How to Grant Anonymous Access *

Antonio M. Larriba and Damián López
vrAIn - Valencian Research Institute for Artificial Intelligence
Universitat Politècnica de València

December 19, 2022

Abstract

In this paper, we propose three protocols to share, among a set of N competing entities, the responsibility to grant anonymous access to a resource. The protocols we propose vary in their settings to take into account central or distributed registration. We prove that any subset of guardian authorities can neither tamper with, nor forge, new access-key tokens. Besides, two of the methods we propose are resistant to the eventual appearance of quantum computers. The protocols we propose permit new approaches for cryptographic applications such as electronic voting or blockchain access.

Keywords: Cryptography; Distributed identification; One-time identification, Anonymous access; Post-quantum security.

1 Introduction

Authentication and subsequent access to a resource are fundamental problems in cryptography. This process is usually considered as the certification (by a verifier) of the identity of a claimant, and the subsequent permission for the claimant to access to a resource.

Identification protocols are those by which entities provide relevant information that guarantee they are who they claim. Thus, identification is usually related to digital signatures, specially with those in which the claimant takes an active role [16]. Another interesting identification approach is based on zero-knowledge proofs [27], where a claimant must proof the knowledge of the key to a verifier, without revealing any details of his key (e.g. [23]). We note that, in this scenario, the verifier does know the identity of the claimant and permits access when the process is successfully completed. Therefore, a malicious verifier could relate the actions carried out by the individual whose access has been granted to.

*Spanish Patent Application: P202130890 Corresponding author: Damián López (email: dlopez@dsic.upv.es).

In a different and enhanced scenario, we consider anonymous identification and/or access to a service or a resource. In this scenario, the claimant would provide enough evidence to the verifier in order him to be consider as a member of a set of authorized people (e.g. users of a resource, electors in a census, or registrars in a blockchain). Among all identification scenarios, anonymous identification is by far the most conceptual idea (the claimant is not forced to reveal his identity in order to gain access to the resource). Up to our knowledge, no proposal has been done to carry out this process.

Anonymous Access

A less restrictive, but also powerful, scenario aims to grant anonymous access to the members of a collective. Anonymous access protocols stand a challenge since they imply to divide the authorities between those allowed to know the identity of the authorized user, and those who are not. In order to provide a clear description of the roles implied, we divide the anonymous access process into three steps: *identification*, as the process by which a verifier certifies that the credentials suffice to guarantee the identity of a claimant, and, therefore, with the right to access the resource; *registration*, as an intermediate step where the claimant is provided with the access-keys (e.g: password, authorization, access token), which are then used by the claimant to *access* the service. This setup allows to see the difference between the identification and the registration stages, where the former is the stage where a yet unidentified claimant connects to a service to validate himself (his identity), and the latter, where the claimant establishes (or gets) the credentials for, possibly anonymously, accessing a service.

In this work, we focus on anonymous distributed registration and access, which, in the vein other cryptographic problems were proposed, the problem can be stated as:

> Let a resource be controlled by a set of competing guards in such a way that no guard trusts any other. Thus, access to the resource implies all the guards agree that the presented access-keys are valid, while the access-keys themselves reveal no details of their owners identity.

This anonymous access scenario is still an open problem and there are approaches that tackle this problem from different angles (we review them in Section 2). Briefly, some systems [32, 4], in the context of electronic voting, consider an authority to solve the problem and add a few layers of obfuscation to protect user's identity [14, 26]. Other proposals are focused on the prevention the individuals to be tracked, thus, several works have studied how to distribute password authentication [34, 39], or how to use multiple servers to generate tokens [2, 22], all these protocols aim to grant access while protecting the system against server breaches and offline attacks.

Non-interactive Access Protocols

In this paper, we propose three different non-interactive access protocols to address the anonymous access problem. Our methods are easy to scale and implement, they require minimal interaction from the users, and both the access-key generation and verification processes are based on the same primitives. Our results are inspired by the secret sharing protocol proposed by Shamir in [42].

The first protocol we present considers the most restrictive scenario, that implies the assumption of a trusted authority responsible for issuing the access keys. The second protocol presented enhances our first approach and distributes the responsibility of issuing the access-keys to a set of accredited authorities. In our third proposal, we present a fully decentralized, anonymous access scheme. The first two protocols we here propose are unconditionally secure, and provide quantum-resistant security features. We prove that our third method preserves the privacy of the users even in a post-quantum scenario, and that it is secure (no access-keys can be forged) while the discrete logarithm problem remains unsolved. All three protocols grant one-time access with minimal interaction required by the user.

Several applied problems, such as the anonymous access to a blockchain, registration to web pages or the process of casting a vote in an electronic voting framework, motivate us to propose these methods.

2 State of the Art

We devote this section to review the most relevant papers in the literature. Some of these papers are similar to our proposal on their methods, others in their objectives, but we found no previous work with the same attributes our approach has. We here review works from different areas such as: distributed threshold token generation; web identification; anonymous credential systems; and, identity management systems[1].

As mentioned above, traditional authentication protocols rely on a registration phase, where, generally, the user identifies himself to a central entity that stores some kind of credential (usually a password). Then, the registered user can obtain a token to access some service. This method, while valid, centralizes all the stored data on a single point of failure, it also requires the user to trust the entity responsible for the identification and access-keys handling. For this reason, and to preserve user's privacy, many works have explored the possibility of distributing user authentication among a set of servers. If the servers do not store hashed information, then this distribution allows for a

[1] We note that our work is not related with identity-based cryptography [43, 9]. As proposed initially by Shamir, identity based cryptography defines a new asymmetric cryptographic scheme in where the public key can be any random string which is intrinsically associated to the user's identity, and hence the name of this approach. Our work is neither related with key-anonymity protocols [6]. These protocols aim to provide anonymity of the key under which the encryption was performed, even under chosen-cipher texts attacks. However, the provisioned anonymity is only against a third-party observer, and does not include the sender.

better resilience against server breaches and offline attacks, and, if they use a proper threshold scheme, it improves the overall availability. Many works have studied distributed token generation through public key threshold signatures [22, 21, 25] and threshold authentication codes [10, 37] to protect the master key against server breaches. Also, a different line of work has studied the use of threshold password authenticated secret sharing [11], and threshold password-authenticated key-exchange [34, 39, 1], as an improvement against the offline dictionary attacks present in more traditional password-authenticated key-exchange [30, 7].

These works share a common goal with our proposal, because they seek to improve user's privacy through the distribution of the authentication process. However, these methods do not provide a real distribution of the trust in the system. We, instead, propose protocols where some authorities are not able to relate the users identity to the actions they carry out once they have gain access to the system.

To tackle the trust distribution problem, Agrawal et al. present a password-based threshold authentication protocol (PASTA) in [2]. PASTA is a general framework for token generation which distributes the task among a set of servers, such that any subset of t servers can verify and generate tokens, while no subset of $t-1$ servers can forge invalid tokens. In their work, Agrawal et al. aim to shield the token generation process against server breaches as well as to reduce the number of interactions needed with respect to more traditional password sharing works. After a (one time) registration, the user provides the credentials to the token-generation servers. If the credentials are valid, then the servers respond with a part of the token. Once the user has interacted with all the servers, the user proceeds to construct the token with the received parts. To prevent from offline attacks, the authors employ a threshold oblivious pseudo-random function (TOPRF) [24, 29] on the server side. Agrawal et al. propose, analyze, and implement PASTA as a complete and general framework compatible with multiple TOPRF functions and various threshold token generations algorithms (both for symmetric and asymmetric cryptography). To the best of our knowledge, this is the most similar work to our proposed protocol, but, while PASTA does not pay attention to the anonymity of users, our approach is mainly focused on their anonymity. Our proposal requires a single round interaction for registration and another one for access, and the same mechanism enables the distributed registration and allows at the same time the distributed access, thus allowing us to reduce the complexity of the protocol, making, at the same time, unnecessary the use of TOPRFs.

The aforementioned threshold protocols provide different approaches for redistributing the workload of a single service from a unique point of failure to multiple servers. However, in a federated scenario such as the Internet, multiple services co-exist, which implies that more complex interactions involving the identification of users are present, and, therefore, other approaches are needed. In order to address this issue, multiple standards have been proposed for managing identity and authorization on the web. These protocols allow for secure access in multiple scenarios such as login in a service with other's service access-

keys without revealing sensitive data or granting access without revealing the access password. OAuth [28] and SAML [38, 3] are open protocols that derive the identification responsibility to a trusted third-party. OpenID [40] is a protocol built on top of OAuth focused on the identification problem through an third-party acting as identity provider. All these standards provide a really useful set of protocols used everyday by multiple and well-known online services, but, neither of them pay attention to the anonymity of the users, and, in this federated environment, the users privacy lies on the assumption that the entities do not share information. Our protocols distribute the identification of the claimants among different entities, which are not able to track the access-key delivered to the users unless all the entities agree to do so.

