Signet-ring: Authentic and Confidential Sharing of Digital Objects

Amangeet Samra, Amrita Mande, Catherine Jimerson, Diamond Rorie, Mahesh Arumugam

W233 Project Group #2

Abstract

In this paper, we investigate the authenticity and confidentiality aspects of sharing digital objects, especially digital photographs, in news publications, given the recent proliferation of fake news. We propose a system that provides: (1) robust confidentiality to the source of a digital object while allowing a publisher to authenticate the object through a trusted authority for authenticity (i.e., original and not fabricated beyond acceptable edits), (2) verify that the source is the owner of the object, (3) construct and maintain a *lineage* of the edits, and (4) allow a reader to verify that the published image is authentic and the lineage is protected.

I. Introduction

The rampant growth of malicious "deepfakes" in the media has created a need for an "anti-disinformation" solution [1]. Stories such as the *Grenfell Tower fire* [2], where posts of missing or deceased individuals were created with photos of social media celebrities or writers who were not actually in the vicinity of the incident, have taken the news by storm [3]. Unfortunately, there is no indication that incidents such as these will cease.

Deepfakes are just part of the problem. When it comes to reporting current events, journalists have confidential sources, but there is a constant push by consumers/readers (henceforth referenced as readers) who want to verify the origin of the sources in order to protect themselves from fake news or bad actors.

A possible solution would allow journalists/publishers to verify the origin of the digital objects their sources provide and allow sources to verify that their digital objects are shared with the intended journalist/publisher. This solution should implement a system with no agenda other than verifying the source of an object and maintaining an object's *lineage*. It would be uninterested in the content it verifies and tracks, providing some privacy to the sources who post within it. Furthermore, the system cannot prioritize one object over any other. Instead, it should allow sources and publishers to determine what is worth sharing while still tracking every change to the original object, thus making the ideal system unbiased. Furthermore, if a vendor, such as a phone manufacturer, were to design and implement such a system, they may be motivated by proprietary requirements and company interests than the universality of use and user privacy. For example, a Google app would be interested in collecting user data and biased towards some content, sources, or publishers.

Currently, an unbiased and uninterested system does not exist. To address this concern, we propose Signet-ring.[1, 2]. Signet-ring allows the sources and the publishers to verify each other. It also provides a mechanism that protects owner-anonymity when a reader verifies a published object. Furthermore, it takes the burden of verification off the readers. Instead, news outlets must provide information/digital objects that are verifiable.

Signet-ring provides robust confidentiality to the source of a digital object while still allowing a publisher to authenticate the source and the object through a trusted entity for authenticity. The protection applies to the digital objects created in a device connected to Signet-ring and the edits (e.g., adjusting lighting) made to that digital object. It verifies that the source is the owner of the object. Additionally, it allows a reader to verify that the published digital object is authentic and that its lineage is protected.

II. Background Survey

In our efforts to provide a means to verify information/images and ultimately tackle the deepfake problem that persists today, we present two use cases: the case for source confidentiality and the case for authenticity.

[1] *signet* .d: "a seal used officially to give personal authority to a document in lieu of signature" [4]

[2] *signet ring* .d: "a finger ring engraved with a signet, seal, or monogram" [5]

A. The Case for Confidentiality

News and media outlets excessively use "anonymous sources" to protect the confidentiality of outside party involvement in reports. However, contrary to popular belief, best journalistic practices and ethics state that source anonymity should only be used when necessary. These ethics deemed the identification of sources to provide the readers a way to gauge the source's credibility and the information [6]. Furthermore, those publications that inordinately cited "anonymous sources" were seen as lazy and undermined the legitimacy of such information gathering as a tool.

