The Complex Shape of Anonymity in Cryptocurrencies: Case Studies from a Systematic Approach

Niluka Amarasinghe, Xavier Boyen, and Matthew McKague

Queensland University of Technology, Brisbane, Australia

Abstract. The modern financial world has seen a significant rise in the use of cryptocurrencies in recent years, in no small part due to convincing levels of anonymity promised by such schemes. Bitcoin, despite being the most widespread, has significant lapses of anonymity. Several recent constructions aim to bridge some of those gaps. Amid such developments, there have been many attempts to evaluate the anonymity prospects of such schemes, but always with a rather narrow view based on metrics tailored to the schemes being studied.

Here, we employ a common universal framework to characterise the many aspects of anonymity achieved, or not, by any (crypto, digital, or physical) currency schemes, irrespective of the underlying implementation. We focus on a few high-profile practical cases of interest (including Bitcoin, Monero, Zcash) and use our common framework to draw detailed and meaningful comparisons.

Keywords: Anonymity · Cryptocurrencies

1 Introduction

Cryptocurrencies are undeniably one of the most attention-grabbing developments in security research of the last decade. They continue to open up new classes of inquiries for the crypto- and distributed-systems communities, while also arguably offering tangible financial benefits as alternatives to traditional fiat currencies.

Thanks to the blockchain technology, *trust*, the grease of financial transactions, can now be *inferential* rather than *axiomatic*. The decentralised nature, ease of conducting cross-border transactions, resistance to censure, and promises (or hopes) of privacy and anonymity, are factors that have contributed towards this popularity. Bitcoin is the first and by far the most widely used *true*[1] cryptocurrency at the time of this writing, and has attracted much attention with respect to its privacy and anonymity aspects.

[1] By which we mean: permissionless, fully decentralized, with democratic governance, and transparently operated—in other words, conducive to trust from first principles.

Anonymity broadly means that an entity cannot be uniquely identified in a given setting. This concept has been widely discussed in the context of anonymous communication and information sharing. Consequently, many terminologies [22, 30] and theoretical models such as k-anonymity have been developed to model anonymity [10, 29]. For better or worse, these available theoretical frameworks have been borrowed for discussing anonymity in cryptocurrencies.

Many traditional currency schemes are centralised systems where customers depend on another party to preserve their privacy. For example, in a banking model, banks are bound by regulation to preserve the confidentiality of customer information. If the transaction history of a particular individual or entity were exposed to an outsider, it could result in many undesirable consequences, from a subjective sense of betrayal, to more concrete abuses such as misuse of information to gain undue advantages in contract bidding. Even worse, if currency units came attached with transaction histories, that could lead to the blacklisting of specific units based on their use in unlawful activities in the past, even though the units may have had only uncontroversial uses afterwards.

Anonymity of cryptocurrencies has received much attention since the current Bitcoin framework is claimed to provide only a form of 'pseudonymity' as transactions are linked to payment addresses in a big graph that is visible to all [15, 9]. Detailed analyses of public transaction data, such as the work presented in [19, 6, 26], have shown that it is possible to uncover behaviour patterns of Bitcoin users and trace their identities in real life.

As a result of this tension between the need for, and the lack of, improved anonymity in cryptocurrencies, a lot of energy has been expended to fulfil that demand. Some solutions are centered around improving the Bitcoin framework (e.g. Zcash) whereas other approaches have sought to revisit the blockchain machinery in the design of new cryptocurrency schemes (e.g. Monero). Despite many such solutions making claims of "anonymity", some studies claim that those could still be subject to deanonymisation [18, 20].

As rationalised in [4], despite a large number of studies on cryptocurrency anonymity, no standardised means are available to evaluate the actual level of privacy achieved by different cryptocurrencies. Many studies have been conducted in isolation using various metrics, with the consequence that it is not feasible to compare and benchmark the anonymity landscape in a reliable manner across various schemes. To make matters worse, it turns out that the very notion of anonymity itself, in such complex multi-party systems as decentralized cryptocurrencies, has been until now very poorly understood, and is anything but clear-cut. We discuss the specifics in a separate report [5].

1.1 Our Contribution

The present study was initially motivated by the works of [3, 4, 9, 15], which lifted the veil on the multiplicity of anonymity notions for cryptocurrencies, but stopped short of actually providing a crisp formalism for defining and using those notions. Over the course of this study, we identified a very *fine-grained* structure for the intuitive notion of payment anonymity, parameterised through

qualitative distinct definitions that are all sensible and justifiable in appropriate scenarios [5].

Our purpose in this work, is to analyse the multiple precise ways in which a broad notion of anonymity can be envisaged, and we provide a common game-based security template that combines a massive group of explicit attacker scenarios. Indeed, our notions generalise many security notions familiar to cryptographers such as known vs. chosen plaintext, forward security, indistinguishability, active vs. passive adversaries, and so on. The fact that we consider all of these security dimensions simultaneously multiplies the number of definitions, but also allows us to meaningfully understand and compare the anonymity of systems that differ along multiple dimensions. However, it should be noted that we do not intend to address the anonymity of the underlying construction of currency schemes in this work i.e. consensus or communication mechanisms.

Our framework is based around the fundamental notion of *distinguishability*, leading to a security concept of *indistinguishability*, likely familiar to readers from other security definitions. These notions are further particularized to certain subjects such as *transaction value*, *sender*, *recipient* and *metadata*, and parameterised across multiple dimensions based on which information and capabilities are given to the adversary [5].

Our main contribution here is to demonstrate the concrete potential of our model by analysing the anonymity landscape from a few major cryptocurrency implementations. We start with a simple Trusted Third Party scheme as a benchmark and show that it is, as expected, anonymous against all adversaries appropriate to the trusted third party model. We then study Bitcoin, which still receives much criticism in relation to anonymity. In addition, we also examine Zcash, Monero and Mimblewimble; three cryptocurrency schemes with diverse implementations, which have recently become popular due to their claims for improved anonymity.

The take-away message from our effort is that (financial) anonymity is not an all-or-nothing binary property; it is far more subtle. We fully intend that our framework be used to clearly spell out what aspects of privacy a certain coin does or does not satisfy, across diverse implementations. Of course, one could be content with asking for *absolute fungibility* (think: isotopically pure melted gold), but that is likely not to lead us anywhere, as no cryptocurrency in existence comes close to reaching that goal. This only makes the need for a (much) more refined model all the more pressing.

Organisation. Subsequent sections of this paper are organised as follows. We first present a brief summary of related studies where theoretical notions of anonymity have been discussed with reference to cryptocurrencies. We then set forth the preliminaries of our framework, while emphasising its features and relevant anonymity definitions. Next, we present the analysis outcomes followed by a detailed discussion on the significance of this work.

1.2 Other Related Work

As mentioned at the outset, many early studies have focused on quantitative analysis of publicly available Bitcoin transaction data such as payment addresses and values as the Bitcoin blockchain records all transaction details publicly. As claimed in [25, 26, 17, 28], such public transaction data can be used to compromise the anonymity of Bitcoin users by studying behavioural patterns and transaction flows etc. Moving forward, some have attempted to formalise anonymity concepts in a theoretical manner, yet such are not standardised across different constructions. For example, Androulaki et al. [6] conducted an analysis of Bitcoin privacy based on *activity unlinkability* and *profile indistinguishability* with respect to addresses and transactions, which was also used in [21] to analyse Bitcoin network data. Conversely, [33] uses the notion of *unlinkability* with respect to linking different entities as formulated by [22].

More recently, new currency schemes have emerged with more promising anonymity expectations, which has led to more concrete formalisation of anonymity concepts. Zcash is one such scheme which offers improved anonymity levels through its 'shielded transactions', which conceal payment addresses and values. Yet, experimental studies in [14, 24, 34] have shown that it is prone to *linkability* of transactions with corresponding payment addresses.

