Sybil-Resistant Pseudonymization and Pseudonym Change without Trusted Third Parties

Martin Florian, Johannes Walter, Ingmar Baumgart
Institute of Telematics
Karlsruhe Institute of Technology (KIT)
76131 Karlsruhe, Germany
{florian,baumgart}@kit.edu, johannes.walter@posteo.de

ABSTRACT

The issuing of pseudonyms is an established approach for protecting the privacy of users while limiting access and preventing sybil attacks. To prevent pseudonym deanonymization through continuous observation and correlation, frequent and unlinkable pseudonym changes must be enabled. Existing approaches for realizing sybil-resistant pseudonymization and pseudonym change (PPC) are either inherently dependent on trusted third parties (TTPs) or involve significant computation overhead at end-user devices. In this paper, we investigate a novel, TTP-independent approach towards sybil-resistant PPC. Our proposal is based on the use of cryptocurrency block chains as general-purpose, append-only bulletin boards. We present a general approach as well as *BitNym*, a specific design based on the unmodified Bitcoin network. We discuss and propose TTP-independent mechanisms for realizing sybil-free *initial access control*, *pseudonym validation* and *pseudonym mixing*. Evaluation results demonstrate the practical feasibility of our approach and show that anonymity sets encompassing nearly the complete user population are easily achievable.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

pseudonym; sybil attack; blacklisting; block chain

1. INTRODUCTION

Privacy-preservation is becoming increasingly important as more and more areas of life are benefiting from the ubiquitous interconnection of humans and autonomous devices. At the same time, allowing users (and their devices) to collaborate and use services anonymously can lead to abuse, degrading the utility of novel systems and services. Additionally, without limiting access in some form, *sybil attacks* [9] become possible where adversaries create large numbers of fake virtual identities (*sybils*). This both enhances the potential magnitude of abuse and enables malicious users to avoid blacklisting.

The issuing of unlinkable *pseudonyms* to users is a common solution to the challenge of hiding user identities while enabling access control and the effective protection against sybil attacks. Additionally, unlinkable *pseudonym changes* must be made possible for reducing the linkability between (potentially privacy-relevant) data samples originating from the same user. If frequent pseudonym changes are realized and an adversary successfully links one pseudonym to a user (e.g., using side channels and correlation), only the data samples observed in connection with this pseudonym become linkable to the user.

Established approaches for enabling sybil-resistant *pseudonymization and pseudonym change* (PPC) inherently require a *trusted third party* (TTP) like a certification authority for enforcing issuing criteria and preventing sybil attacks. Upon compromise of the TTP, large-scale sybil attacks become possible and the trustworthiness of issued pseudonyms is greatly reduced. Thus, centralized pseudonym issuers become attractive targets for attacks, resulting in high operational costs for maintaining their security. Additionally, the notion of universal trust anchors shared by all system participants is questionable when considering mobile users in a globally interconnected world.

In this paper, as an alternative to assuming centralized TTPs, we explore the use of distributed, non-malleable bulletin boards as provided by cryptocurrency block chains. More specifically, our contributions are the following:

- A novel approach towards TTP-free and abuse-resistant pseudonymization and pseudonym change (PPC). To the best of our knowledge, we are the first to propose a complete PPC system that both prevents sybil attacks and doesn't rely on a TTP for ensuring the correctness and security of any of its operations.
- A specific implementation of the approach - *BitNym* - leveraging the existing *Bitcoin* [15] network without requiring any modifications to its underlying protocols.
- A prototype of the proposed system and an evaluation of our approach using simulations of user populations and pseudonym changing behavior.

WPES'15, October 12 2015, Denver, CO, USA

ACM 978-1-4503-3820-2/15/10 . . . $15.00
DOI: http://dx.doi.org/10.1145/2808138.2808145

2. RELATED WORK

Approaches based on blind signatures [12] and zero-knowledge proofs [6] have been proposed for enabling the centralized issuing of unlinkable pseudonyms. Here, a TTP is also required for changing pseudonyms, as the old pseudonym must be marked invalid to prevent sybil attacks. Alternative approaches like e-token-dispensers [5] or the issuing of pools of pseudonyms with non-overlapping validity periods still inherently require a TTP for enforcing issuing criteria and ensuring that the issuing of pseudonyms is sybil-free.

As an alternative to TTP-based access control, decentralized approaches for preventing sybil attacks were proposed. Proof-of-resource schemes like *CAPTCHA* [20] and *proof-of-work* [4] are not effective for systems involving end-users, as determined adversaries can amass multiple orders of magnitude more resources than regular users are ready to spend for continuously using a system [13]. Approaches based on the *social graph* between users [19] are more promising, but either incompatible with privacy requirements or suited only for scenarios in which nearly all users are continuously reachable and active within a large-scale peer-to-peer network.

The usage of *proof-of-burn*, i.e., the provable destruction of a non-replenishable resource, was proposed for establishing sybil-free and trustworthy online identities [11] using the Bitcoin cryptocurrency [15]. However, resulting identities are linkable to the originating users. Even if funded via anonymous sources, changing such identities is not possible without repeating the significant initial investment.

In [10], Garman et al. propose a scheme for realizing decentralized anonymous credentials. As in our proposal, the authors build upon cryptocurrency systems like Bitcoin for avoiding the dependence on TTPs. However, the proposal is based on computation-intensive zero-knowledge proofs, making it less suited for more resource-constrained devices, and requires a TTP during the initial setup phase. The challenge of initially issuing pseudonyms in a TTP-independent and yet sybil-resistant manner is discussed only marginally. Also, the blacklisting of malicious users is not supported. To the best of our knowledge, we are the first to propose a complete pseudonymization system that is fully decentralized, resistant to sybil attacks and supports efficient and unlinkable pseudonym changes as well as the blacklisting of malicious pseudonym holders.

3. GENERAL APPROACH

In the following, we present the goals of our approach as well as technologies and assumptions it is based on. We then give a rough overview over the complete approach.