With the nearly limitless capabilities of the worldwide web, achieving source anonymity is almost impossible. The maturation and continued technological advancement consistently present the possibility that an individual's life could be (and has been) significantly and negatively impacted by perceived slights or affiliations with specific groups [7]). In this technological age, confidentiality and privacy have never been more prized, yet so easily subverted [7, 8]. Frameworks exist that provide sharing pathways where the pathways do not identify the users [9, 10]. Tangentially, we sought to create a system that provides confidentiality to the source (if desired or needed) while allowing the reporter or any other third party to authenticate the originality and volume of edits done to the piece of media (the primary use case being digital photographs).

B. The Case for Authenticity

The proliferation of "deepfakes" in the media has created a need for any third party to take a piece of media and use it to check the provenance thoroughly [1]. Specifically, in [1], a decentralized blockchain adds an object to its ledger after determining the hash of the discriminative features of the object (calculated using multiple LSTMs [11]). This encoding of digital objects into discriminative features is similar to ARCHANGEL [12], a decentralized blockchain-based system for guaranteeing the integrity of archives of digital objects. However, these approaches are specific to particular digital objects (e.g., video frames) and do not provide a mechanism for authenticating the users sharing a digital object.

Provenance and edit checking is available to experts in their related fields [13], but it may be unusable or impractical for journalists or, say, art experts who want to know that they are buying authentic digital art [14]. In addition, other mechanisms (including blockchain, e.g., [15]) address fake news using a verification framework that involves all the actors (the source, the publisher, and the reader). This work builds on the theoretical framework for building trust and curbing fake news [16, 17, 18]. More specifically, [15] proposes a blockchain-based framework that (1) allows publishers to distinguish between authentic sources and fake sources, (2) adopts a smart-contract [19] to publish news articles, and (3) ensures the integrity of published articles through the use of semantic similarity search (e.g., [20, 21]). Verification of a news article can be determined by searching the news stored in a Merkle tree [22].

III. Method

In this section, we present the architecture of Signet-ring, discuss the components involved in the system, and the various users (i.e., actors) interacting with the system. We also present the authentication workflow that allows a user to publish an object with a publisher and the verification workflow that allows a user to verify the authenticity of a published object.

A. Architecture

Signet-ring provides authentication and verification of digital objects created by a user (i.e., *owner*) using a trusted entity called *Trusted Authority* (TA). Any application that can create an object also registers and authenticates with TA (for example, a mobile camera app). Owners and publishers register and authenticate themselves with the TA. A reader can retrieve published objects from the publisher's portal, and may request TA to verify any published objects. Figure 1 shows the architecture of Signet-ring.

While this architecture is similar to [15], we note that Signet-ring provides a mechanism for ensuring that any content-generation application can be made authentic and integrated with the proposed framework. And, Signet-ring simplifies the mutual authentication of an owner and a publisher (without a blockchain).

1) Components of Signet-ring

The main components of Signet-ring are (1) Trusted Authority and (2) Authenticated App.

- *Trusted Authority (TA).* TA is similar to a Certificate Authority that issues and manages

