Proof-of-Personhood: Redemocratizing Permissionless Cryptocurrencies

Maria Borge, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Bryan Ford
École Polytechnique Fédérale de Lausanne (EPFL)
{maria.borgechavez, eleftherios.kokoriskogias, philipp.jovanovic, linus.gasser, nicolas.gailly, bryan.ford}@epfl.ch

***Abstract*—Permissionless blockchain-based cryptocurrencies commonly use proof-of-work (PoW) or proof-of-stake (PoS) to ensure their security, e.g. to prevent double spending attacks. However, both approaches have disadvantages: PoW leads to massive amounts of wasted electricity and re-centralization, whereas major stakeholders in PoS might be able to create a monopoly. In this work, we propose proof-of-personhood (PoP), a mechanism that binds physical entities to virtual identities in a way that enables accountability while preserving anonymity. Afterwards we introduce PoPCoin, a new cryptocurrency, whose consensus mechanism leverages PoP to eliminate the disadvantages of PoW and PoS while ensuring security. PoPCoin leads to a continuously fair and democratic wealth creation process which paves the way for an experimental basic income infrastructure.**

***Keywords*-proof-of-personhood; proof-of-work; proof-of-stake; blockchain; cryptocurrency;**

I. Introduction

One of the main distinguishing features among permissionless cryptocurrencies, such as Bitcoin [11], is the way they enable open participation in the consensus mechanism while ensuring resistance against Sybil attacks [7]. Bitcoin and many of its offspring use proof-of-work (PoW) mechanisms [5] to obtain the above properties and allow pseudonymous, untrusted, external actors to securely extend the blockchain. However, PoW requires costly special-purpose hardware and consumes massive amounts of electricity. This has led to a re-centralization since only a few privileged entities who have access to the necessary resources are capable to mine, whereas regular users who can not afford such hardware and its maintenance are excluded. Consequently, the control over the entire system rests in the hands of a small number of elite users, for example as in Bitcoin; an undemocratic approach.

Proof-of-stake (PoS), where participants use their assets (coins) to create (mint) new assets, is another approach that promises similar properties as PoW but consumes far less energy. However, PoS is essentially nothing but a shareholder corporation where the rich again have an advantage as they possess more assets and thus are able to mint new coins faster than less-privileged participants. As a consequence, the (already) rich become even richer; again, an undemocratic approach.

Our goal in this paper is to create a cryptocurrency that provides not only resistance against Sybil attacks, but, in contrast to the above approaches, also ensures a fair and widely accessible wealth creation process.

In this work we introduce the concept of *proof-of-personhood (PoP)* as a first step towards our goal, which combines *pseudonym parties* [8] with state-of-the-art cryptographic tools like linkable ring signatures [10] and collective signing [13] to create so-called *PoP-tokens*, which are basically *accountable anonymous credentials*.

The core idea of pseudonym parties is to verify *real people*, thereby linking physical and virtual identities and providing a basis to prevent adversaries from mounting Sybil attacks. Pseudonym parties are, as the name suggests, parties which can be organized basically by anyone, from governments to non-profit organizations, or companies to small groups of people in their own village. The participants agree on a set of rules such as specifying a place and time. All parties are recorded for transparency, but attendees are free to hide their identities by dressing as they wish, including hiding their faces for anonymity. By the end of the party each attendee will obtain *exactly one* cryptographic identity token that represents both a physical and virtual identity without revealing sensitive information.

The proof-of-individuality (PoI) project [4] uses similar ideas to create anti-Sybil-tokens by relying on virtual (video) pseudonym parties and Ethereum smart contracts. The PoI-approach, however, has several (security) disadvantages: it does not provide the same security properties as physical pseudonym parties, e.g., it needs to be ensure that videos show an actual livestream and not a recording; it relies on the security of the Ethereum blockchain and in turn on Ethereum's PoW foundation, and thus cannot be used on its own to bootstrap a secure cryptocurrency; finally, the reliance on Ethereum introduces a non-negligible amount of complexity in terms of implementability.