In an effort to provide a system where users can obtain access-keys directly from organizations without the need of third-parties, Anonymous Credential Systems (ACS) [15] were proposed. In an ACS framework, organizations only know users by pseudonyms and users may have various unlinkable pseudonyms. Through digital signatures and zero-knowledge proofs, ACS provide a set of functionalities such as: unforgeability; anonymous validation of credentials; unlinkability; or access-key transference between organizations while preserving user's privacy. There exist several works that propose original approaches with different features [17, 18, 20, 33]. Nonetheless, as a consequence of the heavy use of zero-knowledge proofs, ACS are usually complex in terms of cryptographic primitives and have high time complexity. These factors are an inconvenient for ACS. Among the papers in the literature we distinguish the one by Camenisch and Lysyanskaya, [12], where they propose an ACS with interesting properties such as non-transferable access-keys, optional revocable anonymity, and one-show access-keys. Despite their focus on making a practical system, their extended protocol still needs up three rounds of interaction and heavy use of modular exponentiation operations. Despite these drawbacks, the proposed system was later implemented in [13]. Also, in [5], Belenkiy et al. propose a delegatable ACS, that presents a hierarchical system in which access-keys can be structured in levels as an intent to model real word interactions. Regarding ACS, our protocols cannot be considered as such, since our protocols do not present a framework for connecting multiple organizations to users, nevertheless, our protocols share with ACS the goal of providing unforgeable, unlikable and anonymous access-keys. Besides, our protocols allow for one time access with minimal computational load, the whole process can be distributed, they do not require pseudonyms or multiple rounds of interaction, and, they provide post-quantum security.

To adapt the latest results on identification systems to the most advanced biometric and secure hardware technology, Identity Management Systems (IdMs) were developed. IdMs aim to provide a complete suite of tools that handle all the aspects of the identification ecosystem as well its integration with biometrics, mobiles, hardware identification devices or others IdMs. They are usually based on some of the approximations previously presented (e.g: ACS) to preserve the privacy of the users. Given the magnitude of their goal, IdMs are usually backed by some government or official institution. In [36], Moreno et

al. present an IdMs based on PASTA that provides unlinkability through distributed identity providers and biometric identification. In [8], Bernabe et al. evaluate ARIES, an European IdMs that also includes ID-proofing based on biometrics and breeder documents handling within their framework. We do not mention IdMs as a comparable result, but as the current framework that includes identification methods similar to ours. We note that the scope of our work is much more limited than a complete IdMs.

3 Centralized registration, anonymous access

No matter which identification scheme is proposed, there must be some instant of time when the users provide their identification in order to prove the right to access a protected resource. In this section, we propose a first scheme that considers a central authority (assumed honest), a set of guards commissioned to control the access; and a set of accredited users[2]. The role of the central authority is twofold: first, he sets up the framework by distributing the access-keys to the users, as well as the distributed access control to the guards; and, second, he becomes an audit authority to solve any identification issue between users and guards. We consider the creation of the necessary private channels out of the scope of this paper, thus, we refer the interested reader to the results in the literature to see details about how to implement them (e.g. [41]).

In the following, we will refer to G as the set of N guards that control the access to the resource, where C will denote the collective of users that apply for accessing to it. We consider that subsets of both users and guards can behave maliciously. We now define the Centralized Registration Anonymous Access scheme as the following system of five probabilisitic polynomial time algorithms as follows:

- $SysSetup(1^k, N, C) \rightarrow (\{q_i\}_{i \in G}, pp)$. This algorithm generates a key-generation system $q(x)$ and some public parameters pp. The key-generation system is then randomly partitioned into $\{q_i\}_{i \in G}$ shares, which are privately distributed among the guards. Only the Central Authority has access to whole system $q(x)$. The set of shares of $q(x)$ meet the satisfiability condition described below.

- $Registration(id) \rightarrow a_{key}$. This algorithm generates for a correctly identified user a private access-key a_{key}.

- $AccessRequest(a_{key}) \rightarrow \{i, k_i\}_{i \in G}$. This algorithm is a call to the set of guards, who generate partial keys k_i according their share of the key-generation system. All the guards are committed to share their results with the rest of guards in G.

- $Combine(\{i, k_i\}_{i \in G}) \rightarrow tk$. The algorithm considers the partial keys computed by the guards and obtains an access-token by combining them.

[2]The credentials owned by the users are suposed to be delivered beforehand by some trustworthy institution.

- $GrantAccess(a_{key}, tk) \rightarrow 1/0$. Checks it the token tk corresponds to access-key a_{key}, in which case the guards agree to grant access (output 1).

Consistency is guaranteed if for any valid id and $a_k \leftarrow Registration(id)$, it is hold that:

$$GrantAccess(a_k, Combine(AccessRequest(a_k))) = 1.$$

It is assumed that $SysSetup$ and $Registration$ are exclusive algorithms of the Central Authority, which is trustworthy in this protocol.

We note that, on the one hand, only the central authority can relate the users identity to the access-keys issued to them, and, on the other hand, it is impossible to track the actions carried out by an individual because no information about the use of the resource by the users is provided to the authority (although it is true that malicious guards and central authority can collude to combine their information and gain such knowledge, we stress that the authority is assumed honest).

3.1 Protocol construction

We now describe our protocol for trusted registration and anonymous access (TRA2). The setup of the protocol implies the generation of a random prime p, and a m-degree polynomial modulus p ($m < p - 1$). Let us note that we are interested in the maximum degree of the polynomial, therefore, in order our proposal to work it is not required to consider a polynomial with all the coefficients non-null:

$$q(x) = a_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0 \bmod p.$$

Provided $q(x)$, the authority partitions the polynomial into N polynomials:

$$\begin{aligned}
&q_1(x) = a_{1,m} x^m + a_{1,m-1} x^{m-1} + \ldots + a_{1,0} \bmod p \\
&q_2(x) = a_{2,m} x^m + a_{2,m-1} x^{m-1} + \ldots + a_{2,0} \bmod p \\
&\vdots \\
&q_N(x) = a_{N,m} x^m + a_{N,m-1} x^{m-1} + \ldots + a_{N,0} \bmod p
\end{aligned}$$

such that the partitioned polynomials complement each other in order to obtain the coefficients of $q(x)$, that is:

$$a_i = \sum_{1 \leq j \leq N} a_{j,i}, \quad \forall i,\ 0 \leq i \leq m.$$

Each one of the partitioned polynomials is safely communicated and assigned to a guard. Note that, unless all the guards agree to forge the system, it is impossible for any subset of guards to interpolate $q(x)$.

Once the system has been setup, the system allows users to obtain their access-keys. Users send their identifications to the authority, who uses the

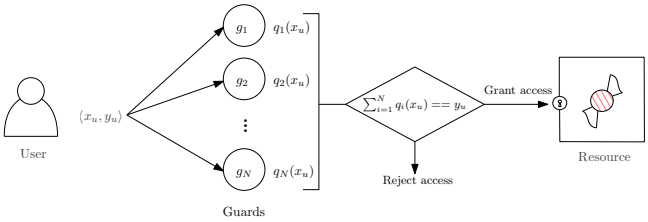

Figure 1: Users provide their access token to the guards to access the resource. Guards distributively decide if the provided token is valid.

polynomial in order to provide them a random point of $q(x)$. These points will play the role of anonymous access-keys of the users against the guards.

Note that unless the number of registered users reaches the bound fixed by m, it is impossible for any set of users to successfully interpolate $q(x)$ and tamper with new access-keys. Let us also note that: first, the size of the modulus p has not influence in the security of the protocol but it must be greater than m; and, second, that the use of modular arithmetic bounds the size of the access-keys while does not prevent the possibility of dealing with a reasonable high number of users[3].

