HAL
open science

Get-your-ID: Decentralized Proof of Identity

Pascal Lafourcade, Marius Lombard-Platet

To cite this version:

Pascal Lafourcade, Marius Lombard-Platet. Get-your-ID: Decentralized Proof of Identity. International Symposium on Foundations & Practice of Security, Nov 2019, Toulouse, France. hal-02412797

HAL Id: hal-02412797

https://hal.science/hal-02412797v1

Submitted on 15 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Get-your-ID: Decentralized Proof of Identity

Pascal Lafourcade[1] and Marius Lombard-Platet[2,3]

[1] Université Clermont-Auvergne, LIMOS CNRS UMR 6158, Aubière, France
pascal.lafourcade@uca.fr
[2] Be-Studys, Geneva, Switzerland
[3] Département
d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France
marius.lombard-platet@ens.fr

Abstract. In most systems without a centralised authority, users are free to create as many accounts as they please, without any harmful effect on the system. However, in the case of e-voting, for instance, proof of identity is crucial, as sybil identities can be used to breach the intended role of the system. We explore the conditions under which a decentralised proof of identity system can exist. We also propose such a scheme, called Get-your-ID (GYID), and prove its security. Our system allows a user to generate and revoke keys, via an endorsement mechanism, and we prove that under some conditions which we discuss, no user can have more than one active key. We then show how voting protocols can be adapted on top of our system, thus ensuring that no user is able to cast a valid vote more than once.

Keywords: Identity System·Decentralized Ledger·Protocol Security·E-voting.

1 Introduction

User account management is a long-time subject of research in computer science. In most cases, the effort is made on account security rather than account unicity: it does not matter that a user has several accounts, as long as no one else can use these accounts. However, in some applications, it is instead required that a user cannot own more than one account. Such applications include e-voting protocols, where a physical person can only vote once (or a definite number of times), and universal income protocols, such as Duniter[**?**].

Preventing these duplicate accounts (also called Sybil accounts) is usually done by referring to a central authority. For instance, by checking a user's ID card, issued by a central government. However, the central authority may deny people's right to get an ID card, sometimes for political or simply technical reasons. Such events have already occurred in 2017 in the Philippines[4], and in 2018 in the United States[5], for instance. In any case, the failure of the central authority to reliably deliver ID to any citizen complying with the rules means that no one, organizing an online vote (or any other application relying on unique accounts), can have perfect faith in the central authority. With a decentralised system, one can trust the honesty of the network, rather than necessarily trusting the honesty of the central user.

[4] https://www.asiatimes.com/2017/11/article/chinese-filipina-denied-identity-card-tells-struggle/

[5] https://www.aclu-nm.org/en/press-releases/groups-seek-immediate-order-stop-states-illegal-denial-non-real-id-licenses-and

As a matter of fact, blockchain-based votes have already been used in real elections, in Issy-les-Moulineaux[6] (France) in August 2018, and Tsukuba[7] (Japan) in September 2018. However, details about these implementations are very sparse and do not allow for security audit. Furthermore in both cases the city hall was in charge of establishing the voters list, which is paradoxical considering that the blockchain is supposed to enhance decentralised applications: the system still relies on the trust of the city hall. As such we introduce the concept of proof of identity: a guarantee that a user only owns at most one valid identity.

Related Work: A first draft for proofs of identity has been proposed in [**?**], but the paper lacks any practical protocol description. A notable work is the Web of Trust, a network of peer certification that has been introduced for PGP.The concept of Web of Trust is the main idea on which relies the blockchain Duniter. However, it is possible (and easy) to obtain several distinct identities in this blockchain.

Despite these caveats, a protocol of proof of personhood has been described in [**?**], in which the physical existence of a user is verified. The certification process is filmed and published online for public verification. However, proof of personhood does not equate proof of identity, as one user can register several times at different parties.

Finally, the idea of storing cryptographic keys on a blockchain is by no means new, as a matter of fact, blockchain PKI has already been proposed in [**?**], but simply as a decentralized ledger rather than a new opportunity of building a network of trust and managing identities.

