AAA: A blockchain-based architecture for ethical, robust, and authenticated anonymity

Luca Sciullo
luca.sciullo@unibo.it
University of Bologna, Department of Computer Science and Engineering
Bologna, Italy

Alberto De Marchi
alberto.demarchi@unibw.de
University of the Bundeswehr Munich, Department of Aerospace Engineering
Neubiberg, Germany

Lorenzo Gigli
lorenzo.gigli@unibo.it
University of Bologna, Department of Computer Science and Engineering
Bologna, Italy

Monica Palmirani
monica.palmirani@unibo.it
University of Bologna, CIRSFID
Bologna, Italy

Fabio Vitali
fabio.vitali@unibo.it
University of Bologna, Department of Computer Science and Engineering
Bologna, Italy

ABSTRACT

In the past years, online anonymity has attracted strong criticism for its role in shielding online crimes such as cyberbullying, fake news, money laundering, and pedo-pornography. Yet, it also has historically strong supporters who emphasize the necessity of a safe haven for carrying out legal and ethical activities that should not be associated with our real-life personas. We define authenticated anonymity as the possibility of using anonymous accounts that cannot be associated with the real identity of their owner unless a criminal act is being performed through them. Blockchain technology represents a good means for managing this complexity in a secure and trustworthy manner. Several solutions exist in the literature and on the market for anonymous identity, but they confer too much power to their owners, who can decide what to reveal about themselves in total autonomy (self-sovereign identities). In this paper, we present the Authenticated Anonymity Architecture (AAA), a blockchain-based solution for creating authenticated anonymous identities, where the mappings between official and anonymous identities can only be revealed after the necessary consensus of multiple different actors on the blockchain, evaluating the appropriateness and ethicality of the request. We mathematically modeled the architecture and conducted some analytical evaluations, showing that our proposal is resilient and fault-tolerant, even in the case of a huge number of identities managed.

CCS CONCEPTS

• **Mathematics of computing** → *Mathematical analysis*; • **Information systems** → **Collaborative and social computing systems and tools**; • **Security and privacy** → **Human and societal aspects of security and privacy**.

GoodIT '24, September 4–6, 2024, Bremen, Germany

ACM ISBN 979-8-4007-1094-0/24/09
https://doi.org/10.1145/3677525.3678676

KEYWORDS

online anonymity, self-sovereign identity, cyberbullying, blockchain-based identity, blockchain

ACM Reference Format:
Luca Sciullo, Alberto De Marchi, Lorenzo Gigli, Monica Palmirani, and Fabio Vitali. 2024. AAA: A blockchain-based architecture for ethical, robust, and authenticated anonymity. In *International Conference on Information Technology for Social Good (GoodIT '24), September 4–6, 2024, Bremen, Germany.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677525.3678676

1 INTRODUCTION

Online anonymity is the ability to act online without revealing our real-life identity, location, and personal details, including any other anonymous identity we may be using simultaneously in other contexts or for other purposes. Online anonymity can be seen as the opposite of (state-issued) digital identities [2], which are identification codes assigned to individuals and companies (possibly by the government) to allow them full access to public and private services online as themselves. Online anonymity is currently a hot discussion topic in various disciplines. For its supporters, it provides a last line of defense against unjust retaliation against individuals who just wish to express themselves and live the life they would like to live. Online anonymity may prevent societal pressures (in the family, workplace, school, neighborhood, etc.), data misuse by online service providers (including connectivity), abuses by unscrupulous law enforcement agents, and oppressive policies from authoritarian states trying to block the free expression of ourselves and our ideas. Yet, at the same time, online anonymity provides criminals with a shield to carry out their deeds undisturbed, leaving law enforcement agencies with just the hope that they commit identifiable mistakes or choose weak technical solutions for their anonymous identities. Of course, waiting for criminals to make technical mistakes is not a reliable method for the prosecution of crimes. Indeed, many different architectures for robust anonymous identities are starting to appear, such as DIDs [22] or ESSIF [3], making such mistakes easier to avoid.

To summarize, we are now at an ideological turning point, with the political discourse neatly divided between strong defenders and

strong attackers of anonymity online, and technical tools being readied for strong decentralized anonymous online identities attracting both legitimate and criminal acts. All in all, vastly different ideological points of view seem to imply that there is limited interest in finding a middle ground that can satisfy both parts. Yet, such middle ground would have a fairly simple characterization: we should be seeking to design *an architecture to provide robust anonymity for all legal uses and yet rapid and precise identification of anonymous individuals responsible for illegal acts, with safeguards against governmental overreach.*

In this paper, we suggest such a possible middle ground by proposing the *Authenticated Anonymity Architecture (AAA)*, a blockchain-based architecture that:

(1) connects public and anonymous identities securely and robustly for all legitimate uses;
(2) provides a swift and reliable way to deanonymize users committing crimes;
(3) guarantees that deanonymization can only be performed by properly authorized actors;
(4) ensures that deanonymization is conducted in a traceable, transparent, dated, authored, and non-repudiable manner to prevent or limit abuses;
(5) allows deanonymization only if a necessary consensus is reached among participating organizations, preventing unethical deanonymizations by "evil" actors.

We envision this architecture to be used by a network of national agencies (UIP, or *Union of Identity Providers*) whose members, after agreeing on technical and ethical baselines, ensure in a robust way that only legal requests for deanonymizations are fulfilled, that law enforcement agencies cannot access users' data covertly, and that evil actors (e.g., coalitions of oppressive countries) can be blocked from deanonymizing political opponents. This is achieved through the systematic use of *authenticated anonymous* identifiers—anonymous accounts that are connected to the real identity of their owners in a manner that is robustly concealed until a criminal act is being performed through them. Given the sensitivity of the topic, we have also formally examined possible attack vectors against the architecture, e.g., coalitions of participating evil actors (i.e., groups of oppressive countries) trying to force an unethical deanonymization (*evil attacks*) or to block an ethical deanonymization, using it as a ransom for unethical means (*node faults*).