Once the access-keys have been delivered, in order for a user to access to the resource or service, he sends his access-key (point of $q(x)$) to all the guards (as illustrates Figure 1). The guards must collaborate in order to decide whether the access-key is correct or not. Thus, given any user u and his point $\langle x_u, q(x_u) \rangle$, access should be granted whenever:

$$q(x_u) = \sum_{1 \leq i \leq N} q_i(x_u) \bmod p$$

In order to prevent the guards misusing the checked access-keys (for instance by distributing them to unauthorized users), we consider access-keys are single-use. Thus guards should also check the uniqueness of x_u to verify the validity of the request. The central authority is responsible for not generating two points with the same x coordinate. An outline of the complete protocol for our trusted registration, anonymous access protocol (TRA2) is summarized in Algorithm 1 and illustrated in Example 3.1.

Example 3.1. *Let us consider an scenario with three guards to control the access to some resource ($N = 3$). Let $p = 7919$ and $m = 665$ be the public integers that allow to set up the system. Let also:*

$$q(x) = 45x^{665} + 22x^{13} + 54x^7 + 1 \bmod 7919$$

be the polynomial the authority (privately) generates in order to generate the access tokens. The final step to set up is to partition the polynomial $q(x)$ into

[3]A value of m of 40 bits is not a big issue in terms of time complexity and is far enough to provide access-keys to all the habitants in Earth.

Algorithm 1 Trusted registration, anonymous access protocol (TRA2).

System setup

(a) The Central authority selects a prime p and generates a m-degree polynomial $q(x)$, where m is greater than the maximum number of users to identify.

(b) The Central authority partitions $q(x)$ into N polynomials $P = \{q_1(x), q_2(x), \ldots, q_N(x)\}$ that, for any x, meet the condition:

$$q(x) = \sum_{1 \leq i \leq N} q_i(x) \bmod p$$

(c) The Central authority distributes a polynomial in P to each of the N guards.

User (trusted) identification and registration

(a) Users send his identification to the central authority.

(b) Central authority verifies the identification credentials.

(c) If valid, central authority generates a random x_u and replies to the user with $\langle x_u, y_u = q(x_u) \mod p \rangle$.

Anonymous access

(a) User sends his access access-keys $\langle x_u, y_u \rangle$ to each one of the guards.

(b) Each guard computes $q_i(x_u)$.

(c) Guards check if $y_u = \sum_{1 \leq i \leq N} q_i(x_u) \bmod p$.

(d) If the access-keys are valid, access is granted.

three complementary polynomials to distribute among the guards, for instance:

$$q_1(x) = 26x^{665} + 4x^7 + 1 \bmod 7919$$
$$q_2(x) = 22x^{13} \bmod 7919$$
$$q_3(x) = 19x^{665} + 50x^7 \bmod 7919$$

Provided that $q(21) = 655$, a valid access token for a correctly identified user could be $\langle x_u = 21, y_u = 655 \rangle$. Once the guards receive the access token, each one can work out the result from its polynomial share:

$$q_1(21) = 26x^{665} + 4x^7 + 1 \bmod 7919 = 7526$$
$$q_2(21) = 22x^{13} \bmod 7919 = 3501$$
$$q_3(21) = 19x^{665} + 50x^7 \bmod 7919 = 5466$$

and must collaborate in order to check whether the token is valid or not. Indeed, $y_u = q_1(x_u) + q_2(x_u) + q_3(x_u) \bmod p$, and access should be granted.

3.2 Security analysis

We devote this section to analyze the security properties of the TRA2 protocol. In order to prove TRA2 unforgeability, for every probabilistic polynomial time adversaries Adv, we propose a security game where:

- The $SysSetup$ is run to get the polynomial $q(x)$, its partition into $\{q_i\}_{i\in N}$, m, and p. The values m and p are given to Adv.
- Let $U \subsetneq G$ be a set of corrupt guards. Give $P = \{q_i\}_{i\in U}$ to Adv.
- Let $V \subseteq C$ be a set of corrupt users where $|V| < m$. Give the set of access-keys $K = \{Registration(id_i)\}_{i\in V}$ to Adv.
- Let $\mathcal{O}_{interp}(P, K)$ be the oracle that, by any method, considers the available information to interpolate the polynomial:

$$q(x) - \sum_{i\in U} q_i.$$

- Let $\mathcal{O}_{newk}(P, K)$ be the oracle that, by any method, considers the available information to obtains new pairs (a, b) such that $b = q(a)$.
- Let $\mathcal{O}_{succ}(p(x))$ be the oracle that returns 1 if $p(x) = q(x)$ and returns 0 otherwise.

Lemma 3.2 proves that, when the conditions are met, our construction of the scheme is secure because adversaries cannot forge malicious credentials.

Lemma 3.2. *Provided that the number of users does not exceed m, and that not all the guards are malicious, TRA2 is unforgeable and no coalition of users and/or guards can forge valid access-keys.*

Proof. We note that there exist no method to interpolate the polynomial under the distributed control of the trusted guards (i.e., the result of $\sum_{i\notin U} q_i$). This is a fact regardless the computational power available to Adv. In the same way, the probability any call to $\mathcal{O}_{newk}$ to output a new valid access-key is also negligible, unless there were enough keys (points of the polynomial) that could eventually allow to interpolate $q(x)$.

We stress that the best the oracles can return is a guess consistent with the available data. We note that there exists a combinatorial number of other many different guesses that are also consistent. Therefore, the probability of $\mathcal{O}_{succ}$ to return 1 is negligible. □

Let us now recall the definition of perfect secrecy, (or information-theoretic security) [44] that, for any message msg in the space of possible messages $\mathcal{M}$ and related ciphertext c in the space of possible ciphertexts $\mathcal{C}$, implies that the a priori probability of msg is equal to the a posteriori probability of msg given c.

$$P(\mathcal{M} = msg) \equiv P(\mathcal{M} = msg|\mathcal{C} = c)$$

This implies that the ciphertexts reveal no information about the message. All messages are equiprobable for a given ciphertext, making the scheme secure since the attacker has no procedure to obtain additional information even with selected ciphertexts.

Despite the TRA2 protocol not employing encryption or signature methods, if the security conditions are met, there is no procedure to confirm the guesses made by an attacker, therefore TRA2 succeeds in providing security derived entirely from information theory, creating a system where partial information does not reveal anything about the scheme's secrets.

4 Distributed registration, anonymous access

Proposed TRA2 protocol provides an opportunity to anonymously access a resource. This can be helpful in some frameworks where an honest authority can be easily distinguished. Nevertheless, we acknowledge that a central authority, which is assumed honest, could be a limitation in some scenarios. For this reason, we address the removal of the need of such a central authority. This way, our protocols distribute not only the access, but also the registration process.

We define the Distributed Registration Anonymous Access scheme as the following system of seven probabilistic polynomial time algorithms as follows:

- $SysSetup(1^k, N, C) \rightarrow pp$. This algorithm generates the public parameters pp for the scheme. These can be, either the result of an agreement, or randomly generated.
- $DealersSetup(pp) \rightarrow \{q^j\}_{j \in G}$. The algorithm is run by every dealer d_i that, privately, generates an access-key generator $q_{d_i}(x)$. This generator is randomly partitioned into shares, which are privately distributed among the guards. The set of shares of $q(x)$ meet the satisfiability condition described below.
- $GuardsSetup(\{q_i\}_{i \in D}) \rightarrow q^{g_j}(x)$. The algorithm is run by every guard g^j that considers the shares received from the dealers to obtain its own share of the key-access generator $q^{g_j}(x)$.
- $Registration(id, u_x) \rightarrow a_{key}$. The users call this algorithm to ask for a set of partial keys from the set of dealers. If correctly identified, the user receives a set of partial access-keys, the combination of which returns the user's access-key a_{key}.
- $AccessRequest(a_{key}) \rightarrow \{i, k_i\}_{i \in G}$. This algorithm is a call to the set of guards, who generate partial keys k_i according their shares. All the guards are committed to share their results with the rest of guards in G.
- $Combine(\{i, k_i\}_{i \in G}) \rightarrow tk$. The algorithm considers the partial keys computed by the guards and obtains an access-token by combining them.
- $GrantAccess(a_{key}, tk) \rightarrow 1/0$. Checks it the token tk corresponds to access-key a_{key}, in which case the guards agree to grant access (output 1).

Consistency is guaranteed if, for all k, for any valid id and $a_k \leftarrow Registration(id)$, it is hold that:

$$GrantAccess(a_k, Combine(AccessRequest(a_k))) = 1.$$

4.1 Trusted distributed registration, anonymous access

In this section we construct a protocol that distributes the registration and the issuing of access tokens, as well as the access to a resource. To do so, we apply to the access-keys issuance the same principle used in TRA2 scheme to distribute the access control. This permits to replace the central authority by a set D of registration authorities which in the following will be referred to as dealers. Under this trusted distributed registration, anonymous access protocol (TDRA2) dealers are considered honests. The access control will be carried out by N guards that will proceed in the same way they do under TRA2 protocol.