3.1 Goals and offered functionality

We propose an approach towards TTP-free and abuse-resistant pseudonymization and pseudonym change (PPC). At its core, our PPC approach aims at providing users with unlinkable *pseudonyms* that can be used in arbitrary applications, e.g., for authenticating to online services or in the context of peer-to-peer networking, vehicular communication and collaborative sensing in the Internet of Things. In accordance to Pfitzman et al. [16], we define a pseudonym as *an identifier of a subject other than one of the subject's real names*. While hiding the identity of users, pseudonyms generated by our approach should also offer some security against abuse, by effectively preventing sybil attacks and supporting the punishment of malicious pseudonym holders. More specifically, we aim at realizing the following properties:

1. Unlinkability of pseudonyms to user identities and to other pseudonyms by the same user.
2. Limitation to a fixed number of simultaneously active pseudonyms per user.
3. Authenticity of the linking between a pseudonym and its holder.
4. Possibility of unlinkable pseudonym changes.
5. Possibility of blacklisting pseudonym holders.
6. Complete independence of any trusted third parties (TTPs) for realizing any of the preceding properties.

Properties 1 to 5 have been widely discussed in the literature [16] and are shared by multiple existing and proposed systems. Their combination with property 6, on the other hand, is, to the best of our knowledge, unique to our proposal.

3.2 Decentralized append-only bulletin boards

Our strategy for avoiding the reliance on TTPs is to build upon recent results on realizing distributed consistency in highly adversarial environments. More specifically, we build upon decentralized cryptocurrency systems like *Bitcoin* [15], in which a globally consistent transaction log is collaboratively maintained within a network of non-colluding peers. Transactions are typically grouped in *blocks*, yielding a cryptographically secured *block chain* as the practical manifestation of the network consensus. In the following, we will also refer to Bitcoin-like networks as *block chain networks*.

Transactions in block chain networks are composed of *outputs* - the number of funds exiting the transaction in combination with a challenge that needs to be solved for spending them - and *inputs* - references to preceding outputs in combination with a valid solution. Typically, outputs contain the public part of an asymmetric cryptographic key pair (respectively a hash thereof) and can be spent upon proving the possession of the corresponding secret key. In addition, the inclusion of arbitrary data in transactions is possible. At the very least, without dedicated support from the underlying block chain protocol, transaction outputs can be used for this purpose (possibly rendering the funds allocated to the output unspendable). In this spirit, and despite the original design goal of facilitating online payments, the block chain paradigm has been previously adapted for a wide variety of different applications. Using *colored coins* [17], for example, the ownership of arbitrary assets can be encoded and transferred by marking (*coloring*) cryptocurrency units. Other examples include name services[1], anonymous credentials [10] and the timestamping of cryptographic commitments [7]. In the context of this trend, we observe that block chain networks provide general-purpose, decentralized append-only bulletin boards.

Despite their independence of TTPs, block chain networks are highly resilient to malicious tampering. This is mainly due to the extensive use of cryptography for securing ownership and the transfer of funds as well as dedicated mechanisms for ensuring the append-only feature of the transaction log in the face of sybil attacks. Sybil attacks are

[1] `https://namecoin.info/`

prevented by tying the levels of influence individual peers are able to exert on the block chain to the possession of limited resources. In Bitcoin, for example, computing power in the form of extensive proofs of work is required for adding new blocks to the block chain (also referred to as *mining*)[2]. An adversary must control more computing resources than the remainder of the network in order to cause significant disturbances. In the case of Bitcoin, this would require a significant investment.

Our general approach can be applied to any type of block chain network. Still, we focus on Bitcoin for developing a specific design that is readily deployable today with high security guarantees.

3.3 Assumptions and adversary model

We assume that the cryptographic building blocks used in the underlying block chain network are secure. For Bitcoin, this includes, most prominently, SHA-256 and ECDSA with the secp256k1 curve. We also assume that users are able to generate secure asymmetric key pairs and maintain the confidentiality of their secret keys. We also assume that users can form communication links to services and between each other without leaking identifying information like IP addresses. This can be realized in practice by building upon anonymous communication services like the *Tor* network [8].

Concerning our adversary model, we focus on adversaries that attempt to either disrupt the PPC system, launch sybil attacks or link pseudonyms to user identities. We consider adversaries that can observe and modify all communications but are unable to identify pseudonym holders based on communication metadata, i.e., are effectively stopped by the used anonymous communication services. Adversaries might collude with (or compromise) individual users, in which case that users' pseudonyms are known to them, but with no more than 50% of the user population. Additionally, we assume that adversaries cannot attack the underlying block chain network, disturbing its functionality as an append-only bulletin board. In the case of Bitcoin, for example, we assume that no adversary can control more than 50% of the computing resources ("hash power") in the network.

3.4 Sybil-resistant PPC without TTPs

Our PPC approach is based on three central building blocks: *genesis pseudonym creation*, *pseudonym change* and *pseudonym validation and use* (see Fig. 1). These directly address the desired properties discussed in Sec. 3.1. A central challenge we tackle is to ensure the resistance to sybil attacks (property 2) in all three building blocks while remaining independent of TTPs (property 6).

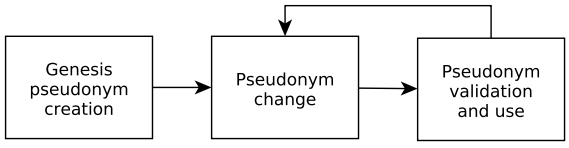

Figure 1: Overview of our PPC approach.

[2]Note that this is different from creating a new transaction. Transactions are simply broadcast to the network. Valid transactions are then later included in blocks.

At the core of our approach, *pseudonyms* are encoded in the outputs of transactions. Ideally, *cryptocurrency addresses* used in the underlying block chain network can be reused. For example, in BitNym (our specific design based on Bitcoin), pseudonyms correspond to Bitcoin addresses and are represented by the hash of an ECDSA public key. Users can authenticate their holding of a pseudonym in the same way they would prove their ownership of the corresponding address (satisfying property 3). The *validity* of a pseudonym, however, depends on additional factors. A pseudonym is only valid if the output it is encoded in exists on the block chain and hasn't been spent. Additionally, a chain of transactions leading from the output to a valid *genesis pseudonym transaction* (GPTx) must be provided. The latter is a special transaction encoding a proof that a predefined set of issuing criteria has been met by a user. A *genesis pseudonym* is included in one of the outputs of a GPTx and might, dependent on the used access control approach, not be completely unlinkable to a user identity. Thus, for ensuring the unlinkability of pseudonyms and allowing unlinkable pseudonym changes (satisfying properties 1 and 4), we adapt state of the art techniques for anonymizing cryptocurrency transactions for realizing *pseudonym mixing*. After a successful mix involving several pseudonym holders, a user will likely begin using a different GPTx for proving his current pseudonym's validity. For ensuring that pseudonym changes do not enable sybil attacks, unambiguous transaction chaining rules are defined so that each GPTx can be used for validating only one currently active pseudonym.