In the rest of the paper, we expand on this vision according to the following structure: in Section 2, we discuss the literature about blockchain identities and blockchain-based anonymity. In Section 3, we provide a lengthier discussion of the motivations behind this work. Section 4 describes the architectural details of AAA, with its analytical modeling in Section 5. Evaluations in Section 6 show the theoretical robustness of our architecture against the malpractice of one or more evil nodes in the UIP intent on disrupting the smooth working of the deanonymization practices. In Section 7, we draw some conclusions and provide some indications of our future activities in this direction.

2 RELATED WORK

Literature on anonymity in blockchain scenarios can be divided into (i) works that explore potential solutions to use blockchain with anonymous identities but with the possibility to account transactions to a specific person, (ii) works that tackle the problem of managing resource access control for anonymous accounts, and (iii) works that leverage blockchain mechanisms for storing the association between anonymous identities and real identities, providing external services the possibility to securely verify the existence of the mapping and allowing users to use such services anonymously.

In the first category, the authors of [7] propose a novel design principle for identity management in blockchain, whose goal is to preserve privacy while ensuring compliance with current regulations and preventing the misuse of blockchain technology for purposes that are contrary to the social good. This is possible thanks to a custom *identity layer* that utilizes cryptographic mechanisms for implementing provably secure protocols that let only authorized parties retrieve the identity behind an account, given the transactions on the blockchain. Similarly, AttriChain [20] is a framework that enables users to interact with the network using identities that are both anonymous and traceable within a permissioned blockchain. It leverages an attribute-based signature system that includes threshold/distributed tag-based encryption for transaction tracing, signatures to ensure unforgeability, and zero-knowledge proofs to maintain anonymity. Also, in [6], the authors work to overcome the native pseudo-anonymity of blockchain for supporting identity-aware applications. They propose a mechanism that mixes public digital identities with Identity-Based Encryption (IBE), which provides a direct connection between cryptographic keys and the relative identity used for signing a transaction.

For the second research direction, the authors of [16] and [14] propose a blockchain solution for access control that supports anonymity and accountability, with the latter delving into data sharing for multiple groups. More in detail, the first propose a protocol for obtaining a verified anonymous identity by sharing the real one only with an Identity Provider in a preliminary phase; the anonymous identity can then query an Access Control Provider to obtain the required rights for accessing resources anonymously. All the operations performed by the anonymous identity are written on the blockchain, giving the Identity Provider the ability to account for these operations to a real identity. The latter propose a mathematical model for enabling multiple groups to share data among each other, leveraging the consortium blockchain technique without storing data in the cloud and excluding third-party audits for verifying what is stored in the cloud.

In the third set of papers, ChainAnchor [13] is an architecture that adds an identity and privacy-preserving layer onto the Bitcoin blockchain, which can hence be considered semi-permissioned, leveraging zero-knowledge proof mechanisms. Everyone can read and verify the transactions, but only verified anonymous identities can write on the blockchain. A user binds her transaction public key to the zero-knowledge proof sent to the Permissions Verifier, creating an anonymous verified identity that can now ask the Permissions Issuer to make a transaction. Working in the same direction, [5] proposes a mechanism for obtaining verified anonymous identities starting from a face-to-face initial proof that can then be

validated against a record on the Bitcoin blockchain. Nevertheless, instead of storing a simple hash that could easily suffer from privacy attacks, the authors use a scheme by Brands to store a commitment against which it is possible to perform zero-knowledge proofs of identity.

It is worth highlighting that, compared to the solutions presented in this section, our proposal is orthogonal to the blockchain used (we exploit the most common features of a blockchain, i.e., a public, secure, and distributed repository), so it does not depend on a specific implementation and does not require any additional layer. Furthermore, our architecture enables the creation of a large number of anonymous authenticated identities connected to a single real identity, increasing privacy and making the system less vulnerable to statistical attacks. Finally, we leverage the decentralization of the blockchain to reveal the real identity behind an anonymous one, making it impossible for a single authority to compromise the anonymity of an account without shared consensus.

3 MOTIVATIONS

As a first approximation, we can define online identity as a digital code that is associated with and singles out a unique individual, and online anonymity as the ability for an individual to act online without being recognized. Yet, the precise meaning of these definitions is complicated to address. Anonymity is often seen as whatever prevents the association of multiple activities with the same person, and pseudonymity is meant for identification without reference to real-life identity (e.g., [19] or [23]). In legislative circles (e.g., [15] and [9]), the key aspect of anonymity is associated with the risk of re-identification of individuals: anonymous data have been *irreversibly* anonymized, forcing the risk of re-identification to zero, while pseudonymization is meant to mitigate non-zero risks (and is subject to legislation such as GDPR). In this paper, on the contrary, we adopt a view of identity and anonymity based on attitude rather than efficacy: on the one hand, we see online identity as *a means to convey the idea of a unique individual*. Even if an account is manned by a team of individuals in a troll factory, the expectation is that we understand it as coming from a single person. Also, we see anonymous identities as those which *are not expected* to be associatable to a real-life identity. Even though some methods are more effective than others in cloaking identities, they were all adopted with the purpose of hiding the real identity of their owners. Fantasy account names in a dating app are as much anonymous as strong DID identifiers: what changes is their robustness, not their justification.

For its defenders, anonymity is seen as the sole defense of individuals from the overreach of powerful third parties:

- **Society overreach**, shielding us from members of our social circles (spouse, family, friends, employers, clergy, etc.) finding our life choices objectionable and possibly retaliating against us.
- **Infrastructure overreach**, shielding us from Internet infrastructure (e.g., connectivity, account providers, social networks and discussion boards) using data about us for undisclosed and possibly unethical commerce both in aggregate form and on us as individuals.
- **Law enforcement overreach**: although most, if not all, liberal democracies have constitutional-level limits to indiscriminate police investigations (e.g., by requiring a properly obtained warrant), excessive practices from law enforcement agencies are widespread and growing [17].
- **State overreach**: oppressive countries use laws to curb dissent and enforce strict control over online activities. Importantly, such governments usually act "legally", i.e., the country's laws themselves are fashioned so as to make such prosecutions formally legal.