To setup the system, the dealers first agree the modulus p, as well as the m-degree of a polynomial $q(x)$. Once the main parameters are chosen, each dealer d_i independently generates a m grade polynomial $q_{d_i}(x)$. In the same way the access-keys of TRA2 protocol were points of a m degree polynomial, we will consider the polynomial $q(x)$ which results of the sum of all the $q_{d_i}(x)$ polynomials generated by the dealers. Note that $q(x)$ is unknown to each individual dealer.

We note the importance $q(x)$ to be unknown to the guards, which prevents a malicious one to forge new access-keys. To guarantee this, each dealer d_i freely partitions its polynomial $q_{d_i}(x)$ into N complementary polynomials $q_{d_i}^{g_j}(x)$ such that:

$$q_{d_i}(x) = \sum_{j=1}^{N} q_{d_i}^{g_j}(x),$$

then, dealers proceed to securely send one of their shares to each guard. Each guard g_j considers the shares from all the dealers and computes its polynomial from the received shares as:

$$q^{g_j}(x) = \sum_{i=1}^{D} q_{d_i}^{g_j}(x),$$

and from this moment on, guards are ready to accept access access-keys. Figure 2 depicts a simple example of the process.

Note that, taking into account the shares of the polynomial known by the guards and the dealers, the following condition is met:

$$\sum_{i=1}^{D} q_{d_i}(x) = \sum_{i=1}^{N} q^{g_i}(x)$$

that is, dealers and guards consider different shares of the same polynomial $q(x)$.

In order to obtain their access-keys, users send their identification along with a chosen x_u to every dealer. Dealers are responsible for checking that no duplicates of x_u occur, in which case, the user must be asked to provide a different x_u. Each dealer d_i checks the credentials to identify the user, if the

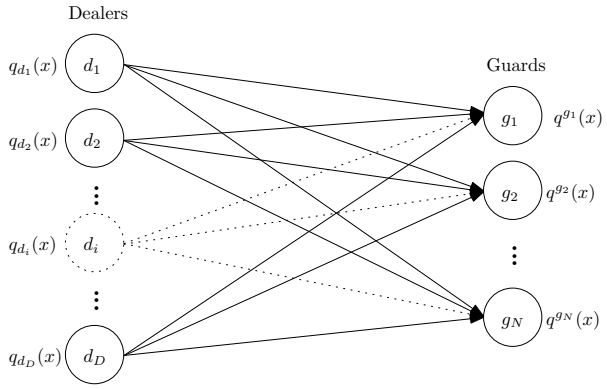

Figure 2: Dealers split their polynomial in N parts and distribute one piece to each guard. Guards use these pieces to construct its own private polynomial.

credentials are valid and x_u is unique, it replies to the user with the result of $q_{d_i}(x_u) \bmod p$, and the users can now compute their access-keys as:

$$\langle x_u, y_u = \sum_{i=1}^{D} q_{d_i}(x_u) \bmod p \rangle.$$

Figure 3, illustrates the process (TDRA2 and TRA2 access stages are the same). Algorithm 2 describes the protocol and Example 4.1 depicts it.

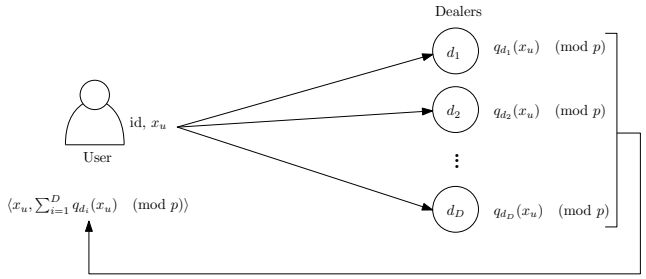

Figure 3: User sends his identification and selected x_u to the dealers. If the identification is valid, dealers respond with the result of applying its polynomial. With these results, the user is capable of independently construct his access token.

Example 4.1. *Let us consider an scenario with two dealers ($D = 2$) and three guards to control the access to some resource ($N = 3$). We consider the same public integers considered in Example 3.1 ($p = 7919$ and $m = 665$). Let also consider the following polynomials (privately) generated by the dealers in order*

Algorithm 2 Trusted distributed registration, anonymous access (TDRA2) protocol

1: System setup
2: (a) Dealers jointly agree on a prime p and m.
3: (b) Each dealer generates an m degree polynomial $q_{d_i}(x)$ modulus p.
4: (c) Each dealer partitions its polynomial into N shares and sends each one to a different guard.
5: (d) Every guard g_j sums the received components to compute its (m degree) polynomial $q^{g_j}(x)$ modulus p.
6: User identification
7: (a) Users send his identification along with a selected x_u to each dealer.
8: (b) Every dealer verifies the identification credentials and check that x_u is unique.
If so, each dealer d replies to the user with $\{x_u, q_d(x_u) \mod p\}$.
9: (d) Users can compute their credential using the received points from the dealers as:

$$\langle x_u, y_u = \sum_{i=1}^{D} q_{d_i}(x_u) \bmod p \rangle.$$

10: Anonymous access
11: (a) User sends his access-keys $\langle x_u, y_u \rangle$ to each one of the guards.
12: (b) Each guard computes $q^{g_j}(x_u)$.
13: (c) Guards check if $y_u = \sum_{1 \le j \le N} q^{g_j}(x_u) \bmod p$.
14: (d) If the access-keys are valid, access is granted.

to generate the access tokens:

$$q_{d_1}(x) = 26x^{665} + 54x^7 + 1 \bmod 7919$$
$$q_{d_2}(x) = 19x^{665} + 22x^{13} \bmod 7919$$

The final set up step implies each dealer to partition his polynomial $q(x)$ into three complementary polynomials to distribute among the guards, for instance:

$$q_{d_1}^{g^1}(x) = 6x^{665} + 1 \bmod 7919$$
$$q_{d_1}^{g^2}(x) = 10x^{665} + 54x^7 \bmod 7919$$
$$q_{d_1}^{g^3}(x) = 10x^{665} \bmod 7919$$

$$q_{d_2}^{g^1}(x) = 19x^{665} \bmod 7919$$
$$q_{d_2}^{g^2}(x) = 10x^{13} \bmod 7919$$
$$q_{d_2}^{g^3}(x) = 12x^{13} \bmod 7919$$

which implies that the polynomials the guards consider are:

$$q^{g_1}(x) = 25x^{665} + 1 \bmod 7919$$
$$q^{g_2}(x) = 10x^{665} + 10^{13} + 54^7 \bmod 7919$$
$$q^{g_3}(x) = 10x^{665} + 12^{13} \bmod 7919$$

A user who sends his identification together with $x_u = 233$ to the dealers will be replied with the values 5048 *and* 6449 *from dealers one and two respectively, and will be able to obtain his access token as:*

$$\langle x_u = 233,\ y_u = 5048 + 6449 \bmod 7919 = 3578 \rangle.$$

Once the guards receive the access token, each one obtains the following results from their polynomial shares:

$$q^{q_1}(233) = 4372; \quad q^{g_2}(233) = 4794; \quad q^{g_3}(233) = 2331$$

and must collaborate in order to check whether the token is valid or not. Actually, $q^{g_1}(x_u) + q^{g_2}(x_u) + q^{g_3}(x_u) = 3578 = y_u \bmod p$, and the token is correct.

This approach avoids the polynomial $q(x)$ to be stored in a single point of failure, and therefore, may lead the users to increase their trust in the system. In Section 4.3 we prove the security of the protocol.

4.2 Anonymous registration, anonymous access

We note that, despite the distribution of the responsibility and the impossibility of the dealers or guards to forge new access-keys, TDRA2 protocol does not suffice to provide complete privacy to the users, because malicious dealer and guard could collude and be able to relate identities and the tokens issued to the users, and, therefore, know the actions carried out by the users once they gain access to the resource.

In order to protect user's privacy during the registration stage, it would be necessary to avoid the possibility of relating user's identity and user's access-keys. In order to achieve this goal, we employ homomorphic cryptography to hide the relationship between identities and access-keys. In the following, we construct a protocol that allows a user to securely hide the components of the access token as long as the discrete logarithm problem remains secure.