4. SPECIFIC DESIGN

In the following, we describe *BitNym*, a specific design of TTP-free and sybil-resistant PPC that is based on the Bitcoin network. The discussed techniques are, for the most part, directly applicable to other block chain networks. In accordance to 3.4, BitNym is divided into the building blocks *genesis pseudonym creation*, *pseudonym validation and use* and *pseudonym change*. We also discuss the possibility for *blacklisting* pseudonym holders based on malicious behavior.

4.1 Genesis pseudonym creation

The number of genesis pseudonyms a user has created forms the upper bound for the number of pseudonyms he is able to simultaneously hold. Thus, *initial access control* (IAC) needs to be performed in the course of genesis pseudonym creation, to ensure that only legitimate users receive pseudonyms and that sybil attacks can be prevented. In the following, we will first introduce a few technical details regarding the realization of GPTxes in BitNym and later discuss approaches for handling IAC.

4.1.1 Genesis pseudonym transaction

As noted in Sec. 3.2, transactions in Bitcoin are composed of one or more *inputs* and one or more *outputs*. For the GPTx, inputs come from regular Bitcoin transactions. These are required for funding - without the provision of *mining fees*, transactions cannot be written to the block chain. As the genesis pseudonym can be changed immediately after creation, we don't require inputs to be from an anonymous source.

The GPTx is required to contain two special outputs: the *marker output* and the *pseudonym output*. The outputs are defined by their ordering within the transaction: the first

output is the marker output and the second the pseudonym output. GPTxes may also have additional outputs, but these are ignored in the context of BitNym. An example GPTx is depicted in Fig. 2.

The marker output is provably unspendable (any funds allocated to it are lost) and has the capacity for holding 40 Bytes of arbitrary data. We realize this by using an *OP_RETURN* code in the beginning of the output script[3]. Similar marker outputs are also used in the colored coin approach. The marker outputs in BitNym contain a magic number in the first two bytes, for easing the detection of GPTxes. The remaining 38 bytes are used for storing a proof that the IAC criteria have been met.

The pseudonym output is based on the *pay-to-pubkey-hash* pattern, i.e., it is a regular output used for funding a Bitcoin address. The destination Bitcoin address is generated by the user (who then also holds the corresponding private key) and forms the genesis pseudonym. The number of Bitcoins allocated to the pseudonym output determines the *value* of the genesis pseudonym. The value of a pseudonym roughly determines the number of pseudonym changes its holder can perform without recharging it with additional funds.

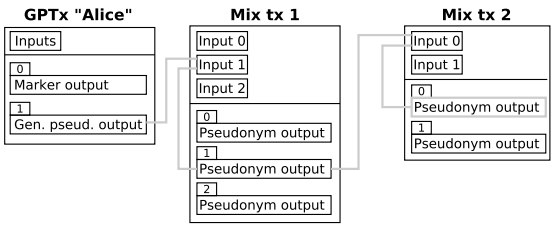

Figure 2: GPTx created by Alice and validation path involving that transaction. The validated pseudonym is not necessarily held by Alice.

4.1.2 Initial access control

The initial access control (IAC) building block defines the criteria based on which the validity of a GPTx is determined. In a sense, it provides a certification service for genesis pseudonyms. Thus, as a naive solution, it can be realized using a TTP that verifies if a user meets some predefined issuing criteria (e.g., that he is human and hasn't received a genesis pseudonym before). Upon verification, the TTP can provide a cryptographically signed validity acknowledgement that the user can include in the marker output of his GPTx.

One of the central advantages of our PPC approach is that a dependence on such a TTP is not inherently given. In the following, we discuss several approaches for realizing IAC without a TTP that are compatible with BitNym. The approaches focus primarily on ensuring that sybil attacks are prevented, i.e., that every human user is able to create only a bounded number of valid genesis pseudonyms.

Proof-of-work. Users can be required to solve a computational puzzle (as in [4]) for producing a valid GPTx. While easy to realize, the security of this approach is questionable. Specifically, regular users often feature both restricted computational resources and a low time budget while adversaries can rent or buy specialized hardware and leave it running for longer periods of time. Thus, the puzzle difficulty will likely be either too high, deterring honest users and slowing adoption, or too easy, making large-scale sybil attacks possible for determined adversaries.

Proof-of-burn. A related approach to proof-of-work is the use of proof-of-burn, i.e., the provable destruction of valuable resources. In the context of Bitcoin, proof-of-burn is implemented by provably rendering a certain amount of funds unusable for future transactions [11]. In BitNym, proof-of-burn can be realized by allocating a certain amount of Bitcoin to the marker output of the GPTx. In contrast to proof-of-work, proof-of-burn enables the instant creation of genesis pseudonyms and adversaries are unable to leverage economy of scale effects: every genesis pseudonym "costs" the same for everybody. Large-scale sybil attacks are thus rendered infeasible even with small proof-of-burn requirements. Additionally, spending "money" for registering an online identity may have a positive effect on behavior if blacklisting is possible [11]. Thus, while not perfect (participation rights and influence become tied to economic well-being), proof-of-burn is a viable option for IAC and is also used in our proof-of-concept prototype.

Social graph based IAC. A third avenue for exploration is to leverage social connections for deciding about the trustworthiness of new users and preventing sybil attacks. For example, genesis pseudonyms might be considered valid only if backed by sufficient "is not a sybil" confirmations from established users. Confirmations can be issued pseudonymously, thus not leaking any social graph information. For ensuring that a small group of users cannot introduce an arbitrary number of sybils, a high threshold must be chosen for the number of required confirmations. This can quickly become unpractical as honest users might not have sufficient social contacts. Thus, a *preselection mechanism* is necessary. We envision the use of a delay-tolerant *darknet* for facilitating this. In a darknet, only trusted users (e.g., users with whom a social connection exists) are accepted as peers and only direct neighbors are aware of each other's identities. In this way, the social relationships between users do not need to be revealed. The darknet can be used by new users to convince a large number of existing users of their own non-sybilness, thus collecting confirmations from them. Unfortunately, a detailed investigation of this approach would exceed the scope of this paper and is thus left for follow-up works.

