The Cryptographic Complexity of Anonymous Coins: A Systematic Exploration

Niluka Amarasinghe, Xavier Boyen, Matthew McKague

Queensland University of Technology, Australia

Abstract

The modern financial world has seen a significant rise in the use of cryptocurrencies in recent years, partly due to the convincing lures of anonymity promised by these schemes. Bitcoin, despite being considered as the most widespread among all, is claimed to have significant lapses in relation to its anonymity. Unfortunately, studies have shown that many cryptocurrency transactions can be traced back to their corresponding participants through the analysis of publicly available data, to which the cryptographic community has responded by proposing new constructions with improved anonymity claims. Nevertheless, the absence of a common metric for evaluating the level of anonymity achieved by these schemes has led to a number of disparate ad hoc anonymity definitions, making comparisons difficult. The multitude of these notions also hints at the surprising complexity of the overall anonymity landscape.

In this study, we introduce such a common framework to evaluate the nature and extent of anonymity in (crypto)currencies and distributed transaction systems, irrespective of their implementation. As such, our work lays the foundation for formalising security models and terminology across a wide range of anonymity notions referenced in the literature, while showing how "anonymity" itself is a surprisingly nuanced concept.

Keywords: Anonymity; Security Models; Cryptocurrencies; Foundations

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 to the common man and woman. Consequently, their emergence as alternatives to traditional fiat currencies is reaching new heights.

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.

Anonymity, from a broad perspective, means that with respect to a given group of entities, it is not possible to uniquely identify one entity from the rest in that group. This concept of

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

anonymity has been widely discussed in the context of anonymous communication and also in anonymous information sharing. Consequently, many theoretical models have been developed to model anonymity, such as k-Anonymity [25] and approaches based on modal logic [26]. Some such studies present formalised terminologies to capture different aspects of anonymity [18], while some propose metrics that could be used to measure a quantitative notion thereof; e.g. as a degree of anonymity [7]. For better or for worse, these available theoretical frameworks have been borrowed for discussing anonymity in cryptocurrencies.

The absence of an acceptable level of anonymity and privacy could hinder the effectiveness of any currency scheme. Many traditional currency schemes are centralised systems where customers depend on another party to preserve the privacy of related information. 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 that 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, or their involvement in boycotted operations, even though the units may have had only uncontroversial uses afterwards. As such, it is paramount to have a tolerable level of anonymity in a currency scheme in order to ensure its *fungibility.*

In relation to the anonymity of Bitcoin, it has been argued that the current Bitcoin framework only provides a level of 'pseudonymity', in place of anonymity, since transactions are linked to payment addresses in a big graph that is visible to all [10, 6]. Detailed analyses of public bitcoin transaction data have shown that it is possible to uncover behaviour patterns of Bitcoin users and trace their identities in real life [14, 3, 22].

As a consequence of this tension between the need for, and the lack of, effective anonymity in cryptocurrencies, a lot of energy has been expended with the primary focus of fulfilling that demand. Some solutions are centered around improving the anonymity of 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). In spite of many such solutions making claims of "anonymity", further studies have shown that a majority of them could still be subject to deanonymisation [13, 15].

As rationalised in [2], despite a large number of studies around the topic of 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 different metrics, with the consequence that it is not feasible to compare and benchmark the anonymity landscape in a reliable manner across various constructions. 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. It is replete with nooks and crannies of special cases and limitations, that could turn into so many vulnerabilities.

1.1 Our Contribution

The present study was initially motivated by the works of [1, 2, 6, 10], 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. Moreover, our definitions follow patterns that make them amenable to being brought to order according to a logical taxonomy.

Our purpose in this work, therefore, is to initiate a comprehensive *formal* study of *fine-grained* notions of anonymity in payment systems. While the multiplicity of notions is truly a by-product of the diversity and complexity of cryptographic cash systems (both existing or envisioned), our framework is general enough to capture familiar instances such as intermediated banking transactions and interpersonal physical payments. It should be noted that we do not intend to address the anonymity of the underlying implementation of currency schemes in this work i.e. consensus or communication mechanisms.

Our main contribution in this context is the formulation of a theoretical framework that can be used to provide a systematic categorisation of terminology related to anonymity of (crypto)currencies and to model anonymity across different instances of such currency schemes.

Before we even start discoursing of anonymity, we create a flexible framework to abstract the generic functionality of nearly arbitrary payment systems, as long as certain basic consistency, security and financial soundness properties that we define, are satisfied. We model notions of *spendability*, *balance* and *indemnification*, among others, considered either in an absolute universal sense for all inputs, or with respect to adversaries granted access to helper oracles.

On this foundation, we then 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 consolidates a massive group of explicit attacker scenarios. 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, and a weaker notion of *unlinkability.* 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, including (or not) the ability to see or set the initial state, to access or choose ancillary public/private keys, to query and/or manipulate the system as it runs and to access or choose other transaction data.

Throughout this rather expansive exercise, we strive to identify similarities between related notions, which allows us in certain instances to "compress" or abstract them according to a common template, cutting size and tedium while boosting descriptive power. Some of the resulting definitions are not distinct; others are mere tweaks in a common template; and yet others will require individual treatment. In order to encapsulate these dispersed scenarios, we present a set of theorems, underpinning the relationships among them.

While a multitude of separate definitions may seem absurdly excessive, we emphasise that these definitions arise naturally from considering the possible interactions between the adversary and the cryptocurrency. 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.

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 provide a preamble to the notation used in this paper. Thereafter, we set forth the preliminaries of our currency scheme, algorithms involved and relevant considerations for correctness and security of the scheme. Next, we present anonymity definitions and theorems which depicts the implications and separations among those definitions, 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.

One of the early studies conducted by Reid et al. [21] on the anonymity of Bitcoin, presents a passive analysis on publicly available transaction data by constructing two topological structures based on the connectivity of users and transactions showing how these data can be analysed in many different ways compromising the anonymity of users. Some have attempted to quantify such data as in [22], where behavioural patterns and transaction flows are studied at the user level. Meiklejohn et al. [12] present a different characterisation of Bitcoin transaction data by clustering user accounts in terms of several heuristics, thereby highlighting the gap between expected vs actual level of anonymity in the Bitcoin network. A similar work done by Spagnuolo et al. [24] proposes a framework (named BitIodine) to extract Bitcoin user information, mainly aiming for forensic purposes. These studies evidently, place more emphasis on the quantitative analysis while we follow a qualitative approach.

On a different note, some have attempted to formalise the anonymity concepts in a theoretical manner. In this regard, a majority of the work conducted in the Bitcoin system evaluates the level of anonymity based on the notion of so called *linkability*, yet with different interpretations. Androulaki et al. [3] conducted an analysis of Bitcoin privacy based on *activity unlinkability* and *profile indistinguishability*. In this work, unlinkability is defined in relation to addresses and transactions (independently) with respective users. This interpretation of unlinkability has been applied in several subsequent studies related to the anonymity of Bitcoin [16, 17, 30]. From these studies, it is apparent that Bitcoin anonymity cannot be defined at the transaction layer since addresses and transactions are linkable by the construction itself. As Bitcoin receives much criticism to that effect, new currency schemes have emerged with more promising anonymity expectations, which has led to the need for more concrete formalisation of anonymity concepts. Zcash is one such scheme which supports two types of transactions, ‘shielded’ and ‘unshielded’. Shielded transactions are encrypted, hence concealing the addresses and values involved, thereby claiming to acquire improved levels of anonymity. However, users have the option to choose the transaction type, and thus they end up creating unshielded transactions at some point, where they work similar to Bitcoin transactions. Several experimental studies have shown that it is prone to *linkability* [9, 20]. Linkability in this context is defined as the ability to link transactions and the corresponding payment addresses, and they claim that shielded transactions eventually end up in transparent addresses [20].

Cryptonote is one of the protocols based on which several currency systems have been constructed with improved anonymity claims. Saberhagen [27], in the original Cryptonote paper, states that a fully anonymous currency scheme should satisfy two properties with respect to anonymity; *unlinkability* and *untraceability*. Unlinkability in this work refers to the property that given two transactions, it is not possible to identify whether both transactions were intended to the same party, whereas untraceability is defined as the inability to identify the corresponding sender among a set of possible senders for a given transaction. Monero, which originated from the Cryptonote protocol was then claimed to provide untraceable transactions and untraceable payments. However, these two properties have fallen to deanonymisation attacks in many subsequent studies through analysis of Monero transaction data [15, 29, 28].

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 [6, 23]. Although it has been claimed [20] that Zcash achieves fungibility through its use of zero-knowledge SNARK proofs, the survey study of [6] makes the countermanding claim that Mimblewimble [19] is the only cryptocurrency scheme to do so. Even so, the original Mimblewimble is insecure, and the fix proposed in [8], by making it preserve a lot more data, reintroduces the coin history removed in the original fungibility claim.

A recent work by Biryukov et al. [4] presents an experimental analysis on deanonymisation of sample transactions in several cryptocurrencies based on network analysis. The outcomes are presented in terms of anonymity degree, which provides an information-theoretic notion of anonymity as a metric external to the scheme being studied. The study however stresses the need for a common mechanism that could be used to measure the effectiveness of various techniques used by different currency schemes in their search of anonymity.

Cachin et al. [5] has proposed a formal model for blockchain systems by modelling the transactions in terms of a graphical structure called Transaction Graphs, focusing on three different blockchain systems; Bitcoin, Ethereum and Hyperledger Fabric. While this work focuses on the semantics of a blockchain, our model deviates from this as our emphasis is on modelling anonymity based on the functionality of a currency scheme.

With this background, many have attempted to evaluate and compare the level of anonymity achieved by different cryptocurrencies through diverse means. Khalilov et al. [10] conducted a comprehensive survey on a wide range of cryptocurrencies. They attempted to group the underlying constructions around three aspects of anonymity; *untraceability*, *hidden values* and *hidden IP addresses*. A similar study was carried out by Conti et al. [6], which discusses the privacy aspects of Bitcoin and other cryptocurrencies as a comparison of advantages and disadvantages in terms of privacy and anonymity. Further, [2] presented a survey of several cryptocurrencies with respect to a set of qualitative anonymity properties such as *fungibility*, *unlinkability*, *untraceability*, *hidden values*, *unlinkability of IP addresses*. Without formally defining those properties, they used them to compare cryptocurrencies over multiple dimensions. In a more recent survey paper, Alsalami et al. [1] presented a systematic grouping of a chosen set of cryptocurrencies in terms of four privacy tiers; *pseudonymity*, *set anonymity*, *full anonymity* and *confidential transactions*, based on two characteristics; ability to break links between transactions and hiding user identities. This categorisation also however, similar to [10], provides a very high level picture of the anonymity levels based on the techniques used by the schemes, which is orthogonal to our work.

Nevertheless, 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 Proposed Model

We start by constructing a model for a cryptocurrency scheme in terms of a set of algorithms which depicts the overall functionality of a generic cryptocurrency scheme.

The currency scheme is defined in terms of a security parameter $\lambda \in \mathbb{Z}^+$ and the initial state of the system, is called the *genesis state*. The scheme consists of a set of payment addresses, each consisting of a private key and a public address or identity. A transaction takes place between multiple senders and recipients, and consists of a private and a public part. A minting operation collects unminted transactions at any given point in time and generates a new state. New currency units are generated as a result of the minting process, as per the underlying implementation of the scheme. The adjudicate operation selects the rightful new state of the system. A system state p is defined by the implementation and will typically record all payment addresses and transactions that are valid in that instance. In Bitcoin, for example, the blockchain is the state. Every valid state descends from a valid checkpoint state, which descends from another checkpoint state or the genesis state. Accordingly, consecutive states of the scheme form a partial ordering with respect to the internal system specifications.