For its critics, on the other hand, anonymity provides an impenetrable shield to the proper prosecution of criminals carrying out objectively horrific and hideous crimes, leaving them outside the reach of the law:

- **Crimes targeting individuals**: hate-driven speech, intimidatory content, cyberbullying, or *impersonation* (i.e., pretending to be someone else): online scams, romantic or sexual catfishing, or impersonating public figures.
- **Crimes targeting information channels**: false data, fake news, and constructed outrage flood social networks and individual mailboxes, expecting recipients not to check the truth of their content given their limited available time and competency, with serious consequences for public discourse, politics, and social stability.
- **Carrying out illegal activities**: illegal commerce and activities use anonymity for personal profit, e.g., drug trafficking, money laundering, terrorism, child abuse (including child pornography), etc.

Online anonymity attracts very polarized points of view, often orthogonal to traditional progressive vs. conservative political positions. For instance, both the EFF [10] and the Cato Institute [21], not frequently aligned on political themes, share the opinion that anonymity is a positive value and that repressing or confining it would be a mistake and a blow to freedom and to a working and functioning democracy. Since its early years of online message boards, indeed, the Internet has been traditionally fairly open-minded about anonymity [18], but episodes of misuse of anonymity and their potential for harm are now present in the political discourse of many liberal democracies. For instance, [11] showed how easy it is for foreign governments, antidemocratic groups, and commercial companies to manipulate public debate through campaigns using networks of fake accounts. Thus, at the opposite end, many legislators have proposed and enacted laws to control or limit anonymous uses of the Internet ([1] or [8]) in the financial sector or even the recent UK Online Safety Act 2023 [4] in social networks. And yet, elsewhere more robust forms of anonymity online are being actively sought. Blockchain technologies frequently figure in the proposed approaches, whether they be decentralized identifiers such as DIDs [22] or self-sovereign identities such as ESSIF [3]. Overall, while the careless adoption of weak anonymity tools can give users a false sense of security, solutions are being deployed that guarantee robust anonymity to careful and competent users.

4 ARCHITECTURE

We here describe the architecture of the AAA system starting from the actors involved. We consider the architecture composed of two

layers with different confidentiality: (i) the secret level and (ii) the confidential level. In the secret layer, each piece of data is retrieved and managed only by the owner through cryptographic systems. In the confidential level, data is collaboratively managed, with each participant responsible for securely storing and protecting the data.

The secret layer includes the blockchain and the smart contracts. The blockchain primarily serves as a secure, public, and distributed data storage, while the smart contracts are used to execute and manage all system operations in a secure and reliable manner. The confidential layer includes the actors that participate in the creation, usage, and verification of anonymous and public identities. A user is a person who can request to authenticate her public identity and to obtain one or more anonymous identities. These requests are submitted to the Union of Identity Providers (UIP), i.e., the network of official national identity providers (NIPs) of each country. Once the user obtains the identities, she can use them to authenticate herself to any online service requiring authentication. We distinguish between public identity services (PIs), used by e.g., healthcare or social services, and anonymous identity services (AIs), used by e.g., online forums and dating applications, which accept/require authenticated public identities and fully anonymous identities to log in, respectively.

Figure 1 shows the AAA architecture and the interactions among the entities of the system. As previously introduced, data shared in the *secret* layer is stored on a blockchain. Due to the blockchain's nature as a public registry, where data can be freely accessed by anyone, any private data must be encrypted. In this paper, we abstract from the specific type of blockchain used — e.g., a permissioned blockchain where participation is controlled. Therefore, we save encrypted data to ensure that it is readable only by specific actors. Thus we call $ENC(payload, K)$ the function that encrypts a payload with the key K, which can be a public or symmetric key, and $SIGN(payload, K)$ the function that signs a payload with a private key.

Data in the confidential layer can be shared among the entities involved in the architecture: it is their duty to keep the data safe and not accessible to any unauthorized actor. To start the process for obtaining one or multiple anonymous identities, the user makes a request to her NIP providing all her personal data (e.g., ID card or passport) and a public key (*STEP #1*). The NIP issues a Public Identity Data (PID), i.e., an anonymous token that identifies the user inside the system without explicitly sharing information, saves it on its local database, and shares it with the user. From this moment on, the real identity of the user in the system is carried by the PID alone, since it has been officially issued by an NIP that verified the identity of the user. Additionally, the NIP will be the only one able to connect a PID to the real information of the respective user.

After obtaining the PID, the next step for the user is to send her PID and a public key to the smart contract to request a seed phrase, i.e., a mnemonic phrase that can be used as a master key to generate many private keys [12] (*STEP #2*). The smart contract initiates a protocol that generates N random words that are encrypted and saved to the blockchain so that the user can retrieve them confidentially through her private key. In addition, the smart contract generates the Secret Identity Data (SID), a token that is the hash of the concatenation of the hashes of N previously generated words. This SID is saved on the blockchain, encrypted with the user's previously provided public key (PK), allowing the user to recover it confidentially.

In order to retrieve a real identity, the UIP must be able to reconstruct the seed phrase of a user. The protocol duplicates each generated word several times, saving the record $PID : ENC(PID, wordnumber, word, PK_i)$, i.e., the association between the PID, the word generated, and its order, encrypted with the PK of the i-th UIP node randomly selected for redundancy. Thus, in order to obtain the seed phrase, there will needs to be consensus among the nodes of the UIP to decrypt the words and reconstruct the seed phrase in the right order. Additional details on the protocol for seed phrase generation and distribution are provided in Section 4.1.

Finally, the smart contract creates a symmetric key ($symK$) using the concatenation of the hash of each word and stores the record $SID : ENC(PID, symK), PK$ on the blockchain, i.e., the PID-SID association encrypted with the symmetric key and the public key. This lets the UIP, once having obtained the seed phrase, recover the PID associated with a SID and retrieve the real identity of a user.