Our approach: We propose a new way of managing identities, in which no user can own more than one non-revoked public key. For auditability reasons, the transactions are public and stored in a blockchain. In order to avoid relying on an authority for deciding who is eligible to the system, we provide a protocol in which a user must be co-opted by several other users, thus allowing a user to bypass possible censorship by a central authority.Hence we propose Get-your-ID, our novel identity management protocol.

Contributions: In this paper, we present several results. First, we offer a formalisation of the problem, giving a clear definition of the tools we use and a formal description of the real-life hypotheses we make. Under these hypotheses, we show that it is impossible to have a proof of identity system without members having more power than others.

Then, we place ourselves in the context where trusted members are allowed, allowing us to build a proof of identity system, in which a user can get at most one valid identity at all times. Any user can obtain a validity through an endorsement process, and for privacy, all endorsements are secret until the user decides to reveal them. In our model, each user can have the same role, i.e. no user has any privilege. We give a full description of our protocol.

Outline: In Section 2, we define the terms we use in the rest of the paper. We then prove that under reasonable assumptions, no verifiable decentralized proof of identity can exist. Furthermore, we exhibit a decentralized protocol of proof of identity, where as much as possible data is publicly verifiable, and we provide a security proof of the protocol. Finally, we discuss about compatibility of our protocol with voting protocols and further improvements.

---

[6] `http://issy.com/taxonomy/term/817/issy-les-moulineaux-va-tester-le-vote-electronique-avec-la-blockchain` (in French)

[7] `https://www.japantimes.co.jp/news/2018/09/02/national/politics-diplomacy/new-online-voting-system-introduced-city-tsukuba/`

2 Preliminaries

2.1 Biometric Authentication

The topic of biometric authentication has been an active field of research in the last decade. Notably, several different biometrics methods have been surveyed in [**?**]. From various biometric methods, we can cite DNA, iris and retina patterns, palm and fingerprints, but also more exotic methods such as facial thermogram, gait analysis or body odor.

Authors of [**?**] also insist on the fact that no method is perfect, and that one should choose a system with several biometric methods, in order to compensate for the deficiencies of each method. For instance, DNA authentication has a very high distinctiveness rate (only twins can reasonably produce duplicates), but its extraction is currently still costly and time-consuming. Faster methods may be, for instance, iris recognition or face recognition.

2.2 Proof of Identity Concepts

We now introduce the concept of proof of identity. The underlying idea is that any physical person is uniquely identified by a set of properties, most likely biometric properties, that is unforgeable and unique to the user. We argue in the next sections that these properties are the only ones that can be used for an identity system, and we explore the limitations around them. Proof of identity system is not an application that is designed to run for itself; rather is it designed to be the support of many other applications. We intend to build an identity repository, that other applications can use for their own purposes. Such other applications can be for instance an e-voting application, a universal income scheme, both or anything else.

Definition 1 (Essential properties). *A set of properties that uniquely define any user is called a set of essential properties. The set of essential properties of the user U is noted P_U.*

For instance, it is most probable that the DNA and fingerprints of a user uniquely define them. We call this set the **essential properties**.

Definition 2 (Digital Fingerprint). *If a user U has for essential properties P_U, then its digital fingerprint F_U is defined as $F_U = F(P_U)$ where F is a hash function*[8].

For both security and efficiency, we use the digital fingerprint rather than the essential properties. For instance, in case of DNA authentication, the data can be several gigabytes big, and also sensitive data. Moreover, errors can occur during the acquisition of biometric data. This inherent fuzziness can be lessened with several techniques described in [**?**]. Put together, these techniques can be incorporated in F.

For these reasons, the raw use of biometrics is neither practical nor secure. On the other hand, a hash of these data is short and does not leak any information since it is preimage resistant. Finally, users are still uniquely defined by their digital fingerprint because the hash function is collision resistant, so it is extremely unlikely that two user's essential fingerprint hashes shall collide.

[8] We require that it is hard to find a collision on F. In the random oracle model, this is an immediate consequence, but in more general models, we say that F is collision-resistant (see [**?**]).