The Cryptonote protocol, which forms the foundation for several currency systems, is claimed to satisfy anonymity in terms of *unlinkability* and *untraceability* [31]. Their interpretation of *unlinkability* is more specific in that, given two transactions, it should not be possible to identify whether both transactions were intended to the same party. *Untraceability* on the other hand is defined as the inability to identify the corresponding sender for a given transaction. Nevertheless, subsequent studies in [20, 32] claim that Monero, which originated from the Cryptonote protocol, is prone to deanonymisation attacks through analyses of public transaction data.

Fungibility, which is the property of every currency unit being identical, is regarded by many as an elementary requirement of any currency scheme, but it is a tall order. It is well accepted that Bitcoin is not fungible [9, 27]. Although it has been claimed in [24] that Zcash achieves fungibility through its use of zero-knowledge SNARK proofs, the survey study of [9] makes the countermanding claim that Mimblewimble [23] is the only cryptocurrency scheme to do so. Even so, the original Mimblewimble is insecure, and the fix proposed in [11], by making it preserve a lot more data, reintroduces coin history thereby negating the original fungibility claim.

Methods such as network analysis proposed in [7] and transaction graph-based analysis in [8] provide means for modelling anonymity through experimental analysis, which however may not be possible across different constructions. In comparison, our model deviates from this as our emphasis is on modelling anonymity from first principles in any currency scheme.

With the increasing complexity, comparing anonymity of cryptocurrencies has become a challenge. Surveys conducted in [4, 9, 15] present independent categorisations of cryptocurrencies based on different anonymity properties such as

untraceability, *unlinkability*, *fungibility*, *hidden values* and *hidden IP addresses*. In a different approach, [3] provides a systematic grouping of a subset of cryptocurrencies in terms of four privacy tiers; *pseudonymity*, *set anonymity*, *full anonymity* and *confidential transactions*, based on unlinkability and hidden user identities. Yet, all such categorisations provide a very high level picture of anonymity levels based on the techniques used by the schemes, which is orthogonal to our work.

Nonetheless, these studies, mostly based on experimental analyses or specific constructions, do not necessarily facilitate the assessment and comparison of cryptocurrencies in terms of a common, *fine-grained*, *formal* qualitative model of anonymity.

2 Anonymity framework

Our work is based on an abstract model of a cryptocurrency scheme, depicting the overall functionality of a generic cryptocurrency scheme. We construct an anonymity framework for this scheme through a game-based approach. We chose game-based definitions over the UC framework because the former are intuitive and can be agreed upon by non-specialists (much less non-cryptographers). This is essential as a bridge between theory and applications. Further, UC, though a very nice theoretical methodology, is best suited for small primitives whose ideal functionalities may still have a clean description, which is certainly not the case with cryptocurrencies. This abstract anonymity model is formalised in detail in [5]; here we summarise the points of interest for our purposes in this paper of analysing concrete cryptocurrency schemes.

2.1 A generic cryptocurrency scheme

We define a currency scheme in terms of a security parameter $\lambda \in \mathbb{Z}^+$, and a system state consisting of payment addresses, each having a public key (a_{pk}) and a private key (a_{sk}), and transaction history. A transaction takes place between senders and recipients with inputs such as values and other metadata (such as IP addresses). Each transaction comprises private and public parts (t_s and t_p), with the latter being broadcast to the network. A mint operation collects new transactions on the network at any given time and generates a new state. New currency units may be created as a result of minting, as per its underlying implementation. Then an adjudicate operation selects the rightful new state of the system. Accordingly, consecutive states of the scheme form a partial ordering.

It should be noted that we model only the generic functionality of a cryptocurrency scheme in this scheme. Hence, we do not consider the specifics of the underlying consensus mechanism or the communication here.

Further details of the algorithms are included in Appendix A. The full theoretical study [5] details how correctness and security of this abstract currency scheme are established.

Definition 1. *A cryptocurrency scheme Π, is defined in terms of security parameter $\lambda \in \mathbb{Z}^+$ and with the functionality prescribed by means of a set of algorithms; {Init, CreateAddr, IsValidPubAddr, IsValidSecAddr, GetBalance, CreateTxn, IsValidPubTxn, IsValidSecTxn, ExtractSenderPubAddr, ExtractRecipientPubAddr, ExtractInputVal, ExtractOutputVal, IsMintable, Mint, Adjudicate, IsValidState, IsGenesisState, CreateCheckpointState, RetrieveCheckpointState}.*

2.2 A comprehensive adversarial capability model

We define a comprehensive adversarial model to include a wide range of capabilities for adversarial power and knowledge, represented by a set of parameters (Table 1). Adversarial knowledge of public/secret keys of senders/recipients, values, metadata and other transaction-related data are modelled by ψ. Here, metadata refers to implementation specific data that appear in a transaction such as IP addresses, while the knowledge of a transaction represents other related information as shown in Table 1. Adversarial power is modelled by the adversary's ability to modify the state (δ), to control state initialisation in the experimental setup (α), and to cause minting to fail during the game (β). This parametrisation accommodates a wide range of adversaries; passive with all parameters equal to '0', static with δ, $\alpha \leq 1$ and adaptive with parameter values greater than 1. It is assumed that the adversary ($\mathcal{A}$) has oracle access to hidden entities via opaque handles.

Table 1: Parameters of the adversarial capability model

Param. value	Adversarial knowledge					Adversarial power		
	Sender public/ secret keys ψ_{pk_s}/ψ_{sk_s}	Recipient public /secret keys ψ_{pk_r}/ψ_{sk_r}	Transaction value ψ_v	Transaction metadata ψ_m	Transaction ψ_t	State manipulation δ	State initialisation α	Cause mint to fail β
0	Hidden	Hidden	Hidden	Hidden	Hidden	Hidden	Hidden randomness, honest init	Not allowed
1	Hidden but revealed at the end	Hidden but revealed at the end	Hidden but revealed at the end	Hidden but revealed at the end	t_p is revealed	Can view the state	Public randomness, honest init	Allowed
2	Access public keys through oracle	Access secret keys through oracle	Chosen by Oracle and known	Chosen by oracle and known	t_s is revealed	Can manipulate the state	Public randomness, adversarial init	-
3	$\mathcal{A}$ chooses identity, oracle creates addresses	$\mathcal{A}$ chooses randomness, oracle creates addresses	$\mathcal{A}$ chooses values	$\mathcal{A}$ chooses metadata	Randomness is revealed	-	Hidden randomness, adversarial init	-
4	$\mathcal{A}$ generates address	$\mathcal{A}$ generates address	-	-	$\mathcal{A}$ chooses randomness	-	-	-
5	-	-	-	-	$\mathcal{A}$ creates transaction	-	-	-

$\underline{\mathrm{Exp}^{Anonymity}_{\pi,\mathcal{A},\mathcal{C},\omega,\psi,\delta,\alpha,\beta}(\lambda)}$

1. Initialises the state based on the parameter α.
2. $\mathcal{A}$ provides inputs based on the capabilities decided by the parameters ψ and δ.
3. $\mathcal{C}$ checks the inputs against the parameters and if there is any discrepancy, $\mathcal{A}$ loses the game.
4. If $\psi_t \neq 5$, $\mathcal{C}$ creates two transactions t_{p_0} and t_{p_1} based on the test variable/s parameterised by ω. $\mathcal{A}$ continues to evolve the state through appropriate oracle calls. If $\beta = 0$, then $\mathcal{A}$ loses the game if any mint operation fails. If $\psi_t = 5$, $\mathcal{C}$ accepts the transactions provided by the adversary.
5. $\mathcal{C}$ picks a bit $b \in \{0, 1\}$ and mints the transaction t_{p_b}.
6. Based on the parameter values of ψ, corresponding data are revealed to $\mathcal{A}$, but $\mathcal{A}$ is not allowed to create or mint any further transactions.
7. $\mathcal{A}$ makes a guess for the bit and wins the game if the guess is correct.