Table 1: Notation

Description	**Notation**
Security parameter	$\lambda : \lambda \in \mathbb{Z}^+$
A system state/Current state	p
A set of states	P
p_0 is an earlier state in time than p_1 or $p_0 = p_1$ i.e. p_0 is in p_1's history	$p_0 \preceq p_1$
p_0 is not in the history of p_1	$p_0 \npreceq p_1$
$p_0 \preceq p_1 \quad \wedge \quad p_0 \neq p_1$	$p_0 \prec p_1$
A payment address	a
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	$S,\ R$
Number of items in a tuple S	$\lvert S \rvert$
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
Concatenation of tuples A and B, Set minus operation of tuples A, B	$A \Vert B,\ A \setminus B$
Empty set, empty tuple	$(\emptyset/\{\}),\ ()$
a_{pk} is an element of tuple R	$a_{pk} \in R$
a_{pk} is not an element of tuple R	$a_{pk} \notin R$
Every element in tuple $R^{'}$ is in tuple R	$R^{'} \subseteq R$
If [*condition*] is false after $<$ *statement* $>$, then return 1	$<$ *statement* $>$ [*condition*]
If $\langle$ *condition* $\rangle$ is false after $<$ *statement* $>$, then return 0	$<$ *statement* $>$ $\langle$*condition*$\rangle$
If $a = \perp$ then return c, else return b	$a?b : c$
If $a = \perp$ then return b, else return a	$a?\_ : b$
Return X if y, otherwise return 1	X^y
Standard operations on Associative Arrays	$\texttt{Operation}_{AA}$
Set of all possible system states	$\mathbb{P}$
Set of all possible addresses (both public and secret parts)	$\mathbb{A}$
Set of all possible transactions (both public and secret parts)	$\mathbb{T}$
Set of all possible transaction values of the form (V_{old}, V_{new}, m)	$\mathbb{V}$
Set of all possible mint data values of the form (R_m, V_M)	$\mathbb{M}$

2.0.1 Notation

We use the notation given in Table 1 throughout the document in order to represent the scheme and its operations mathematically.

2.1 A Generic Cryptocurrency Scheme

We define a generic cryptocurrency scheme as follows:

Definition 2.1. A cryptocurrency scheme Π, is defined in terms of security parameter λ 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$\}$.

2.1.1 Functionality

Table 2: 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`	$(outputs) \leftarrow \texttt{AdditionalFunctionality}(inputs)$

Table 2 summarises the structure of the algorithms of the scheme. The initial setup of the scheme is defined by the `Init` algorithm in terms of a security parameter λ and this process generates the genesis state. Payment address creation process, `CreateAddr` takes and identity and some randomness, and generates a public, private key pair (a_{pk}, a_{sk}) and a transaction, which can be minted to register the addresses. Public and private keys can be validated with respect to a given state, p. A transaction (t_p, t_s) is created with unspent funds from one or more senders (V_{old}) and corresponding funds for recipients (V_{new}), together with transaction related metadata m such as corresponding IP addresses or other system specific data. The validity of a transaction can be defined with respect to its public part as well as both public and private parts taken together. The difference between the total input value and the total output value is considered as transaction fees. Further, transaction related data (input output values and public keys of senders and recipients) can be extracted from a given transaction, if both public and private parts of the transaction are known.

A minting operation takes place on a set of public parts of transactions $\{t_p\}$ and new currency units may be generated through this process, whose value is decided by the implementation specifications, internally. These minted currency units and respective transaction fees, collectively termed as excess value (V_x), are collected by the miners. The preferred state out of a set of candidate states is chosen to be the subsequent state of the system through the `Adjudicate` operation by preserving the precedence of states. `IsValidState` algorithm checks the validity of a given state with respect to a given security parameter. A given state can be designated as a checkpoint state through the `CreateCheckpointState` function based on the particulars of the state, which can be retrieved later through the `RetrieveCheckpointState` operation. The genesis state is considered as the first checkpoint and the algorithm `IsGenesisState` can be used to identify the genesis state corresponding to a given security parameter.

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 network in this work. However, 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.

3 Correctness

Table 3: List of experiments for correctness.

Correctness property	Experiment
Correctness of state initialisation	$\mathtt{Exp}_{\pi}^{init}$
Correctness of address creation	$\mathtt{Exp}_{\pi}^{create\text{-}addr}$
Correctness of transaction creation	$\mathtt{Exp}_{\pi}^{create\text{-}txn}$
Correctness of minting	$\mathtt{Exp}_{\pi}^{mint}$
Correctness of extracting transaction data	$\mathtt{Exp}_{\pi}^{extract\text{-}txn\text{-}data}$
Correctness of adjudicate operation	$\mathtt{Exp}_{\pi}^{adjudicate}$
Correctness of checkpoint creation	$\mathtt{Exp}_{\pi}^{create\text{-}checkpoint}$
Correctness of the verification of genesis state	$\mathtt{Exp}_{\pi}^{genesis\text{-}state}$
Monotonicity of checkpoint states	$\mathtt{Exp}_{\pi}^{checkpoint\text{-}monotonicity}$
Monotonicity of states with respect to adjudicate operation	$\mathtt{Exp}_{\pi}^{adj\text{-}monotonicity}$
Correctness of the checkpoint retrieval	$\mathtt{Exp}_{\pi}^{retrieve\text{-}checkpoint}$

In this section we establish the correctness of our model in terms of the functionality. We consider the correctness of individual functions as well as their collective functionality, in terms of a set of experiments as listed in Table 3.

3.1 Generating input data

We define several functions to generate input data for the correctness experiments in terms of λ and a tuple of bit strings $\rho \in (\{0,1\}^*)^*$. Bit strings are mapped to required datasets through separate and arbitrary surjective functions with following mappings:
$\mathtt{Deserialise}_{\mathtt{W}} : \{0,1\}^* \times \mathbb{N} \rightarrow \{\bot\} \cup \mathbb{W}$ where $\mathbb{W} \in \{\mathbb{P}, \mathbb{A}, \mathbb{T}, \mathbb{V}, \mathbb{M}\}$

```
GenerateState($\lambda, \rho_k$)
  1. $p \leftarrow \mathtt{Deserialise}_{\mathtt{P}}(\lambda, \rho_k)$
  2. if $\neg(\mathtt{IsValidState}_{\pi}(p, \lambda))$
  3.     return $\bot$
  4. return $p$
```

```
GenerateTxns(p, $\lambda, \rho_k$)
  1. $\{(t_p, t_s)\} \leftarrow \mathtt{Deserialise}_{\mathtt{T}}(\lambda, \rho_k)$
  2. for all $(t_p, t_s) \in \{(t_p, t_s)\}$ do
  3.     if $\neg(\mathtt{IsValidSecTxn}_{\pi}(t_p, t_s, p))$
  4.         return $\bot$
  5. return $\{(t_p, t_s)\}$
```

```
GenerateTxnValues(R, $\bar{S}$, $\lambda, \rho_k$)
  1. $(V_{old}, V_{new}, m) \leftarrow \mathtt{Deserialise}_{\mathtt{V}}(\lambda, \rho_k)$
  2. if $(|V_{old}| \neq |\bar{S}|) \vee (|V_{new}| \neq |R|)$
  3.     return $\bot$
  4. return $(V_{old}, V_{new}, m)$
```

```
GenerateAddr(p, $\lambda, \rho_k$)
  1. $((a_{pk}, a_{sk})) \leftarrow \mathtt{Deserialise}_{\mathtt{A}}(\lambda, \rho_{k+1}$
  2. for all $(a_{pk}, a_{sk}) \in ((a_{pk}, a_{sk}))$ do
  3.     if $\neg(\mathtt{IsValidPubAddr}_{\pi}(a_{pk}, p))$
  4.         return $\bot$
  5.     if $\neg(\mathtt{IsValidSecAddr}_{\pi}(a_{pk}, a_{sk}, p))$
  6.         return $\bot$
  7. return $((a_{pk}, a_{sk}))$
```

```
GenerateMintAddr(p, $\lambda, \rho_k$)
  1. $R_m \leftarrow \mathtt{Deserialise}_{\mathtt{M}}(\lambda, \rho_{k+1})$
  2. for all $a_{pk} \in R_m$ do
  3.     if $\neg(\mathtt{IsValidPubAddr}_{\pi}(a_{pk}, p))$
  4.         return $\bot$
  5. return $R_m$
```

Figure 1: Functions used to generate input data from bitstring ρ

Here we list the functions that generate input data for the correctness experiments from a given ρ, which can be thought of as random coins (Figure 1). ρ is represented as $\rho = (\rho_1, \rho_2, ...)$

within experiments where ρ_1, ρ_2 etc. are bit strings of arbitrary finite length of which the combined length is equal to the length of ρ. i.e. $\sum_i |\rho_i| = |\rho|$. Each ρ_i is used to generate required inputs for the experiments through respective `Deserialise` functions.

```
SelectSubsetofStates(P_set, k, λ, ρ)
 1. $n \leftarrow (\{1,..,|P_{set}|\}; \rho_{k+1})$
 2. for  $i \in \{1,..,n\}$  do
 3.    $p_i \leftarrow (P_{set}; \rho_{k+1+i})$
 4.    $P_{set} \leftarrow P_{set} \setminus \{p_i\}$
 5. return  $(P_{set}, k+1+i)$

EvolveState(p, k, λ, ρ)
 1. $W \leftarrow$ GenerateTxns$(p, \lambda, \rho_{k+1})$
 2. if $W = \perp$, return $\perp$
 3. $\{(t_p, t_s)\} \leftarrow W$
 4. if $\neg$(IsMintable$_\pi(\{t_p\}, p)$), return $\perp$
 5. $Y \leftarrow$ GenerateMintData$(p, \lambda, \rho_{k+2})$
 6. if $Y = \perp$, return $\perp$
 7. $R_m \leftarrow Y$
 8. $(p_1, V_x) \leftarrow$ Mint$_\pi(\{t_p\}, R_m, p; \rho_{k+3})$
 9. return  $(p_1, k+3)$

GenerateSetofStates(k, λ, ρ)
 1. $P_{set} \leftarrow \{\}$
 2. $m \leftarrow |\rho| - k$
 3. if $m < 1$, return $\perp$
 4. $n \leftarrow (\{1,..,m-1\}; \rho_{k+1})$
 5. $i \leftarrow k+2$
 6. for  $j \in \{1,..,n\}$  do
 7.    $p_j \leftarrow$ GenerateState$(\lambda, \rho_i)$
 8.    if IsValidState$_\pi(p_j, \lambda)$, $P_{set} \leftarrow P_{set} \cup \{p_j\}$
 9.    $i \leftarrow i+1$
10. return  $(P_{set}, i)$
```