3 Impossibility Theorem

We assume the following hypotheses to be true:

Hypothesis 1 *The essential properties rely on physical properties that the system stores in digital memory. These physical properties can be considered as PUFs.*

Hypothesis 2 *In order to be sure that a digital fingerprint maps to a physical person, a user must compute the digital fingerprint by herself. Notably, she must acquire the biometric data herself, with a device she trusts (home made or from a trusted manufacturer). We further assume that the user must be physically next to the person in order to obtain their essential properties.*

Hypothesis 3 *The current state of the art does not allow for an acquisition of fewer than 10 seconds on average, travel time included.*

Hypothesis 4 *A system with N users requiring at least $N \cdot t$ amount of time from each user individually is not scalable.*

From these hypotheses, we get the following result:

Theorem 1 (Impossibility Result). *There exists no system in which each user can verify by herself the validity of all the digital fingerprints.*

Proof. If we want the system to be fully verifiable, then any user is able to verify every other user's digital fingerprint. Given that the essential properties are physical (hypothesis 1) and that the verifying user must acquire the data by themselves (hypothesis 2). For this reason, each user will require $O(N)$ of their time for verifying each other user. Even worse, a user can be solicited for verification $O(N)$ times.

As a conclusion, in such a system, a user must dedicate $O(N)$ of their own time for being verified by other users. If the user also wants to verify other users, they must dedicate another $O(N)$ amount of time. Given hypothesis 3, this is not scalable as this amount of time grows unrealistic with N. □

For instance, assuming a system of 1,000,000 users, a verification process (according to hypothesis 4) requires 16 months of 40-hours weeks purely dedicated to the verification process. The 1,000,000 value is arbitrary but represents well the number of inhabitants in a country, according to 2017 UN censuses[9,10], and as such is a relevant threshold for any nationwide system. Even though this result was quite straightforward, it leads to the following conclusion:

Corollary 1. *Scalable, decentralized and verifiable proof of identity systems do not exist.*

The next conclusion is that in a (scalable) identity system, a user must trust at least one person, which we call an authority, to operate at least some of the verifications. A system in which all users trust one person is very close to the current governmental system, where all citizen trust the government (although many people in the government have the power to issue identities). However, such a system is not perfect, as the trust is centralized: a user needs to be sure the authority will never turn malicious.

[9] `https://population.un.org/wpp/Download/Standard/Population/`

[10] In 2017, 158 countries out of the 195 (81%) UN member or observer countries have more than 1 million inhabitants.

4 Get-your-ID Protocol

Even though a 'fully verifiable' solution (in which each user has a proof of the validity of the system) cannot exist, we propose a solution, that we call Get-your-ID, in which a user does not need to trust a central authority, but that allows any user to become an authority, and as importantly, in which no authority can craft a malicious identity without collusion. Moreover, as authorities may want to preserve some kind of monopoly, we provide a system in which a user may keep their endorsements secret until they decide to unveil them. Thus, no authority can prevent a user to become an authority as well.

Note that our protocol requires some participants to gather at the same place at a given time (for signing parties). Even though some might find this solution impractical, we remark that because a system only allowing for a single identity per user must somehow rely on the physical properties of said user, and because there is no clear possibility of securely obtaining these physical properties in an other way than a physical meeting. Hence, we argue that the signing parties are a necessary part of our protocol.

4.1 Intuition of the protocol

A user is able to get endorsements on their identity. The endorsements are secret until disclosed by the user. After some number of endorsements, the user reveals said endorsements, which are verified by the network before being accepted as such. If they are valid, the user is granted an active identity on the system. Endorsements can only be delivered by special users, which we call authorities. An authority verifies that the user is who she claims to be (by verifying their essential properties) before handing them an endorsement. Any user can become an authority, simply by gathering some number of endorsements (this threshold being higher than the threshold to get an identity)[11]. Because authorities may want to keep their monopoly, endorsements are made secret until a user decides to reveal them. Thus, authorities have no way of knowing who is close to become an authority (thus having no incentive to deviate from the protocol).

For security reasons, endorsements are only valid when they are carried during what we call a signing party: several authorities announce they will host a party at a given place and date. A user then has to get endorsements from all (or at least some fraction f, with $f > 0.5$) of the present authorities. We require a majority of endorsements, so that dishonest authorities cannot influence the signing party, as long as they are in minority . Endorsements are recorded in public blockchain. The blockchain behaves like a list of publications. Each publication is given a unique ID by the blockchain, that can be considered as an index.