Fig. 1: Anonymity Game

2.3 All-in-one generic flexible Anonymity game

We now formulate a generic game, that captures different attacker scenarios, each depicting a unique aspect of anonymity. We use the variable $\omega = (\omega_s\omega_r\omega_v\omega_m)$ to set the test variable/s (the attacker's goal); sender (s), recipient (r), value (v) and metadata (m). Accordingly, we develop a set of definitions around the fundamental concept of *indistinguishability*, which requires the adversary to distinguish between two known entities in the game. We also define a weaker notion, *unlinkability*, in which case, the two entities to choose between are not known to the attacker explicitly, but rather by their history in previous transactions. We define the *Anonymity game* between a challenger ($\mathcal{C}$) and an adversary ($\mathcal{A}$) as given in Figure 1 and further explained in Appendix B.

2.4 Notions of anonymity

Unsurprisingly, there are over 600,000 distinct combinations of ω, ψ, δ, α and β alone, resulting in different attacker scenarios, which reveal the complexity of what it means for a currency scheme to be anonymous. While some notions may not result in apprehensible real world scenarios, others may assist in assessing different aspects of anonymity. Each notion is defined based on a unique adversary, as per the goal, knowledge and power (i.e. GOAL-KNOWL-POWER), which is also given as a unique parameter vector, ω-ψ-(δ,α,β) setting the game. The strongest adversary is assigned the full power (to manipulate the state setup, the state, and minting) and the full knowledge (of secret keys of senders/recipients, values, metadata, transaction), which we call a FULL-FULL adversary. The weakest is named a NIL-NIL adversary, with no power nor knowledge. Others are named accordingly to reveal relevant adversarial capabilities.

3 Analysis

Formally proving consistency and all implications and separations that exist across our 600,000 flavours of anonymity would be far beyond the scope and space available in this paper (but see [5] for details). Instead, we will focus on specific notions of interest towards our purpose here to demonstrate how our framework can be deployed to very precisely characterise concrete properties of actual cryptocurrencies. We consider *Indistinguishability* (IND) and *Unlinkability* (ULK) notions related to sender (S), recipient (R) and value (V) (not

metadata which may be different in each implementation), in a bid to provide a meaningful comparison across real-world currency schemes.

We start by analysing a Trusted Third Party scheme, which has a very high level of anonymity, as a benchmark for comparison. Then, we study the Bitcoin system, followed by Zcash, Monero and Mimblewimble, all three of which claim to have convincing anonymity levels, yet have very diverse implementations.

3.1 A Trusted Third Party (TTP) scheme

Consider a TTP scheme where a trusted Central Authority (CA) operates a currency scheme. The CA registers users, validates, creates and mints transactions upon request by users. We also assume that the CA communicates with all other parties over authenticated channels and only honours requests from the rightful owners of accounts. A user registers one or more accounts with the CA and maintains funds under those registered identities. No negative fund balances are allowed at any given time. A user can request the CA to create a transaction, and subsequently to mint the transactions and the CA performs corresponding fund transfer/s and creates a transaction record internally. The CA can view the transaction history at any time. With this functionality, there are no public/private keys involved in the scheme and transactions will always be secret, hence the system state is always internal and private.

Adversarial capabilities In the TTP model, CA can have its own state variables outside the challenger and the adversary, and thus is not required to accept the adversarial state. Also, the initial state will be an empty list of transactions, accounts etc., allowing any method of state initialisation possible. Hence we can allow the adversary to take any value for δ and α in our model (Table 1). Further, we assume that transactions are encrypted with an asymmetric system using CA's public key, and hence can be revealed in the end without revealing any information. We model user identities in terms of a single address thereby setting $a_{pk} = a_{sk}$ in our model. To enable the adversary to supply sender/recipient addresses to the challenger, we provide access to an additional oracle `DelegateAccess` to transfer the authority of the addresses controlled by the adversary to the challenger. Thus, the challenger is able to create the transactions required for different scenarios. Note that this oracle is only specific to the TTP functionality, and is reminiscent of how ideal functionalities must be augmented with corruption functions in the UC model.

Analysis of anonymity First, we consider a FULL power adversary (denoted by $(2_\delta,3_\alpha,1_\beta)$), who has the complete knowledge of recipients, value and metadata, but knows senders only by public keys and provides the input transactions to the game (named as PUBS knowledge denoted by $((3,0)_s,(4,4_r),3_v,3_m,5_t)_\psi$) against the goal of S-IND. We name this adversary as S-IND-PUBS-FULL, who in this case cannot learn any new information about the sender corresponding to the minted transaction as the state is private, and thus has negligi-

ble advantage of winning the Anonymity game (given by the parameter vector $(1_s000)_\omega$-$((\mathbf{3},\mathbf{0})_\mathbf{s}, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$). Hence, the TTP scheme is secure against S-IND-PUBS-FULL adversary and also against a S-ULK-NILS-FULL adversary having no knowledge of senders (NILS knowledge) represented by $(1_s000)_\omega$-$((\mathbf{0},\mathbf{0})_\mathbf{s}, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta,3_\alpha,1_\beta)$ by implication. Similar anonymity notions hold for R and V as well. Accordingly, we can say that the scheme is secure even against an adversary with FULL-FULL capabilities, for all entities; i.e. ALL-IND-FULL-FULL setting, as the scheme does not leak any information to the adversary. This is modelled by the vector $(1111)_\omega$-$((4,4)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta,3_\alpha,1_\beta)$, which depicts '*absolute fungibility*' demonstrating the strongest possible level of anonymity in our model.

3.2 Bitcoin

The Bitcoin peer-to-peer cryptocurrency relies on a public blockchain where transaction data are public. Users are identified via public addresses and they initiate transactions using their private keys to spend funds (unspent transaction outputs). Transaction inputs include references to unspent transaction outputs and a set of new outputs with corresponding values, which later becomes inputs to another transaction. In addition, transactions also contain additional data which help in the verification. Participating network nodes compete to create new blocks (mining) to include new transactions in the blockchain and a qualifying block is accepted by the network based on a Proof-of-Work system.

Adversarial capabilities Most parameters in our model directly corresponds to Bitcoin except that there is no secret transaction part of the transaction t_s. Since the scheme does not have a private state, this can be modelled with $\delta \neq 0$ in our model. Similarly, honest state initialisation with hidden randomness is not allowed, and hence we model this by setting $\alpha \neq 0$.

Analysis of anonymity As all Bitcoin transaction details are public, any adversary has non-negligible advantage in winning the game against any test variable (i.e. S, R or V), since they can observe the topology of the transaction graph. Adversaries can create a specific set of transactions (through the oracle) chosen in a way that they can correctly identify the graph (by analysing starting balances of inputs etc.). Hence, it is not secure in any adversary with respect to indistinguishability or unlinkability of S, R or V.

Conversely, consider a weak adversary in our game against an empty test variable, who has no information of the transaction (NIL knowledge), but can view the state setup and the state (VIEW power), denoted by NIL-IND-NIL-VIEW and parameterised by $(0000)_\omega$-$((0,0)_s, (0,0)_r, 0_v, 0_m, 0_t)_\psi$-$(1_\delta,1_\alpha,0_\beta)$. Here, the adversary has to distinguish between two identical transactions carrying same data, except with different randomness. Despite the public transaction history, the adversary cannot identify the correct transaction with a substantial probability, thus making the Bitcoin system secure against this attacker. However, if

we increase at least one capability, the scheme becomes insecure. Thus, we conclude that Bitcoin only satisfies an extremely weak notion in our model, which only provides anonymity against two identical transactions that only differ in the randomness. It should be noted however that we make this claim subject to the computational and operational assumptions of the Bitcoin construction. In fact, the only way to make the scheme anonymous is to make the state private (i.e. $\delta = 0$), which is impossible with the current Bitcoin construction.