Figure 2: Helper functions for correctness

```
$\mathbf{Exp}_{\Pi}^{checkpoint\text{-}monotonicity}(\lambda, \rho)$
 1. $p_0 \leftarrow$ GenerateState$(\lambda, \rho_1)$ $[p_0 \neq \perp]$
 2. $p_0^c \leftarrow$ CreateCheckpoint$_\pi(p_0, \lambda)[p_0^c \neq \perp]$
 3. $p_1 \leftarrow p_0^c$
 4. $i, n_c \leftarrow 1$
 5. while $(i < |\rho|)$ do
 6.   $X \leftarrow$ EvolveState$(p_1, i, \lambda, \rho)$ $[X \neq \perp]$
 7.   $(p_2, k) \leftarrow X$
 8.   $b \xleftarrow{\$} \{0,1\}$
 9.   if  $b = 1$
10.     $H \leftarrow$ CreateCheckpoint$_\pi(p_2, \lambda; \rho_{k+1})$
11.     if  $H \neq \perp$, $p^c \leftarrow H$; $n_c \leftarrow n_c + 1$
12.     $i \leftarrow k+1$
13.   else
14.     $i \leftarrow k$
15.   $p_1 \leftarrow p_2$
16. for  $j \in \{1,..,n_c - 1\}$  do
17.   $p_1^c \leftarrow$ RetrieveCheckpointState$(p^c)$
18.   if $p_1^c = \perp$, return 0
19.   $p^c \leftarrow p_1^c$
20. return  $(p^c \stackrel{?}{=} p_0^c)$

$\mathbf{Exp}_{\Pi}^{adj\text{-}monotonicity}(\lambda, \rho)$
 1. $p_0 \leftarrow$ GenerateState$(\lambda, \rho_1)$  $[p_0 \neq \perp]$
 2. $X \leftarrow$ GenerateSetofStates$(2, \lambda, \rho)$ $[X \neq \perp]$
 3. $(P_{set}, k) \leftarrow X$
 4. $(P_{test}, j) \leftarrow$ SelectSubsetofStates$(P_{set}, k, \lambda, \rho)$
 5. $p_1 \leftarrow$ Adjudicate$_\pi(P_{set}, p_0)$
 6. $X \leftarrow$ EvolveState$(p_1, j, \lambda, \rho)$  $[X \neq \perp]$
 7. $(p_2, j) \leftarrow X$
 8. $p_3 \leftarrow$ Adjudicate$_\pi(P_{test} \cup \{p_2\}, p_1)$
 9. return  $P_3 \stackrel{?}{=} P_2$

$\mathbf{Exp}_{\Pi}^{retireve\text{-}checkpoint}(\lambda, \rho)$
 1. $p_0 \leftarrow$ GenerateState$(\lambda, \rho_1)$  $[\ p_0 \neq \perp\ ]$
 2. $p_1 \leftarrow p_0$
 3. while RetrieveCheckpointState$_\pi(p_1) \neq p_1$ do
 4.   $X \leftarrow$ RetrieveCheckpointState$_\pi(p_1)$
 5.   if  $X = \perp$,  return 0
 6.   $p' \leftarrow X$
 7.   if  $\neg$(IsValidState$_\pi(p', \lambda)$),  return 0
 8.   if  $p' \not\preceq p_1$,  return 0
 9.   $p_1 \leftarrow p'$
10. return  IsGenesisState$_\pi(p_1, \lambda)$
```

Figure 3: Correctness experiments 1

Further, ρ_i strings are also used to introduce randomness to the operations performed within experiments with the notation, `Function`(*parameters* ; ρ_i). In the case where the length of ρ is not sufficient to produce required number of ρ_i bit strings, corresponding experiment is terminated by returning 1 (i.e. experiment terminates with success). In addition, we also introduce several helper functions that help improve the readability of the correctness experiments

(Figure 2). Figures 3 and 4 list all experiments that establish the correctness of the proposed scheme.

$\mathbf{Exp}_{\Pi}^{init}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{Init}_\pi(1^\lambda; \rho_1)$
2. $b \leftarrow \texttt{IsGenesisState}_\pi(p_0, \lambda)$
3. $b' \leftarrow \texttt{IsValidState}_\pi(p_0, \lambda)$
4. return $(b \wedge b')$
```

$\mathbf{Exp}_{\Pi}^{create\text{-}addr}(\lambda, d, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}_\pi(\lambda, \rho_1) \quad [\, p_0 \neq \perp \,]$
2. $(a_{pk}, a_{sk}, t_p, t_s) \leftarrow \texttt{CreateAddr}_\pi(p_0, d; \rho_2)$
3. $b \leftarrow (\texttt{IsValidPubAddr}_\pi(a_{pk}, p_0)$
4. $b' \leftarrow \texttt{IsValidSecAddr}_\pi(a_{pk}, a_{sk}, p_0))$
5. $b'' \leftarrow (\texttt{IsMintable}_\pi(\{t_p\}, p_0)$
6. return $(b \wedge b' \wedge b'')$
```

$\mathbf{Exp}_{\Pi}^{mint}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}(\lambda, \rho_1) \quad [p_0 \neq \perp]$
2. $X \leftarrow \texttt{GenerateTxns}(p_0, \lambda, \rho_2) \quad [X \neq \perp]$
3. $\{(t_p, t_s)\} \leftarrow X$
4. $b \leftarrow \texttt{IsMintable}_\pi(\{t_p\}, p_0)$
5. $Y \leftarrow \texttt{GenerateMintData}(p_0, \lambda, \rho_3)\ [Y \neq \perp]$
6. $(V_m, R_m) \leftarrow Y$
7. $(p_1, V_x) \leftarrow \texttt{Mint}_\pi(\{t_p\}, R_m, p_0; \rho_4)$
8. $b' \leftarrow \texttt{IsValidState}(p_1, \lambda) \ \wedge \ (p_0 \prec p_1)$
9. return $(\, b \stackrel{?}{=} b' \,)$
```

$\mathbf{Exp}_{\Pi}^{adjudicate}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}(\lambda, \rho_1) \quad [\, p_0 \neq \perp \,]$
2. $X \leftarrow \texttt{GenerateSetofStates}(1, \lambda, \rho)\ [X \neq \perp]$
3. $(P_{set}, k) \leftarrow X$
4. $p' \leftarrow \texttt{Adjudicate}_\pi(P_{set}, p_0)$
5. if $p' = p_0$
6.    for all $p_i \in P_{set}$ do
7.       if $\texttt{IsValidState}_\pi(p_i, \lambda) \wedge (p_0 \prec p_i)$
8.          return 0
9. else
10.    if $\neg(\texttt{IsValidState}_\pi(p', \lambda)) \vee (p' \notin P_{set})$
11.       return 0
12.    if $p_0 \not\prec p'$, return 0
13.    for all $p_i \in P_{set}$ do
14.       if $(p' \prec p_i)$, return 0
15. return 1
```

$\mathbf{Exp}_{\Pi}^{create\text{-}checkpoint}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}(\lambda, \rho_1) \quad [p_0 \neq \perp]$
2. $p_0^c \leftarrow \texttt{CreateCheckpoint}_\pi(p_0, \rho_2) \quad [p_0^c \neq \perp]$
3. $X \leftarrow \texttt{EvolveState}(p_1, 2, \lambda, \rho) \quad [\, X \neq \perp \,]$
4. $(p_1, k) \leftarrow X$
5. return $(p_0^c \preceq \texttt{RetrieveCheckpointState}_\pi(p_1))$
```

$\mathbf{Exp}_{\Pi}^{create\text{-}txn}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}(\lambda, \rho_1) \quad [\, p_0 \neq \perp \,]$
2. $\bar{R} \leftarrow \texttt{GenerateAddr}(p_0, \lambda, \rho_2) \quad [\, \bar{R} \neq \perp \,]$
3. $\bar{S} \leftarrow \texttt{GenerateAddr}(p_0, \lambda, \rho_3) \quad [\, \bar{S} \neq \perp \,]$
4. $Z \leftarrow \texttt{GenerateTxnValues}(\lambda, \rho_4) \quad [\, Z \neq \perp \,]$
5. $(V_{old}, V_{new}, m) \leftarrow Z$
6. for $i \in \{0, .., |\bar{S}| - 1\}$ do
7.    $(a_{pk_i}, a_{sk_i}) \leftarrow \bar{S}[i]$
8.    if $\texttt{GetBalance}_\pi(a_{pk_i}, a_{sk_i}, p_0) < V_{old}[i]$
9.       return 1
10. $(t_p, t_s) \leftarrow \texttt{CreateTxn}_\pi(R, V_{new}, \bar{S}, V_{old}, m,\ p_0; \rho_5)$
11. $b \leftarrow \texttt{IsValidPubTxn}(t_p, p_0)$
12. $b' \leftarrow \texttt{IsValidSecTxn}(t_p, t_s, p_0)$
13. return $(b \wedge b')$
```

$\mathbf{Exp}_{\Pi}^{extract\text{-}txn\text{-}data}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}(\lambda, \rho_1) \quad [\, p_0 \neq \perp \,]$
2. $\bar{R} \leftarrow \texttt{GenerateAddr}(p_0, \lambda, \rho_2) \quad [\, \bar{R} \neq \perp \,]$
3. $\bar{S} \leftarrow \texttt{GenerateAddr}(p_0, \lambda, \rho_3) \quad [\, \bar{S} \neq \perp \,]$
4. $Z \leftarrow \texttt{GenerateTxnValues}(\lambda, \rho_4) \quad [\, Z \neq \perp \,]$
5. $(V_{old}, V_{new}, m) \leftarrow Z$
6. for $i \in \{0, .., |\bar{S}| - 1\}$ do
7.    $(a_{pk_i}, a_{sk_i}) \leftarrow \bar{S}[i]$
8.    if $\texttt{GetBalance}_\pi(a_{pk_i}, a_{sk_i}, p_0) < V_{old}[i]$
9.       return 1
10. $(t_p, t_s) \leftarrow \texttt{CreateTxn}_\pi(R, V_{new}, \bar{S}, V_{old}, m,\ p_0; \rho_5)$
11. $W \leftarrow \texttt{GenerateMintData}(p_0, \lambda, \rho_6) \ [W \neq \perp]$
12. $(V_m, R_m) \leftarrow W$
13. if $\neg(\texttt{IsMintable}_\pi(\{t_p\}, p_0))$, return 0
14. $(p_1, V_x) \leftarrow \texttt{Mint}_\pi(\{t_p\}, R_m, p_0; \rho_7)$
15. $S' \leftarrow \texttt{ExtractSenderPubAddr}_\pi(t_p, t_s, p_1)$
16. $R' \leftarrow \texttt{ExtractSenderPubAddr}_\pi(t_p, t_s, p_1)$
17. $V'_{old} \leftarrow \texttt{ExtractInputVal}_\pi(t_p, t_s, p_1)$
18. $V'_{new} \leftarrow \texttt{ExtractOutputVal}_\pi(t_p, t_s, p_1)$
19. $b \leftarrow (R' \stackrel{?}{=} R) \wedge (S' \stackrel{?}{=} S)$
20. $b' \leftarrow (V'_{old} \stackrel{?}{=} V_{old}) \wedge (V'_{new} \stackrel{?}{=} V_{new})$
21. return $(b \wedge b')$
```

$\mathbf{Exp}_{\Pi}^{genesis\text{-}state}(\lambda, \rho)$