Each transaction written on the blockchain carries some new data; for simplicity, we assume that our ledger is composed of five registries. One stores announcements, another stores endorsements, another one the *active keys* of users with a valid identity, the last registry stores the *revoked keys*, the identities that have been revoked (compromise, loss...). The fifth registry stores the authority keys.

Note that the authority keys are, in our protocol, user keys with enough endorsements. Therefore, the fifth registry only contains keys that are also stored in the user registry, and is used for performance reasons: without this registry, each user would need to verify that a key is indeed an authority by checking, from the endorsement registry, if there are enough endorsements.

[11] This definition leads to an initialization problem: at the beginning, how can we endorse a user if there is no authority? We suppose that a given number of authorities are named as such when the blockchain is first created, in the genesis block.

4.2 Description of Get-your-ID

We use a hash function H that we consider inside the random oracle model and an existential-forgery resistant signing function Sign. Each user U possess a keypair $(\mathsf{sk}_U, \mathsf{pk}_U)$. We also use two distinct messages ENDORSE,REVOKE, and the existence of a uniform random number generator, whose outputs are more than 120 bits long, in accordance to current NIST's security recommendations [**?**].

The blockchain ledger is consisting of five registries[12]: an authority key registry, an active key registry, a revoked key registry, an announcement registry and an endorsement registry. Depending on the nature of the message, a user will write on either of these registries. In each of these registries, one entry will automatically be assigned a unique integer id.

Definition 3 (Authority). *An authority is a user who has been designated as an authority in the genesis block, or whose public key has been validly endorsed by at least S distinct authorities.*

As mentioned earlier, given that a user must gather some endorsements by an authority before becoming an authority themselves, some authorities must be generated during the initialisation of the blockchain (in the genesis block). When several authorities decide to gather in the same physical place at a given date and time, the event is called a **signing party**. In this signing parties, any user can receive endorsements on their public key.

Definition 4 (Endorsement). *An endorsement of a user U by an authority A_i is a claim made from A_i that they have verified U's essential properties. An endorsement of U is valid if at least a fraction f of the authorities present at the signing party has endorsed U.*

Definition 5 (Active key, Revoked key). *A public key* pk *is considered as active when there is an entry* $(F,\mathsf{pk},proof)$ *in the active key registry, with F a digital fingerprint, but no entry* pk *in the revoked key registry.*[13] *If* pk *is present in the revoked key registry, then the key is considered revoked. The key* pk *is linked to a user U of digital fingerprint F_U if* (F_U,pk) *is in the active key registry.*

Our proof of identity protocol has three main steps, namely the endorsement step, the registration step, and the revocation step.

Endorsement Step. In order to receive endorsements, a user must present herself to signing parties, where authorities will endorse the user's key.

Announcement. First, any group of authorities $A_1,\ldots,A_n$ can decide to host a signing party at the place of their convenience. They broadcast their intent in the form of a message Event$=$ $([\mathsf{pk}_{A_1},\ldots,\mathsf{pk}_{A_n}]), place, date)$, signed by all authorities. $[\mathsf{pk}_{A_1},\ldots,\mathsf{pk}_{A_n}]$ is the list of the authorities $A_1,\ldots,A_n$'s public keys, and *place* and *date* are indications about where and when the event will be held. The announcement, being stored in the blockchain, is assigned an id *party_id*.

[12] Registries are an abstraction that helps to separate data. It can be implemented by a 3-bits prefix on each transaction message.

[13] *proof* is a proof that the user with digital fingerprint F_U has indeed received enough endorsements on their public key pk_U. We give more details about this proof later on.

Individual endorsement. Any user can join the signing party. If a user wants to have the public part pk_U of her keypair $(\mathsf{sk}_U,\mathsf{pk}_U)$ endorsed, she proceeds as follows, for each authority A_i. Given that U and A_i are physically at the same place, we assume the existence of a confidential channel between both entities, i.e. a channel that cannot be eavesdropped or altered. Which is immediate to achieve as the user can directly interact with the authority's measurement tools.