After the mapping between PID and SID has been correctly stored on the blockchain, both identities are considered authenticated. When the user wants to use a service that requires an authenticated public identity, she can request a Public Authentication Code (PAC) from the NIP (*STEP #3*). This is a one-time code that is used to authenticate the user as an *identified user* in the system without sharing any information about her identity. More precisely, the user sends a message to the NIP containing $SIGN(PID, sk)$, namely the PID signed with the private key of the public-private key pair used to obtain the PID at the beginning. The NIP verifies that she is indeed the key holder and returns the PAC, saving it in its local repository. When logging onto the public service, the user can show the PAC, and the service has only to query the NIP to verify that the code is associated with an *authenticated user*.

Similarly, when the user wants to use a service that requires an authenticated anonymous identity, she can request a Secret Authentication Code (SAC) from the UIP (*STEP #4*). The SAC is a one-time code used to authenticate the user as an *anonymous user*, at the same time guaranteeing that a real user—one that has been registered within the system—exists behind the anonymous identity. To obtain the SAC, the user queries the NIP by sending a message containing $SID : ENC(SID, sk)$, i.e., her SID signed with the private key associated with the public key saved on the blockchain at the moment of seed phrase creation and used in the record where the SID was stored. The NIP retrieves the SID record from the blockchain and checks that it was actually signed by that user via the PK saved in the record. This certifies that the user is the true owner of that SID.

Now the NIP saves the mapping $SAC : SID$ in its local repository and then the SAC on the blockchain. Through the seed phrase, a user can create as many PK-sk pairs as wished. Abstracting from the implementation details, each pair can be considered an anonymous identity that must be authorized by the system. For this purpose, the user sends to the smart contract the PK of the account she intends to use together with her SAC. The smart contract checks the SAC existence on the blockchain and saves a $SAC : PK$ record on the blockchain to store the association between the SAC and the public key of the anonymous account. To log in, the user provides

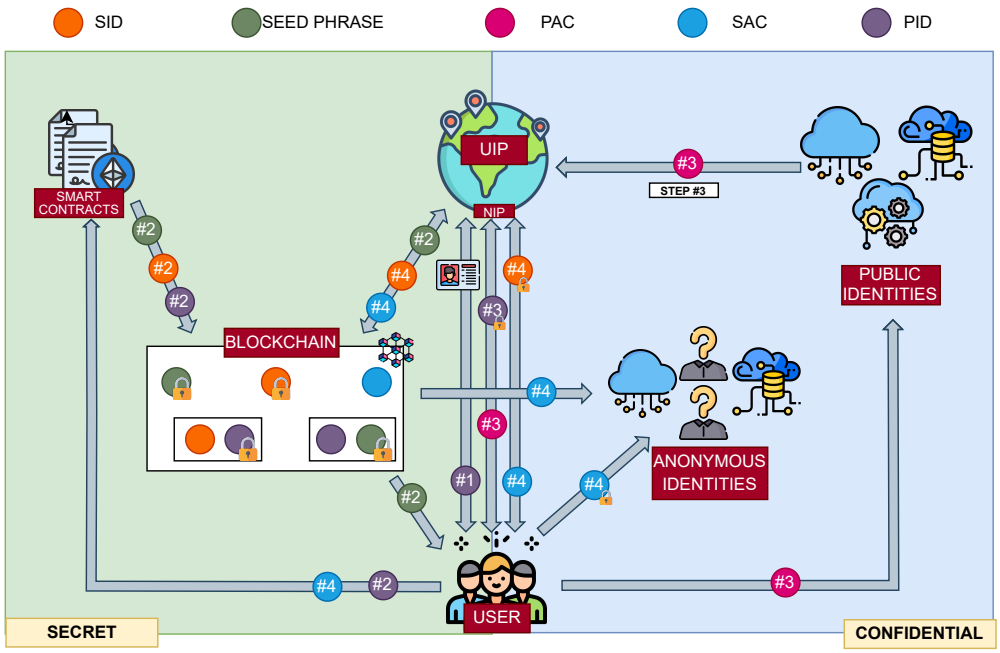


Figure 1: Overview of the AAA architecture and interactions between components. The user makes a request and the NIP issues a PID (*STEP #1*), which is used to generate a seed phrase through a smart contract (*STEP #2*). For services that require an authenticated public identity, the user makes a request for a public authentication code that allows authenticated access without sharing any information about her identity (*STEP #3*). For an authenticated anonymous identity the user can request a secret authentication code (*STEP #4*), with the guarantee that a real user is behind the anonymous identity.

Table 1: Glossary of relevant terms.

Acronym	Meaning
PK	Public key
sk	Secret key associated to a PK
symK	Symmetric key
PID	Public Identity Data: an anonymous token that identifies the real user inside the system
NIP	National Identity Provider: national institution that releases a PID after verification of the real identity of a person
UIP	Union of Identity Providers: the network of official national identity NIPs of each country
PIs	Public Identity services: online services that require public identities for logging in
AIs	Anonymous Identity services: online services that accept fully anonymous identities for logging in
ENC (payload, K)	Function that encrypts a certain payload with the key *K*, which can be a public or symmetric key
SIGN (payload, sk)	Function that signs a payload with a private key *sk*
SID	Secret Identity data: a token that is the HASH of the concatenation of the 24-words HASH
PAC	Public Authentication Code: one-time code used to authenticate the user as an identified user in the system
SAC	Secret Authentication Code: one-time code used to authenticate the user as an anonymous user

a record containing $SAC, PK, SIGN(SAC, sk)$ to the anonymous service, namely the SAC, the PK that identifies the anonymous account, and the SAC signed with the private key associated with the PK. The service retrieves the SAC record from the blockchain, verifies the signature—and hence that the user owns the public-private key pair—and the association between the SAC and the anonymous account. If everything matches, the service provides anonymous access to its features.