3.3 Zcash

Zcash emerged as a result of the efforts of improving the anonymity of Bitcoin. We consider the Zcash Sapling specification for this study. This scheme consists of shielded and transparent payment addresses where transparent addresses and related transactions operate similar to Bitcoin [12]. Here we only consider the transactions between shielded addresses (referred to as addresses hereafter). Each address has a private spending key that allows the owner to spend the coins (notes) sent to that address. Each note is coupled with a unique nullifier generated using the spending key and a note commitment, which is publicly revealed when the note is created. Without the private key, it is infeasible to link a note commitment to its nullifier. An unspent note in Zcash is a note with a publicly revealed commitment and a hidden nullifier. When a shielded transaction is created, nullifiers of input notes and commitments of output notes are revealed. In addition, the value of a shielded transaction is also hidden, and is revealed through value commitments related to input and output notes, and relevant balancing operations are carried out as homomorphic operations. Further, zk-SNARK primitives are used for functions such as proving the ownership of notes, verifying and validating transactions [12].

Adversarial capabilities Similar to Bitcoin, we can model Zcash addresses through the payment addresses a_{pk}, a_{sk} in our model. As the state is public, it can be modelled by setting $\delta \in \{1, 2\}$ and $\alpha \in \{1, 2, 3\}$. In shielded transactions, senders and recipients correspond to the nullifiers of input notes and to the commitments of output notes respectively. Further, the values of input/output notes are also concealed as value commitments. t_p represents nullifiers of input notes, output note commitments and value commitments whereas actual input/output notes and relevant data can be modelled by t_s. The knowledge of secret keys (i.e. (ψ_{sk_s})) is required to link the nullifiers of input notes to their owners (senders) and the private keys of recipients (i.e. ψ_{sk_r}) should be known to link the note commitments of output notes to their owners (recipients).

Analysis of anonymity We begin by analysing the unlinkability property. Although the linkability of Zcash transactions is explored in literature such as [24] with respect to transactions involving both shielded and transparent addresses, we only consider shielded addresses here. Consider an adversary for S-ULK who has all powers except to cause minting to fail (ACTIVE power), and has full

knowledge of recipients, values, metadata and public transaction data (output note commitments), except the senders (NILS-PUBT knowledge) which we capture in a parameter vector $((0,0)_s,(4,4)_r,3_v,\ 3_m,1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$. The adversary cannot obtain any additional knowledge of the transaction as output note commitments do not leak any information about the sender, and thus has a negligible advantage over winning the game. Hence, Zcash scheme is secure in S-ULK-NILS-PUBT-ACTIVE. If the adversary is given more powers to cause minting to fail (i.e. FULL power), then he may gain additional information about account balances etc., making the system insecure against S-ULK-NILS-PUBT-FULL. Further, for any $\psi_t > 1$, the adversary has access to additional knowledge about the transaction which makes the system insecure. Hence, we can also show that Zcash is secure in R-ULK-NILR-PUBT-ACTIVE, but not in R-ULK-NILR-PUBT-FULL.

The scheme also satisfies S-IND-PUBST-ACTIVE security, as the knowledge of senders' public keys and public transaction data (PUBST knowledge) does not reveal any information about the nullifiers of input notes (i.e. $((3,0)_s,(4,4)_r,3_v$, $3_m,1_t)_\psi$-2_δ-3_α-$0_\beta)$. Yet, with the same reasoning as with S-ULK, Zcash fails in S-IND-PUBST-FULL. Similarly, Zcash is secure in R-IND-PUBRT-ACTIVE, but not in R-IND-PUBRT-FULL.

When testing for the value (i.e. $\omega_v = 1$), the system is secure against a FULL power adversary, having only the knowledge of public keys of senders, recipients and public transaction, but with no knowledge of the values (NILV-PUBSRT knowledge) as in V-ULK-NILV-PUBSRT-FULL (given by $(001_v0)_\omega$-$((3,0)_s,(3,0)_r,\ \mathbf{0_v},3_m,1_t)_\psi$-$(2_\delta$,$3_\alpha$,$1_\beta))$, since failed minting attempts do not reveal any information despite knowing public keys. Hence, the level of anonymity with respect to V depends on the knowledge of secret keys as the value is hidden. Therefore, V-IND property holds only for an adversary with ACTIVE power and PUBSRT knowledge; i.e. V-IND-PUBSRT-ACTIVE notion denoted by $(001_v0)_\omega$-$((3,0)_s,(3,0)_r,\ \mathbf{3_v},3_m,1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$.

Accordingly, we can say that Zcash satisfies the strongest level of anonymity against a PUBSRT-ACTIVE adversary for all test variables given by ALL-IND-PUBSRT-ACTIVE setting and parameterised by $(1111)_\omega$-$((3,0)_s,(3,0)_r,3_v,3_m,$ $1_t)_\psi$-$(2_\delta$,3_α,$0_\beta)$. Hence, Zcash achieves higher anonymity prospects compared to Bitcoin, and is bounded by the knowledge of secret keys of payment addresses.

3.4 Monero

Monero is another cryptocurrency that claims improved anonymity based on several cryptographic primitives such as ring signatures and stealth addresses to achieve anonymity with respect to senders and recipients [32]. In addition, Ring Confidential Transactions (RingCT) are used to conceal transaction values through value commitments [32]. Each user has two pairs of private/public keys as spend and view keys. A sender creates a one-time public key (stealth address) for each output using recipients' public keys. The sender mixes the actual inputs with a set of additional random public keys (aka mixins) using ring signatures, to produce a signature for the ring of inputs. The one-time public key, the

signature and the public keys of inputs (in the ring) are submitted to the network along with other transaction data [2, 32]. A sender can include an (optional) pre-agreed, encrypted payment ID, enabling respective recipients to identify the sender using their private keys. The recipients can retrieve outputs using both their private/public view keys and can spend them using the spending keys. Outsiders can only view the public keys in the ring (of probable senders), with each being an equally probable input to the transaction.

Adversarial capabilities Similar to others, the Monero Blockchain state is also public, thus we set the parameters $\delta \in \{1, 2\}$ and $\alpha \in \{1, 2, 3\}$ as before. However, most of the transaction data (e.g. actual senders, recipients, values etc.) are hidden from the public. We map public and private keys of both spend and view keys collectively to public/private keys in our model. We represent mixin data by metadata in our model, i.e. ψ_m. The knowledge of secret keys of a sender/recipient is sufficient to identify the respective sender/recipient of a transaction, respectively. Conversely, the knowledge of ψ_{sk_r} alone is not enough to identify the sender, if the transaction does not contain a payment ID.

Analysis of anonymity First we look at the unlinkability property of Monero, which is analogous to the notion of traceability of Monero, referred to in [16, 32]. We consider the S-ULK-NILS-PUBT-ACTIVE notion as with Zcash, without the knowledge of likely senders (i.e. $((0,0)_s, (4,4)_r, 3_v, 3_m, 1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$). The state only reveals the public keys of a possible set of senders, but not the recipients nor the value. Yet, if the adversary chooses the mixins, then he has additional information about the sender as ring participants are public. Thus, Monero cannot be secure if the adversary knows the mixins in the ring. Hence, we define a weaker adversary by setting $\psi_m = 0$ in our model, with an adversary having no knowledge of sender or metadata (NILSM-PUBT knowledge), in which case Monero is secure in S-ULK-NILSM-PUBT-ACTIVE (modelled by $(1_s000)_\omega$-$((0,0)_s, (4,4)_r, 3_v, 0_m, 1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$).

With S-IND-PUBSMT-ACTIVE, Monero cannot be secure as the knowledge of mixins along with the public keys of probable senders may leak information about the actual sender. Further, the knowledge of the transaction t_p can also leak information about the mixins. Hence, we consider a weaker adversary with no knowledge of metadata or the transaction (i.e. NILMT-PUBS knowledge) with S-IND-NILMT-PUBS-ACTIVE, represented by $(1_s000)_\omega$-$((3,0)_s, (4,4)_r, 3_v, 0_m, 0_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$, who fails against Monero. However, these claims may be broken if the mixins are not chosen carefully by the sender.