4.2 Pseudonym validation and use

In order for a user to be able to use a pseudonym, he must prove (1) that he is the holder of the pseudonym and (2) that the pseudonym is valid. For (1), it is sufficient to use the pseudonym as an identity certificate, i.e., using the corresponding private key for signing messages. For (2), the user needs to construct a *proof* based on a valid GPTx and a transaction in which the Bitcoin address corresponding to the pseudonym has received funds. For ensuring the unlinkability of pseudonyms to user identities, validity proofs shouldn't disclose any additional information about a pseudonym holder other than the fact that his pseudonym is valid. We will introduce our specific approach for building such proofs in the following.

Validation path. In Bitcoin, an input is always linked to one specific output of a preceding transaction. Through

[3]Bitcoin uses a scripting system for encoding the requirements for spending an output.

this, transactions become linked. By defining a mapping between the inputs and outputs within the same transaction, a path in the transaction graph can be defined. In BitNym, we define such a mapping via the ordering of inputs and outputs, i.e., via the input and output indices. The i'th output is mapped to the i'th input of the same transaction. We refer to the resulting path as the *validation path*. An example validation path is depicted in Fig. 2. The marked pseudonym is validated using a path containing two *mix* transactions (*mix tx 1* and *2*, see also Sec. 4.3) and a GPTx created by Alice. However, it is not necessarily held by Alice.

Constructing a proof. Pseudonyms holders must be able to present a validation path from their current pseudonym to a valid genesis pseudonym. More specifically, a *proof message* must be constructed that includes: (1) a transaction with an output containing the pseudonym, (2) a GPTx and (3) a list of transactions that form a valid validation path between the output containing the pseudonym and the provided GPTx. Depending on the scenario, the communication overhead can be greatly reduced by including only the address of the output (i.e., a transaction hash in combination with an output index) that contains the pseudonym to be validated. A recipient with efficient reading access to the block chain (e.g., a full Bitcoin node) can then obtain the necessary transactions and form the validation path himself.

Verifying a proof. The recipient of a proof message needs to verify the following criteria:

1. The provided transactions form a valid validation path.
2. The transaction containing the pseudonym is included in the block chain.
3. The output containing the pseudonym has not been spent, i.e., hasn't been used in a follow-up transaction.
4. The provided GPTx is valid.

Criteria 2 and 3 require reading access to the block chain. For clients with constrained resources, that cannot form full nodes in the Bitcoin network, such access can be provided by querying full nodes (the de facto standard operating mode of end-user Bitcoin clients). Note that the verification of criteria 1 and 2 is sufficient for ensuring that all transactions in the proof message are included in the block chain. Transaction inputs include cryptographic hashes of preceding transactions, so that no fake validation path involving a transaction on the block chain can likely be constructed. The verification of the GPTx (criterion 4) depends entirely on the used IAC mechanism (see Sec. 4.1.2). When using proof-of-burn, for example, the recipient needs to verify if the amount of Bitcoins allocated to the marker output of the GPTx meets a previously established burn requirement.

4.3 Pseudonym change

As we do not require genesis pseudonyms to be unlinkable to user identities, unlinkable pseudonym changes are necessary for ensuring that pseudonyms hide the identities of their holders. Additionally, pseudonym changes help with the prevention of correlation attacks where pseudonyms are broken following longer periods of observation.

4.3.1 Simple change and Bitcoin mixing

As discussed previously, each valid pseudonym is encoded in an output of a transaction on the block chain. Changing a pseudonym involves creating a new transaction that spends that output. However, consecutive transactions on the Bitcoin block chain are, by design, completely linkable to each other. Thus, for ensuring the unlinkability between old and new pseudonyms, pseudonym change transactions must be created in a coordinated fashion so that an external observer cannot determine if two pseudonyms linked on the block chain are also held by the same user.

This challenge is tightly related to the anonymization, or *mixing*, of Bitcoin funds, which has already been tackled in a wide range of works [1–3, 14, 18]. While our strategy is to build upon such works, existing approaches cannot be leveraged directly. For one, the unlinkable payment of transaction fees is non-trivial when considering the double role of transaction outputs as encoders of pseudonyms and holders of funds. Secondly, it must be ensured that only users with valid pseudonyms are able to contribute to mixes and that no entity is able to "steal" access rights.

4.3.2 Pseudonym mixing protocol

In the following, we propose a specific pseudonym mixing protocol based on the Bitcoin mixing approach *CoinShuffle* [18]. In CoinShuffle, groups of users collaboratively form *mix transactions* to which every user contributes inputs and outputs. The mapping between inputs and outputs is randomized so that both external adversaries and mix group members (for mix groups with 3 or more non-colluding members) cannot conclusive determine who provided funds to which output. In addition to being fully TTP-independent, CoinShuffle has the benefit of being fully compatible with Bitcoin, requiring only standard output scripts. In comparison to commitment-based approaches like *Xim* [2], CoinShuffle mixes are faster and cheaper, as only one transaction per mix needs to be written to the block chain.

Our pseudonym mixing protocol is divided into four phases that we will discuss in the following: *discovery of mixing partners*, *peer verification*, *creation of a mix transaction* and, finally, the *construction of new proofs*.

Discovery of mixing partners. For performing the CoinShuffle protocol and creating a mix transaction, at least one additional pseudonym holder is required that also wishes to change his pseudonym. Several options exist for discovering such mixing partners. A generic approach involves the establishment a block chain-based broadcast channel, using OP_RETURN outputs to mark transactions belonging to the channel and include arbitrary announcement data. Pseudonym holders monitor the broadcast channel and either respond to announcements or place one themselves. A similar approach to mixing partner discovery is used in [2].

Alternatively, in some scenarios, suitable mixing partners can be found via side channels. In peer-to-peer networking or vehicle-to-vehicle communication, for example, pseudonym holders are typically already aware of a number of other participants. Additionally, in some scenarios, side channels that might weaken the unlinkability of pseudonym mixing must be taken into account. In vehicular networking, for example, mixing partners should be spatially close to each other to avoid position-based linking.

Verification of mixing partners. All members of a mixing group must prove the validity of their respective pseudonyms to each other as described in Sec. 4.2. As an additional benefit, this reduces the danger from denial-of-service attacks via uncooperative mix group members as

malicious pseudonym holders face the danger of being blacklisted (we discuss blacklisting in Sec. 4.4).