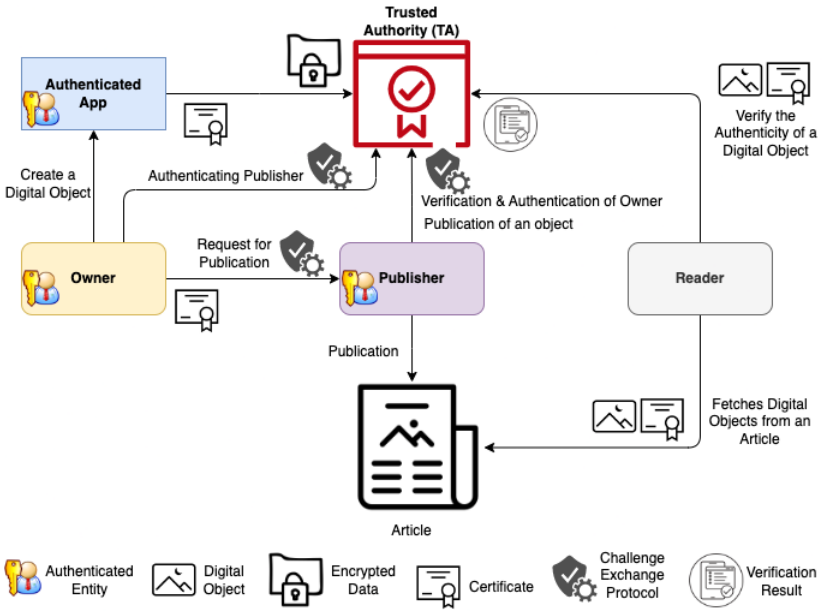

Figure 1: Architecture of Signet-ring

digital certificates to certify the ownership of public-key of named entities (e.g., website). Likewise, TA is responsible for issuing and certifying public-private keys to registered entities. TA is also responsible for issuing certificates to digital objects created by an owner using an authenticated app. Furthermore, TA maintains the lineage of edits to a digital object. TA allows the publisher and owner to mutually authenticate each other before publishing the owner's digital object. This process does not require any entity to share any confidential information explicitly. Finally, TA verifies the certificate of a published object.

- *Authenticated App.* Authenticated App creates a digital object on request from an owner. When the owner requests (e.g., *clicks*) the app to create an object, app submits the created object to the TA along with the credentials of the app and the owner for certification.

2) Actors of Signet-ring

The main actors of Signet-ring are owner, publisher, and reader.

- *Owner.* Owner is an authenticated user of TA and is responsible for the creation (or edits) of a digital object and publishing the digital object with a publisher.
- *Publisher.* Publisher is an authenticated user of TA and is responsible for publishing a digital object in its portal. In addition, the publisher makes the certificate available for all published images in the portal.
- *Reader.* Reader can verify the published object's authenticity and lineage with the TA.

B. Workflows

In this section, we discuss the primary workflows of Signet-ring.

1) Creating a Digital Object

Figure 2 shows the creation workflow. Owner requests authenticated app to create a new digital object (or save the edits performed on an existing digital object). App then sends a certificate request to TA with the name of the object, its payload (i.e., contents), and the identity of the owner. If the object has lineage (i.e., the object is edited from an existing object), the app includes the identity of the parent object and the certificate of the parent object in the request. TA verifies authentication of the app and the owner. Subsequently, if the lineage is part of the message, TA verifies the certificate of the parent object. Finally, TA issues a certificate for the object.

2) Publishing a Digital Object

To publish a digital object, owner and publisher execute a challenge protocol modeled based on

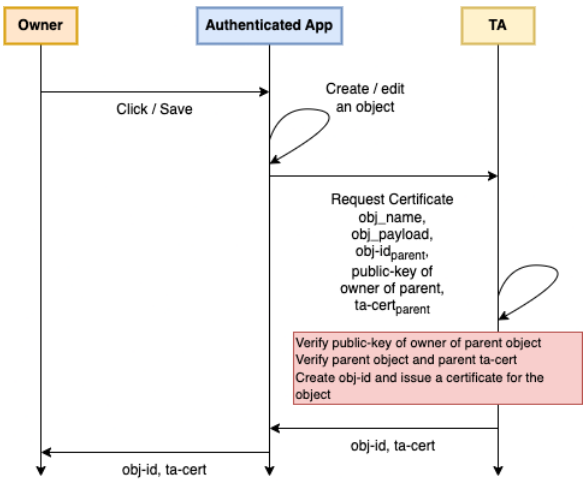

Figure 2: Creation of a digital object

the Diffie-Hellman Key Exchange Protocol [23].[3] The protocol ensures mutual authentication of the owner and publisher before the publication of the digital object.

Step 1: Challenge Creation: An owner initiates a challenge with the TA when they decide to share an object. The challenge includes a random secret (*challenge-text1*) that is encrypted with the publisher's public key. TA returns an ID for the challenge as shown in Figure 3.