The resulting protocol for anonymous registration and anonymous access (ARA2) permits the distribution of the registration among D dealers, while the access control will be carried out by a team of N guards. We note that in this protocol, security is based on the discrete logarithm problem. Thus, in order to prevent a coalition of users to use multiplicative properties to forge new access-keys, a redundancy function is included in the protocol. Algorithm 4.2 describes the whole protocol, but, for the sake of brevity, and without loss of generality, the use of the redundancy function has not been taken into account in the description of the protocol nor in Example 4.2. We will prove in Section

4.3 that, in this protocol, malicious dealers are not able to reveal the identity of the users nor forge new access tokens.

The setup of the scheme implies the dealers to agree a prime p, as well as every dealer d_i to generate a random integer m_{d_i}. Then, each dealer partitions his integer m_{d_i} into N parts $m_{d_i}^{g^j}$ such that:

$$m_{d_i} = \sum_{j=1}^{N} m_{d_i}^{g^j},$$

and distribute each share to the guards through a secure channel. Thus, a guard g_j can compose an integer m^{g^j} as the sum of the received shares from the dealers:

$$m^{g^j} = \sum_{i=1}^{D} m_{d_i}^{g^j}$$

Note that, at the end of the distribution phase, both the set of dealers and the set of guards have different shares of the same secret integer m which is never stored anywhere and result from the sums:

$$m = \sum_{i=1}^{D} m_{d_i} = \sum_{j=1}^{N} m^{g^j}$$

Before the registration, each user generates a pair of private integers v and s such that $vs \equiv 1 \mod (p-1)$. Then, the user u selects an integer x_u and sends his identification together with a pretoken $pt_u = x_u^v \mod p$ to each dealer.

If the identification is valid, then each dealer d_i can compute and reply to the user $pt_u^{m_{d_i}} \mod p$. Note that dealers do not have access to x_u, and therefore, no dealer can track the way the tokens are used by the users. We also note that, for big enough values of p, the probability that two users could generate the same x_u, which would lead two different users to have the same access token, is extremely low[4].

Once received a reply from all the dealers, the user can now compute his token $\langle x_u, y_u \rangle$ where:

$$y_u = \prod_{i=1}^{D} (pt_u^{m_{d_i}})^s \mod p.$$

Note that the process is such as it guarantees that:

$$\begin{aligned} y_u = \prod_{i=1}^{D} (pt_u^{m_{d_i}})^s \mod p = \prod_{i=1}^{D} (x_u^v)^s)^{m_{d_i}} \mod p = \\ = x_u^{\sum_{i=1}^{D} m_{d_i}} \mod p = x_u^m \mod p \end{aligned}$$

where m is an agreed but unknown integer.

[4]Negligible for p values of 1024 bits, which is nowadays a very conservative modular size.

Note that, first, both v and s are private values, enrolled in masking/unmasking processes in an homomorphic cryptography framework; and, second, that the fact that the modulus p is a known prime is not a security issue because: both values v and s will remain secret for everyone but the user who generated them; and, $pt_u = x_u^v \bmod p$ is the only transmitted value which is not enough to reveal the hidden x_u value.

In the access phase the user provides his access token to each guard who can collectively check if it is valid. Example 4.2 illustrates the protocol taking into account artificially low setup values. We summarize the ARA2 protocol in Algorithm 4.2 and Example 4.2 illustrates the procedure.

Example 4.2. *Let us consider an scenario with two dealers ($D = 2$) and three guards ($N = 3$) to control the access of users to some resource. As in previous examples, let $p = 7919$.*

Let $m_{d_1} = 3401$ and $m_{d_2} = 1034$ be the (private) integers generated by the dealers. Each dealer partitions his integers into N shares and (securely) send them to the guards, for instance, as follows:

$$m_{d_1}^{g^1} = 1400 \quad m_{d_1}^{g^2} = 1001 \quad m_{d_1}^{g^3} = 1000$$
$$m_{d_2}^{g^1} = 34 \qquad m_{d_2}^{g^2} = 500 \qquad m_{d_2}^{g^3} = 500$$

and, subsequently, the guards can obtain:

$$m^{g^1} = 1434; \quad m^{g^2} = 1501; \quad m^{g^3} = 1500$$

Consider a user willing to obtain an access token that randomly generates $x_u = 103$, $v = 7717$ and, $s = 1103$. Note that $sv \equiv 1 \pmod{p-1}$. All three values are kept secret. The user then sends his identification together with the pretoken $pt_u = x_u^v \bmod 7919 = 2976$ to each dealer, who reply:

$$pt_u^{m_{d_1}} \bmod p = 1646; \quad pt_u^{m_{d_2}} \bmod p = 5625$$

and the user now can compute his access token as:

$$\langle x_u = 103,\ y_u = 1646^{1103} \cdot 5625^{1103} \bmod 7919 = 4271 \rangle,$$

note that $m = 3401 + 1034 = 4435$, and $103^{4435} = 4271$.

Once the guards receive the access token, each one can work out the (partial) modular exponentiation from its integer share:

$$103^{1434} \bmod 7919 = 4898$$
$$103^{1501} \bmod 7919 = 6001$$
$$103^{1500} \bmod 7919 = 2211$$

and must collaborate in order to check whether the token is valid or not. Indeed, $4898 \cdot 6001 \cdot 2211 \bmod p = 4271 = y_u$, and access should be granted.

ARA2 protocol provides a fully decentralized and anonymous way to obtain access tokens. In Section 4.3 we prove that no coalition of guards and/or dealers and/or users, can reveal the identity of registered users; and, unless the discrete logarithm problem is solved, they are unable to forge new credentials.

4.3 Security analysis

We analyze in this section the security properties of TDRA2 and ARA2 protocols. We will name D the set of dealers, G the set of N guards that control the access to the resource, and C will denote the colective of users that apply for accessing to the resource. We consider that any combination of a subset of dealers, users and/or guards can collude to forge the protocol.

To analyze the unforgeability of TDRA2, for every probabilistic polynomial time adversary Adv, we propose a security game where:

- $SysSetup$ is run to get the m and p values that are given to Adv.
- $DealersSetup$ is run by each dealer that, individually generate an m-degree polynomial mod p, and divide it into shares that safely send to each guard.
- Let $E \subsetneq G$ be a set of corrupt dealers. Give $R = \{q_i\}_{i \in E}$ to Adv.
- $GuardsSetup$ is run by each guard that, individually combine all the polynomials received from the dealers to obtain its own m-degree polynomial mod p.
- Let $U \subsetneq G$ be a set of corrupt guards. Give $P = \{q_i\}_{i \in U}$ to Adv.
- Let $V \subseteq C$ be a set of corrupt users where $|V| < m$. Give the set of access-keys $K = \{Registration(id_i)\}_{i \in V}$ to Adv.
- Let $\mathcal{O}_{interp}(P, R, K)$ be the oracle that, by any method, consider the available information to interpolate either the polynomial:

$$q(x) - \sum_{i \in D} q_i,$$

  or the polynomial:

$$q(x) - \sum_{i \in U} q_i.$$

- Let $\mathcal{O}_{newk}(P, R, K)$ be the oracle that, by any method, consider the available information to obtains new pairs (a, b) such that $b = q(a)$.
- Let $\mathcal{O}_{succ}(p(x))$ be the oracle that returns 1 if $p(x) = q(x)$ and returns 0 otherwise.

Lemma 3.2 proves that, when the conditions are met, our TDRA2 protocol is unforgeable, because, no matter the computational power available to the adversaries, it is not possible to forge malicious credentials.

Lemma 4.3. *Provided that the number of users does not exceed m, and that neither all the guards, nor all the dealers are malicious, TDRA2 is unforgeable and no coalition of users, and/or dealers, and/or guards can forge valid access-keys.*

Proof. We note first that, after the dealers and guards setup, both collectives distributively guard the same secret polynomial $q(x)$. Second, that, individually, dealers and guards only have access to shares of $q(x)$. Third, that the shares from corrupt dealers/guards provided to Adv do not allow to interpolate $q(x)$, because, no matter the computational power, there exists no method to do it. Finally, that any number of access-keys provided to Adv do not change the scenario, unless the number of keys provided exceed the polynomial degree m. Thus, in any case, the best the oracles can return is a consistent guess with the available data of $q(x)$, but with no available method to check the plausibility of the guess against all the other many different guesses that, given any amount of information, are consistent. Therefore, the probability of $\mathcal{O}_{succ}$ to return 1 is negligible.