Creation of a mix transaction. This step is largely based on the CoinShuffle protocol [18]. The participants within a mix group exchange inputs and outputs (containing freshly generated Bitcoin addresses), out of which one transaction is cooperatively formed. Every participant verifies if his own inputs and outputs are correctly included in the resulting transaction and, if so, provides the necessary signature for his inputs. The exchange of inputs, outputs and signatures is performed anonymously using techniques reminiscent to decryption mix nets.

As an important modification to CoinShuffle, we change the mechanism for distributing the value of inputs and paying mining fees. Every participant is allowed to contribute only one output to the resulting transaction. The value of outputs is not chosen freely by mix group members, ensuring that they "don't mix at a loss", but divided equally between all outputs. In this way, the value-based linking between inputs and outputs is prevented. The value v' used for all outputs is equal to the total value available after the payment of the transaction's mining fee f divided by the number of participants m. With v_i denoting the value associated to the input with index i, we arrive at the following formula:

$$v' = \frac{(\sum_{i=0}^{m} v_i) - f}{m} \quad (1)$$

The resulting mix transaction is broadcast to the Bitcoin network. Once it has been included in the block chain, the old pseudonyms become invalid and the new ones (encoded in the outputs of the mix transaction) valid.

Construction of new proofs. Once all pseudonyms are exchanged, each participant needs to construct a proof for his new pseudonym. The proofs for all old pseudonyms participating in the mixing have already been exchanged during the verification of mixing partners. According to the validation path logic introduced in Sec. 4.2, every new proof consists of one of these old proofs in combination with the currently formed transaction. Which proof should be used is determined by the index of the output containing a user's new pseudonym - if it has the index i, the proof starting with the output referenced in the i'th input of the transaction is used. It is often the case that users use a different genesis pseudonym for validation after each mix.

Fig. 3 shows an example of such a path change. The holder of the marked pseudonym in the mix transaction *mix tx 2* participates in *mix tx 4*. He contributes input 2 and output 1 of that transaction. Thus, his new pseudonym has an entirely different validation path with a different GPTx.

4.3.3 Parametrization of mixing protocol

Larger mixing groups lead to higher anonymity gains per mix. However, larger minimum mixing group sizes may also lead to longer waiting times until a group has been formed. The acceptable difference of the pseudonym values to other pseudonym holders in a mix group also has an effect on this delay. A low acceptable difference leads to a lower number of possible mixing partners. Note that there is no reason to refuse mixing with a pseudonym whose value is higher than one's own, as this would result in a value gain. In Sec. 6, we further investigate the effect of these parameters.

Ideally, pseudonym changes should be possible instantly. However, coordination with other users is necessary and mix transactions need to be written to the block chain. This leads to unavoidable waiting times. A straightforward approach for enabling fast pseudonym changes is to allow the simultaneous holding of two or more pseudonyms for each user. For example, genesis pseudonym transactions might spawn two instead of one pseudonym. In this way, one pseudonym can be actively used while the pseudonym mixing protocol is conducted opportunistically using the other. Once a change is appropriate, the roles of the pseudonyms can be swapped.

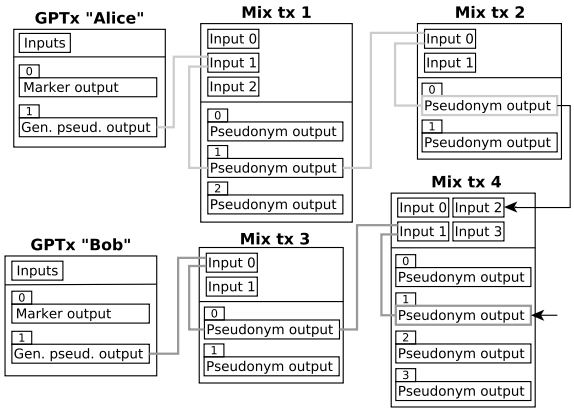

Figure 3: Validation path change after mix.

4.4 Blacklisting

In many scenarios it is necessary for users to be punishable upon malicious behavior. Even in the context of BitNym itself, it is beneficial if participants that deliberately disrupt the pseudonym mixing protocol can be blacklisted for a certain time. Blacklisting is easily supported by our approach. Pseudonyms can be marked as malicious, e.g., using a block chain-based broadcast channel as in [2]. Blacklisted pseudonyms will fail the pseudonym validation step. Additionally, once a pseudonym is blacklisted, other participants will refuse to cooperate with it in the scope of pseudonym mixes. Thus, the pseudonym holder is permanently blacklisted, respectively needs to create a new genesis pseudonym.

Open challenges with blacklisting in our PPC approach include ensuring that punishment cannot be easily evaded by carefully timing pseudonym changes. Lock times per pseudonym might need to be enforced, during which no changes are possible. Additionally, it must be ensured that the blacklisting mechanism cannot be used for maliciously censoring users. Voting-based approaches might be a possible solution for making correct blacklisting decisions without a TTP. Given a sybil-free IAC mechanism, sybil-free and robust voting can easily be realized.

5. ANALYSIS

In the following, we discuss the degree of anonymity provided by our approach as well as dangers arising from the external payment of transaction fees.

5.1 Provided degree of anonymity

A metric is required for quantifying the degree to which pseudonyms in our approach are unlinkable to the identities of their holders. We construct such a metric based on

the notion of anonymity as defined by Pfitzman et al. [16], i.e., the indistinguishability within a set of subjects - the *anonymity set*. The size of the anonymity set of a user is a common metric for the unlinkability of his pseudonym. In the following, we will refer to it simply as *anonymity*.

In the context of BitNym, different approaches for defining anonymity exist that depend on the assumed goals of the adversary. When considering *backward anonymity*, the adversary aims at determining which genesis pseudonym was previously held by the holder of a pseudonym. A somewhat opposite approach, *forward anonymity*, assumes an adversary that, given a specific inactive pseudonym, wishes to determine the pseudonyms which that pseudonym's holder held afterwards. We focus on backward anonymity here, but most of the analysis applies for forward anonymity as well. Also, the evaluation of forward anonymity produced very similar results to that of backward anonymity.