Figure 3: Challenge creation

Step 2: Challenge Exchange: As shown in Figure 4, owner sends the challenge ID to the publisher. Publisher retrieves the challenge from the TA and decrypts the secret using the publisher's private key. Then, it creates a new random secret (*challenge-text2*) and appends it to the decrypted secret. Subsequently, it encrypts the combined secret with the owner's public key. Publisher updates the combined secret at the TA and replies to the owner with an opaque value that the owner should return to the publisher for verification. This opaque value contains the value of the publisher's secret that is encrypted using the publisher's public key. (As a result, only the publisher can decrypt this value.)

[3]We would like to note that there are many different implementations possible for the challenge protocol, including the implementation where the owner and publisher do not communicate directly. In our design, we let the owner and the publisher securely communicate while participating in the challenge protocol to authenticate each other mutually.

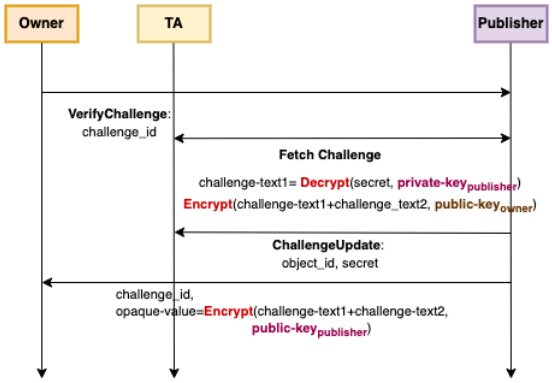

Figure 4: Challenge exchange

Step 3: Challenge Verification: In the final step (cf. Figure 5), owner decrypts the secret in the challenge using its private key. Owner authenticates publisher if it sees *challenge-text1* in the message. When owner decrypts the secret, it also gets the text that publisher added in Step 2. To complete the authentication process, owner sends a message to the publisher that it accepted the challenge along with both the secrets and publisher's opaque value. Publisher decrypts the opaque value and retrieves the original secrets. If the secrets match the texts in the message, publisher authenticates owner.

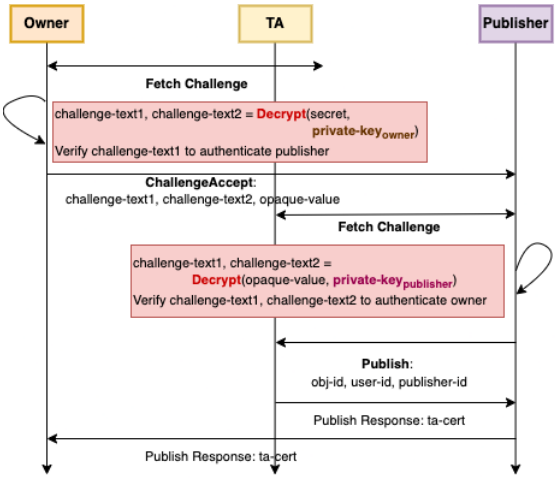

Figure 5: Challenge verification

On successful authentication, publisher requests the TA to publish the object. The TA validates provided information for publication and issues a signed certificate along with the owner's and publisher's public keys. Finally, the publisher publishes the certified object in its portal.

3) Verification of a Published Object

A random reader of the publisher's portal can request verification of the digital objects published in the portal with the TA. Figure 6 outlines the verification process. First, TA validates the digital signature of the certificate. Subsequently, TA verifies that the public keys of the owner and publisher match. Finally, if the object has a lineage, TA ensures the lineage is authentic and returns the validation result to the reader. Thus, Signet-ring lets a reader authenticate and verify the published objects without disclosing the owner and any attributes associated with the owner.