```
1. $p_0 \leftarrow \texttt{GenerateState}(\lambda, \rho_1) \quad [\, p_0 \neq \perp \,]$
2. if $\texttt{RetrieveCheckpointState}_\pi(p_0) = p_0$
3.    return $(\texttt{IsGenesisState}_\pi(p_0, \lambda))$
4. $X \leftarrow \texttt{EvolveState}(p_1, 1, \lambda, \rho) \quad [\, X \neq \perp \,]$
5. $(p_1, k) \leftarrow X$
6. $b \leftarrow \texttt{IsValidState}_\pi(p_1, \lambda)$
7. $b' \leftarrow \neg\, \texttt{IsGenesisState}_\pi(p_1, \lambda)$
8. return $(b \wedge b')$
```

Figure 4: Correctness Experiments 2

Accordingly, the correctness of the proposed scheme is defined as follows:

Definition 3.1. (Correctness of the Cryptocurrency Scheme) A currency scheme Π is correct if, for all security parameters $\lambda \in \mathbb{Z}^+$, for all sufficiently long bit strings $\rho \in (\{0,1\}^*)^*$ and for all $X \in \{$*init, create-addr, create-txn, extract-txn-data, mint, adjudicate, adj-monotinicty, create-checkpoint, ret- rieve-checkpoint, genesis-state, checkpoint-monotinicity*$\}$, $\mathtt{Exp}_\pi^X(\lambda, \rho)$ returns 1.

4 Security

In this section, we establish the security requirements for the proposed framework 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 is a very nice theoretical methodology which is best suited for small primitives whose ideal functionalities may still have a clean description, which is certainly not the case in the context of cryptocurrencies.

We define a comprehensive adversarial model to accommodate a wide range of capabilities on the part of the adversary. Then we define security requirements for the functionality of the proposed currency scheme. Anonymity aspects, although related to security, are discussed in a separate section as it is the main focus of this paper.

4.1 Adversarial Model

We consider several parameters to define the adversarial model in depth. These parameters represent various levels of adversary capabilities, which include the level of knowledge of public/secret keys, transaction values, metadata and transactions (ψ), the ability to view and manipulate the state (δ), the nature of state initialisation in the experimental setup (α), and whether failed minting is allowed during the execution of the game (β). These symbols are used with different subscripts denoting which entity is being referred to, as listed in Table 4. These values scale from the least capability (0) to the strongest on the part of the adversary in game based experiments.

The adversary's level of knowledge ψ is modelled in the following manner. When any knowledge parameter has a value of 0, corresponding entity of that parameter is considered to be hidden from the adversary. We assume that the adversary has oracle access through opaque handles to those hidden entities using which desired activities can be initiated through relevant oracles. A value of 1 in these parameters represents the situation where the adversary learns the corresponding entity at the end of the game, just before he makes his choice. Beyond that point, the adversary is not allowed to create or mint any transactions involving those entities. With the parameter ψ_t on the other hand, the public part of the transaction t_p is revealed to the adversary when $\psi_t = 1$. When $\psi_t = 2$, the secret part t_s is revealed and with $\psi_t = 3$, the randomness of the actual coins is revealed. Further, when $\psi_t = 4$, the adversary gets to choose the randomness for the transaction and finally the adversary gets to create the transaction when $\psi_t = 5$. For other ψ parameters, with a value of 2, relevant information is known to the adversary throughout the game in real time via appropriate oracle access. However, for all those cases, the adversary does not have control over the entities. Conversely, for any value higher than 2, the adversary has some form of control over the relevant entity as explained in Table 4. With this parameterisation, we can capture a wide range of adversaries ranging from passive (with all parameters equal to zero) to static (with δ, $\beta \leq 1$) and adaptive adversaries (with parameter values greater than 1).

Table 4: Parameters of the adversarial model

Parameter 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 (HIDH)	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 (PUBH)	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 (PUBA)	-
3	Adversary chooses the identity, the oracle creates addresses	Adversary chooses the randomness, the oracle creates addresses	Adversary chooses the values	Adversary chooses metadata	Randomness of the coins revealed, oracle creates transaction	-	Hidden randomness adversarial Init (HIDH)	-
4	Adversary generates the address	Adversary generates the address	-	-	Adversary chooses the randomness	-	-	-
5	-	-	-	-	Adversary creates the transaction	-	-	

Helper functions We define a group of oracle functions to provide the adversary with access to honest functionality during the execution of the game (Figure 5). These include $\mathcal{O}_{addr}$ for creating addresses, $\mathcal{O}_{hidaddr}$ for creating hidden addresses, $\mathcal{O}_{txn}$ for creating transactions and $\mathcal{O}_{mint}$ for minting. Another oracle is defined to generate hidden metadata ($\mathcal{O}_{hidMdata}$). The history of the activities of the oracles are maintained globally within the games; i.e. $A_{\mathcal{O}}, T_{\mathcal{O}}$ as associative arrays and $M_{\mathcal{O}}$ as a set to store all addresses, transactions and minting history respectively. In addition, $A^*_{\mathcal{O}}, T^*_{\mathcal{O}}$, and $D^*_{\mathcal{O}}$ are maintained as sets to store hidden addresses, transactions and metadata. In order to cater for the addresses created with different adversarial inputs, the oracle keeps track of different groups of addresses in $A_{\mathcal{O}_{jk}}$ with binary values j and k, and a value of 0 representing adversarial identity and adversarial randomness, respectively. $\mathcal{O}_{mint}$ sets the flag $f_{\mathcal{O}} = 1$ globally, if a minting operation fails, in which case the adversary loses the game, unless β=1. The adversary has access to all available oracles, unless specifically mentioned with a specific subscript in the games. Table 5 summarises the variables used by the oracles.

Further, the current state of the system is denoted by $p_{\mathcal{O}}$ for these games. It is assumed that $p_{\mathcal{O}}$ is updated as the state evolves within the game (e.g. through oracle calls with side effects, which is what the subscript $\mathcal{O}$ tries to convey), except where a new state is generated through a mint operation, in which case the new state is denoted with a different subscript. e.g. p_1.

Table 5: A summary of oracle variables

Variable	**Description**
$A_{\mathcal{O}}$	All addresses created by the oracle i.e. all (a_{pk}, a_{sk})
$A^*_{\mathcal{O}}$	All hidden addresses created by the oracle i.e. all hidden a_{pk}
$A_{\mathcal{O}_{11}}$	All addresses created by the oracle with randomly chosen d and ρ
$A_{\mathcal{O}_{10}}$	All addresses created by the oracle with adversarial randomness (ρ)
$A_{\mathcal{O}_{01}}$	All addresses created by the oracle with adversarial identity (d)
$A_{\mathcal{O}_{00}}$	All addresses created by the oracle with adversarial identity (d) and randomness (ρ)
$T_{\mathcal{O}}$	All transactions created by the oracle
$T^*_{\mathcal{O}}$	All hidden transactions created by the oracle
$T'_{\mathcal{O}}$	Randomness of the coins involved in transactions created by the oracle
$D^*_{\mathcal{O}}$	All hidden metadata generated by the oracle
$M_{\mathcal{O}}$	Minting details of all mint operations performed by the oracle
$p_{\mathcal{O}}$	Current state

```
$\underline{\mathcal{O}_{mint}(\{t_p\}, R_m)}$
 1. $X \leftarrow$ Mint$_\pi(\{t_p\}, R_m, p_{\mathcal{O}})$
 2. if $X = \perp,\ f_{\mathcal{O}} \leftarrow 1$
 3. else
 4.    $(p_1, V_x) \leftarrow X$
 5.    $M_{\mathcal{O}} \leftarrow M_{\mathcal{O}} \cup \{(p_1, \{t_p\}, V_x, R_m)\}$
 6.    $p_{\mathcal{O}} \leftarrow p_1$
 7. return $p_{\mathcal{O}}$

$\underline{\mathcal{O}_{txn}(R, V_{new}, S, V_{old}, m, \rho')}$
 1. $k \leftarrow (\rho' = \emptyset);\ \rho \leftarrow [k\ ?\ \$ : \rho']$
 2. $R \leftarrow$ LookupPubAddr$(R, A^*_{\mathcal{O}})$
 3. $\bar{S} \leftarrow$ LookupSecAddr$(S, A^*_{\mathcal{O}}, A_{\mathcal{O}})$
 4. if $\psi_v \in \{0,1,2\}$ then
 5.  $(V_{old}, V_{new}) \leftarrow$ GenerateTxnVals$(S, R, A^*_{\mathcal{O}}, A_{\mathcal{O}})$
 6. if $\psi_m \in \{0,1,2\}$ then
 7.  $m \leftarrow$ GenerateMetadata$(\lambda)$
 8. $(t_p, t_s) \leftarrow$ CreateTxn$_\pi(R, V_{new}, \bar{S}, V_{old}, m, p_{\mathcal{O}}; \rho)$
 9. $T^*_{\mathcal{O}} \leftarrow T^*_{\mathcal{O}} \| (t_p)$
10. $T_{\mathcal{O}} \leftarrow$ AddKeyVal$_{AA}(t_p, t_s, T_{\mathcal{O}})$
11. $T'_{\mathcal{O}} \leftarrow$ AddKeyVal$_{AA}(t_p, \rho, T'_{\mathcal{O}})$
12. return $t_p$

$\underline{\mathcal{O}_{hidaddr}()}$
 1. $d \xleftarrow{\$} \{0,1\}^\lambda;\ \rho \xleftarrow{\$} \{0,1\}^*$
 2. $(a_{pk}, a_{sk}, t_p, t_s) \leftarrow$ CreateAddr$_\pi(p_{\mathcal{O}}, d; \rho)$
 3. $A_{\mathcal{O}} \leftarrow$ AddKeyVal$_{AA}(a_{pk}, a_{sk}, A_{\mathcal{O}})$
 4. $T_{\mathcal{O}} \leftarrow$ AddKeyVal$_{AA}(t_p, t_s, T_{\mathcal{O}})$
 5. $A^*_{\mathcal{O}} \leftarrow A^*_{\mathcal{O}} \| (a_{pk})$
 6. return $(|A^*_{\mathcal{O}}|, t_p)$

$\underline{\mathcal{O}_{addr}(d', \rho')}$
 1. $j \leftarrow (d' = \emptyset);\ k \leftarrow (\rho' = \emptyset)$
 2. $d \leftarrow [j\ ?\ \$ : d'];\ \rho \leftarrow [k\ ?\ \$ : \rho']$
 3. $(a_{pk}, a_{sk}, t_p, t_s) \leftarrow$ CreateAddr$_\pi(p_{\mathcal{O}}, d; \rho)$
 4. $A_{\mathcal{O}} \leftarrow$ AddKeyVal$_{AA}(a_{pk}, a_{sk}, A_{\mathcal{O}})$
 5. $A_{\mathcal{O}_{jk}} \leftarrow$ AddKeyVal$_{AA}(a_{pk}, a_{sk}, A_{\mathcal{O}_{jk}})$
 6. $T_{\mathcal{O}} \leftarrow$ AddKeyVal$_{AA}(t_p, t_s, T_{\mathcal{O}})$
 7. if $(\psi_{s_{sk}} \in \{2,3\}) \vee (\psi_{r_{sk}} \in \{2,3\})$
 8.    return $(a_{sk}, t_p)$
 9. else return $(a_{pk}, t_p)$

$\underline{\mathcal{O}_{HidMdata}()}$
 1. $m \xleftarrow{\$} poly(\lambda);\ D^*_{\mathcal{O}} \leftarrow D^*_{\mathcal{O}} \| m$
 2. return $|D^*_{\mathcal{O}}|$
```

Figure 5: Oracle functions

Additionally, we also define a set of helper functions to be used in the security games as given in Figure 6 to improve the clarity of the games. The `SetupState` function performs the state initialisation based on γ, whereas the `RunAdversary` function executes an instance of the adversary $\mathcal{A}$ denoted by different subscripts based on δ. `LookupPubAddr` and `LookupSecAddr` functions are used to obtain public keys and private keys from hidden addresses. In addition, `LookupPubTxn` outputs the t_p corresponding to a hidden transaction when $\psi_t = 0$. `GenerateTxnVals` function is used when $\psi_v \in \{0, 1\}$, to generate required input and output transaction values, based on the the maximum transaction value given by the adversary. Further, `LookupMdata` function is used to reveal hidden metadata when $\psi_m \in \{0, 1\}$.

4.2 Security Properties

First, we define a set of security properties to ensure the functional security of the proposed scheme. These are defined by means of game-based experiments around several attributes; *Unforgeability*, *Transaction binding property*, *Spendability*, *Balance property*, *Descendency* and

```
$\texttt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$
1. if $(\alpha = 3)$
2.   $(p, s) \leftarrow \mathcal{A}_1(\lambda)$
3.   return $(p, \emptyset, s)$
4. else if $(\alpha = 2)$
5.   $(r, s) \leftarrow \mathcal{A}_1'(\lambda);\ p \leftarrow \texttt{Init}_\pi(\lambda; r)$
6.   return $(p, r, s)$
7. $r \leftarrow \$;\ p \leftarrow \texttt{Init}_\pi(\lambda; r)$
8. if $(\alpha = 1)$, return $(p, r, \emptyset)$
9. else return $(p, \emptyset, \emptyset)$
```