The *backward anonymity set* of a pseudonym is the set of genesis pseudonyms that could have been created by that pseudonym's holder. Consequently, the backward anonymity of a pseudonym is the size of that set. For analysis, we assume that all genesis pseudonyms are fully linkable to user identities, e.g., due to an insufficient anonymization of the funds used for creating them[4]. Thus, by measuring backward anonymity, we quantify the linkability of pseudonyms to user identities.

Genesis pseudonyms have a backward anonymity of 1. It is usually increased following pseudonym mixes, as the backward anonymity set after a mix is the union of all anonymity sets of the participating pseudonyms. Given a pseudonym mix transaction with the set of participating pseudonyms M and the anonymity sets A_p for all $p \in M$, the *anonymity increase* Δa_p for a $p \in M$ can be written as:

$$\Delta a_p = \left| \bigcup_{q \in M} A_q \setminus A_p \right| \tag{2}$$

The increase of backward anonymity across pseudonym mixes is also depicted in Fig. 4. At first, the creator of the genesis pseudonym *GP 1* has no anonymity. Through the mix transaction *mix tx 1* his anonymity set increases to 3, because the pseudonyms *P 1.1*, *P 2.1* and *P 3.1* cannot be unambiguously linked to any of *GP 1* to 3. After *mix tx 4* the anonymity set of the creator of *GP 1* increases to 8, as *P 1.2* can be held by the creator of any one of the 8 GPTxes.

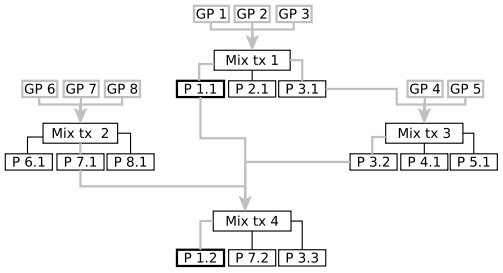

Figure 4: Backward anonymity increase for the creator of genesis pseudonym *GP 1*.

[4] Note that, depending on the used IAC mechanism, this assumption may only hold for very strong adversaries.

5.2 Transaction fees and pseudonym value

In the predominant majority of cases, the creation of block chain transactions requires the provision of transaction fees. If fees are not payed from pseudonym inputs as proposed in Sec. 4.3, they need to be provided from other sources via additional transaction inputs. If these additional inputs are not anonymized, the breaking of pseudonyms becomes possible. For example, if a user contributes fees to two linked mix transactions, it becomes clear which pseudonym belonged to the user between mixes. This is also depicted in Fig. 5.

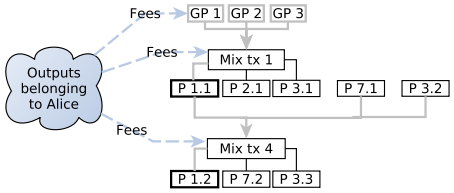

Figure 5: Anonymity loss through external payment of transaction fees.

More complex alternatives to the proposed approach of reusing pseudonym outputs for funding are possible. Pseudonym holders can contribute to pools that are then used for paying fees. Or a parallel coin mixing mechanism can be implemented that ensures that fee payment funds are anonymous. However, the benefit of such schemes is questionable while the increased complexity is significant.

Our approach of reusing pseudonym outputs also supports the *recharging* of pseudonyms. Here, a non-mixing change is conducted, with additional inputs increasing the pseudonym's value. For analysis purposes, this is similar to creating a new genesis pseudonym - the pseudonym created in such a transaction might become linkable to the holder of the provided funds. However, recharging doesn't reduce only the anonymity level of the current pseudonym, but also that of preceding pseudonyms. Thus, depending on the context, the creation of new genesis pseudonyms upon value depletion in combination with an overprovisioning of pseudonym value in GPTxes might be a more appropriate option.

6. EVALUATION

We evaluated BitNym by implementing a proof-of-concept prototype and measuring its communication and computation overhead. Additionally, we developed a lightweight simulator for evaluating the effect of pseudonym mixing in scenarios with several thousand participants.

6.1 Prototype

For gaining insights about the overhead and applicability of PPC in practical scenarios, we implemented a proof-of-concept prototype compatible with the official Bitcoin network. Our prototype is capable of creating GPTxes using proof-of-burn as an IAC mechanism, and performing the pseudonym mixing protocol with mixing partners given as input. The discovery of mixing partners as well as blacklisting are not part of our prototype.

6.1.1 Implementation

For communicating with the Bitcoin network and accessing the Bitcoin block chain, our prototype uses the *BitcoinJ*

library[5] (version 0.12). We used the SPV mode of BitcoinJ, i.e., our prototype did not maintain a complete block chain but contacted full nodes for required information. Thus, our prototype is already suited for resource-constrained devices. For the cryptographic primitives required, amongst other things, for realizing the CoinShuffle protocol, the cryptographic library *Bouncy Castle*[6] (version 1.51) was used.

As one of the steps of validating a pseudonym, it must be verified that the transaction output holding the pseudonym has not been spent. This can be performed efficiently by checking if the output is part of Bitcoin's *unspent transaction output* (UTXO) set that is maintained by all full nodes. Unfortunately, regular full nodes currently do not share their view on the UTXO set with SPV clients. An extension to the Bitcoin protocol has been proposed for solving this problem[7]. However, at the time of writing, the proposed extension was not yet accepted and deployed at full nodes. In our prototype, we implemented a workaround that is based on, in essence, querying a full node block by block for transactions that spend that output. However, this process is unrealistically time consuming in comparison to directly querying the UTXO set. Thus, we deactivated such checks for the performance evaluation of our prototype and instead requested an additional block from a full node to approximate the latency of making a UTXO query. For a feature complete and yet efficient implementation, a possible alternative can also consist of monitoring all active pseudonym addresses, effectively maintaining a subset of the UTXO including only pseudonym outputs.

6.1.2 Performance

In the following, we present evaluation results gathered with our prototype. We restrained from evaluating the time requirement of pseudonym changes, as this is predominantly based on the time until mixing partners are found, the CoinShuffle mixing protocol and the time until the final mix transaction is included in the block chain (around 9 minutes on average [10]).