With recipient anonymity, we can see that Monero complies with R-ULK-NILR-PUBT-ACTIVE as funds are received by stealth addresses which can be claimed only by the recipient with the matching private key. This notion of unlinkability closely relates to the notions described in [16, 32]. Similarly, Monero is also secure in R-IND-PUBRT-ACTIVE as the knowledge of recipients' public keys do not reveal anything about the stealth addresses. Further, as values are hidden, Monero's anonymity in V reduces to the knowledge of the secret

keys of the senders/recipients similar to Zcash and hence it satisfies V-ULK-NILV-PUBSRT-FULL and V-IND-PUBSRT-ACTIVE notions. As with Zcash, S-IND, S-ULK, R-IND, R-ULK and V-IND goals fail against a FULL power adversary with the information leakage from failed minting. Thus, we can see that the maximal anonymity level satisfied by Monero is the ALL-IND-NILMT-PUBSR-ACTIVE security given by the parameter vector $(1111)_{\omega}$-$((3,0)_s, (3,0)_r, 3_v, 0_m, 0_t)_{\psi}$-$(2_{\delta}$,$3_{\alpha}$,$0_{\beta})$.

3.5 Mimblewimble

The Mimblewimble protocol focuses on improving anonymity and scalability through confidential transactions and transaction aggregation [13, 11]. We study the Grin implementation for this analysis [1]. A coin in this is a commitment, $C = vH + rG$ where v is the value, r is the randomness (hence the private key of the coin), and H, G are generators of a discrete logarithm [11]. The opening of the commitment of a coin is necessary to spend that coin, which requires the corresponding secret key (r). The sender sends the input coins (commitments) to the recipient over an authenticated channel, who then adds the commitments to the output coins (by including individual private keys) and a partial signature for the transaction (using a random nonce), which is sent back to the sender. The sender validates the received signature and generates his portion of the signature and broadcasts the transaction on the network, which is verified (via the relevant public key generated through public transaction data) and minted by the network nodes subsequently. Transactions are included in the blockchain, subject to transaction aggregation which hides the actual transaction graph [1]. A typical transaction consists of input/output coins (commitments) and relevant range proofs (proving that values are positive), transaction fee and a signature.

Adversarial capabilities As before, the Mimblewimble state is public. However, transactions hide the senders, recipients and the values while revealing only the commitments required to validate a given transaction by any third-party. The knowledge of the secret key (r) of the coins is required to produce a valid signature for a transaction, allowing the rightful owners to spend the coins. Hence we model the knowledge of secret keys of inputs and outputs as ψ_{sk_s} and ψ_{sk_r} respectively in our model. The knowledge of the public key of the transaction can be modelled as ψ_{pk_s} and ψ_{pk_r}. As the sender initiates a transaction by communicating with relevant recipients, when the adversary knows any of the secret keys, there there is no anonymity (i.e. when $\psi_{sk_s}, \psi_{sk_r} > 0$ in the model).

Analysis of anonymity Consider the S-IND-PUBSRT-ACTIVE notion, which is parameterised by $(1_s000)_{\omega}$-$((3,0)_s, (3,0)_r, 3_v, 3_m, 1_t)_{\psi}$-$(2_{\delta}$,$3_{\alpha}$,$0_{\beta})$. Despite learning the value, metadata and public transaction data, the adversary is not able to distinguish between any sender, as secret keys are not known, thus making Mimblewimble secure against this adversary. However, any further leakage of information (i.e. private keys of recipients) would compromise anonymity. Similarly, the

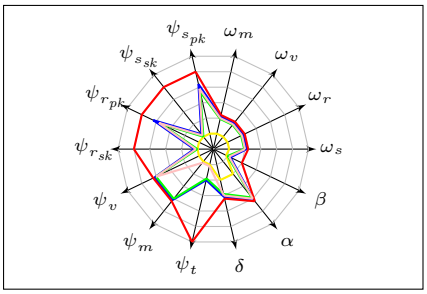

Fig. 2: Maximal anonymity notions satisfied by: TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)

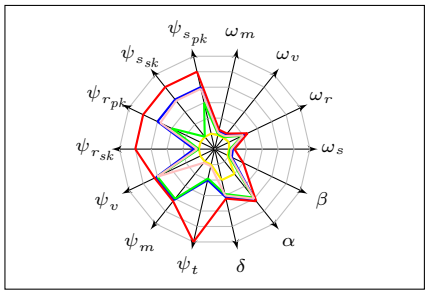

Fig. 4: Recipient indistinguishability satisfied by: TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)

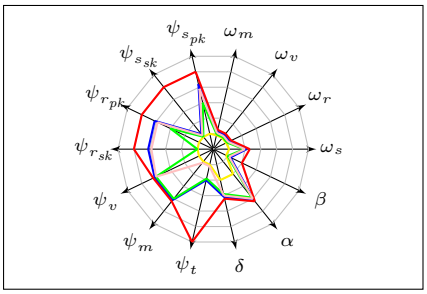

Fig. 3: Sender indistinguishability satisfied by: TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)

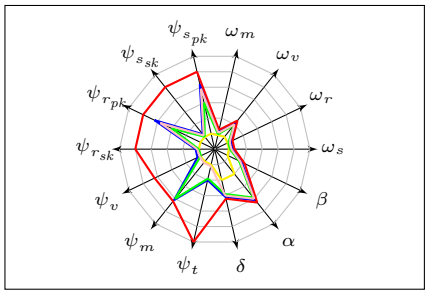

Fig. 5: Value unlinkability satisfied by: TTP (red), Bitcoin(yellow), Zcash(blue), Monero(pink), Mimblewimble(green)

notion of S-ULK-NILS-PUBRT-ACTIVE denoted by $(1_s000)_\omega$-$((0,0)_s,(3,0)_r, 3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$ is also satisfied by implication. With a similar argument, we can show that it also satisfies R-IND-PUBSRT-ACTIVE and R-ULK-NILR-PUBST-ACTIVE. With V-IND, we can see that it is secure in V-IND-PUBSRT-ACTIVE as the value is hidden similar to Zcash and Monero, and hence also secure in V-ULK-NILV-PUBSRT-FULL. Thus, we can conclude that Mimblewimble satisfies strongest anonymity with respect to ALL-IND-PUBSRT-ACTIVE, denoted by the vector $(1111)_\omega$-$((3,0)_s,(3,0)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$.

4 Discussion

While anonymity on the surface looks like an atomic notion, it is evident from the above analysis that it is actually quite quirky and splits apart under a powerful microscope. These findings reveal how minute differences of anonymity exist among the currency schemes studied, as illustrated by the figures 2 to 5. Figure 2 compares the schemes with respect to the maximal anonymity notion[2]. Compared to the TTP scheme, the other four schemes show weaker anonymity

[2] I.e. where the adversary has to distinguish between two transactions that differ in all aspects: sender, receiver, value and metadata

prospects, proving that they do not meet the criteria for "absolute fungibility". As expected, Bitcoin demonstrates the weakest anonymity of all. Conversely Zcash, Monero and Mimblewimble demonstrate improved anonymity with minor deviations among them. Zcash shows the highest level while Mimblewimble shows weaker anonymity with respect to the participants of a transaction and in Monero, anonymity is compromised when details of the choice of mixins are leaked to the adversary. Nevertheless, the knowledge of the randomness of the coins (i.e. $\psi_t > 1$) hinders the anonymity in all three schemes above. Figures 3 to 5 compare three individual anonymity notions related to S-IND, R-IND and V-ULK, and accordingly Zcash is secure against a stronger adversary, compared to other two. However, we only consider shielded addresses here whereas in reality Zcash users have the option to choose transparent addresses, in which case its anonymity is degraded to that of Bitcoin.

On that account, our work shows the very complex nature of the level of anonymity realised by various currency schemes. Consequently, our analysis demonstrates how one can effectively evaluate anonymity in a unified manner across dissimilar implementations as opposed to different categorisations presented in studies such as [4, 9, 15]. Hence, claims for anonymity cannot be made lightly in the presence of such granularity.