After defining Proof-of-Personhood, we then introduce *PoPCoin*, a cryptocurrency that leverages PoP to move from Bitcoin's ultimately unsuccessful "one CPU one vote" decentralization principle to a minting mechanism that embodies a *one PoP-token one vote* principle. PoPCoin provides a fair decentralized wealth creation mechanism tied to actual people, ensuring that every participant has the same chance of being chosen to create new assets, and essentially implements an instance of *basic income*. The value of that income is volatile and not specified by anyone since the whole currency will "float" to whatever represents the collective

value that the currency is providing to its population of users.

We suggest two different approaches to tweak existing cryptocurrencies to create PoPCoin instances. The first is a simple substitution of PoS with PoP where the system randomly selects the next minter from the list of PoP-token holders, while the second extends ByzCoin [9] in order to remove PoW but preserve the same performance benefits. A PoP-token contains the necessary information to prove that it was issued at a pseudonym party. State-of-the-art cryptographic building blocks enables the detection and prevention of double-spending attempts using PoP-tokens.

II. Background

A. Collective Signing (CoSi)

Scalable collective signing [13] enables an authority to request the validation of a statement by a decentralized group of witnesses. The resulting collective signatures are comparable in size and verification cost to individual signatures. The result of a collective signing (CoSi) protocol run is a standard Schnorr signature that anyone can verify efficiently against the corresponding aggregate public key.

B. Linkable Ring Signatures

A ring signature [12] is a type of signature that can be created by any member of a group of users. The public keys of the group members form a so-called *anonymity set*. One of the security properties of ring signatures is that it is infeasible to determine which of the keys from the anonymity set was used to compute a given signature. Linkable ring signatures [10] introduce the notion of *signature linkability* to a traditional ring signature scheme, by allowing to identify whether two signatures were issued by the same member without uncovering the member's identity.

C. ByzCoin

ByzCoin [9] is a Bitcoin-like cryptocurrency having a consensus mechanism inspired by the well-known PBFT algorithm [6]. ByzCoin leverages CoSi to reduce the message overhead in comparison to traditional PBFT approaches and uses Bitcoin's PoW mechanism to offer open-membership and Sybil attack resistance. Thanks to these modifications ByzCoin enables realization of a strong-consistency blockchain system that scales to hundreds of nodes while improving transaction rate.

D. RandHound

RandHound [14] is a distributed protocol that provides scalable, unbiasable, publicly-verifiable randomness against Byzantine adversaries. RandHound is a client-server protocol where a group of cooperating servers provide randomness to a client.

III. Design of PoP: Proof-of-Personhood

The first and most important step of PoPCoin is to bind the coin minting mechanism with real people, so that each person can mint new coins at the same rate. We achieve this property by introducing the notion of *proof-of-personhood*.

Proof-of-personhood is based on the concept of accountable pseudonyms [8]. The idea is to link virtual and physical identities in a real-world gathering (e.g., a party) while preserving users' anonymity. At the party every attendee is issued one and only one proof-of-personhood token, without their needing to disclose any identifying information.

A. Assumptions and Threat Model

A pseudonym party is organized by a set of volunteer *organizers*. Organizers might be a group of independent persons, part of an institution or an entity. In any case organizers are selected via a process that is external and independent from the pseudonym party. Each organizer is an independent person in charge of an independent server, henceforth denoted a *conode*, which together form a collective authority or *cothority*.

We assume an *anytrust* model in which attendees trust only that *at least one* organizer and one conode is trustworthy, that is they are uncompromised and not colluding; we do not require that the untrusted organizers and conodes are known. Ideally, in the future different pseudonym parties will be organized by different organizers allowing attendees to select the pseudonym party that they trust most.

The party occurs at a certain place and date, established by the organizers. Attendees are allowed to enter the party during a period of time; once this period ends no one is allowed to enter. However, for safety reasons anyone is free to leave at any moment.