Instead, we focused on the evaluation of validity checks. These need to be performed not only by pseudonym holders, but also by all entities wishing to validate BitNym pseudonyms. On a regular notebook computer (2.5 GHz CPUs, the tested functionality was single-threaded), verifying a proof consistently took less or insignificantly above 200ms for multiple different proof messages. Proof sizes had no discernible effect on validation duration. The vast majority of time is spent on requesting block data from a Bitcoin full node (for verifying that the pseudonym transaction was included in the block chain) and verifying, with the help of a full node, if the pseudonym output is unspent (which we approximated by requesting an additional block, see Sec. 6.1.1). The sizes of proof messages grew linearly with the number of transactions, starting from 4075 Bytes for a proof containing only a GPTx and increasing by 380 Bytes for a mix transaction with a mixing group size of 2 (and around 570 Bytes for a mix transaction with 3 participants, etc.). We used the standard serialization routines of Java in our prototype, so that improvements to the proof message sizes are likely possible. Still, even with the current application, proof chains containing more than hundred entries are possible at a proof message size of below 50 Kilobytes.

6.2 Large-scale pseudonym mixing

For answering question concerning the achievable levels of anonymity as well as the large-scale feasibility of BitNym, we conducted a series of simulation studies.

6.2.1 Simulation environment

We developed a lightweight time-discrete simulator for evaluating different properties of pseudonym mixing on a macroscopic scale. No actual interfacing with the block chain is simulated. Time is divided in slots. During each slot (simulated) users either create new GPTxes, make announcements that they are looking for mixing partners or answer already existing announcements and instantly perform a pseudonym mix. The simulator can be used for measuring, amongst other things, the influence of mixing on the degree of anonymity and the average size of proof messages (affecting in communication overhead).

The desired number of users per simulation needs to be configured beforehand. The simulation is separated in different phases. In the *warmup phase*, every user creates a GPTx. Creation times are uniformly distributed within the warmup phase. The length of the warmup phase corresponds to the average expected *lifetime* of a pseudonym. The lifetime of a pseudonym is determined by the value of the user's genesis pseudonym. In every change, an average of 5.000 Satoshi[8] is subtracted from the pseudonym's value (we require 5.000 Satoshi per participant for the payment of transaction fees). After the pseudonym's value has dropped to below 5.000, it is considered *dead* in the simulated context. Thus for a configured pseudonym start value of 200.000 Satoshi, the average pseudonym lifetime is 40 changes. After all pseudonyms are created, the simulation continues for 2 more lifetimes until measurements start to be made. Thus, the warmup phase consists of 3 lifetimes. Measurements are made during the *evaluation phase*, using only pseudonyms created during that phase.

Once a pseudonym dies a new one is created. The exact point in time when the new pseudonym is created is determined using the same logic as the decision about when a change should be made.

Mixing groups are formed centrally by the simulator in every time step. The simulator is honest and attempts to satisfy all requirements optimally. Pseudonym change and mixing is not implemented in detail. Only the outward effect of the change protocol outlined in Sec. 4.3 is reproduced.

Unless noted otherwise, the following simulation parameters were constant for all studies presented in the following:

- **Pseudonym start value:** 200.000 Satoshi (0.002 Bitcoin; worth less than 0.5 Euro at the time of writing).
- **Simulation duration:** 3x lifetime for warmup and 10x lifetime for evaluation phase.
- **Number of users:** 1.000

6.2.2 Parametrization

For determining a suitable parametrization for BitNym in the simulated scenario, we performed an extensive *performance versus cost* (PvC) study. For the performance metric,

[5] `https://bitcoinj.github.io/`

[6] `https://www.bouncycastle.org/`

[7] `https://github.com/bitcoin/bips/blob/master/bip-0064.mediawiki`

[8] 10^8 Satoshi amount to one Bitcoin.

we used the average anonymity increase per change. For the cost metric, we used the average size of proof messages calculated based on the results gathered with our prototype. We based the translation of validation path length (in number of changes) to proof message sizes (in Byte) on the results gathered with our prototype.

We evaluated different parameter combinations, i.e., varied average change rates and change rate deviations, mixing group sizes and the acceptable difference to the pseudonym values of mixing partners. For every parameter combination, we performed 10 simulation runs, averaging the results.

The optimum parameter combination according to our PvC study included a high change rate with a high deviation in the change interval (288 time slots with a deviation of 8). Also, mix groups of size 2 performed significantly better than parametrizations with larger mix group sizes.

6.2.3 *Evaluation of individual parameters*

In the following, we investigate the impact of individual parameters on backward anonymity. The simulation parametrizations are based on the optimal settings determined during our PvC study (Sec. 6.2.2). Only parameters with a significant influence on anonymity are discussed here.

The following figures depict the backward anonymity (i.e., the size of the backward anonymity set) for a single user in relation to the number of performed pseudonym changes starting from a genesis pseudonym. The depicted results are averaged over all "alive" users, i.e., users whose pseudonyms are still funded, as well as several separate simulation runs. At change 40, which is marked as a vertical line in the diagrams, pseudonyms, on average, run out of funds. Thus, the values after and closely before 40 are significantly less representable as only few pseudonyms make it past the 40th change. Additionally, drops in anonymity can be seen in some plots. These result from the fact that, once a pseudonym dies and a new one is created, the old pseudonym is removed from all anonymity sets. With this, we model the assumption that GPTxes are fully linkable to user identities.

Fig 6 shows the impact of the change rate deviation *crd* (the standard deviation of change intervals in simulation time slots) on the development of the anonymity set. As can be seen, it is of significant importance that the change rate is varied. This is understandable, as a low change rate variation leads to a limiting of mixing groups to members of small communities.

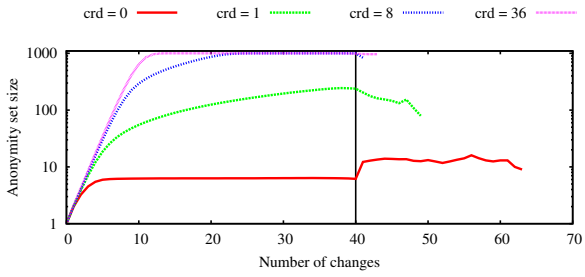

Figure 6: Impact of change rate deviation *crd* (in time slots) on backward anonymity.

The impact of the acceptable difference *ad* to the pseudonym values of mixing partners (in Satoshi) is shown on Fig. 7. As expected, a higher acceptable difference, i.e., a higher readiness to lose Bitcoins during a mix, leads to a faster anonymity growth. Again, the reason is that the pool of users out of which mixing partners are likely taken is significantly enlarged.