4.1 Seed phrase generation and distribution

In this section, we provide further details about the protocol run by the smart contract in *STEP #2* for seed phrase generation and its distribution in a redundant way. In Algorithm 1, each UIP node initializes the same Pseudo Random Number Generator (PRNG) using the user's PID as a seed, ensuring that the entire process is completely verifiable and reproducible by all the actors. Each node executes the selection process independently to determine if it is

Algorithm 1: Seed phrase Generation and Distribution Protocol

Input: User PID pid, Public Key pk_{user}, UIP Nodes Pool $\mathcal{Q}$, Redundancy Factor M
Output: Blockchain record of encrypted words and their redundancy across UIP nodes

```
// Select UIP nodes
```
$\mathcal{N} \subseteq \mathcal{Q}, |\mathcal{N}| = 24 \leftarrow$ SelectNodes$(\mathcal{Q}, pid)$
```
// Distribute words to blockchain
```
foreach $n_i \in \mathcal{N}$ **do**
  $word_i \leftarrow$ GenerateRandomWord()
  $enc\_word_i \leftarrow$ Encrypt$(word_i, pk_{user})$
  Write (pid, i, enc_word_i) to blockchain asynchronously
```
// Generate the final hash of all words
```
$complete\_phrase \leftarrow$ Concatenate and Hash all enc_word_i
Write $complete\_phrase$ to blockchain
```
// Redistribute words for redundancy
```
foreach $n_i \in \mathcal{N}$ **do**
  $\mathcal{M}_i \subseteq \mathcal{Q}, |\mathcal{M}_i| = M - 1 \leftarrow$ SelectRedundancyNodes$(\mathcal{Q}, \mathcal{M}, pid, id_{n_i})$
  foreach $m \in \mathcal{M}_i - 1$ **do**
    $enc\_word_{im} \leftarrow$ Encrypt$(word_i, pk_m)$
    Write (pid, i, enc_word_{im}) to blockchain

part of the group of N UIP nodes that need to participate in the seed phrase generation process.

The selection of the N nodes is achieved by generating a sequence of N unique indices from the range $[0, \text{size}(UIP)]$ using the PRNG initialized with the PID, which ensures deterministic and reproducible node selection. Each UIP node has a pre-assigned index based on its registration order on the blockchain. Once the group of N UIP nodes is determined, each node generates a random word, encrypts it using the user's PK, and asynchronously writes this encrypted word to the blockchain. The smart contract hashes and concatenates all received words once all N entries are recorded, finalizing the seed phrase generation process.

To further secure the system against node failures or attacks, each node redistributes its word to $M - 1$ other nodes. This secondary selection uses a PRNG initialized with a combination of the PID and the node ID to select $M - 1$ unique nodes from the pool, ensuring deterministic redundancy. The words are encrypted for the target nodes and recorded on the blockchain, providing a redundant layer of security and ensuring the availability of the phrase on the network.

The time complexity of Algorithm 1 can be studied from the perspective of one of the elected nodes and hence participating in the entire process of seed phrase generation and redundancy. The function SelectNodes requires the initialization of the PRNG using the user's PID as a seed. The complexity of initializing the PRNG is $O(1)$. The following selection of N nodes from the pool $size(UIP)$ is performed by generating a sequence of N unique indexes using the PRNG. Generating each index is $O(1)$, and ensuring uniqueness requires some set operations, resulting in an overall complexity of $O(N)$, which can be considered $O(1)$.

During the second phase each elected node generates a random word and encrypts it using the user's public key. Assuming that the word is selected from a common dictionary, we can consider the complexity of the GenerateRandomWord function as $O(1)$. The encryption function Encrypt depends on the encryption algorithm used, and we can define it as $O(E)$. Finally, the encrypted word is written on-chain with a cost of $O(1)$. The key redistribution process involves an additional selection function, SelectRedundancyNodes, similar to the previous one, with a complexity of $O(M)$. Encrypting the word for M nodes involves $O(M \times E)$ operations. Given that, the overall time complexity for the execution of the algorithm by one of the elected nodes is linear.

5 ANALYTICAL MODEL

This section is dedicated to providing a mathematical model of the proposed AAA. In particular, we are interested in analyzing how the system's robustness and fault tolerance are affected by its parameters, such as the number N of words in a seed phrase or the number M of copies. As performance metrics, we define as *evil attacks* the possibility that a set of (evil) actors could recover entirely —without the consensus of the rest of the UIP— a seed phrase. Additionally, we define as *node faults* the case in which a certain seed phrase is no longer recoverable because of the collapse of a set of (faulty) nodes, despite redundancy. Given that, we also analyze, from a global perspective, the potential risk of evil attacks or node faults as the number of created identities continues to increase.

The following mathematical model and expressions rely on some blanket assumptions and simplifications. We denote by Q the total number of national agencies, by N the number of words each seed phrase is composed of, and by M the number of copies shared of each word. Since word copies are distributed to different actors, M is also the number of actors storing each word. We will treat all phrases, words, and actors as equal: parameters N and M are taken constant over all seed phrases and words, and the actors storing word copies are selected randomly with uniform probability.

Given a pair of integers n and k such that $n \geq k \geq 0$, we make use of the so-called *binomial coefficient*, compactly denoted by $C(n, k)$ and defined with factorial notation as

$$C(n,k) := \binom{n}{k} := \frac{n!}{k!(n-k)!}.$$

The positive number $C(n, k)$ is also known as "n choose k", since it is the number of possible ways to choose an (unordered) subset of k elements from a fixed set of n items. Thus $C(n, k)$ is the number of k-combinations of n elements.

Evil attacks. The evil attack event coincides with a subset of q actors collecting at least one copy of all N words of an arbitrary seed phrase. This situation stems from *STEP #2*, when the different actors can collect some of the N words. Let us start by considering just one word and how it is shared for redundancy: what is the probability that none of the q actors collect the word? Out of all redundancy combinations (M-combinations of Q elements), this happens only for M-combinations of $Q - q$ elements, hence with (frequentist)

probability $p_{\text{none}}(q)$. Then, since just one copy is sufficient, the probability that the q nodes collectively get access to the word is the complement $1 - p_{\text{none}}(q)$. Finally, since all N words are needed to reconstruct a seed phrase, a prearranged coalition of q actors has access to it with probability

$$\pi_{\text{evil}}(q) \coloneqq [1 - p_{\text{none}}(q)]^N, \quad \text{where} \quad p_{\text{none}}(q) = \frac{C(Q-q, M)}{C(Q, M)}.$$