B. Pseudonym Party Setup

At some point before the party, organizers gather in order to establish the main details of the event which include: the place p, the date d, the expected times of start and end of the party t_{start} and t_{end}, the barrier time at which tokens will start to be issued $t_{barrier}$, and the PoP-token's expiration date t_{exp}. After fixing these parameters, organizers exchange their public keys org_{PubKey} and select a group of observers to record the party and produce a video file $file_{video}$.

The observers are a different set of people selected, via an independent process, by the organizers. The tapes recorded by observers reinforce transparency and, e.g., help to capture the moment in which an attendee attempted to get more than one PoP-token or when an organizer refused to issue a PoP-token to an attendee. Afterwards, organizers create a configuration file $file_{config}$, that contains: p, d, t_{start}, t_{end}, $t_{barrier}$, t_{exp}, org_{PubKey} and $file_{video}$. Following this process, organizers establish a test period in which they set up the environment for the party, including the servers, to do a set of dry runs without actual attendees. After

thorough testing, organizers publish the configuration file and advertise the pseudonym party.

Once a person has decided to attend the pseudonym party he or she downloads $file_{config}$, where the party details are available and creates his own public and private keys. Additionally, the public key used for the pseudonym party should not be used on any other service that may compromise the attendee's anonymity. In order to ensure this we propose using $file_{config}$ for the creation of an ephemeral public key, to be used only at the party. The ephemeral public key will be used as the basis for the PoP-token. We propose two options to create these keys:

1) An attendee downloads third-party software, similar to a Bitcoin paper wallet, which takes the configuration file as an input and outputs a file containing: an ephemeral public key, a public key, and a private key.
2) An attendee downloads third-party software, a mobile application. The software receives the configuration file as an input and outputs QR codes presented to the mobile application.

C. Pseudonym Party Operation

On the start of a pseudonym party, doors open and attendees are allowed to enter the place for a certain period of time. Once that period of time has expired the *barrier point* has been reached and nobody is allowed to enter whereas anyone may leave. Shortly after this point organizers also start the PoP-token issuing process.

For this purpose, organizers and attendees form a line and each organizer confirms one-by-one each attendee's *personhood* by storing the attendee's ephemeral public key and marking the attendee with an ink stamp. After visiting all organizers, attendees enter a separate room and they are not allowed to go back to the previous room. The ink stamp is a visual confirmation, aiding organizers to recognize the attendee's that have already visited the first line, preventing an attendee from going back to the first room and obtaining a second PoP-token.

D. Pseudonym Party End

At the end of the pseudonym party the organizers stop collecting ephemeral public keys and enable their conodes to sign a party transcript, which is publicly available. The party transcript contains the following parameters: Hash of the video files recorded by the observers $file_{video}$, list of attendees' public keys att_{PubKey}. Each PoP-token is composed of the attendee's ephemeral public key, the attendee's private key and the list of attendee's public keys available in the party transcript.

IV. PoPCoin

Proof-of-personhood can be used as a defense against Sybil attacks and as a membership mechanism for achieving consensus, hence it is a suitable mechanism to be used in blockchain systems that use PoW, PoS, or a combination of the two. In this section we describe two approaches to use PoP in blockchain systems and our vision for the deployment of PoPCoin in the real world.