1. U generates a random number r and sends $\mathsf{Sign}(\mathsf{sk}_U,\mathsf{pk}_U||r)$ to A_i.
2. A_i receives $\mathsf{Sign}(\mathsf{sk}_U,\mathsf{pk}_U||r)$, checks the validity of the signature and collects U's essential properties P_U. If the signature is incorrect, then emit an error and stop. Otherwise, A_i transforms P_U into U's fingerprint F_U.
3. A_i sends U her digital fingerprint F.
4. A_i broadcasts the endorsement $\mathsf{Sign}(\mathsf{sk}_{A_i},\ \mathit{party_id}||\mathsf{ENDORSE}||H(F_U||\mathsf{pk}_U||r))$.

An emitted endorsement is only accepted by the blockchain if it is emitted by an authority, and if the endorsement comes on the day of the related signing party, by one of the authorities who signed the party announcement. A schema of this protocol is shown in Fig. **??**.

Key activation. Once a user has gathered the required number of endorsements for her key pk_U, she broadcasts her intention of activating her key. For activating her key, both following conditions must be met:

- U discloses at least T valid endorsements by at least T distinct authorities. In order for the network to determine whether the endorsements are valid or not, U must provide at least $f \cdot N$ endorsements for each signing party with N authorities.
- No active key is currently linked to U.

Disclosing an endorsement is made by announcing F_U,pk_U,r and the id of the endorsement on the blockchain. For instance, if an endorsement $E = \mathsf{Sign}(\mathsf{sk},\mathit{party_id}||\ \mathsf{ENDORSE}||H(F_U||\mathsf{pk}_U||r))$ for some private key sk has been broadcast during a signing party, the blockchain automatically assigns an id to the endorsement. Then U claims ownership of the endorsement by publishing id,F_u,pk_U,r. Given that an endorsement is valid only if at least $f \cdot N$ of the N authorities hosting the signing party have indeed endorsed the same user, U has to prove ownership of at least $f \cdot N$ endorsements for each signing party-related endorsement she wishes to use.

Even though U can own several endorsed keys, only one of them can be considered valid at a given time. For this reason, before adding a new key to the valid keys registry, the network checks that no other active key is linked to F_U. Fig **??** summarizes these exchanges.

Authority activation. Similarly, in order to become an authority, a user discloses at least T endorsements. An authority being a user, the endorsed key must be an active key.

Key Revocation. Key revocation is critical for most systems. Due to this, a user must be able to revoke their current active public key, even if they have lost the matching private key. Classical key revocation mechanisms require knowledge of the private key, as is the case in GPG, for instance. We argue that such a protocol is not suited for a wide public adoption, as most users will not generate their revocation message until too late. Given that Get-your-ID does not allow several accounts to be owned by the same user, such a mistake would be permanent. Therefore, we propose another key revocation protocol, in which the user does not need to store anything for revocation purposes.

During a signing party held by authorities $A_1,\ldots,A_n$, a user U wanting to revoke her active key pk_U will proceed as follows:

1. U computes a random number r and sends REVOKE,pk_U,r to A_i.

2. A_i receives $\mathsf{REVOKE}, \mathsf{pk}_U, r$ and collects U's essential properties P_U, and transforms it into U's fingerprint F_U.
3. A_i publishes an endorsement of the revocation $\mathsf{Sign}(\mathsf{sk}_{A_i}, party_id \| H(F_U \| \mathsf{pk}_U) \| \mathsf{REVOKE} \| r))$.

When U has more than R revocation endorsements, she can send a new entry to the revocation registry, consisting of their active public key, F_U, the revocation endorsement ids and the random numbers used in each endorsement, similarly as what she did for the key endorsement.

Before accepting the revocation the network checks that the key pk_U is indeed an active key, mapped to F_U and that the revocation endorsements provided by the user are valid, i.e., that these were emitted during a signing party by the organizing authorities, and that for each party at least a fraction f of the authorities have emitted a similar statement.

5 Conclusion

In this paper, we showed that even though a fully verifiable decentralized proof of identity is impossible for all practical applications, we can still achieve a proof of identity with a maximum amount of decentralization and verifiability, if we defer some of the verification to what we called authorities. We thus described Get-your-ID, a protocol in which no user can have more than one account (one active key), and we proved that voting protocols are immediately pluggable on top of our system, thus allowing an effective decentralized voting protocol.

Even though perfect prevention against sybil identities is unlikely to exist, we point several improvements that can be made on our protocol, that could be investigated in some future work.