```
$\texttt{LookupPubAddr}(H, A^*_\mathcal{O})$
1. $S \leftarrow ()$
2. for all $x \in H$ do
3.   if $x \in \mathbb{Z}^+$ then
4.     $x \leftarrow [A^*_\mathcal{O}[x]?_- : x]$
5.   $S \leftarrow S \| x$
6. return $S$
```

```
$\texttt{LookupSecAddr}(H, A^*_\mathcal{O}, A_\mathcal{O})$
1. $\bar{S} \leftarrow ()$
2. $S \leftarrow \texttt{LookupPubAddr}(H, A^*_\mathcal{O})$
3. for all $a_{pk} \in S$ do
4.   $a_{sk} \leftarrow [AA.\texttt{Lookup}(a_{pk}, A_\mathcal{O})?_- : a_{pk}]$
5.   $\bar{S} \leftarrow \bar{S} \| a_{sk}$
6. return $\bar{S}$
```

```
$\texttt{GenerateMetadata}(\lambda)$
1. $m \xleftarrow{\$} \{0,1\}^{|\lambda|}$
2. return $m$
```

```
$\texttt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_i, p_0, \boldsymbol{inputVal}, r, s, \delta)$
1. if $\delta = 0$
2.   $(\emptyset, returnVal, s) \leftarrow \mathcal{A}_i(\emptyset, inputVal, r, s)$
3.   return $(p_\mathcal{O}, returnVal, s)$
4. else
5.   $(p_1, returnVal, s) \leftarrow \mathcal{A}_i(p_0, inputVal, r, s)$
6. if $\texttt{Adjudicate}_\pi(\{p_1\}, p_0) \neq p_1$
7.   return $(\perp, \perp, \perp)$
8. return $(p_1, returnVal, s)$
```

```
$\texttt{LookupPubTxn}(t, T^*_\mathcal{O})$
1. $t_p \leftarrow [T^*_\mathcal{O}[t]?_- : t]$
2. return $t_p$
```

```
$\texttt{GenerateTxnVals}(V_{\boldsymbol{max1}}, V_{\boldsymbol{max2}}, S, R)$
1. $V_{old}, V_{new}, X, W \leftarrow ()$
2. $v_0, w_0, \ell_1, \ell_2 \leftarrow 0;\ j, m \leftarrow 1$
3. $n_s \leftarrow |S|;\ n_r \leftarrow |R|$
4. for $i = \{1, .., n_s - 1\}$ do
5.   $x_i \xleftarrow{\$} \{0, .., V_{max1}[0]\};\ X \leftarrow X \| \{x_i\}$
6. while $(X \neq ())$ do
7.   $x \leftarrow \texttt{Min}(X);\ V_{old}[j-1] \leftarrow x - \ell_1$
8.   $\ell_1 \leftarrow x;\ j \leftarrow j + 1$
9.   $X \leftarrow X \setminus x$
10. for $k = \{1, .., n_r\}$ do
11.   $w_k \xleftarrow{\$} \{0, .., V_{max2}[0]\};\ W \leftarrow W \| \{w_k\}$
12. while $(W \neq ())$ do
13.   $w \leftarrow \texttt{Min}(W);\ V_{new}[m-1] \leftarrow w - \ell_2$
14.   $\ell_2 \leftarrow w;\ m \leftarrow m + 1$
15.   $W \leftarrow W \setminus w$
16. return $(V_{old}, V_{new})$
```

Figure 6: Helper functions

Anonymity. Each property is demonstrated with respect to attacker's goals and we construct appropriate games to model adversarial behaviour explained earlier.

4.2.1 Unforgeability

Unforgeability property ensures that it is not possible to spend the funds associated with a payment address without the knowledge of the secret key corresponding to that payment address. We define a security game to model this property as listed in Figure 7.

Game : In this game, the initial state is setup according to the input parameters and the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs a transaction (t_p, t_s) and the current state $p_\mathcal{O}$ based on the capabilities defined by the parameters δ and α. The challenger verifies whether the given state is valid. Subsequently, the challenger extracts the public addresses of the senders from the given transaction and performs a check to see if those addresses were created by the oracle (i.e. to ensure that the adversary does not have the knowledge of any of the secret keys). Further, the challenger also checks whether the transaction was created by the oracle and also whether the transaction is valid. This experiment is listed in Figure 7.

Winning Condition : Adversary wins this game, if he is able to produce a valid spending transaction (which is not a transaction created by the Oracle) with at least one sender address in S which was created by the oracle, for which he does not know the corresponding secret key a_{sk}.

```
$\mathtt{Exp}^{\textit{unforgeability}}_{\Pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda)$
 1. $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\mathtt{Init}()$ ;   $M_{\mathcal{O}} \leftarrow \{\}$; $f_{\mathcal{O}} \leftarrow 0$
 2. $(p_{\mathcal{O}}, r, s) \leftarrow \mathtt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
 3. $(p_{\mathcal{O}}, (t_p, t_s), s) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
 4. $S \leftarrow \mathtt{ExtractSenderPubAddr}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
 5. return  $\mathtt{IsValidPubTxn}_{\pi}(t_p, p_{\mathcal{O}}) \wedge (AA.\mathtt{Lookup}(t_p, T_{\mathcal{O}}) \overset{?}{=} \perp) \wedge (S \cap AA.\mathtt{Keys}(A_{\mathcal{O}}) \neq \emptyset)$
```

Figure 7: Experiment for Unforgeability.

4.2.2 Transaction binding property

This property establishes that the secret part of a transaction t_s cannot be tampered with and ensures that t_s binds with a unique t_p. i.e. A given t_s cannot correspond to two different t_p's. Figure 8 lists the corresponding game.

```
$\mathtt{Exp}^{\textit{txn-binding}}_{\pi,\mathcal{A},\mathcal{O},\Phi,\psi,\delta,\alpha,\beta}(\lambda)$
 1. $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\mathtt{Init}()$ ;   $M_{\mathcal{O}} \leftarrow \{\}$
 2. $(p_{\mathcal{O}}, r, s) \leftarrow \mathtt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
 3. $(p_{\mathcal{O}}, t_s, s) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle p_{\mathcal{O}} \neq \perp \rangle$
 4. $t_p \leftarrow AA.\mathtt{Lookup}(t_s, T_{\mathcal{O}})$
 5. return  $(t_p \neq \perp) \wedge (\mathtt{IsValidSecTxn}_{\pi}(t_p, t_s, p_{\mathcal{O}}))$
```

Figure 8: Experiment for Transaction binding property.

Game : The game starts with the initial state generated as per the parameters. Then the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ outputs a secret part of a transaction t_s and the current state $p_{\mathcal{O}}$ according to his capabilities. The challenger checks whether the current state is valid. Then the challenger checks whether t_s corresponds to a transaction created by the oracle with t_p and the validity of transaction t_s with respect to t_p. The corresponding game is listed in Figure 8.

Winning condition : If t_s is present in the list of transactions created by the oracle with corresponding public part t_p and t_s is a valid binding with t_p in the given state, then adversary wins the game.

4.2.3 Spendability

The property of spendability guarantees that the funds associated with a payment address (a) cannot decrease unless the corresponding secret keys are known (Figure 9). i.e. $\mathtt{Balance}_{aft}(a) < \mathtt{Balance}_{bef}(a)$ only if secret key of a (a_{sk}) is known $\forall a$

Game : After the initial setup, the adversary $A = (A_1, A_2, A_3)$ outputs the current state $p_{\mathcal{O}}$ to start the game. The challenger then records the fund balances of all addresses created by the oracle (all addresses in $A_{\mathcal{O}}$). In addition, the challenger obtains a list of unminted transactions created by the oracle, and takes away all V_{new} values from corresponding payment addresses in the stored balances, in order to ensure that the adversary cannot mint those transactions later (Figure 9). Then the adversary evolves the state from that point onwards and the oracle does not create any new addresses or transactions during that period. The adversary has access to the minting oracle only. Subsequently, the adversary outputs the evolved state $p_{\mathcal{O}}$ and the challenger then checks the balances of each address in $A_{\mathcal{O}}$ again in that state and compares with the corresponding initial balances stored (Figure 9).

Winning condition : Adversary wins if there is at least one address in $A_{\mathcal{O}}$ for which the closing balance is less than the starting balance.

```
ExtractUnmintedTxns($T_{\mathcal{O}}, M_{\mathcal{O}}$)
 1. $T_M \leftarrow \bigcup_{m \in M_{\mathcal{O}}} m[1]$
 2. $T \leftarrow AA.\texttt{Keys}(T_{\mathcal{O}})$
 3. $T' \leftarrow T \setminus T_M$
 4. return $T'$


$\texttt{Exp}^{spendability}_{\pi,\mathcal{A},\mathcal{O},\Phi,\psi,\delta,\alpha,\beta}(\lambda)$
 1. $A_{\mathcal{O}}, T_{\mathcal{O}}, B \leftarrow AA.\texttt{Init}()$ ;  $M_{\mathcal{O}} \leftarrow \{\}$; $f_{\mathcal{O}} \leftarrow 0$
 2. $(p_{\mathcal{O}}, r, s) \leftarrow \texttt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
 3. $(p_{\mathcal{O}}, \emptyset, s) \leftarrow \texttt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
 4. for all $a_{pk} \in A_{\mathcal{O}}$ do
 5.       $a_{sk} \leftarrow AA.\texttt{Lookup}(a_{pk}, A_{\mathcal{O}})$; $bal \leftarrow \texttt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})$
 6.       $B \leftarrow AA.\texttt{Insert}(a_{pk}, bal, B)$
 7. $T' \leftarrow \texttt{ExtractUnmintedTxns}(T_{\mathcal{O}}, M_{\mathcal{O}})$
 8. for all $t_p \in T'$ do
 9.    $t_s \leftarrow AA.\texttt{Lookup}(t_p, T_{\mathcal{O}})$
10.    $R \leftarrow \texttt{ExtractRecipientPubAddr}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
11.    $V_{new} \leftarrow \texttt{ExtractOutputVal}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
12.    for $i \in \{0, .., |R|\}$ do
13.       $a_{pk} \leftarrow R[i]$; $bal \leftarrow AA.\texttt{Lookup}(a_{pk}, B)$
14.       $B \leftarrow AA.\texttt{Update}(a_{pk}, bal - V_{new}[i], B)$
15. $(p_{\mathcal{O}}, \emptyset, s) \leftarrow \texttt{RunAdversary}_{\pi,\mathcal{O}_{mint}}(\mathcal{A}_3, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
16. for all $a_{pk} \in A_{\mathcal{O}}$ do
17.       $a_{sk} \leftarrow AA.\texttt{Lookup}(a_{pk}, A_{\mathcal{O}})$; $bal' \leftarrow \texttt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})$
18.       if $bal' < AA.\texttt{Lookup}(a_{pk}, B)$, return 1
19. return 0
```

Figure 9: Experiment for Spendability.

4.2.4 Balance

This property requires that the fund balances of participants in a transaction are updated correctly. Further, the balances of miners' addresses should also be updated correctly with relevant transaction fees and mint values (V_x). These goals can be summarised for a set of transactions involved in one minting operation as follows:

$$\sum_{a \in \bar{S}} V_{old}(a) - \sum_{a \in \bar{R}} V_{new}(a) + \textit{Minted units} = \sum_{a \in \bar{R}_m} V_x(a) \tag{1}$$

$$\texttt{Balance}_{bef}(a) + V_{new}(a) - V_{old}(a) = \texttt{Balance}_{aft}(a) \quad \forall a \in \bar{S}, \bar{R} \tag{2}$$

$$\texttt{Balance}_{bef}(a) + \texttt{Excess}(a) = \texttt{Balance}_{aft}(a) \quad \forall a \in \bar{R}_m \tag{3}$$

A single experiment is defined to capture all three properties in (Figure 10).

Game : In this game, the adversary $A = (A_1, A_2, A_3)$ outputs a tuple of sender addresses $\bar{S}$, a tuple of recipient addresses $\bar{R}$, a tuple of miner addresses $\bar{R}_m$ together with the current state $p_{\mathcal{O}}$. The challenger records the balances of all addresses in the three groups of addresses and the minting history of the oracle $M_{\mathcal{O}}$. Then, the state evolves and the adversary outputs a set of transactions $\{(t_p, t_s)\}$ and the updated state $p_{\mathcal{O}}$. The challenger then records the new minting history M_2 and checks whether only one mint operation has taken place between M_1 and M_2, and also checks whether the minted transactions corresponds to the transactions returned by the