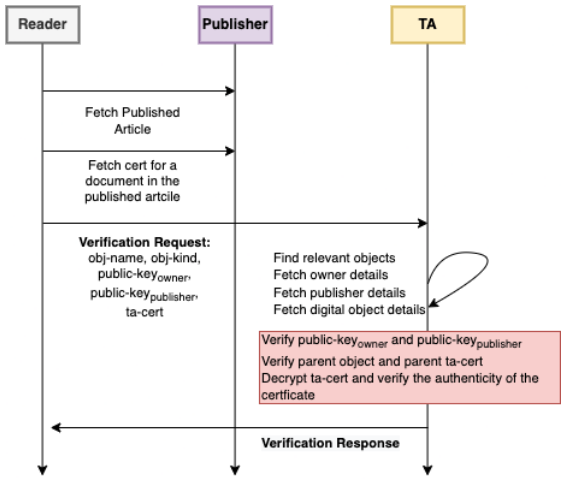

Figure 6: Verification of a published object

IV. Results

In this section, we discuss a proof-of-concept implementation of Signet-ring and our findings.

A. Implementation

We implemented a proof-of-concept (POC) system of Signet-ring.[4] We modeled the various components and actors of Signet-ring using Python FastAPI [24] web services that expose HTTP REST [25] endpoints and use a PostgreSQL as the backend database.

[4]The implementation of this architecture is available at https://github.com/aumahesh-mids/signetring.

1) Trusted Authority Application[5]

We implemented TA as a Python FastAPI web server that exposes the following REST APIs.

- *apps*: Register a new app and get a list of registered apps.
- *user*: Register a new user and get a list of registered users.
- *objects*: Request a certificate for a new object, find the lineage of an object, and publish a certified digital object.
- *publication*: Implement the challenge protocol discussed in Section III-B2.
- *verification*: Verify the certificate of a published object.

Note that we have implemented additional APIs that are not listed above (for brevity).

2) User[6]

The user application exposes endpoints to initialize an owner or a publisher and register with the TA. For an owner, the user application exposes REST endpoints that: (1) create a new digital object (by forwarding the request to the source app) and (2) initiate the challenge protocol for publication of the object. For a publisher, it exposes a REST endpoint to trigger a challenge for the publication of an object.

3) Source App[7]

We implemented the source app as a server that exposes a REST endpoint to initialize the app (type of the application) and register it with the TA. When the owner submits a request to create a new object, the user app invokes a REST API on the source app that triggers the creation and submission of the object to the TA.

B. Findings

We measured success based on the following criteria: (1) verification of published objects, (2) maintenance of accurate lineages for each digital object, and (3) robustness against imitation attacks.

We know that some users will try to subvert the system, for example, by taking pictures of pictures or trying to represent other users. The

[5]OpenAPI specification for TA is available at: https://app.swaggerhub.com/apis/AUMAHESHMIDS/signet-ta/0.1.0

[6]OpenAPI specification for user application is available at: https://app.swaggerhub.com/apis/AUMAHESHMIDS/user/0.1.0

[7]OpenAPI specification for source app application is available at: https://app.swaggerhub.com/apis/AUMAHESHMIDS/app/0.1.0

challenge protocol protects against the imitation of users and publishers.

Signet-ring certifies the authenticity of digital objects upon creation (e.g., clicking a photograph). Those certifications are viewable by the publisher, the owner, and any reader. Signet-ring also tracks all edits made by both the owner and any other users in our system, which is essential in providing another layer of authenticity as a reader can track the edited object back to the original. Note that a reader cannot track the published object back to its owner.

A challenge protocol that allows mutual verification of owners and publishers was added to the scope of the work to deal with imitation attacks. We believe our design and the POC implementation met the goals. We discuss several improvements and enhancements to our design (and our implementation) in the next section.

V. Discussion

A. Staged Objects

Signet-ring ensures that when the owner creates an object through an app, the app submits the object to the TA and requests a certificate. The action of the creation of an object is the trigger for the request for a certificate. However, the architecture does not control the environment at the time of the creation of the object. As an example, consider the process of clicking a photograph using a camera app. When the user clicks, the app shoots the image and submits it to the TA for a certificate. However, the environment/context where the image was shot is beyond the purview of Signet-ring. Signet-ring only guarantees that an object submitted to TA and published by a publisher is authentic, i.e., it is certified at the time of the creation of the object.