Node Faults. Node faults correspond to q nodes failing (or not collaborating) to reconstruct a seed phrase: we define $\pi_{\text{fault}}(q)$ as the probability that, despite the redundancy, a certain seed phrase is no longer recoverable because of the collapse of q nodes. For this circumstance to take place it is enough to have one seed word missing, namely all M nodes possessing a copy of that seed word have to fail. Let us focus on an arbitrary seed word first, considering a subset of q faulty nodes: the probability that all M copies are collected by the q actors out of Q is $p_{\text{all}}(q)$. Then, the probability that all copies of at least one word are collected is

$$\pi_{\text{fault}}(q) \coloneqq 1 - [1 - p_{\text{all}}(q)]^N, \quad \text{where} \quad p_{\text{all}}(q) = \frac{C(q, M)}{C(Q, M)}.$$

Evil attacks and node faults in time. We should consider not only the impact of evil players and disruptions on individual users but also the overall effect on the entire architecture. In particular, we focus on the probability $P_{\text{evil}}(q, T, k)$ that q evil nodes can reconstruct k seed phrases after T seed phrases have been created and on the probability $P_{\text{fault}}(q, T, k)$ that q faulty nodes could block the reconstruction of those k seed phrases.

The growth process can be modeled as a Bernoulli process: a sequence of a fixed number T of statistically independent trials, each with its own Boolean-valued outcome, with a probability of 'success' p. Then, the probability of *exactly* k successes in the experiment follows a binomial distribution, given by

$$P(p, T, k) \coloneqq C(T, k) p^k (1-p)^{T-k}.$$

In our context, we consider the probabilities $p \in \{\pi_{\text{evil}}(q), \pi_{\text{fault}}(q)\}$ for the respective events. Thus, we have simply $P_{\text{evil}}(q, T, k) \coloneqq P(\pi_{\text{evil}}(q), T, k)$ and $P_{\text{fault}}(q, T, k) \coloneqq P(\pi_{\text{fault}}(q), T, k)$. Finally, to evaluate the probability of *at least* k successes out of T trials it suffices to compute the cumulative distribution function of $P(p, T, \cdot)$.

6 EVALUATION

The goal of the evaluation carried out in this section is to verify the robustness of the AAA system given its formal modeling described in Section 5. More specifically, we investigate how the architectural parameters and the size of evil coalitions (or faulty nodes) impact the functioning and efficacy of the AAA. For our analysis we assume the size Q of the UIP to be in the order of 50-100 elements.

Optimal redundancy. As a first analysis, we investigate the sensitivity to the number M of copies shared of each word in the system, which represents a trade-off: a lower value of M implies a lower number of different NIPs involved, hence a lower probability of evil attacks but a higher probability of node faults (i.e., a lower fault tolerance). On the contrary, a higher value of M represents a higher possibility of evil attacks and a higher fault tolerance. Hence, our goal is to find a good balance between these two criteria to determine an optimal interval for M for the rest of the evaluations.

For this purpose, we defined the optimization target

$$\Theta(q) \coloneqq 1 - [1 - \pi_{\text{evil}}(q)][1 - \pi_{\text{fault}}(q)]$$

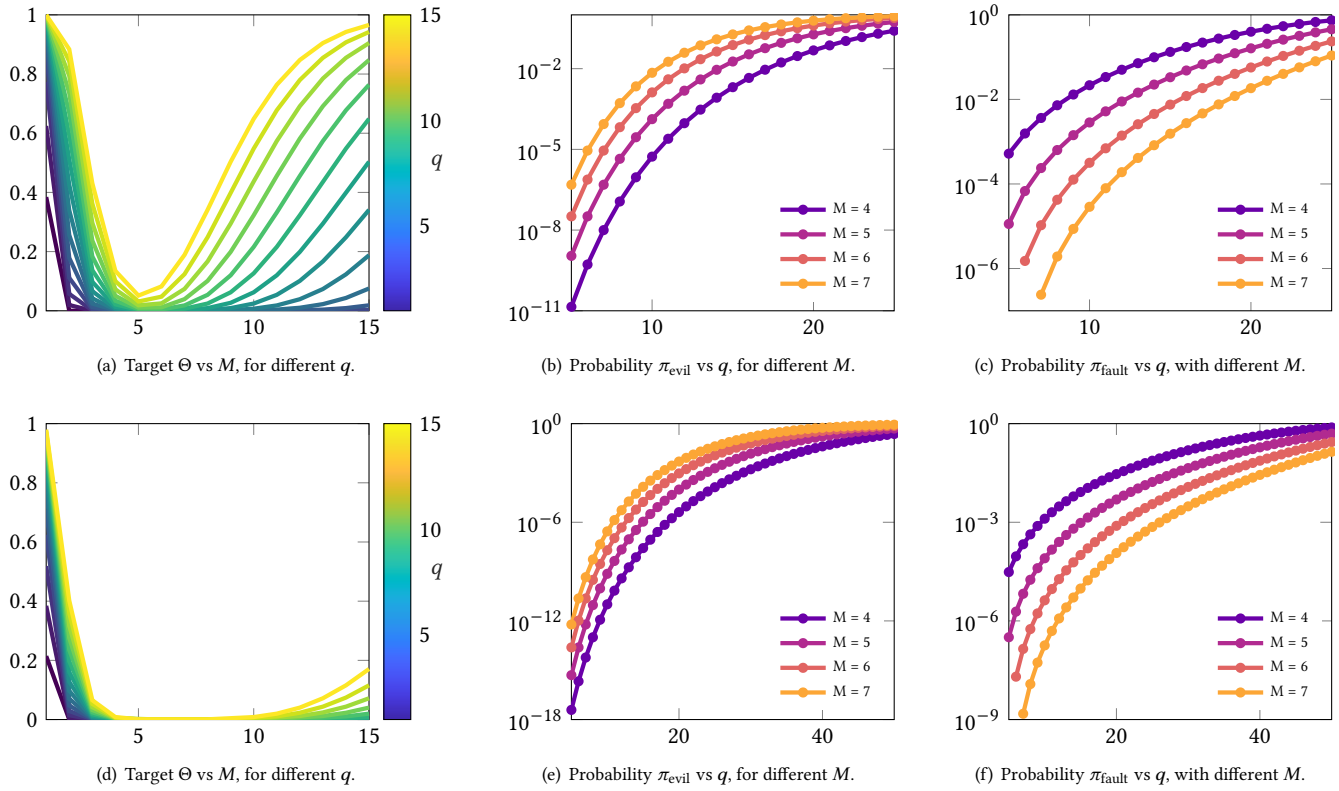
with the goal of minimizing its value for a range of q. The target Θ is constructed to represent the susceptibility of the architecture to both evil attacks and faults: the value $\Theta(q)$ is the probability that a seed phrase can be either recovered, or blocked, or both, by a set of q evil actors.

