

Li, D., Han, D., Zheng, Z., Weng, T. -H., Li, H., Liu, H., et al. (2022). MOOCsChain: A blockchain-based secure storage and sharing scheme for MOOCs learning. *Computer Standards & Interfaces*, *81*, Article 103597. http://dx.doi.org/10.1016/j.csi.2021.103597.

Lin, C., He, D., Huang, X., Khan, M. K., & Choo, K. -K. R. (2020). DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain. *IEEE Transactions on Information Forensics and Security*, *15*, 2440–2452. http://dx.doi.org/10.1109/TIFS.2020.2969565.

Liu, Y., He, D., Obaidat, M. S., Kumar, N., Khan, M. K., Choo, K. -K. R., et al. (2020). Blockchain-based identity management systems: A review. *Journal of Network and Computer Applications*, Article 102731.

Liu, D. Y., Liu, J. K., Mu, Y., Susilo, W., & Wong, D. S. (2007). Revocable ring signature. *Journal of Computer Science and Technology*, *22*, 785–794.

Lundkvist, C., Heck, R., Torstensson, J., Mitton, Z., & Sena, M. (2017). uPort: A platform for self-sovereign identity. Retrieved from https://blockchainlab.com/pdf/uPort_whitepaper_DRAFT20161020.pdf. (Accessed 18 November 2019).

Mastodon GmbH (2021). Mastodon. Retrieved from https://joinmastodon.org. (Accessed 21 September 2021).

Moore, D., & Rid, T. (2016). Cryptopolitik and the darknet. *Survival*, *58*, 7–38.

Pandey, V., & Kulkarni, U. (2017). Effective data sharing with forward security: Identity based ring signature using different algorithms. In *2017 international conference on intelligent computing and control* (pp. 1–6). http://dx.doi.org/10.1109/I2C2.2017.8321818.

Peepeth (2021). Peepeth. Retrieved from https://peepeth.com. (Accessed 21 September 2021).

Politou, E., Alepis, E., Patsakis, C., Casino, F., & Alazab, M. (2020). Delegated content erasure in IPFS. *Future Generation Computer Systems*, *112*, 956–964. http://dx.doi.org/10.1016/j.future.2020.06.037.

Qureshi, A., Garcia-Font, V., Rifà-Pous, H., & Megías, D. (2020). Collaborative and efficient privacy-preserving critical incident management system. *Expert Systems with Applications*, Article 113727.

Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M., et al. (2020). Decentralized identifiers (DIDs) v1.0. Retrieved from https://www.w3.org/TR/did-core. (Accessed 5 February 2020).

Rivest, R. L., Shamir, A., & Tauman, Y. (2001). How to leak a secret. In *Advances in cryptology - ASIACRYPT 2001* (pp. 552–565). Springer Berlin Heidelberg.

Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression. *SRI International*, 1–19.

Sapien Network, Inc. (2021). Sapien network. Retrieved from https://www.sapien.network. (Accessed 21 September 2021).

Shao, W., Jia, C., Xu, Y., Qiu, K., Gao, Y., & He, Y. (2020). AttriChain: Decentralized traceable anonymous identities in privacy-preserving permissioned blockchain. *Computers & Security*, *99*, Article 102069. http://dx.doi.org/10.1016/j.cose.2020.102069.

Shen, J., Li, Y., Zhou, Y., & Wang, X. (2019). Understanding I/O performance of IPFS storage: A client's perspective. In *Proceedings of the international symposium on quality of service* (p. 17). ACM.

Shen, H., Liu, A. X., Liu, G., & Zhao, L. (2015). Freeweb: P2P-assisted collaborative censorship-resistant web browsing. *IEEE Transactions on Parallel and Distributed Systems*, *27*, 3226–3241.

SocialX Pte. Ltd (2019). SocialX. Retrieved from https://socialx.network. (Accessed 21 September 2021).

Sovrin (2018). Sovrin: A protocol and token for self-sovereign identity and decentralized trust. Retrieved from https://sovrin.org/wp-content/uploads/2018/03/Sovrin-Protocol-and-Token-White-Paper.pdf. (Accessed 18 November 2019).

Sporny, M., Longley, D., & Chadwick, D. (2019). Verifiable credentials data model v1.1. Retrieved from https://www.w3.org/TR/vc-data-model. (Accessed 10 June 2020).

Steemit (2021). Steemit. Retrieved from https://steemit.com. (Accessed 21 September 2021).

Su, Q., Zhang, R., Xue, R., & Li, P. (2020). Revocable attribute-based signature for blockchain-based healthcare system. *IEEE Access*, *8*, 127884–127896. http://dx.doi.org/10.1109/ACCESS.2020.3007691.

Van Saberhagen, N. (2013). CryptoNote v 2.0.

Van Saberhagen, N. (2014). Monero. Retrieved from https://www.getmonero.org. (Accessed 8 October 2019).

Wang, D., Zhao, J., & Wang, Y. (2020). A survey on privacy protection of blockchain: The technology and application. *IEEE Access*, *8*, 108766–108781. http://dx.doi.org/10.1109/ACCESS.2020.2994294.

Wilcox, Z. (2016). Zcash. Retrieved from https://z.cash. (Accessed 8 October 2019).

Xiong, H., Chen, Z., & Li, F. (2012). Bidder-anonymous English auction protocol based on revocable ring signature. *Expert Systems with Applications*, *39*, 7062–7066.

Zhang, X., & Ye, C. (2022). A novel privacy protection of permissioned blockchains with conditionally anonymous ring signature. *Cluster Computing*, 1–15. http://dx.doi.org/10.1007/s10586-021-03529-4.