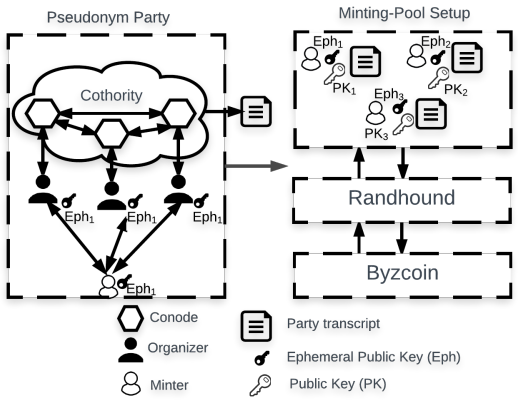

Figure 1: Interaction between components

A. Implementing PoPCoin

The integration and implementation of PoP in current blockchain systems is divided into two phases: *Setup* and *Minting*. The *Setup* phase involves carrying out the pseudonym party and validating the PoP-tokens. First, a set of organizers arranges a pseudonym party in which they issue PoP-tokens to be used for minting in blockchain systems. Organizers must follow the process presented in Section III. Once the party ends, organizers hand the *party transcript* to all attendees and upload it to a set of trusted servers for a period of time, e.g., one week, during which attendees that want to become minters are allowed to authenticate their PoP-tokens, following the process presented in Section V. After a successful authentication process an attendee deposits a public key to identify him or her and become an eligible minter. At the end of the validation process organizers generate a *minting-pool* formed by the public keys of attendees that have successfully authenticated their PoP-tokens.

In the *minting* phase all minters who are part of the minting-pool are eligible to create new blocks and as a result mint new coins. During this phase the last N minters run RandHound [14] with the last minter behaving as the client. If he fails to generate randomness, he is deprived of his block-reward. This randomness is used to select the member of the minting pool that is allowed to create the next block. All minters are equally eligible, since each of them owns only one token. Minters repeat the process every M minutes. If a minter fails to announce a new block within the M minutes, then RandHound is run again to select a new one. The RandHound output contains the last randomness it

issued and if a block was created, as a result there is a clear ordering of minters and there is only one eligible minter at any given time. For this reason forks cannot occur by accident. If a minter is caught extending two chains, he is punished, resolving the nothing-at-stake problem of PoS.

As an improvement over the minting phase we propose the use of ByzCoin [9], a protocol that achieves deterministic consensus in blockchain systems. ByzCoin defines a sliding *share* window, of fixed size W, in which shares are represented by blocks. Each round of ByzCoin runs among the share holders of the last W blocks. In this scenario minters of the last W blocks, i.e., the consensus group, run RandHound and randomly select the next minter to become a member of the window for W rounds. Fig. 1 presents a diagram describing the interaction of the system. After the membership is defined the protocol continues the same way as ByzCoin, by collectively signing the microblocks the leader (who is the last minter) proposes.

B. Deploying PoPCoin in the Real World

Although PoPCoin presents multiple interesting technical challenges, it is only fair to discuss its interaction with the real world, as it is based on the real world limitation of owning only one body.

We envision PoPCoin to start as a local cryptocurrency similar to multiple efforts like Ithaca Hours [2], Berk-Shares [1], or the Leman [3]. At such a small scale it can be feasible to have one pseudonym party every few months which is easily accessible from the community that uses PoPCoin. Also this small scale will allow the coin to quickly evolve without the need for "permission" or financial support from government or industry, and without close cooperation from any other pseudonym party group. Those different PoPCoin currencies would form completely separate blockchains with separate currencies, which might eventually be tradeable with each other and/or with existing currencies. Nevertheless their value will float to different levels depending on a variety of factors like the level of trust the community has on the currency or it's wider adoption.

This local form of PoPCoin would be decentralized in at least two respects: within a given group or PoPCoin instance, it is decentralized among the organizers and users in that group; but also globally, it is decentralized because all such local PoPCoin instances are in principle independent – one might fail completely (due to a successful attack or merely disinterest within the local community), but that will not prevent other local PoPCoin groups elsewhere succeeding and being reliable and secure.

If this experiment succeeds and gains enough traction these local PoPCoin groups can eventually federate and standardize enough so that they can merge their currencies into one larger currency shared by many groups, in which any person can show up to any cooperating pseudonym party anywhere and get one (but only one) PoP-token good for one minting share within this larger federation. A possible way to scale the PoP-token issuing process is for organizers to throw simultaneous pseudonym parties in different regions at the same or close enough times such that it is infeasible for a person to attend both, e.g., because physically traveling from Europe to the US west coast within a handful of hours is infeasible or because the expected gain is less than the cost of traveling; their frequency depends on coordination among local PoPCoin groups.