B. Trustworthiness of Applications

Staged objects raised the question about how to deal with applications or owners that are not trustworthy. Consider the following scenario. A third-party authenticated photo editing software submits an edited version of an already certified digital object without providing lineage. To deal with this problem, we propose using semantic similarity hashing (SSHash) [20, 21], similar to architecture proposed in [15]. Specifically, the TA runs a semantic similarity search across the objects it has already certified. In the case where an existing certified object has a significant similarity match (above some predefined threshold) with the submitted object, TA flags the object as a violation and does not issue a certificate.

C. Verification of Real Identities

When the TA registers a user, it has to ensure that the user is whom they claim to be. For example, consider a user who registers with the TA as CNN or Fox News. TA should not accept the request for registration without proper verification. Verifying the real identity is beyond the scope of this work and is usually a manual or a semi-automated process through verification vendors. Moreover, this verification of real identity is similar to verifying subject names by a certificate authority when a subject requests a digital certificate [26]. Also, this is similar to how various social media platforms verify users' identities and add a verified icon to their profiles (e.g., Twitter Blue [27]). Therefore, TA has to verify a user's real identity before accepting their registration request.

D. Anonymity in Challenge Protocol

As mentioned in Section III-B2, the challenge protocol can be implemented in many different ways. In our implementation, for quick prototyping and demonstration of our approach, we let the owner communicate with the publisher directly during the challenge exchange protocol. Instead, we can implement the challenge protocol such that the owner submits the challenge request for the TA to manage. Specifically, the owner initiates the challenge, and the TA executes the challenge by forwarding the challenge to the requested publisher. The owner and the publisher do not communicate directly and do not know the other party's public key. Instead, TA manages the whole protocol and sends only the result to the respective parties. Thus, it is possible to publish a digital object anonymously.

E. Anonymity Everywhere

In the current design, the owner of a digital object has to authenticate with the TA to get a certificate and publish the object. One obvious question is whether it is possible to protect the identity of the owner from the TA itself. One approach is to create virtual entities every time a digital object is created. The virtual entities include a virtual app and a virtual user, with no traceability to the original app and the owner. Nevertheless, registration and verification of such virtual entities

is an open question. Current architecture requires the TA to verify and authenticate the app and the owner. We do not yet know if the anonymous creation and publication of digital objects are possible. And this question is beyond the scope of this work.

F. Inference Threats

All communication in Signet-ring is encrypted with appropriate keys negotiated as part of the TLS exchange [28]. However, inference threats on encrypted traffic are still possible (as studied in [29, 30]). As a result, it is possible to make some inferences based on the communication between the various components and users of Signet-ring. To minimize such attacks, we propose to extend Signet-ring to implement the anonymity-preserving challenge protocol for publishing a digital object, as discussed in Section V-D. However, a third party can still make inferences by just observing communication patterns between the owner and the TA, or the TA and the publisher. For example, an attacker could correlate the sequence of communication to infer that a particular owner is trying to publish an object with a particular publisher. To address this issue, TA will bulk challenge messages for a publisher and dispatch them at a fixed time of the day, minimizing the correlation. In addition, we note that a given user, app, and TA communicate for various reasons (authentication, creation, verification, publication) over a channel created using a negotiated symmetric key between the respective parties. Thus, the possibility of any kind of inference is very low.

G. Signet-ring in Production

The POC implementation discussed in Section IV is very limited. To bring Signet-ring to production, we have the following options:

- *End-to-end ecosystem.* We design and develop all components of the architecture. We would invest in the development of various content-creation applications such as camera and word processor. However, there are existing applications such as the native mobile camera app with huge adoption. As a result, getting consumers to adopt native Signet-ring content-generation applications will be the main challenge.
- *Connectors.* We design connectors for existing third-party content-generation applications. Connectors allow Signet-ring to provide an unbiased and uninterested privacy-centric framework for such applications. Furthermore, to increase awareness of consumers to use applications that have connectors for Signet-ring, we plan to work with popular App Stores (e.g., Apple App Store, Google Play Store) to add a seal of approval (e.g., *Verified by Signet-ring*) to such applications.