Therein, we have presented a qualitative recap of anonymity of a subset of real world cryptocurrency schemes as our major contribution in this work. One may wonder why we need such granularity in modelling anonymity in the context of cryptocurrencies, yet the findings of our case studies show how a minute change such as varying one value along a single dimension, could drastically affect the level of anonymity. The study of such impact and the interdependencies can be regarded as a separate study by itself and hence is recommended as a future work in this context.

As noted earlier, this study does not investigate the privacy aspects of the underlying consensus mechanism or the network of a cryptocurrency scheme, which may leak information independently from the currency scheme in which case it may affect the achievable level of anonymity. Our model already provides a way of capturing this leak as an instance of metadata, but the exact mechanisms by which such leaks occur would have to be studied on a case by case basis and it would be another direction for further study.

5 Conclusion

In this paper, we have demonstrated how anonymity of cryptocurrency schemes can be analysed rigorously by means of a common framework, regardless of the implementation method. Our analysis is centered around an extensive group of anonymity properties based on the fundamental property of indistinguishability, further particularised to varying security subjects and adversarial models. Together, these represent a precise and exhaustive recount of true anonymity achieved by any currency scheme. We are first to be surprised by the richness of this formal financial anonymity landscape, which is unlike other formal notions

of security and privacy seen in cryptography. This reality is well demonstrated through the case studies presented in this work.

Acknowledgements

Xavier Boyen is the recipient of an Australian Research Council Future Fellowship and acknowledges generous support from the grant, number FT140101145. Authors also thank the anonymous reviewers for their comments.

Appendix A Anonymity framework

We provide a summary of the framework here while a comprehensive explanation is available in the report in [5]. We use the notation in Table 2.

Table 2: Notation

Description	**Notation**
Security parameter	$\lambda : \lambda \in \mathbb{Z}^+$
A tuple of random bit strings	$\rho : \rho \in (\{0,1\}^*)^*$
A system state/Current state, a set of states	$p,\ P$
Public key/Private key of a payment address	$a_{pk},\ a_{sk}$
Ordered tuple of one/more addresses (senders/recipients) of secret keys	$\bar{S},\ \bar{R}$
Ordered tuple of one/more addresses containing only public keys	$S,\ R$
Public and private parts of a transaction	$t_p,\ t_s$
Ordered tuples of input and output values of a transaction	$V_{old},\ V_{new}$
Metadata for a transaction	m
Excess value of a transaction (fees + minted value)	V_x
A tuple of addresses of miners	R_m
Return X if y, otherwise return 1	X^y
If $a = \perp$ then return c, else return b	$a?b : c$
If $a = \perp$ then return b, else return a	$a?\_ : b$

Table 3: Functions.

Algorithm	**Syntax**
`Init`	$p_0 \leftarrow \texttt{Init}_\pi(1^\lambda)$
`CreateAddress`	$\perp \vee (a_{pk}, a_{sk}, t_p, t_s) \leftarrow \texttt{CreateAddr}_\pi(p, d; \rho)$
`IsValidPubAddr`	$\{0,1\} \leftarrow \texttt{IsValidPubAddr}_\pi(a_{pk}, p)$
`IsValidSecAddr`	$\{0,1\} \leftarrow \texttt{IsValidSecAddr}_\pi(a_{pk}, a_{sk}, p)$
`GetBalance`	$\perp \vee Bal \leftarrow \texttt{GetBalance}_\pi(a_{pk}, a_{sk}, p)$
`CreateTxn`	$\perp \vee (t_s, t_p) \leftarrow \texttt{CreateTxn}_\pi(R, V_{new}, \bar{S}, V_{old}, m, p, \rho)$
`IsValidPubTxn`	$\{0,1\} \leftarrow \texttt{IsValidPubTxn}_\pi(t_p, p)$
`IsValidSecTxn`	$\{0,1\} \leftarrow \texttt{IsValidSecTxn}_\pi(t_p, t_s, p)$
`ExtractSenderPubAddr`	$\perp \vee S \leftarrow \texttt{ExtractSenderPubAddr}_\pi(t_p, t_s, p)$
`ExtractRecipientPubAddr`	$\perp \vee R \leftarrow \texttt{ExtractRecipientPubAddr}_\pi(t_p, t_s, p)$
`ExtractInputVal`	$\perp \vee V_{old} \leftarrow \texttt{ExtractInputVal}(t_p, t_s, p)$
`ExtractOutputVal`	$\perp \vee V_{new} \leftarrow \texttt{ExtractOutputVal}(t_p, t_s, p)$
`IsMintable`	$\{0,1\} \leftarrow \texttt{IsMintable}_\pi(\{t_p\}, p)$
`Mint`	$\perp \vee (p', V_x) \leftarrow \texttt{Mint}_\pi(\{t_p\}, R_m, p)$
`Adjudicate`	$p' \in P : \quad p \vee p' \leftarrow \texttt{Adjudicate}_\pi(P, p)$
`IsValidState`	$\{0,1\} \leftarrow \texttt{IsValidState}_\pi(p, \lambda)$
`IsGenesisState`	$\{0,1\} \leftarrow \texttt{IsGenesisState}_\pi(p, \lambda)$
`RetrieveCheckpointState`	$\perp \vee p_c \leftarrow \texttt{RetrieveCheckpointState}_\pi(p)$
`CreateCheckpointState`	$\perp \vee p_c \leftarrow \texttt{CreateCheckpointState}_\pi(p)$
`AdditionalFunctionality`	$(outputs) \leftarrow \texttt{AdditionalFunctionality}(inputs)$

Functionality of a generic cryptocurrency scheme We define the algorithms of the currency scheme in Table 3. There may be additional functionality associated with real world cryptocurrency systems, e.g. Smart contracts with Ethereum. In order to capture such additional features, we define a supplementary function `AdditionalFunctionality`. This enables us realise the security implications of functionality of a scheme that may be outside our base model.

Appendix B Anonymity

We present the Anonymity game and required helper functions here. Helper functions check the adversarial conditions of inputs at the start of the game (`CheckAdvConditions`) and reveals data in the end (`RevealData`) based on the parameter ψ (Figure 6). Moreover, the test variable, $\omega = (\omega_s, \omega_r, \omega_v, \omega_m)$ with each $\omega_x \in \{0, 1\}$ indicates which entity is being tested in a given instance of the game. The adversarial inputs are crafted based on the ω, ψ, δ, α and β parameters. Figure 7 illustrates the game.

$\texttt{RevealData}(t_p, \omega, \psi, A^*_{\mathcal{O}}, T^*_{\mathcal{O}}, T_{\mathcal{O}}, p_1)$

1. $(\psi_{s_{pk}}, \psi_{s_{sk}}, \psi_{r_{pk}}, \psi_{r_{sk}}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$; $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \omega$
2. $t_p \leftarrow \texttt{LookupPubTxn}(t_p, T^*_{\mathcal{O}})$
3. $t_s \leftarrow AA.\texttt{Lookup}(t_p, T_{\mathcal{O}})$
4. $S \leftarrow \texttt{ExtractSenderPubAddr}_\pi(t_p, t_s, p_1)$
5. $R \leftarrow \texttt{ExtractRecipientPubAddr}_\pi(t_p, t_s, p_1)$
6. $V_{old} \leftarrow \texttt{ExtractInputVal}_\pi(t_p, t_s, p_1)$
7. $V_{new} \leftarrow \texttt{ExtractOutputVal}_\pi(t_p, t_s, p_1)$
8. $m \leftarrow \texttt{ExtractMetadata}_\pi(t_p, t_s, p_1)$
9. $U_s \leftarrow (S^{\psi_{s_{pk}}}, (\texttt{LookupSecAddr}(S, A^*_{\mathcal{O}}))^{\psi_{s_{sk}}})$
10. $U_r \leftarrow (R^{\psi_{r_{pk}}}, (\texttt{LookupSecAddr}(R, A^*_{\mathcal{O}}))^{\psi_{r_{sk}}})$
11. $U_v \leftarrow ((V_{old}, V_{new})^{\psi_v})$; $U_m \leftarrow (m)^{\psi_m}$
12. $U_t \leftarrow (t_p^{\psi t}, t_s^{(\psi_t=2)}, \rho_t^{(\psi_t=3)})$
13. **return** $(U_s \| U_r \| U_v \| U_m \| U_t)$