V. Technical and Security Challenges

In this section we discuss some security challenges that need to be addressed in order to correctly implement a pseudonym party not only for PoPCoin, but also for creating Sybil resistance applications like anonymous but accountable e-voting, editing of a wiki or participating in online forums.

A. Public and Private Key Generation

The first challenge is trusting the attendees to generate their public-private key pair, especially to somehow ensure that they will not use the same public key in another service, thus compromising their anonymity. We propose a non-standard form of public-private key pair creation in which the main output is an ephemeral public key that would be used just for the party, but if it is used in other service it would not compromise user's anonymity. First, attendees generate their private key x and public key $X = G^x$ using Ed25519, where G is the generator of the elliptic curve group. Next, attendees download the $file_{config}$ and hash it into a point in the elliptic curve $\mathrm{H}(file_{config}) = H$. After, attendees generate ephemeral public key using the point H obtained in the previous step, $X' = H^x$. Finally, attendees obtain (x, X') and (x, X), the first tuple contains the ephemeral public key and it is the one to be used at the party. The key generation process occurs on the user side. The process described above intends to preserve user privacy by generating an ephemeral public key for every party.

B. PoP-token Authentication

Applications and services need to authenticate and validate PoP-tokens, that is, check that the token was generated at a pseudonym party, its expiration date and its unique usage (per service). To achieve this goal we propose the use of a linkable ring signature scheme [10] as an aid in PoP-token authentication mechanism. This signature scheme links signatures (in a particular service) issued by the same user by means of a *tag*, i.e. the verification process of a signature outputs a *tag*, a signature produced by a particular user always produces the same tag.

The services that rely on PoP-token-based authentication to create new accounts are required to use the *party transcript*, handed at the end of the party and it is also publicly available (in trusted servers provided by organizers). Additionally they must keep a simple database to store

the tags generated during the signature verification process. When a user requests authentication to a service using his PoP-token the service requests the *party transcript*, which allows the service to match the PoP-token to a particular party and check the expiration date of the PoP-token. Once the service verifies the token validity it must ask the user to sign a message (e.g., "empty message") within a certain context (e.g., www.service.com). Next, the user must reply to the service with the signature (both are outputs of the signature process). The service verifies the signature (outputs a tag), if the signature is authentic the service checks the presence of the tag in the database. In the case of a first use PoP-token the tag is not in the database, then the service stores the tag. However, if the tag is already registered in the database the service refuses authentication to the user, because the PoP-token has already been used to create an account in the service, hence preventing Sybil attacks.

Acknowledgments

This research was supported in part by DHS grant FA8750-16-2-0034, and by the AXA Research Fund.

REFERENCES

[1] Berkshares inc. local currency. http://www.berkshares.org/.

[2] Ithaca hours community currency. http://www.ithacahours.com/.

[3] Le léman votre monnaie. http://monnaie-leman.org/.

[4] Proof of individuality. http://proofofindividuality.online. Consulted on October 2016.

[5] A. Back. Hashcash – A Denial of Service Counter-Measure, Aug. 2002.

[6] M. Castro et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.

[7] J. R. Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

[8] B. Ford and J. Strauss. An offline foundation for online accountable pseudonyms. In *Proceedings of the 1st Workshop on Social Network Systems*, pages 31–36. ACM, 2008.

[9] E. K. Kogias et al. Enhancing Bitcoin security and performance with strong consistency via collective signing. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 279–296. USENIX Association, 2016.

[10] J. K. Liu et al. Linkable spontaneous anonymous group signature for ad hoc groups. In *Australasian Conference on Information Security and Privacy*, pages 325–335. Springer, 2004.

[11] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[12] R. L. Rivest et al. How to leak a secret. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 552–565. Springer, 2001.

[13] E. Syta et al. Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning. In *37th IEEE Symposium on Security and Privacy*, May 2016.

[14] E. Syta et al. Scalable bias-resistant distributed randomness. In *38th IEEE Symposium on Security and Privacy (to appear)*, May 2017. Preprint: http://eprint.iacr.org/2016/1067.