VI. Conclusion

In this paper, we presented Signet-ring, an unbiased and uninterested system for sharing digital objects that ensures authenticity and preserves the confidentiality of the sources. We implemented a proof-of-concept (POC) system and showed how a challenge protocol (like the Diffie-Hellman Key Exchange [23]) could provide mutual authentication for owners and publishers. Our POC certified digital objects upon creation and preserved their lineage. While there are circumstances outside the scope of Signet-ring that allow fake objects to be created, Signet-ring provides the framework for authentic and confidential sharing of digital objects. We discussed several open problems and opportunities for enhancements in Section V. Bringing Signet-ring to production by integrating with third-party applications using connectors is the essential first step towards our goal.

References

[1] Christopher Chun Ki Chan, Vimal Kumar, Steven Delaney, and Munkhjargal Gochoo. Combating deepfakes: Multi-lstm and blockchain as proof of authenticity for digital media. In *2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G)*, pages 55–62. IEEE, 2020.

[2] Wikipedia contributors. Grenfell tower fire — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Grenfell_Tower_fire&oldid=1125574144, 2022. [Online; accessed 8-December-2022].

[3] Bharati Bharali and Anupa Lahkar. Fake news: Credibility, cultivation syndrome and the new age media. *Media Watch*, 9, 03 2018.

[4] Merriam-Webster. Signet. https://www.merriam-webster.com/dictionary/signet. [Online; accessed 30-November-2022].

[5] Merriam-Webster. Signet ring. https://www.merriam-webster.com/dictionary/signet%20ring. [Online; accessed 30-November-2022].

[6] K. Tim Wulfemeyer. Use of anonymous sources in journalism. *Newspaper Research Journal*, 4(2):43–50, 1983.

[7] Svana Calabro. From the message board to the front door: Addressing the offline consequences of race-and

gender-based doxxing and swatting. *Suffolk UL Rev.*, 51:55, 2018.

[8] Laura Durity. Shielding journalist-"bloggers": The need to protect newsgathering despite the distribution medium. *Duke L. & Tech. Rev.*, 5:1, 2005.

[9] Stefan Contiu, Sébastien Vaucher, Rafael Pires, Marcelo Pasin, Pascal Felber, and Laurent Réveillère. Anonymous and confidential file sharing over untrusted clouds. In *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, pages 21–2110, 2019.

[10] Danan Thilakanathan, Rafael Calvo, Shiping Chen, and Surya Nepal. Secure and controlled sharing of data in distributed computing. In *2013 IEEE 16th International Conference on Computational Science and Engineering*, pages 825–832, 2013.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[12] Tu Bui, Daniel Cooper, John P. Collomosse, Mark Bell, Alex Green, John Sheridan, Jez Higgins, Arindra Das, Jared Keller, Olivier Thereaux, and Alan W. Brown. Archangel: Tamper-proofing video archives using temporal content hashes on the blockchain. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2793–2801, 2019.

[13] Alan J Cooper. Detecting butt-spliced edits in forensic digital audio recordings. In *Audio Engineering Society Conference: 39th International Conference: Audio Forensics: Practices and Challenges*. Audio Engineering Society, 2010.

[14] Qin Wang, Rujia Li, Qi Wang, and Shiping Chen. Non-fungible token (nft): Overview, evaluation, opportunities and challenges. *arXiv preprint arXiv:2105.07447*, 2021.

[15] Adnan Qayyum, Junaid Qadir, Muhammad Umar Janjua, and Falak Sher. Using blockchain to rein in the new post-truth world and check the spread of fake news. *IT Professional*, 21(4):16–24, 2019.