Summarizing, regardless the computational power available to Adv, there exists no method to interpolate the polynomial under the control of the dealers and guards, and, therefore, it is not possible to forge new credentials unless the number of registered users reaches the grade of the polynomial, or all the dealers/guards collude to forge the system. □

We now analyze the security properties of ARA2 protocol. To do so, for every probabilistic polynomial time adversary Adv, we propose a security game where:

- $SysSetup$ is run to get the m and p values that are given to Adv.
- $DealersSetup$ is run by each dealer that, individually generate an exponent m_{d_i}, and divide it into shares that safely send to each guard.
- Let $E \subsetneq G$ be a set of corrupt dealers. Give $R = \{m_i\}_{i \in E}$ to Adv.
- $GuardsSetup$ is run by each guard that, individually combine all the exponents received from the dealers to obtain its own exponent m^{g_j}.
- Let $U \subsetneq G$ be a set of corrupt guards. Give $P = \{m_i\}_{i \in U}$ to Adv.
- Let $V \subseteq C$ be a set of corrupt users. Give the set of access-keys $K = \{Registration(id_i)\}_{i \in V}$ to Adv.
- Let $\mathcal{O}_{disclog}(P, R, K)$ be the oracle that, by any method, consider the available information to solve the discrete logarithm problem.
- Let $\mathcal{O}_{newk}(P, R, K)$ be the oracle that, by any method, consider the available information to obtains a new pair (a, b) such that $b = a^m \bmod p$.

First, we prove in Lemma 4.4 that the protocol as described protects the users' identity.

Lemma 4.4. *ARA2 protocol ensures users' privacy.*

Proof. Let us stress that the access-key delivered to any user take into account a x_u value which, before sending them to the dealers, is masked using a (private exponent) modular exponentiation. Since the user is the only one who knows the operation to reverse this mask, all the possible values of x_u are equally probable for an *Adv*. Thus, it is not possible to relate a user identity to an access-key, even if an *Adv* intercepts all the dealers' responses to the user.

Guards receive no information about the identity of the users, and they grant access using exclusively the issued access-keys. Therefore, neither a subset of guards, nor a coalition of dealers and guards can track user identities. □

We now prove in Lemma 4.5 that no one can forge new access-keys.

Lemma 4.5. *According ARA2 protocol's description, the probability of forging new access-keys is negligible.*

Proof. The protocol states that a user access-keys is of the form:

$$\langle x_u,\ y_u = x_u^m \bmod p \rangle.$$

In order an adversary to forge a new access-key from an existing one, it would be necessary to partition x_u into two different components a_u and b_u, in order to, afterwards, derive the related a_u^m and b_b^m, that is:

$$\langle x_u = a_u b_u,\ y_u = (a_u b_u)^m \bmod p = a_u^m b_u^m \rangle.$$

We note that, since m is distributed and unknown, there is not enough information to factorize y_u to obtain a_u^m and b_u^m. Nevertheless, if eventually the adversary could carry out this process, it would be necessary, that at least two out of x_u, a_u and b_u, would consider the redundancy function f, which is highly unfeasible.

An adversary could try the following approach. Let him to call algorithm *Registration* for a chosen $x_u = ab$, and for $x_v = a^{-1}b$ as well. Given the respective values $y_u = (ab)^m \bmod p$ and $y_v = (a^{-1}b)^m \bmod p$, let the adversary to proceed as follows:

$$\begin{gathered}\langle x_u x_v \bmod p = b,\ (ab)^m (a^{-1}b)^m \bmod p \rangle \ \Rightarrow \\ \Rightarrow\ \langle b,\ (aba^{-1}b)^m \bmod p = b^{2m} \rangle\end{gathered}$$

from which it is possible to efficiently obtain $\langle b,\ b^m \bmod p \rangle$. Nevertheless, in order the access-keys to agree with the redundancy function, x_u, x_v and b should capture correctly the redundancy established by f, which again is highly unfeasible. □

Previous lemmas prove that, under ARA2 protocol, the identity of the users is preserved even in a post-quantum scenario, and that it is not possible for the partners to forge new access-keys as the discrete logarithm problem remains unsolved. Thus, we conclude that ARA2 protocol is secure.

5 Performance Evaluation

In this section we study the the time behaviour of our three protocols. We analyse in Appendix A the time complexity of the protocols, and show that, for all three protocols, the complexity is $\mathcal{O}(log^3 p)$, where p is the modulus. We also show that this complexity scales linearly with respect the number of users.

We have also implemented a Proof of Concept (PoC) of our protocols in order to prove their feasibility, as well as a complement of our time complexity analysis, because empirical data illustrates aspects that cannot be considered in the theoretical complexity analysis (specially those related to communication issues). The protocols has been implemented in Javascript due to the distributed and asynchronous nature of the processes involved, using the communication layer provided by ZeroMQ concurrency framework[5]. We have implemented the PoC such that, in the experimentation of TRA2 and TDRA2, the higher the number of guards and/or dealers, the higher the number of terms in the polynomials involved. This allows to check also the effect of the size of the polynomial.

For the interested reader, the source code is available in Github[6]. Please note that the code has not been audited and should not be used in production. Furthermore, as a PoC, the implementation has introduced some simplifications (e.g. the consideration of a proxy) that do not affect the results, but reduce the time and resources needed to carry out the experimentation, while allows to capture a similar behaviour to that of the deployed application. The experiments have been run under Linux, in a AMD Ryzen 7 3700X CPU computer (16 cores) with 32 GB RAM. All the different configurations have been run 100 times in order to obtain representative data.

The experimentation carried out is summarized in Figures 4, 5, and 6. The Figures show the total time for the user to have access divided into the time for the user to register, and the time needed for the guards to grant access the resorce. We note the effect of the polynomial in the performance. In this regard, we note that a 5D/8G TDRA2 scenario considers polynomials with 5 times more terms than a 1D/8G TRA2 scenario. In this regard, please note that the protocol could be implemented, without lack of security, to consider smaller polynomials. We also note that, the more guards, the more messages to exchange in order to allow access, and, therefore, the more time needed. The number of terms in the polynomial also plays a relevant role in performance. This is confirmed by some experiments that considered different configurations of dealers/guards and polynomials of the same size (data not shown).

Regarding ARA2, we note that it obtains tolerable delays while providing anonymous access Note also that ARA2 needs extra time to address synchronization issues, due to non-deterministic message ordering between the authorities (also present in other protocols but with lower effect in the overall time).

[5] https://zeromq.org/

[6] https://github.com/Fantoni0/ara2

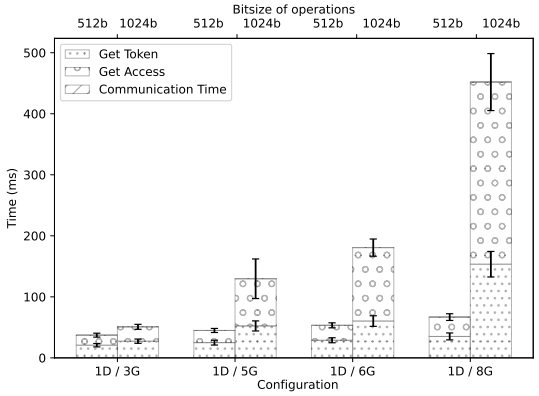

Figure 4: TRA2 protocol. Experimental time results considering different configurations and bitsize of the operations.

6 Applications

As we mentioned in Section 1, our protocols enable multiple applications on different fields. We here show how our anonymous access protocols can be used to create an untraceable blockchain airdrop, as well as an electronic voting scheme. We will assume that we are working with the ARA2 protocol for the anonymous registration property, but the presented procedures can be easily modify to consider the other protocols we propose.

6.1 Blockchain Airdrop System

Airdrops are a common strategy in blockchain, in which projects that aim to increase engagement, gain popularity and/or enlarge market capitalization by distributing free tokens between participants. One of the challenges airdrops need to address is to reward only legitimate and loyal users, avoiding to reward bots, or recently created accounts with the sole purpose of draining the airdrop's liquidity. One of the most used techniques to do it is the use of a whitelist. With this approach, interested participants provide information about their experience and history in blockchain projects, this is later reviewed by airdrop organizers to grant access. The problem with this technique is twofold: requires manual revision, and violates all forms of anonymity of interested participants.

We show below a procedure that shows this can be solved by using our ARA2 protocol, and smart contracts that play the role of dealers. The result preserves and respects anonymity. The procedure is based on participants that prove their activity in the blockchain space enough time to qualify themselves as valid users.