$\texttt{CheckAdvConditions}(\omega, \psi, S_0, S_1, R_0, R_1, V_{old_0}, V_{new_0}, V_{old_1}, V_{new_1}, m_0, m_1, A^*_{\mathcal{O}}, A_{\mathcal{O}_{jk}}, D^*_{\mathcal{O}})$

1. $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \Omega$; $(\psi_{pk_s}, \psi_{sk_s}, \psi_{pk_r}, \psi_{sk_r}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$
2. **if** $(\psi_{pk_s} \in \{0,1\}) \wedge (\psi_{sk_s} \in \{0,1\}) \wedge \neg(S_0, S_1 \subseteq A^*_{\mathcal{O}})$, **return** 0
3. **if** $(\psi_{pk_s} \in \{0,1,2\}) \wedge (\psi_{sk_s} \in \{0,1,2\}) \wedge \neg(S_0, S_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{11}}))$, **return** 0
4. **if** $(\psi_{pk_s} = 3) \wedge (\psi_{sk_s} \notin \{3,4\}) \wedge \neg(S_0, S_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{01}}))$, **return** 0
5. **if** $(\psi_{pk_s} \notin \{3,4\}) \wedge (\psi_{sk_s} = 3) \wedge \neg(S_0, S_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{00}}))$, **return** 0
6. **if** $(\psi_{pk_s} = 3) \wedge (\psi_{sk_s} = 3) \wedge \neg(S_0, S_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{10}}))$, **return** 0
7. **if** $(\psi_{pk_r} \in \{0,1\}) \wedge (\psi_{sk_r} \in \{0,1\}) \wedge \neg(R_0, R_1 \subseteq A^*_{\mathcal{O}})$, **return** 0
8. **if** $(\psi_{pk_r} \in \{0,1,2\}) \wedge (\psi_{sk_r} \in \{0,1,2\}) \wedge \neg(R_0, R_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{11}}))$, **return** 0
9. **if** $(\psi_{pk_r} = 3) \wedge (\psi_{sk_r} \notin \{3,4\}) \wedge \neg(R_0, R_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{01}}))$, **return** 0
10. **if** $(\psi_{pk_r} \notin \{3,4\}) \wedge (\psi_{sk_r} = 3) \wedge \neg(R_0, R_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{00}}))$, **return** 0
11. **if** $(\psi_{pk_r} = 3) \wedge (\psi_{sk_r} = 3) \wedge \neg(R_0, R_1 \subseteq AA.\texttt{keys}(A_{\mathcal{O}_{10}}))$, **return** 0
12. **if** $(\psi_m \in \{0,1\}) \wedge \neg(m \in D^*_{\mathcal{O}})$, **return** 0
13. **return** 1

Fig. 6: Additional helper functions for the Anonymity game

$\underline{\mathtt{Exp}^{Anonymity}_{\pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda)}$

1. $A_{\mathcal{O}}, A_{\mathcal{O}_{11}}, A_{\mathcal{O}_{10}}, A_{\mathcal{O}_{01}}, A_{\mathcal{O}_{00}}, T_{\mathcal{O}}, T'_{\mathcal{O}} \leftarrow AA.\mathtt{Init}();\ A^*_{\mathcal{O}}, T^*_{\mathcal{O}}, D^*_{\mathcal{O}} \leftarrow ()$
2. $U \leftarrow \emptyset;\ M_{\mathcal{O}} \leftarrow \{\};\ f_{\mathcal{O}} \leftarrow 0$ ▷ Initialise variables
3. $(p_{\mathcal{O}}, r, s) \leftarrow \mathtt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \gamma) \quad \langle\ p_{\mathcal{O}} \neq \perp\ \rangle$ ▷ State initialisation
4. $(p_{\mathcal{O}}, (S_0, S_1, R_0, R_1, V_{old_0}, V_{new_0}, V_{old_1}, V_{new_1}, T, R_m, m_0, m_1, t_0, t_1, \rho_0, \rho_1), s) \leftarrow$ $\mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, (\emptyset), r, s, \delta) \quad \langle\ p_{\mathcal{O}} \neq \perp\rangle$ ▷ Adversary inputs
5. $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \omega$ ▷ Testing entities
6. $(\psi_{pk_s}, \psi_{sk_s}, \psi_{pk_r}, \psi_{sk_r}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$ ▷ Adversary capabilities
7. **if** $\neg\{\mathtt{CheckAdvConditions}(\omega, \psi, S_0, S_1, R_0, R_1, V_{old_0}, V_{new_0}, V_{old_1}, V_{new_1}, m_0, m_1, A^*_{\mathcal{O}}, A_{\mathcal{O}_{jk}}, D_{\mathcal{O}})\}$ **then return** 0 ▷ Check adversarial conditions on inputs
8. **if** $(\psi_t = 5)$ **then**
9. $\quad (t_{p_0}, t_{s_0}) \leftarrow t_0 \quad \langle \mathtt{IsMintable}_{\pi}(\{t_{p_0}\} \cup T, p_{\mathcal{O}})^{\bar{\beta}}\ \rangle$
10. $\quad (t_{p_1}, t_{s_1}) \leftarrow t_1 \quad \langle \mathtt{IsMintable}_{\pi}(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\bar{\beta}}\ \rangle$
11. **else**
12. $\quad t_{p_0} \leftarrow \mathcal{O}_{txn}(R_0, V_{new_0}, S_0, V_{old_0}, m_0, \psi, p_{\mathcal{O}}, \rho_0) \quad \langle \mathtt{IsMintable}_{\pi}(\{t_{p_0}\} \cup T, p_{\mathcal{O}}, \rho_1)^{\bar{\beta}}\ \rangle$
13. $\quad t_{p_1} \leftarrow \mathcal{O}_{txn}(R_{\omega_r}, V_{new_{\omega_v}}, S_{\omega_s}, V_{old_{\omega_v}}, m_{\omega_m}, \psi, p_{\mathcal{O}}) \quad \langle \mathtt{IsMintable}_{\pi}(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\bar{\beta}}\ \rangle$
14. $b \xleftarrow{\$} \{0, 1\}$ ▷ Challenger picks a bit
15. $(p_1, V_x) \leftarrow \mathtt{Mint}_{\pi}(\{t_{p_b}\} \cup T, R_m, p_{\mathcal{O}})$
16. $U \leftarrow \mathtt{RevealData}(t_{p_b}, \psi, \omega, A^*_{\mathcal{O}}, T^*_{\mathcal{O}}, T_{\mathcal{O}}, p_1)$
17. $(\cdot, b', \cdot) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_3, p_1, (U), r, s, \delta) \quad \langle\ \beta \vee (f_{\mathcal{O}} \neq 1)\ \rangle$
18. **return** $b' \stackrel{?}{=} b$

Fig. 7: Anonymity Game

In this game, we use '$\langle condition \rangle$' notation after an action to check if a valid outcome is obtained and if the condition inside the brackets is false, then the game terminates and the adversary loses the game. Upon submission of valid inputs, the adversary continues to evolve the current state through appropriate oracle queries. If $\psi_t \neq 5$, then the challenger creates two transactions (Fig 7 - lines 12 and 13), or chooses the transactions provided by the adversary otherwise. Out of the two transactions, only one transaction is minted based on the chosen bit b (line 15). Failed mint operations are not allowed except when $\beta = 1$ and to check this condition, the notation '$\langle \mathtt{IsMintable}_{\pi}(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\bar{\beta}}\ \rangle$' is used. In this case, when β= 0, $\bar{\beta} = 1$ and the game continues if $\mathtt{IsMintable}() = 1$. When $\beta = 1$, $\bar{\beta} = 0$ and hence $\mathtt{IsMintable}()^0 = 1$ always and hence the game proceeds. After revealing the relevant data (line 16), the adversary is not allowed to create any transactions involving revealed addresses. The adversary wins the game if the chosen bit is guessed correctly, subject to the condition $\beta \vee (f_{\mathcal{O}} \neq 1)$.