```
$\underline{\texttt{Exp}^{balance}_{\pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda)}$
1.  $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\texttt{Init}()$ ;   $M_{\mathcal{O}} \leftarrow \{\}$;   $S^{''}, R^{''} \leftarrow ()$
2.  $B_{bef}, B_{old}, B_{new}, B_{mint}, B_{adj} \leftarrow AA.\texttt{Init}()$
3.  $v_{in}, v_{out}, v_a \leftarrow 0$
4.  $(p_{\mathcal{O}}, r, s) \leftarrow \texttt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
5.  $(p_{\mathcal{O}}, (\bar{R}, \bar{S}, \bar{R}_m), s) \leftarrow \texttt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle p_{\mathcal{O}} \neq \perp \rangle$
6.  $M_1 \leftarrow M_{\mathcal{O}}$
7.  for all  $(a_{pk}, a_{sk}) \in \bar{R} \parallel \bar{S} \parallel \bar{R}_m$  do
8.      $bal \leftarrow \texttt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})$;   $B_{bef} \leftarrow AA.\texttt{Insert}(a_{pk}, bal, B_{bef})$
9.  $(p_{\mathcal{O}}, (\{(t_p, t_s)\}), s) \leftarrow \texttt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_3, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
10. $M^{'} \leftarrow M_{\mathcal{O}} \setminus M_1$   $\langle\ |M^{'}| = 1\ \rangle$
11. $(p^{'}, \{t_p\}^{'}, V^{'}_x, \bar{R}^{'}_m) \leftarrow M^{'}$   $\langle\ (p^{'} = p_{\mathcal{O}}) \wedge (\{t_p\}^{'} = \{t_p\} \wedge (\bar{R}^{'}_m = \bar{R}_m)\ \rangle$
12. for all  $(t_p, t_s) \in \{(t_p, t_s)\}$  do
13.     $V^{'}_{old} \leftarrow \texttt{ExtractInputVal}(t_p, t_s, p_{\mathcal{O}})$;   $V^{'}_{new} \leftarrow \texttt{ExtractOutputVal}(t_p, t_s, p_{\mathcal{O}})$
14.     $S^{'} \leftarrow \texttt{ExtractSenderPubAddr}(t_p, t_s, p_{\mathcal{O}})$;   $R^{'} \leftarrow \texttt{ExtractRecipientPubAddr}(t_p, t_s, p_{\mathcal{O}})$
15.     for  $i \in \{0, .., |S^{'}| - 1\}$  do
16.         $v_1 \leftarrow AA.\texttt{Lookup}(S^{'}[i], B_{old})$
17.         if  $v_1 = \perp$,   $B_{old} \leftarrow AA.\texttt{Insert}(S^{'}[i], V^{'}_{old}[i], B_{old})$
18.         else  $B_{old} \leftarrow AA.\texttt{Update}(S^{'}[i], V^{'}_{old}[i] + v_1, B_{old})$
19.         $v_{old} \leftarrow v_{old} + V^{'}_{old}[i]$
20.     for  $j \in \{0, .., |R^{'}| - 1\}$  do
21.         $v_2 \leftarrow AA.\texttt{Lookup}(R^{'}[j], B_{new})$
22.         if  $v_2 = \perp$,   $B_{new} \leftarrow AA.\texttt{Insert}(R^{'}[j], V_{new}[j], B_{new})$
23.         else  $B_{new} \leftarrow AA.\texttt{Update}(R^{'}[j], V^{'}_{new}[j] + v_2, B_{new})$
24.         $v_{new} \leftarrow v_{new} + V^{'}_{new}[j]$
25.     $S^{''} \leftarrow S^{''} \parallel S^{'}$;  $R^{''} \leftarrow R^{''} \parallel R^{'}$
26. if  $(S^{''} \neq S) \vee (R^{''} \neq R)$,   return  0
27. for  $k \in \{0, .., |R^{'}_m| - 1\}$  do
28.     $B_{excess} \leftarrow AA.\texttt{Insert}(R^{'}_m[k], V^{'}_x[k], B_{excess})$;   $v_x \leftarrow v_x + V^{'}_x[k]$
29. for all  $(a_{pk}, a_{sk}) \in \bar{R} \parallel \bar{S} \parallel \bar{R}_m$  do
30.     $w_{aft} \leftarrow \texttt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})$
31.     $w_{bef} \leftarrow AA.\texttt{Lookup}(a_{pk}, B_{bef})$
32.     $w_{old} \leftarrow (AA.\texttt{Lookup}(a_{pk}, B_{old})\ ?\_ :\ 0)$
33.     $w_{new} \leftarrow (AA.\texttt{Lookup}(a_{pk}, B_{new})\ ?\_ :\ 0)$
34.     $w_{excess} \leftarrow (AA.\texttt{Lookup}(a_{pk}, B_{excess})\ ?\_ :\ 0)$
35.     if  $w_{aft} \neq w_{bef} + w_{new} + w_{excess} - w_{old}$,   return  1
36. return  0
```

Figure 10: Experiment for the Balance property.

adversary. In addition, another check is performed to see if the sender and recipient addresses involved in all transactions are the same as the sender and recipient addresses returned by the adversary. If any of these checks fails, adversary loses the game. For each transaction returned by the adversary, V_{old} and V_{new} values are recorded separately with the corresponding addresses. In addition, V_x values are also recorded with the miners' address details. Finally, the challenger records respective balances of all involved addresses and checks whether above conditions are satisfied for all the addresses.

Winning condition : Adversary wins the game if there is at least one address in which the individual balances do not satisfy the above three conditions, based on the formula below:

$$\texttt{Balance}_{bef}(a) + V_{new}(a) - V_{old}(a) + V_x(a) = \texttt{Balance}_{aft}(a) \quad \forall a \in \bar{S}, \bar{R}, \bar{R}_m$$

Figure 10 lists the corresponding game ($\texttt{Exp}^{balance}_{\pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}$) which demonstrates this property.

4.2.5 Indemnification

requires that fund balances associated with the payment addresses that are not involved in a transaction should remain unchanged. We define this experiment in Figure 11.

Game : In this game, the adversary $A = (A_1, A_2, A_3)$ outputs the current state first. The challenger records the balances of all addresses in $A_{\mathcal{O}}$ in B_{bef} and also records the mint history M_1. Then, adversary evolves the state and outputs a set of transactions $\{(t_p, t_s)\}$ together with the current state $p_{\mathcal{O}}$ and the challenger ensures that there has been only one mint operation since the previous state, and also whether the set of transactions corresponding to that mint operation matches the transactions returned by the adversary. Then the challenger records the sender and recipient addresses corresponding to the given transactions S and R. Subsequently, he checks the closing balances of all addresses in $A_{\mathcal{O}}$ and ensures none of these addresses are in S or R (Figure 11).

Winning condition : Adversary wins if the balance of at least one address in $A_{\mathcal{O}}$ has changed.

```
$\mathtt{Exp}^{\textit{indemnification}}_{\pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda)$
 1. $A_{\mathcal{O}}, T_{\mathcal{O}}, B_{bef} \leftarrow AA.\mathtt{Init}();\quad M_{\mathcal{O}} \leftarrow \{\};\quad S, R \leftarrow ();\ f_{\mathcal{O}} \leftarrow 0$
 2. $(p_{\mathcal{O}}, r, s) \leftarrow \mathtt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)\quad \langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
 3. $(p_{\mathcal{O}}, \emptyset, s) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)\quad \langle\ (p_{\mathcal{O}} \neq \perp)\ \rangle$
 4. $M_1 \leftarrow M_{\mathcal{O}}$
 5. for all $a_{pk} \in AA.\mathtt{Keys}(A_{\mathcal{O}})$ do
 6.    $a_{sk} \leftarrow AA.\mathtt{Lookup}(a_{pk}, A_{\mathcal{O}})$
 7.    $bal \leftarrow \mathtt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})$
 8.    $B_{bef} \leftarrow AA.\mathtt{Insert}(a_{pk}, bal, B_{bef})$
 9. $(p_{\mathcal{O}}, (\{(t_p, t_s)\}), s) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_3, p_{\mathcal{O}}, \emptyset, r, s, 0)\quad \langle\ p_{\mathcal{O}} \neq \perp\ \rangle$
10. $M' \leftarrow M_{\mathcal{O}} \setminus M_1\quad \langle\ |M'| = 1\ \rangle$
11. $(p', \{t_p\}', V'_a, V'_m, \bar{R}'_m) \leftarrow M'\quad \langle\ (p' = p_{\mathcal{O}})\ \wedge\ (\{t_p\}' = \{t_p\})\ \rangle$
12. for all $(t_p, t_s) \in (\{t_p, t_s\})$ do
13.    $S \leftarrow S \parallel \mathtt{ExtractSenderPubAddr}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
14.    $R \leftarrow R \parallel \mathtt{ExtractRecipientPubAddr}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
15. for all $a_{pk} \in AA.\mathtt{Keys}(A_{\mathcal{O}})$ do $\langle\ a_{pk} \notin S \parallel R\ \rangle$
16.    $a_{sk} \leftarrow AA.\mathtt{Lookup}(a_{pk}, A_{\mathcal{O}})$
17.    if $\mathtt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}}) \neq AA.\mathtt{Lookup}(a_{pk}, B_{bef})$, return 1
18. return 0
```

Figure 11: Experiment for Indemnification.

4.2.6 Positivity

ensures that the fund balance corresponding to each payment address in the system is non-negative at all times. Figure 12 defines the corresponding experiment for this property.

Game : In this game, the adversary $A = (A_1, A_2)$ outputs an address (a_{pk}, a_{sk}) and the state $p_{\mathcal{O}}$. The challenger checks whether the given address is valid and checks the corresponding balance of that address (Figure 12).

Winning condition : Adversary wins if the given address is valid and has a negative balance.