1. Participant generates at random v and s such that $vs \equiv 1 \pmod{p-1}$.
2. Participant chooses a old and valid transaction t from its wallet history.

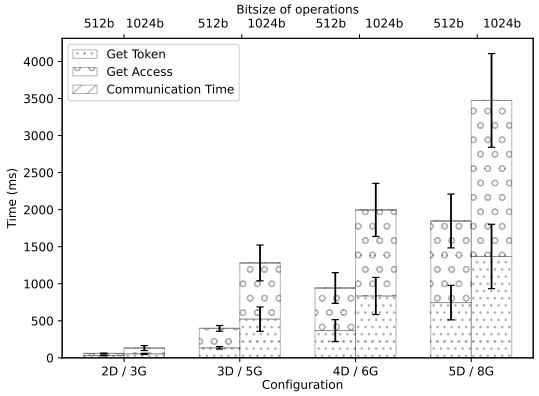

Figure 5: TDRA2 protocol. Experimental time results considering different configurations and bitsize of the operations.

3. Participant signs a message containing t and the address of the airdrop. This will be used as identification $id = sign(t, address)$.
4. Participant chooses x_u at random and masks it as ${x_u}^v \bmod p$.
5. Participant sends the identification and ${x_u}^v \bmod p$ to the airdrop address.
6. The smart contract verifies the identification's signature and the antiquity of t and stops if false.
7. The smart contract returns the access-key to the participant.
8. Participant sends the access-key to the airdrop organizers and a new receiving blockchain address t_n.
9. Organizers check the access-key and stop if false.
10. Organizers send tokens to t_n.

6.2 Electronic Voting Scheme

Electronic voting requires two properties that seem counterintuitive at first: to ensure that no double-voting occurs, while preserving elector's privacy. In order to achieve these two properties, electronic voting schemes usually employ different cryptographic techniques, among which there are the use of blind signatures (e.g. [31]), ring signatures (e.g. [45]), or homomorphic encryption (e.g. [19]).

We present a simple sketch of a possible voting system that achieves both privacy and democracy. We separate the registration and the voting processes, and it implies the existence of an identification authorities that checks the membership of the electors to the census (and also provides the access-keys), and a remote polling station whose access is controlled by controllers that verify

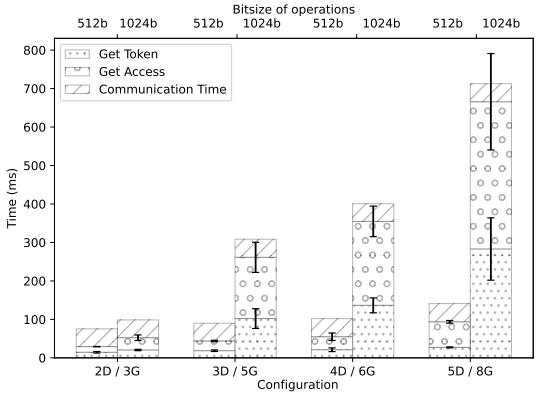

Figure 6: ARA2 protocol. Experimental time results considering different configurations and bitsize of the operations.

access-keys and count votes. The same scheme can be modified to consider a set of identification authorities or another protocol we propose.

1. Elector generates at random v, s such that $vs \equiv 1 \pmod{p-1}$.
2. Elector chooses x_u at random and masks it as ${x_u}^v \bmod p$.
3. Elector sends his personal identification and ${x_u}^v \bmod p$ to the identification authority.
4. Identification authority checks the identification belongs to a valid elector in the census. It stops if false.
5. Identification authority computes the access-key and returns it to the elector.
6. Elector sends his vote and access-key to the controllers that guard the remote polling station.
7. Remote polling station controllers check the validity of the access-key. They stop if false.
8. Controllers grant electors access to the ballot box, and, once finished the election, count and publish the received votes.

7 Conclusions

We here propose three anonymous access protocols. The schemes we propose decouple identification and the actions to carry once the access is granted. They are based on a basic mathematical primitives and allow readily escalation. The

protocols are lightweight and suitable to be implemented on any type of platform, and a wide range of applications. We briefly sketch how they could be useful in blockchain or e-voting applications.

The third protocol we propose cancels the possibility of tracking the identity of a user of the service and the actions carried out once access to the service is granted. Under the other two protocols here proposed, the users identities are accesible only to a set of (assumed honest) authorities that are entitled to issue the access-keys that grant access to the resource. All the protocols we propose allow immediate non-interactive registration, two of them provide post-quantum security, and all of them prevent users' multiple access. As far as we now, they are the first distributed registration schemes with these properties.

The extension of this method in order to allow each credential to be used more than one time is very interesting and we will study it in the future. Another possible extension is to introduce some error-tolerance mechanism or failure suport on the dealers/guards, because the protocols availability is affected when one of the dealers/guards is unreachable. An initial solution can be implemented considering separated sets of dealers/guards, that can be implemented to, either mirror every individual authority, or mirror the whole authority system. In both cases the time needed for the user should not be affected.

We finally note the three protocols offer different solutions adaptable to diverse scenarios. If post-quantum security is desired, then TRA2 or TDRA2 could be the option, while ARA2 is suitable when honesty of authorities could be an issue.

A Time Complexity Analysis

As it is usual, we choose bit operation as the unit in our time-complexity analysis. Since all the operations are carried out modulus a prime p, the complexity of the operations will be expressed in terms of $\log p$. We refer the interested reader to [35] to consult the complexity of modular operations. The presented methods require a variety set of steps and different sequential phases to operate. We do not consider complexity derived from auxiliary procedures (for instance, that from calls to the hash function), nor the time devoted to communication between the parties. Since those functions have usually low complexity, and their use is not extensive, we refer the interested reader to the empirical analysis.

A.1 TRA2/TDRA2 time complexity analysis

Due to the role of the polynomials in these protocols, we first stress that dealing with an m-degree polynomial does not imply to deal with $m+1$ terms. Security is not affected by considering disperse polynomials in which the number of terms is bounded by a constant $t << m$ (and, therefore, also such that $t << p$). We analyze the time complexity of the processes carried out by each partner once the parameters have been setup and the polynomials have been generated.

Note that all these procedures can be carried out off-line and do not affect the complexity of the whole process.

First, regarding the dealers, for any given user registration request, each dealer d computes the result of his share of the polinomial, that can be computed with $\mathcal{O}(t \log^3 p) \approx \mathcal{O}(\log^3 p)$. Second, regarding the users, once they have received all the messages from the dealers, they simply carry out an addition modulus p, process with time complexity $\mathcal{O}(t \log p) \approx \mathcal{O}(\log p)$. Finally, regarding the guards, each one has to compute the result of his own share of the polynomial, with time complexity $\mathcal{O}(\log^3 p)$, and then collaboratively add the partial results of the rest of guards to grant or revoke the access, with time complexity $\mathcal{O}(n \log p) \approx \mathcal{O}(\log p)$. The whole identification process is a sequence of these procedures, therefore, for any single user requesting access, the complexity of the identification process is $\mathcal{O}(\log^3 p) + \mathcal{O}(d \log p) + \mathcal{O}(n \log^3 p) \approx \mathcal{O}(\log^3 p)$. Let us note that, regarding the time complexity analysis, TRA2 can be considered as TDRA2 with $d = 1$. Note also that the number of bit operations scales linearly with the number of users in the scenario.

A.2 ARA2 time complexity analysis

Again, we will analyze the time complexity of the processes carried out by each partner once the parameters have been setup and the polynomials have been generated.

Regarding the time complexity of the procedure carried out by ARA2 dealers, they only need to compute a single modular exponentiation, with complexity $\mathcal{O}(\log^3 p)$. Regarding the users, to mask the token implies a modular exponentiation, and, to reconstruct the access-key from the received shares implies the product of d values and a modular exponentiation, with time complexity $\mathcal{O}(d \log^2 p) + \mathcal{O}(\log^3 p) \approx \mathcal{O}(\log^3 p)$. Finally, each guard carry out a modular exponentiation to compute his partial result, with complexity $\mathcal{O}(\log^3 p)$, and then a set of n multiplications modulus p to combine all the partial results from all the guards in the collective, with complexity $\mathcal{O}(\log^3 p) + \mathcal{O}(n \log^2 p) \approx \mathcal{O}(\log^3 p)$. Therefore, for any single user requesting access, the complexity of the whole identification process is $\mathcal{O}(\log^3 p) + \mathcal{O}(\log^3 p) + \mathcal{O}(\log^3 p) \approx \mathcal{O}(\log^3 p)$, and it is possible to see that the number of bit operations scale linearly with the number of users.