Figures 2(a) and 2(d) show the dependence of Θ on M and q, respectively for $Q = 50$ and $Q = 100$. In both cases, for any fixed M, the target $\Theta(q)$ monotonically increases with the number q of evil nodes, as one would expect. In contrast, for any fixed q, the target does not display a monotonic behavior with respect to the number M of copies: it appears that values of M around $[4, 7]$ provide a robust minimization of the target, making these valid choices for M over a wide range of possible q (and Q).

Notice that for the visualization in Figures 2(a) and 2(d), we varied $M, q \in [1, 15]$ to zoom in on the zone of interest represented by the sequence of minima for the target. Intuitively, larger values of M can be excluded because they lead to a larger percentage of nodes holding parts of information for each user, undermining security against evil attacks. Once we identify an interval of reasonable values for M, we consider larger values of q to examine the performance of the proposed architecture in severe situations.

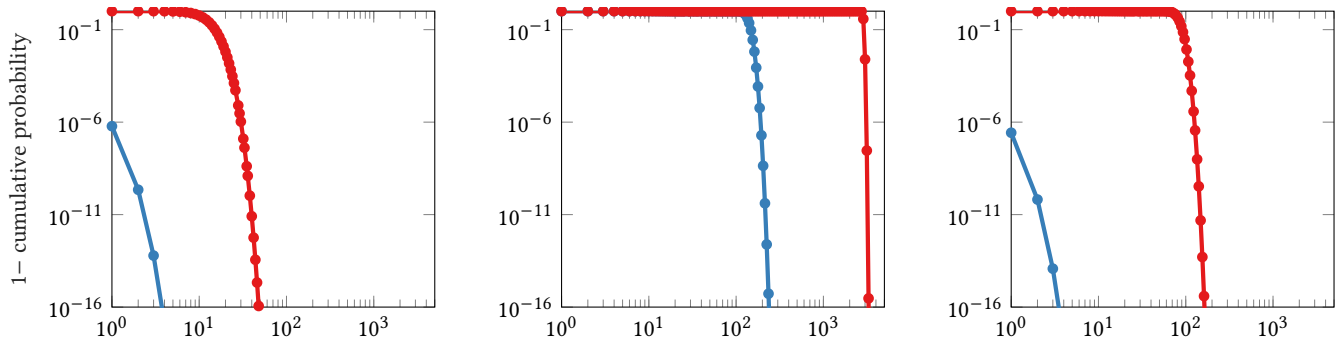
Individual level. Our next analysis investigates how the probabilities π_{evil} and π_{fault} change while varying the q and the M values, both for $Q = 50$ and $Q = 100$. In particular, we want to understand until what values the system can guarantee robustness: in Figure 2(b) we vary $q \in [5, 25]$, hence with the number of evil nodes up to 50% of the total. When a large portion of the nodes is corrupted or not collaborating, both probabilities $\pi_{\text{evil}}(q)$ and $\pi_{\text{fault}}(q)$ approach 1, regardless of the redundancy M, capturing the plausible sabotage or breakdown of the AAA. However, for evil coalitions of reasonable size (say $q \leq Q/5$), the system behaves more reliably and yields better performance metrics. Figure 2(b) shows that bigger values of M increase $\pi_{\text{evil}}(q)$, that is, negatively impact on security, whereas fault tolerance monitored by $\pi_{\text{fault}}(q)$ is improved, see Figure 2(c). It is interesting to note that, with $q = 5$ evil nodes, the system guarantees a $10^{-12} < \pi_{\text{evil}}(q) < 10^{-6}$, which translates into the fact that, with $M = 7$, just one real identity in a million is likely to be recovered without the consensus of the UIPs. With the same configuration, $\pi_{\text{fault}}(q) < 10^{-3}$, hence less than one account in a thousand will be not recoverable in the worst case. For the case $Q = 100$, we vary $q \in [5, 50]$ and the results show the same pattern. In particular, with $q = 5$, the system grants that less than one real identity over 10^{12} can be discovered by evil nodes and that less than one account over 10^4 can be unrecoverable with the lowest M. These analyses suggest that the system parameters must be tuned according to the requirements of the AAA administrators, based on the current values of Q, the expected q, and a custom threshold of evil and fault acceptability.

System level. In Figure 3 we visualize the susceptibility of the system from a global perspective, under different circumstances, probing its state after a number $T \gg 1$ of seed phrases have been created. For these simulations, we take $M = 5$ and $T = 10^6$. Let us focus first on the case with $Q = 50$ and $q = 5$ (10% evil players),



(a) Target Θ vs M, for different q.
(b) Probability π_{evil} vs q, for different M.
(c) Probability π_{fault} vs q, with different M.
(d) Target Θ vs M, for different q.
(e) Probability π_{evil} vs q, for different M.
(f) Probability π_{fault} vs q, with different M.

Figure 2: Illustration of performance metrics as a function of number of copies M and number of evil nodes q in different scenarios, for $Q = 50$ (top) and $Q = 100$ (bottom) national agencies and $N = 24$ words per seedphrase. Left: Target Θ vs M, for different values q (colorbar). Middle: Probability of evil attacks π_{evil} vs q, for different values M. Right: Probability of node faults π_{fault} vs q, for different values M.



(a) $q = 5$, $Q = 50$.
(b) $q = 10$, $Q = 50$.
(c) $q = 10$, $Q = 100$.