B.1 Anonymity Notions

We summarise some useful anonymity notions with their corresponding parameter vectors in Table 4 below. Formal definitions of these notions are given in [5].

Table 4: Some useful anonymity notions

Goal	**Adversarial Knowledge**	**Adversarial Power**	**Parameter vector**
ALL-IND	FULL	FULL	$(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s,(4,4)_r,3_v,3_m,5_t)_\psi$-$(2_\delta,3_\alpha,1_\beta)$
ALL-IND	NILMT-PUBSR	ACTIVE	$(1_s 1_r 1_v 1_m)_\omega$-$((3,0)_s,(3,0)_r,3_v,0_m,0_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
ALL-IND	PUBSRT	ACTIVE	$(1_s 1_r 1_v 1_m)_\omega$-$((3,0)_s,(3,0)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
S-IND	PUBST	ACTIVE	$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s,(4,4)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
S-IND	NILMT-PUBS	ACTIVE	$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s,(4,4)_r,3_v,0_m,0_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
S-ULK	NILS-PUBT	ACTIVE	$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s,(4,4)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
R-IND	PUBRT	ACTIVE	$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s,(3,0)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
R-ULK	NILR-PUBT	ACTIVE	$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s,(0,0)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
V-IND	PUBSRT	ACTIVE	$(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s,(3,0)_r,3_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,0_\beta)$
V-ULK	NILV-PUBSRT	FULL	$(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s,(3,0)_r,0_v,3_m,1_t)_\psi$-$(2_\delta,3_\alpha,1_\beta)$
NIL-IND	NIL	VIEW	$(0_s 0_r 0_v 0_m)_\omega$-$((0,0)_s,(0,0)_r,0_v,0_m,0_t)_\psi$-$(1_\delta,1_\alpha,0_\beta)$
NIL-IND	NIL	NIL	$(0_s 0_r 0_v 0_m)_\omega$-$((0,0)_s,(0,0)_r,0_v,0_m,0_t)_\psi$-$(0_\delta,0_\alpha,0_\beta)$

References

1. Introduction to mimblewimble and grin (Aug 2020), `https://github.com/mimblewimble/grin/blob/master/doc/intro.md`
2. Alonso, K.M.: Zero to Monero (2020), `https://src.getmonero.org/library/Zero-to-Monero-1-0-0.pdf`
3. Alsalami, N., Zhang, B.: SoK: A systematic study of anonymity in cryptocurrencies. In: 2019 IEEE Conference on Dependable and Secure Computing (DSC) (2019)
4. Amarasinghe, N., Boyen, X., McKague, M.: A survey of anonymity of cryptocurrencies. In: Proceedings of the Australasian Computer Science Week Multiconference. pp. 2:1–2:10. ACSW 2019, ACM, New York, NY, USA (2019)
5. Amarasinghe, N., Boyen, X., McKague, M.: The cryptographic complexity of anonymous coins: A systematic exploration. Cryptology ePrint Archive, Report 2021/036 (2021), `https://eprint.iacr.org/2021/036`
6. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: International Conference on Financial Cryptography and Data Security. pp. 34–51. Springer (2013)
7. Biryukov, A., Tikhomirov, S.: Deanonymization and linkability of cryptocurrency transactions based on network analysis. In: 2019 IEEE European Symposium on Security and Privacy (EuroS P). pp. 172–184 (June 2019)
8. Cachin, C., De Caro, A., Moreno-Sanchez, P., Tackmann, B., Vukolic, M.: The transaction graph for modeling blockchain semantics. IACR Cryptology ePrint Archive **2017**, 1070 (2017)
9. Conti, M., Kumar, S., Lal, C., Ruj, S.: A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials (2018)
10. Díaz, C., Seys, S., Claessens, J., Preneel, B.: Towards measuring anonymity. In: Dingledine, R., Syverson, P. (eds.) Privacy Enhancing Technologies. pp. 54–68. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
11. Fuchsbauer, G., Orrù, M., Seurin, Y.: Aggregate cash systems: A cryptographic investigation of mimblewimble. In: EUROCRYPT (2019)
12. Hopwood, D., Bowe, S., Hornby, T., Wilcox, N.: Zcash protocol specification version 2020.1.3. Tech. rep., Electric Coin Company (2020)
13. Jedusor, T.E.: Mimblewimble (2017), `https://scalingbitcoin.org/papers/mimblew-imble.txt`

14. Kappos, G., Yousaf, H., Maller, M., Meiklejohn, S.: An empirical analysis of anonymity in zcash. CoRR **abs/1805.03180** (2018)
15. Khalilov, M.C.K., Levi, A.: A survey on anonymity and privacy in bitcoin-like digital cash systems. IEEE Communications Surveys Tutorials pp. 1–1 (2018)
16. Kumar, A., Fischer, C., Tople, S., Saxena, P.: A traceability analysis of monero's blockchain. In: Computer Security – ESORICS. Springer (2017)
17. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: Characterizing payments among men with no names. In: Proceedings of the 2013 Conference on Internet Measurement Conference. pp. 127–140. IMC '13, ACM, New York, NY, USA (2013)
18. Miller, A., Moeser, M., Lee, K., Narayanan, A.: An empirical analysis of linkability in the monero blockchain. arXiv preprint arXiv:1704.04299 (2017)
19. Morris, L.: Anonymity Analysis of Cryptocurrencies. Ph.D. thesis, Rochester Institute of Techology (2015)
20. Möser, M., Soska, K., Heilman, E., Lee, K., Heffan, H., Srivastava, S., Hogan, K., Hennessey, J., Miller, A., Narayanan, A., et al.: An empirical analysis of traceability in the monero blockchain. Proc. Privacy Enhancing Technologies (3) (2018)
21. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the bitcoin transaction graph. Future Internet **5**(2), 237–250 (2013), copyright - Copyright MDPI AG 2013; Last updated - 2014-07-30
22. Pfitzmann, A., Hansen, M.: 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
23. Poelstra, A.: Mimblewimble (2016)
24. Quesnelle, J.: An Analysis of Anonymity in the Zcash Cryptocurrency. Master's thesis, University of Michigan-Dearborn (2018)
25. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Security and privacy in social networks, pp. 197–223. Springer (2013)
26. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Financial Cryptography and Data Security. pp. 6–24. Springer (2013)
27. Ruffing, T., Moreno-Sanchez, P.: Valueshuffle: Mixing confidential transactions for comprehensive transaction privacy in bitcoin. In: Financial Cryptography and Data Security. pp. 133–154. Springer, Cham (2017)
28. Spagnuolo, M., Maggi, F., Zanero, S.: Bitiodine: Extracting intelligence from the bitcoin network. In: International Conference on Financial Cryptography and Data Security. pp. 457–468. Springer (2014)
29. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002)
30. Tsukada, Y., Mano, K., Sakurada, H., Kawabe, Y.: Anonymity, privacy, onymity, and identity: A modal logic approach. In: 2009 International Conference on Computational Science and Engineering. vol. 3, pp. 42–51 (Aug 2009)
31. Van Saberhagen, N.: Cryptonote v 2. 0 (2013), `https://cryptonote.org/whitepaper.pdf`
32. Wijaya, D.A., Liu, J., Steinfeld, R., Liu, D., Yuen, T.H.: Anonymity reduction attacks to monero. In: Information Security and Cryptology. Springer (2019)
33. Wijaya, D.A., Liu, J.K., Steinfeld, R., Sun, S.F., Huang, X.: Anonymizing bitcoin transaction. In: Information Security Practice and Experience. Springer (2016)
34. Zhang, Z., Li, W., Liu, H., Liu, J.: A refined analysis of zcash anonymity. IEEE Access **8**, 31845–31853 (2020)