References

[1] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In International Workshop on Public Key Cryptography, pages 65–84. Springer, 2005.

[2] Shashank Agrawal, Peihan Miao, Payman Mohassel, and Pratyay Mukherjee. PASTA: password-based threshold authentication. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications

Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018, pages 2042–2059. ACM, 2018.

[3] Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, and Llanos Tobarra. Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps. In Proceedings of the 6th ACM workshop on Formal methods in security engineering, pages 1–10, 2008.

[4] Ahmed Ben Ayed. A conceptual secure blockchain-based electronic voting system. International Journal of Network Security & Its Applications, 9(3):01–09, 2017.

[5] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable proofs and delegatable anonymous credentials. In Annual International Cryptology Conference, pages 108–125. Springer, 2009.

[6] Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security volume 2248 of Lecture Notes in Computer Science, pages 566–582. Springer, 2001.

[7] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In International conference on the theory and applications of cryptographic techniques, pages 139–155. Springer, 2000.

[8] Jorge Bernal Bernabe, Martin David, Rafael Torres Moreno, Javier Presa Cordero, Sébastien Bahloul, and Antonio Skarmeta. Aries: Evaluation of a reliable and privacy-preserving european identity management framework. Future Generation Computer Systems, 102:409–425, 2020.

[9] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In Annual international cryptology conference, pages 213–229. Springer, 2001.

[10] Dan Boneh, Kevin Lewi, Hart Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. In Annual Cryptology Conference, pages 410–428. Springer, 2013.

[11] Jan Camenisch, Anja Lehmann, Anna Lysyanskaya, and Gregory Neven. Memento: How to reconstruct your secrets from a single password in a hostile environment. IACR Cryptol. ePrint Arch., 2014:429, 2014.

[12] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In International conference on the theory and applications of cryptographic techniques, pages 93–118. Springer, 2001.

[13] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In Proceedings of the 9th ACM conference on Computer and communications security, pages 21–30, 2002.

[14] David Chaum. Blind signatures for untraceable payments. In Advances in cryptology, pages 199–203. Springer, 1983.

[15] David Chaum. Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM, 28(10):1030–1044, 1985.

[16] David Chaum. Zero-knowledge undeniable signatures. In Advances in Cryptology - EUROCRYPT '90, Workshop on the Theory and Application of of Cryptographic Techniques , pages 458–464, 1990.

[17] David Chaum and Jan-Hendrik Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. In Conference on the Theory and Application of Cryptographic Techniques, pages 118–167. Springer, 1986.

[18] Lidong Chen. Access with pseudonyms. In International Conference on Cryptography: Policy and Algorithms, pages 232–243. Springer, 1995.

[19] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. European transactions on Telecommunications, 8(5):481–490, 1997.

[20] Ivan Bjerre Damgård. Payment systems and credential mechanisms with provable security against abuse by individuals. In Conference on the Theory and Application of Cryptography, pages 328–335. Springer, 1988.

[21] Alfredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung. How to share a function securely. In Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, pages 522–533, 1994.

[22] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In Gilles Brassard, editor, Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, volume 435 of Lecture Notes in Computer Science, pages 307–315. Springer, 1989.

[23] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. Journal of Cryptology, 1(2):77–94, 1988.

[24] Michael J Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword search and oblivious pseudorandom functions. In Theory of Cryptography Conference, pages 303–324. Springer, 2005.

[25] Rosario Gennaro, Shai Halevi, Hugo Krawczyk, and Tal Rabin. Threshold rsa for dynamic and ad-hoc groups. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 88–107. Springer, 2008.

[26] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. SIAM Journal on Computing, 25(1):169–192, 1996.

[27] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In Proceedings of the 17th Annual ACM Symposium on Theory of Computing, pages 291–304, 1985.

[28] Dick Hardt et al. The oauth 2.0 authorization framework. Technical report, RFC 6749, October, 2012.

[29] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. Highly-efficient and composable password-protected secret sharing (or: How to protect your bitcoin wallet online). In 2016 IEEE European Symposium on Security and Privacy (EuroS&P), pages 276–291. IEEE, 2016.

[30] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Efficient password-authenticated key exchange using human-memorable passwords. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 475–494. Springer, 2001.

[31] Antonio M Larriba, José M Sempere, and Damián López. A two authorities electronic vote scheme. Computers & Security, 97:101940, 2020.

[32] Chun-Ta Li, Min-Shiang Hwang, and Yan-Chi Lai. A verifiable electronic voting scheme over the internet. In 2009 Sixth International Conference on Information Technology: New Generations, pages 449–454. IEEE, 2009.

[33] Anna Lysyanskaya, Ronald L Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In International Workshop on Selected Areas in Cryptography, pages 184–199. Springer, 1999.

[34] Philip D. MacKenzie, Thomas Shrimpton, and Markus Jakobsson. Threshold password-authenticated key exchange. In Moti Yung, editor, Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, volume 2442 of Lecture Notes in Computer Science, pages 385–400. Springer, 2002.

[35] Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996.

[36] Rafael Torres Moreno, Jorge Bernal Bernabe, Antonio Skarmeta, Michael Stausholm, Tore Kasper Frederiksen, Noelia Martínez, Nuno Ponte, Evangelos Sakkopoulos, and Anja Lehmann. Olympus: Towards oblivious identity management for private and user-friendly services. In 2019 Global IoT Summit (GIoTS), pages 1–6. IEEE, 2019.

[37] Moni Naor, Benny Pinkas, and Omer Reingold. Distributed pseudo-random functions and kdcs. In International Conference on the Theory and Applications of Cryptographic Techniques, pages 327–346. Springer, 1999.

[38] OASIS Security Services TC. Security Assertion Markup Language. `https://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf`, 2008. Online; accessed 20 January 2021.

[39] Mario Di Raimondo and Rosario Gennaro. Provably secure threshold password-authenticated key exchange. In Eli Biham, editor, Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, volume 2656 of Lecture Notes in Computer Science, pages 507–523. Springer, 2003.

[40] David Recordon and Drummond Reed. Openid 2.0: a platform for user-centric identity management. In Proceedings of the second ACM workshop on Digital identity management, pages 11–16, 2006.

[41] Eric Rescorla. The transport layer security (TLS) protocol version 1.3. RFC, 8446:1–160, 2018.

[42] Adi Shamir. How to share a secret. Communications of the ACM, 22(11):612–613, 1979.

[43] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, Advances in Cryptology, Proceedings of CRYPTO '84, volume 196 of Lecture Notes in Computer Science, pages 47–53. Springer, 1984.

[44] Claude E. Shannon. Communication theory of secrecy systems. Bell Syst. Tech. J., 28(4):656–715, 1949.

[45] José Luis Tornos, José Luis Salazar, Joan Josep Piles, Jose Saldana, Luis Casadesus, José Ruíz-Mas, and Julián Fernández-Navajas. An evoting system based on ring signatures. Network Protocols & Algorithms, 6(2):38–54, 2014.

Algorithm 3 Anonymous registration anonymous access protocol (ARA2).

System setup

(a) Dealers jointly agree on a prime p and m.

(b) Dealers agree a redundancy function f. //e.g. a hash function

(c) Each dealer computes its own degree m_{d_i}.

(d) Each dealer decomposes its own degree m_{d_i} and sends a component to each guard.

(e) Every guard g_j adds the received components to compute its own share m^{g_j}.

User identification

(a) Each user generates two private values s and v such that $sv \equiv 1 \pmod{p-1}$.

(b) Users generate a random value r and set x_u as the concatenation of r and $f(r)$.

(c) Users send his identification along with $pt_u = x_u^v$ (the masked x_u) to each dealer.

(d) Dealers verify the identification credentials.

(e) If the credentials are valid, each dealer d_i replies to the user with:

$$pt_u^{m_{d_i}} \bmod p.$$

(e) Users can compute their credential (point) from the received points from the dealers as:

$$\langle x_u,\ y_u = \prod_{i=1}^{D} (pt_u^{m_{d_i}})^s \bmod p \rangle.$$

Anonymous access

(a) User sends his access-keys $\langle x_u, y_u \rangle$ to each one of the guards.

(b) Each guard check that x_u capture the redundancy correctly.

(b) Each guard computes $(x_u)^{m^{g_j}} \bmod p$.

(c) Guards check if $y_u = \prod_{1 \le j \le N} (x_u)^{m^{g_j}} \bmod p$.

(d) If the access-keys are valid, access is granted.