Figure 3: Comparison of global performance metrics after $T = 10^6$ seedphrases have been created in different scenarios, with $N = 24$ words per seedphrase and $M = 5$ copies of each word. Each plot indicates the complement to one of the cumulative probability distribution of $P_{\text{evil}}(q, T, \cdot)$ and $P_{\text{fault}}(q, T, \cdot)$, namely the probability that more than k identities are affected by the q evil nodes.

depicted in Figure 3(a), where we illustrate the cumulative probability distribution to estimate how many accounts may be affected by an evil coalition, or faulty players. Notice that the complement (to one) of the cumulative distribution gives the probability that *more*

than k identities are affected, out of T. Thus, we observe that the most likely outcome is to have approximately 10-50 unrecoverable and 0-3 cracked identities, out of $T = 10^6$, since the two curves are most steep around these values. In this case, the probability of having more than 40 identities lost or 3 cracked is less than 10^{-12}. Figure 3(b) illustrates the scenario with 20% evil players: in this exaggerated situation, the most likely outcome is around 3000 unrecoverable accounts and 100 cracked ones, out of $T = 10^6$. Instead, with $q = 10$ evil actors out of $Q = 100$, Figure 3(c) shows up to 100 unrecoverable identities and 3 cracked ones.

7 CONCLUSIONS

In this paper, we presented AAA, an Anonymous Authenticated Architecture based on blockchain solutions, through which it is possible to create authenticated anonymous identities that can be used for logging into online services anonymously. The mathematical model of the architecture we provided has been used to evaluate the risk of evil attacks and faults, demonstrating that AAA can provide a trustworthy approach for the public good, even with the numbers of a potential worldwide adoption. Future works include the possibility of reconstructing the list of all the anonymous identities connected to the same physical person, a strategy for credential recovery, and real testbed implementation.

REFERENCES

[1] 2001. Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism (USA PATRIOT ACT) Act of 2001. Public Law 107-56.

[2] 2011. ISO/IEC 24760–1:2011: Information technology – Security techniques – A framework for identity management – Part 1: Terminology and concepts. http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html.

[3] 2022. *European Self Sovereign Identity Framework Laboratory.* https://doi.org/10.3030/871932

[4] 2023. *Online Safety Act 2023.* http://www.legislation.gov.uk/ukpga/2023/50

[5] Daniel Augot, Hervé Chabanne, Olivier Clémot, and William George. 2017. Transforming face-to-face identity proofing into anonymous digital identity using the bitcoin blockchain. In *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 25–2509.

[6] Francesco Buccafurri, Gianluca Lax, Antonia Russo, and Guillaume Zunino. 2018. Integrating digital identity and blockchain. In *On the Move to Meaningful Internet Systems. OTM 2018 Conferences: Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I.* Springer, 568–585.

[7] Ivan Damgård, Chaya Ganesh, Hamidreza Khoshakhlagh, Claudio Orlandi, and Luisa Siniscalchi. 2021. Balancing privacy and accountability in blockchain identity management. In *Cryptographers' Track at the RSA Conference*. Springer, 552–576.

[8] European Parliament and Council of the European Union. 2015. *Directive (EU) 2015/849 of the European Parliament and of the Council.* https://data.europa.eu/eli/dir/2015/849/oj

[9] European Parliament and Council of the European Union. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council.* https://data.europa.eu/eli/reg/2016/679/oj

[10] Electronic Frontier Foundation. 2024. *Anonymity.* https://www.eff.org/issues/anonymity

[11] Rolf Fredheim, Sebastian Bay, Anton Dek, Iryna Dek, and Singularex. 2020. *Social Media Manipulation Report 2020.* Report. NATO Strategic Communicaions Centre of Excellence. https://stratcomcoe.org/publications?aid[]=43

[12] Gus Gutoski and Douglas Stebila. 2015. Hierarchical Deterministic Bitcoin Wallets that Tolerate Key Leakage. In *Financial Cryptography and Data Security*, Rainer Böhme and Tatsuaki Okamoto (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 497–504.

[13] Thomas Hardjono, Ned Smith, and Alex Sandy Pentland. 2014. Anonymous identities for permissioned blockchains.

[14] Hui Huang, Xiaofeng Chen, and Jianfeng Wang. 2020. Blockchain-based multiple groups data sharing with anonymity and traceability. *Science China Information Sciences* 63 (2020), 1–13.

[15] ISO 29100:2024 2024. *Information technology — Security techniques — Privacy framework.* Standard. International Organization for Standardization, Geneva, CH.

[16] Gianluca Lax and Antonia Russo. 2020. Blockchain-Based Access Control Supporting Anonymity and Accountability. *Journal of Advances in Information Technology* 11, 4 (2020), 186–191. https://doi.org/10.12720/jait.11.4.186-191

[17] Alexandra Mateescu, Douglas Brunton, Alex Rosenblat, Desmond Patton, Zachary Gold, and Danah Boyd. 2015. Social media surveillance and law enforcement. *Data & Civil Rights* 27 (2015), 2015–2027.

[18] Jacob Palme and Mikael Berglund. 2002. *Anonymity on the Internet.* https://people.dsv.su.se/~jpalme/society/anonymity.html

[19] Andreas Pfitzmann and Marit Hansen. 2010. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf v0.34.

[20] Wei Shao, Chunfu Jia, Yunkai Xu, Kefan Qiu, Yan Gao, and Yituo He. 2020. Attrichain: Decentralized traceable anonymous identities in privacy-preserving permissioned blockchain. *Computers & Security* 99 (2020), 102069.

[21] Ilya Shapiro and Randal John Meyer. 2015. *The Right to Anonymous Speech and Association.* https://www.cato.org/blog/right-anonymous-speech-association

[22] Manu Sporny, Amy Guy, Markus Sabadello, and Drummond Reed. 2022. *Decentralized Identifiers (DIDs) v1.0.* Technical Report. W3C. https://www.w3.org/TR/did-core/

[23] Cryptopedia staff. 2021. Anonymity vs. Pseudonymity In Crypto. https://www.gemini.com/cryptopedia/anonymity-vs-pseudonymity-basic-differences