```
$\mathtt{Exp}^{positivity}_{\pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda)$
 1. $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\mathtt{Init}()$;   $M_{\mathcal{O}} \leftarrow \{\}$; $f_{\mathcal{O}} \leftarrow 0$
 2. $(p_{\mathcal{O}}, r, s) \leftarrow \mathtt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$   $\langle\, p_{\mathcal{O}} \neq \perp \,\rangle$
 3. $(p_{\mathcal{O}}, (a_{pk}, a_{sk}), s) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle\, p_{\mathcal{O}} \neq \perp \,\rangle$
 4. return  $(\mathtt{IsValidSecAddr}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})) \wedge (\mathtt{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}}) < 0)$
```

Figure 12: Experiment for Positivity.

4.2.7 Descendancy

requires that an adversary should not be able to produce a valid system state, which does not descend from the genesis state. We define this property in Figure 13.

Game : In this game, adversary $A = (A_1, A_2)$ gives a state to the challenger. The challenger retrieves the checkpoint state of the given state and attempts to loop back to the genesis state by retrieving the checkpoint state iteratively (Figure 13).

Winning condition : Adversary wins if the loop ends up in an invalid state. Experiment for Descendancy:

```
$\mathtt{Exp}^{descendancy}_{\pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda)$
 1. $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\mathtt{Init}()$ ;   $M_{\mathcal{O}} \leftarrow \{\}$; $f_{\mathcal{O}} \leftarrow 0$
 2. $(p_{\mathcal{O}}, r, s) \leftarrow \mathtt{SetupState}_{\pi,\mathcal{O},\mathcal{A}}(\lambda, \alpha)$   $\langle\, p_{\mathcal{O}} \neq \perp \,\rangle$
 3. $(p_{\mathcal{O}}, \emptyset, s) \leftarrow \mathtt{RunAdversary}_{\pi,\mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \emptyset, r, s, \delta)$   $\langle\, p_{\mathcal{O}} \neq \perp \,\rangle$
 4. $p' \leftarrow p_{\mathcal{O}}$
 5. while  $\mathtt{IsValidState}_{\pi}(p', \lambda) \neq 0$  do
 6.    $p^c \leftarrow \mathtt{RetrieveCheckpoint}(p')$
 7.    if $\mathtt{IsGenesisState}_{\pi}(p^c, \lambda)$, return 0
 8.    $p' \leftarrow p^c$
 9. return  1
```

Figure 13: Experiment for Descendancy.

4.2.8 Security of the currency scheme

Having constructed different games to capture a comprehensive set of attacker scenarios related to the functionality of our scheme, now we formally define the security of the proposed currency scheme in terms of those experiments.

Definition 4.1. (**Security of the currency scheme**) For $Y \in \{$*unforgeability, txn-binding, spendability, balance, indemnification, positivity, descendancy*$\}$ the currency scheme Π is said to be $(\psi, \delta, \alpha, \beta)$-secure with respect to Y if for every PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, the advantage of winning the security experiment $\mathtt{Exp}^{Y}_{\Pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda)$ is negligible in $\lambda \in \mathbb{Z}^+$, i.e.

$$\mathbf{Adv}^{Y}_{\Pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}(\mathbf{Exp}^{Y}_{\Pi,\mathcal{A},\mathcal{O},\psi,\delta,\alpha,\beta}(\lambda) = 1) \right| \leq \varepsilon(\lambda)$$

where ε is a negligible function[2] in λ.

5 Anonymity

In this section, we demonstrate how we can model different aspects of anonymity of a currency scheme, in terms of the proposed model. Initially, we formulate a game to capture different

[2]$\forall$ *positive polynomials* $p(\lambda)$, $\exists\, N$ *such that* $\forall\, \lambda > N$, $\varepsilon(\lambda) \leq 1/p(\lambda)$

attacker scenarios each of which represents a different aspect of anonymity. Then, we provide a group of definitions for several anonymity properties stemming from the fundamental concept of *indistinguishability.* The term *indistinguishability* means that it is not possible to distinguish between two known entities in a given situation, e.g. inability to distinguish the sender of a transaction from two possible sender addresses.

We also define a weaker notion of anonymity, *unlinkability*, which is similar to indistinguishability, except the two entities to choose between are not known to the attacker explicitly, but rather by their history in previous transactions. For example, value unlinkability refers to the inability to decide which of two transactions has the same value as a transaction of interest.

We define these anonymity notions around a set of entities in a typical currency scheme. These entities can be categorised as topological and non-topological where topological entities directly correspond to entities in the transaction graph of the scheme. Senders and recipients form the topological category whereas value and other relevant metadata are categorised as non-topological entities, without having a direct relationship to the transaction graph. We parameterise different scenarios where an attacker can manipulate these entities at various levels.

```
$\underline{\texttt{RevealData}(t_p, \omega, \psi, A^*_{\mathcal{O}}, T^*_{\mathcal{O}}, T_{\mathcal{O}}, T'_{\mathcal{O}}, p_1)}$
 1. $(\psi_{pk_s}, \psi_{sk_s}, \psi_{pk_r}, \psi_{sk_r}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$; $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \omega$
 2. $t_p \leftarrow \texttt{LookupPubTxn}(t_p, T^*_{\mathcal{O}})$;  $t_s \leftarrow AA.\texttt{Lookup}(t_p, T_{\mathcal{O}})$;  $\rho_t \leftarrow AA.\texttt{Lookup}(t_p, T'_{\mathcal{O}})$
 3. $S \leftarrow \texttt{ExtractSenderPubAddr}_\pi(t_p, t_s, p_1)$;  $R \leftarrow \texttt{ExtractRecipientPubAddr}_\pi(t_p, t_s, p_1)$
 4. $V_{old} \leftarrow \texttt{ExtractInputVal}_\pi(t_p, t_s, p_1)$;  $V_{new} \leftarrow \texttt{ExtractOutputVal}_\pi(t_p, t_s, p_1)$
 5. $m \leftarrow \texttt{ExtractMetadata}_\pi(t_p, t_s, p_1)$
 6. $U_s \leftarrow (S^{\psi_{pk_s}}, (\texttt{LookupSecAddr}(S, A^*_{\mathcal{O}}))^{\psi_{sk_s}})$
 7. $U_r \leftarrow (R^{\psi_{pk_r}}, (\texttt{LookupSecAddr}(R, A^*_{\mathcal{O}}))^{\psi_{r_{sk}}})$
 8. $U_v \leftarrow ((V_{old}, V_{new})^{\psi_v})$; $U_m \leftarrow (m)^{\psi_m}$
 9. $U_t \leftarrow (t_p^{\psi_t}, t_s^{(\psi_t=2)}, \rho_t^{(\psi_t=3)})$
10. return $(U_s \| U_r \| U_v \| U_m \| U_t)$

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

Figure 14: Additional helper functions for the Anonymity game

5.1 Anonymity Game

In order to facilitate the execution of the Anonymity game in a more transparent manner, we define a few additional helper functions to check the adversarial conditions of inputs at the start of the game (`CheckAdvConditions`) and to reveal data to the adversary at the end (`RevealData`) based on the parameter ψ (Figure 14). Note that for a particular security notion, ψ is constant. Moreover, we also introduce another variable; ω to represent test variable/s. We define $\omega = (\omega_s \omega_r \omega_v \omega_m)$ with each $\omega_x \in \{0,1\}$ to indicate which entity (sender/recipient/value/metadata) is being tested in a given instance of the game. The adversarial inputs are crafted based on the

ω and ψ parameters. We now define a common game to capture all possible attacker scenarios in this setting. Figure 15 illustrates the game.

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

Figure 15: Anonymity Game

Execution of the Game The state initialisation takes place at the beginning of the execution of the game based on α (line 3). The game continues if the returned state $p_{\mathcal{O}}$ is valid. Here we use '$\langle$*condition*$\rangle$' notation to check this condition. In this notation, if the condition inside the brackets is false, then the game terminates and the adversary loses the game. In this case, if the state is valid, the adversary outputs a set of data to the challenger, based on the values of ω and ψ. These outputs include two sets of senders S_0, S_1, recipients R_0, R_1, input/output values V_{old}, V_{new}, miners' addresses R_m, metadata m_0, m_1, a set of transactions T (to be minted), two additional transactions t_0, t_1, two sets of randomness of coins ρ_0, ρ_1 (line 4). If sender/recipient addresses are hidden, respective outputs S/R should be handles to those hidden addresses created by the oracle. i.e. ψ_{pk_s}, ψ_{sk_s}, ψ_{pk_r}, $\psi_{sk_r} \in \{0,1\}$. In addition, if transaction values are hidden (i.e. $\psi_v \in \{0,1\}$), then the adversary provides maximum values for respective input and output values (through V_{old}, V_{new}), and the oracle chooses appropriate transaction values accordingly. If $\psi_m \in \{0,1\}$, then adversary returns a handle for hidden metadata. The transactions t_0 and t_1 represent two transactions created by the adversary when $\psi_t = 5$. For other values of ψ_t, these will remain null. The inputs ρ_0 and ρ_1 provide the randomness (of coins) required to create the transactions when $\psi_t = 4$, or will be empty, otherwise. According to the values of ω and ψ, the challenger checks whether the values returned by the adversary meet the expected criteria and terminates the game if any of the inputs are invalid (line 7). Upon submission of valid inputs, the adversary continues to evolve the current state through appropriate oracle queries.

Next, the challenger checks whether $\psi_t = 5$, in which case the adversary produces the required transactions (which will be considered as t_{p_0} and t_{p_1}). Otherwise, the challenger creates a transaction t_{p_0}, using the input values through the oracle $\mathcal{O}_{txn}$ as given in step 13. Based on the entity/entities being tested as defined by ω, a second transaction t_{p_1} is also created as appropriate (line 14). If the transactions are not mintable and the parameter $\beta = 0$ (i.e. failed mint operations are not allowed), then the game is terminated and the adversary loses the game. We use the notation '$\langle \texttt{IsMintable}_\pi(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\bar{\beta}} \rangle$' to represent this condition. In this case, when β= 0, $\bar{\beta} = 1$ and the game continues if $\texttt{IsMintable}() = 1$. When $\beta = 1$, $\bar{\beta} = 0$ and hence $\texttt{IsMintable}()^0 = 1$ always and hence the game proceeds.

Subsequently, the challenger picks a bit b and chooses to mint t_{p_b} together with the list of transactions T returned by the adversary (line 15). Next, the challenger calls the `RevealData` function to prepare the information that needs to be revealed to the adversary based on ψ and this information is then passed to the adversary. At this point, the adversary is not allowed to create any further transactions involving the revealed addresses. Then he makes a guess ($b^{'}$) for the bit b, based on the revealed data U, minted state p_1 and the adversarial state s. The challenger checks whether the guess is correct, subject to the condition $\beta \vee (f_{\mathcal{O}} \neq 1)$. The adversary wins the game if his guess is correct.

Unsurprisingly, there are over 600,000 different combinations of ω, ψ, δ and α alone, resulting in different attacker scenarios, which reveal the atomicity of anonymity in a currency system. This game helps one to assess which combinations are satisfied by a given currency scheme, by proving that the attacker has negligible advantage of winning the game. In order to simplify this task, in the next section, we come up with a set of anonymity notions which can be related to the terminology discussed in the literature.

5.2 Notions of anonymity

As previously mentioned, different combinations of the parameters in the *Anonymity Game* yields a large number of unique scenarios with respect to anonymity. While some notions may not result in apprehensible real world scenarios, others may assist in assessing different levels in achievable anonymity. In this section, we identify a set of some useful anonymity notions with respect to *indistinguishability* (IND) and *unlinkability* (ULK) of entities; senders (S), recipients (R), value (V) and metadata (M) in a currency scheme.

We define each notion in terms of a unique adversary, based on the adversary's goal, knowledge and power as GOAL-KNOWL-POWER, which is also represented as a unique parameter vector ω-ψ-(δ, α, β). The strongest adversary is modelled with the highest power (to manipulate the state initialisation and the state, and to make minting to fail) and the maximum knowledge (full knowledge of secret keys of senders/recipients, values and metadata), which we name as a FULL-FULL adversary. The weakest adversary has no power and no knowledge of transaction data, hence we name as a NIL-NIL adversary. Other intermediate adversaries are named accordingly to emphasise the capabilities in power and knowledge specific to a given setting. Hence, the highest level of anonymity modelled by the game is the notion ALL-IND-FULL-FULL and the weakest is the notion of NIL-IND-NIL-NIL. Accordingly, Table 6 lists some useful anonymity notions and their corresponding parameter vectors.