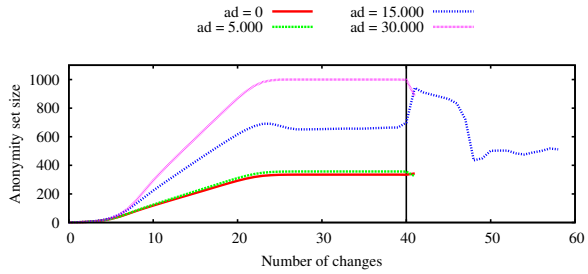

Figure 7: Impact of acceptable difference *ad* to the pseudonym values of mixing partners (in Satoshi) on backward anonymity.

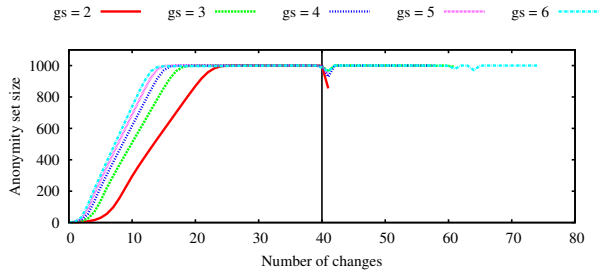

Figure 8: Impact of mixing group size *gs* on backward anonymity.

The impact of the mixing group size *gs* on anonymity increase is shown on Fig. 8. While mixing group size does not influence the maximum reachable level of anonymity in the evaluated scenario, the maximum is reached quicker for larger groups. However, with larger groups the size of mix transactions grows as well. Ultimately, this results in significantly larger proof messages if mixes are performed continuously independently of the achieved level of anonymity (as is the case in our evaluation setup). A practical benefit of low mixing group sizes is also that small mixing groups can be formed faster, as fewer partners are needed. However, it should also be noted that mixing groups of size 2 offer no anonymity increase between group members.

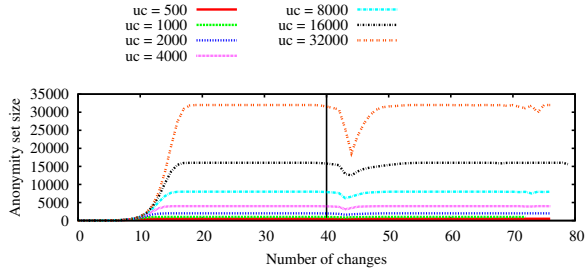

Figure 9: Backward anonymity development for larger user counts *uc*.

Finally, we also conducted simulations involving larger user populations. Fig. 9 shows the development of anonymity

set sizes in dependence of the number of simultaneously active users, the user count *uc*. The maximum achievable level of anonymity is equal to this number. Due to the exponential growth of backwards anonymity with every mix, the maximum is reached quickly even for larger user populations.

7. CONLCUSION AND FUTURE WORK

In this paper, we tackle the challenge of enabling unlinkable and TTP-independent pseudonymity while remaining resilient to large-scale sybil attacks. We outline an approach that is suitable for offering privacy-preserving and abuse-resistant authentication for online services, peer-to-peer networking and cooperative services in the Internet of Things. Robustness and sybil-resistance is achieved by leveraging cryptocurrency block chains as decentralized append-only bulleting boards. We furthermore present BitNym, a specific realization of our approach based on the Bitcoin network and present mechanisms for initial access control and pseudonym mixing. Via a prototype of BitNym, we demonstrate the practical feasibility of our approach. Using simulations of larger user populations, we demonstrate that large anonymity sets can be built quickly and discuss the influence of different parameters on anonymity and overhead.

We view this publication as a base for a wide range of further works. Amongst other things, we plan to further investigate social-graph based initial access control for BitNym. Additionally, we will design and evaluate specific blacklisting mechanisms and investigate to what extend reputation scores can be introduced without breaking anonymity.

Acknowledgements

This work was supported by the German Ministry for Education and Research (BMBF) through funding for the Center of Excellence for Applied Security Technology (KASTEL).

References

[1] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on. IEEE*, 2014.

[2] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore. Sybil-resistant mixing for bitcoin. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 149–158. ACM, 2014.

[3] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *Financial Cryptography and Data Security 2014*, 2014.

[4] N. Borisov. Computational puzzles as sybil defenses. In *Peer-to-Peer Computing, 2006. P2P 2006. Sixth IEEE International Conference on*, pages 171–176. IEEE, 2006.

[5] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 201–210. ACM, 2006.

[6] J. Camenisch and E. 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. ACM, 2002.

[7] J. Clark and A. Essex. CommitCoin: Carbon dating commitments with bitcoin. In *Financial Cryptography and Data Security*, pages 390–398. Springer, 2012.

[8] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *13th USENIX Security Symposium*. Usenix, 2004.

[9] J. R. Douceur. The sybil attack. In *Peer-to-peer Systems*, pages 251–260. Springer, 2002.

[10] C. Garman, I. Miers, and M. Green. Decentralized anonymous credentials. In *Network and Distributed System Security (NDSS) Symposium*, 2014.

[11] M. Hearn. Creating bitcoin passports using sacrifices. Bitcoin Forum, February 2013. `https://bitcointalk.org/index.php?topic=140711.0`.

[12] J. E. Holt and K. E. Seamons. Nym: Practical pseudonymity for anonymous networks. *Internet Security Research Lab Technical Report*, 4:1–12, 2006.

[13] B. Laurie and R. Clayton. "proof-of-work" proves not to work. In *Workshop on Economics and Information Security*, 2004.

[14] G. Maxwell. CoinJoin: Bitcoin privacy for the real world. Bitcoin Forum, August 2013. `https://bitcointalk.org/index.php?topic=279249.0`.

[15] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 1(2012):28, 2008. `http://nakamotoinstitute.org/bitcoin/`.

[16] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. `http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf`, Aug. 2010. v0.34.

[17] M. Rosenfeld. Overview of colored coins. `https://bitcoil.co.il/BitcoinX.pdf`, 2012.

[18] T. Ruffing, P. Moreno-Sanchez, and A. Kate. CoinShuffle: Practical decentralized coin mixing for Bitcoin. In *Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS'14)*, volume 8713 of *Lecture Notes in Computer Science*, pages 345–364. Springer, 2014.

[19] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 41(4):363–374, 2011.

[20] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford. CAPTCHA: Using hard ai problems for security. In *Advances in Cryptology - EUROCRYPT 2003*, pages 294–311. Springer, 2003.