[16] S. Huckle and M. White. Fake news: A technological approach to proving the origins of content, using blockchains. *Big Data*, 5(4):356–371, 2017.

[17] Wenqian Shang, Mengyu Liu, Weiguo Lin, and Minzheng Jia. Tracing the source of news based on blockchain. *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pages 377–381, 2018.

[18] Tee Wee Jing and Raja Kumar Murugesan. A theoretical framework to build trust and prevent fake news in social media using blockchain. In Faisal Saeed, Nadhmi Gazem, Fathey Mohammed, and Abdelsalam Busalim, editors, *Recent Trends in Data Science and Soft Computing*, pages 955–962, Cham, 2019. Springer International Publishing.

[19] Alexander Savelyev. Contract Law 2.0: *Smart* Contracts As the Beginning of the End of Classic Contract Law. *Higher School of Economics Research Paper*, (WP BRP 71/LAW/2016), 2016.

[20] Giulio Ermanno Pibiri. Sparse and skew hashing of K-mers. *Bioinformatics*, 38(Supplement_1):i185–i194, 06 2022.

[21] Jiaming Xu, Pengcheng Liu, Gaowei Wu, Zhengya Sun, Bo Xu, and Hongwei Hao. A fast matching method based on semantic similarity for short texts. In Guodong Zhou, Juanzi Li, Dongyan Zhao, and Yansong Feng, editors, *Natural Language Processing and Chinese Computing*, pages 299–309, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[22] R. C. merkle. A digital signature based on conventional encryption function. *Advances in Cryptology (CRYPTO '87)*, 293:369–378, 1988.

[23] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[24] FastAPI. https://fastapi.tiangolo.com/.

[25] Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly Media, Inc., 2007.

[26] Medha Mehta. Understanding the SSL Validation Process with FAQs. https://sectigostore.com/blog/understanding-the-ssl-validation-process-with-faqs/. [Online; accessed 8-December-2022].

[27] Twitter. Legacy verification policy. https://help.twitter.com/en/managing-your-account/legacy-verification-policy. [Online; accessed 8-December-2022].

[28] T. Dierks and E. Rescorla. The transport layer security (tls) protocol version 1.2. RFC 5246, RFC Editor, August 2008. http://www.rfc-editor.org/rfc/rfc5246.txt.

[29] Charles V. Wright, Lucas Ballard, Fabian Monrose, and Gerald M. Masson. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, USA, 2007. USENIX Association.

[30] Brad Miller, Ling Huang, A. D. Joseph, and J. D. Tygar. I know why you went to the clinic: Risks and realization of https traffic analysis. In Emiliano De Cristofaro and Steven J. Murdoch, editors, *Privacy Enhancing Technologies*, pages 143–163, Cham, 2014. Springer International Publishing.

Contributions of the Team

Amangeet Samra: She helped code the FastAPI implementation of user (with Catherine) and source app applications. She wrote Section V section of the paper (with Mahesh). She put-together the slide-deck for the project presentation. She edited the final paper for grammar, correctness and style.

Amrita Mande: She worked on the writeup for Section III (with Mahesh). She contributed to the overall architecture of Signet-ring and the high-level design of the proof-of-concept implementation. She identified various scenarios to break/attack the system.

Catherine Jimerson: She helped code the FastAPI implementation of user application (with Geet). She worked on the writeup for Sections I (with Diamond) and IV. She contributed to the animation of the slide-deck and fine-tuned the intuitive explanation of the methods.

Diamond Rorie: She did the research/reading for literature review. She worked on the introduction (Section I) with Catherine and literature survey (Section II) of the paper. She helped finalize the slide-deck.

Mahesh Arumugam: He worked on proof-of-concept implementation of Signet-ring. He designed the POC system and developed the story for the demo. Mahesh coded the FastAPI implementation of TA. In addition, He helped write Section III (with Amrita) and Section V (with Geet). And, he helped typeset the paper in LaTeX.