Table 6: Some useful anonymity notions

Goal	**Adversarial Knowledge**	**Adversarial Power**	**Parameter vector**
ALL-IND	FULL	FULL	$(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$
S-IND	PUBS	ACTIVE	$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
S-ULK	NILS	ACTIVE	$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
R-IND	PUBR	ACTIVE	$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (3,0)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
R-ULK	NILR	ACTIVE	$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (0,0)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
V-IND	PUBSRV	ACTIVE	$(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 2_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
V-ULK	PUBSR-NILV	FULL	$(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 0_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$
M-IND	PUBM	ACTIVE	$(0_s 0_r 0_v 1_m)_\omega$-$((3,0)_s, (3,0)_r, 2_v, 2_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
M-ULK	PUBSR-NILM	ACTIVE	$(0_s 0_r 0_v 1_m)_\omega$-$((3,0)_s, (3,0)_r, 2_v, 0_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$
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	$(0_s 0_r 0_v 0_m)_\omega$-$((0,0)_s, (0,0)_r, 0_v, 0_m, 0_t)_\psi$-$(0_\delta, 0_\alpha, 0_\beta)$

5.2.1 Topological Entities

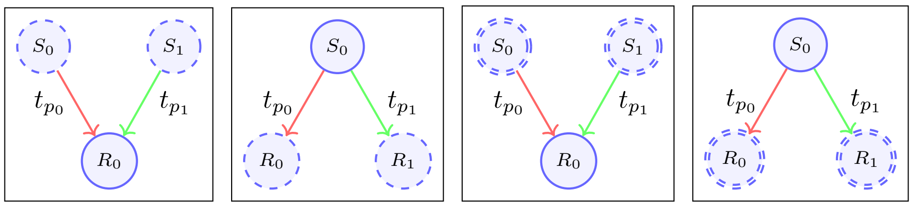
As already mentioned, the identification of topological entities such as senders and recipients participating in a transaction can directly contribute towards constructing the corresponding relationships among those entities. As a result, one can trace the flow of transactions of a particular entity, affecting the level of anonymity. Several studies have been conducted in this regard, especially in the case of Bitcoin, where a transaction graph could be built using publicly available data related to senders and recipients [17]. As such, topological entities play a vital role in the achievable level of anonymity of a currency scheme. Therein, we define a set of useful anonymity properties around these entities in this section (Figure 16).

Sender Indistinguishability (S-IND): We define this property to represent a case where given two possible senders and a transaction, it is not possible to distinguish the correct sender. Figure 16(a) illustrates this scenario. In the anonymity game, only the public keys of the senders will be known to the adversary with $\psi_{pk_s} = 3$ and $\psi_{sk_s} = 0$ with same transaction values and other metadata, and the challenger will create two transactions t_{p_0} and t_{p_1} with same value and metadata. Based on the chosen bit b, the challenger mints the transaction t_{p_b} and the adversary gets to see the data related to the minted transaction, based on ψ_t and has to guess the challenger's choice. The knowledge of recipient addresses can vary based on ψ_{pk_r} and ψ_{sk_r}.

We can see that the game represents the strongest attacker scenario when the recipient addresses are fully controlled by the adversary in a setting with an adversarial hidden state initialisation and the ability to manipulate the state, as well as with the highest knowledge of the transaction (i.e. $\psi_t = 5$). However, having β=1 enables the adversary to craft messages in a manner so that failed mint operations can be used to learn about account balances etc., thus revealing the transaction graph, which will be trivial. Hence, the strongest notion of this property is S-IND-PUBS-ACTIVE which is represented by "$(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s, (4,4)_r, 3_v, 3_m,$ $5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" in the *Anonymity game* with the following formal definition.

Definition 5.1. (**S-IND-PUBS-ACTIVE**) A currency scheme Π is said to satisfy the anonymity notion S-IND-PUBS-ACTIVE with respect to Sender Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameter vector $(1_s 0_r 0_v 0_m)_\omega$-$((3,0)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{S-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$



(a) Sender Indistinguishability (S-IND) (b) Recipient Indistinguishability (R-IND) (c) Sender Unlinkability (S-ULK) (d) Recipient Unlinkability (R-ULK)

Figure 16: Topological anonymity notions. (Dashed outline: addresses with hidden secret keys, double-dashed outline: addresses with hidden public/private keys, Solid outline: both keys known)

Sender Unlinkability (S-ULK): The notion of sender unlinkability is defined to be the property that it is not possible to link a transaction with its corresponding sender in a given setting. As Figure 16(c) illustrates, the adversary has to guess the correct transaction as with S-IND scenario, but without knowing both public/private keys of the senders. i.e. Senders in this case are hidden with $\psi_{pk_s}, \psi_{sk_s} = 0$. The strongest notion in this sense is given by S-ULK-NILS-ACTIVE with the parameter vector "$(1_s 0_r 0_v 0_m)_\omega$-$((0,0)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$"and the corresponding formal definition is as follows.

Definition 5.2. (**S-ULK-NILS-ACTIVE**) A currency scheme Π is said to satisfy the anonymity notion S-ULK-NILS-ACTIVE with respect to Sender Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(1_s 0_r 0_v 0_m)_\omega$-$((0,0)_s, (4,4)_r, 3_v,\ 3_m, 5_t)_\psi$-$(2_\delta,\ 3_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{S-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Recipient Indistinguishability (R-IND): This notion is similar to sender indistinguishability, except with recipient addresses. Hence, it is defined to be one's inability to distinguish the correct recipient out of two given recipients in a given situation. As shown in the Figure 16(b), public keys of the recipients ($\psi_{pk_r} = 3, \psi_{sk_r} = 0$) are known and the senders could be hidden or known as per the parameters ψ_{pk_s} and ψ_{sk_s}. The two transactions t_{p_0} and t_{p_1} both carry the same sender, values and metadata, yet two different recipients. The adversary needs to guess which transaction out of t_{p_0} and t_{p_1} was minted. The strongest adversarial scenario in this case is R-IND-PUBR-ACTIVE, denoted as "$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (3,0)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 0_\alpha, 0_\beta)$". We define the notion formally as below.

Definition 5.3. (**R-IND-PUBR-ACTIVE**) A currency scheme Π is said to satisfy the anonymity notion R-IND-PUBR-ACTIVE with respect to Recipient Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (3,0)_r, 3_v,\ 3_m, 5_t)_\psi$-$(2_\delta, 0_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{R-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

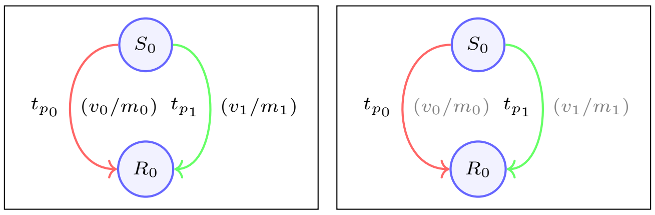
Recipient Unlinkability (R-ULK): This property is referred to as the inability to link a transaction to the correct recipient. Figure 16(d) shows the basic setup for this game where the adversary needs to guess the correct transaction out of the two options t_{p_0} and t_{p_1}, without any knowledge about the corresponding recipients, i.e. ψ_{pk_r},$\psi_{sk_r} = 0$. The strongest notion in this setting is represented as R-ULK-NILR-ACTIVE given by the vector "$(0_s 1_r 0_v 0_m)_\omega$-$((4,4)_s, (0,0)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" and the formal definition is given below.

Definition 5.4. (**R-ULK-NILR-ACTIVE**) A currency scheme Π is said to satisfy the anonymity notion R-ULK-NILR-ACTIVE with respect to Recipient Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s1_r0_v0_m)_\omega$-$((4,4)_s,\ (0,0)_r,\ 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible.

$$\mathbf{Adv}^{R-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

5.2.2 Non-topological Entities

As opposed to topological entities, non-topological entities such as value and metadata in a currency scheme do not directly affect the structure of the transaction graph. However, if made public, these entities also could hinder the privacy of users. Hence, these entities can also be regarded as equally important in determining the level of anonymity in a currency scheme. In this section, we provide formal definitions for major anonymity notions involving non-topological entities; value and metadata (Figure 17).



(a) Value/Metadata indistinguishability (b) Value/Metadata unlinkability (hidden values/metadata)

Figure 17: Non-topological Anonymity notions

Value Indistinguishability (V-IND): The notion of indistinguishability with respect to transaction values refers to the fact that it is impossible to distinguish the correct value from two given values for a given transaction. In the game, the challenger creates two transactions t_{p_0} and t_{p_1}, with two different values v_0 and v_1, while having other entities the same (Figure 17(a)). In this case, the adversary knows what the two values are and other entities can vary according to their ψ values. The challenger then picks a bit b and mints the transaction t_{p_b} and the adversary has to guess which transaction it is. We represent the strongest adversary as PUBSR-ACTIVE as the knowledge of secret keys would leak information about the value. Hence the strongest notion in this scenario is given by V-IND-PUBSR-ACTIVE which is represented by the vector "$(0_s0_r1_v0_m)_\omega$-$((3,0)_s, (3,0)_r, 3_v, 3_m\ , 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" with the following formal definition.

Definition 5.5. (**V-IND-PUBSR-ACTIVE**) A currency scheme Π is said to satisfy V-IND-PUBSR-ACTIVE with respect to Value Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s0_r1_v0_m)_\omega$-$((3,0)_s$, $(3,0)_r, 3_v, 3_m\ , 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{V-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Value Unlinkability (V-ULK): We define the property of unlinkability related to transaction value as the inability correctly identify value of the minted transaction from two possible hidden values. In order to realise this scenario, failed minting operations have to be allowed in the game with the parameter β set to 1, as it would be impossible for the adversary to win the game otherwise. As $\psi_v = 0$, the adversary gives maximum values for V_{new} and V_{old} values from which the challenger generates corresponding values required for the transaction using the `GenerateTxnVals` helper function (Figure 6). Further, as in the case of V-IND, we restrict the knowledge of secret keys of senders/recipients as otherwise the transaction is trivial. As shown in Figure 17(b) in this context, the challenger creates two transactions t_{p_0} and t_{p_1} with hidden transaction values v_0 and v_1, respectively.The challenger then picks a bit b and mints the transaction t_{p_b} and the adversary makes a guess to identify the correct scenario. Accordingly, we have "$(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 0_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$" as the combination of parameters required to achieve the strongest level of anonymity notion V-ULK-PUBSR-NILV-FULL in this sense and the corresponding definition is as follows.

Definition 5.6. (**V-ULK-PUBSR-NILV-FULL**) A currency scheme Π is said to satisfy the anonymity notion V-ULK-PUBSR-NILV-FULL with respect to Value Unlinkability against an adversary $\mathcal{A}$ under a hidden adversarial initialisation, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 0_r 1_v 0_m)_\omega$-$((3,0)_s, (3,0)_r, 0_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{V-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

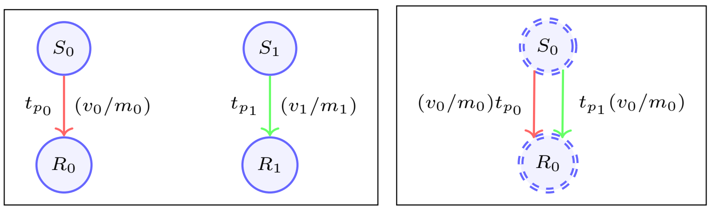
Metadata Indistinguishability (M-IND): Other transaction related data such as scripts, IP addresses etc. also pose a risk to anonymity since, they can be linked to addresses or transactions in many different ways. Although such metadata can be specific to a given implementation, it might be useful in modelling the effects imposed by the other layers of implementations such as the consensus scheme. Hence, in this case, we discuss metadata in general without linking to any specific data, for the completeness of this work.

In this context, we define *Metadata Indistinguishability* to represent the scenario where it is not possible to correctly identify the metadata corresponding to a given transaction, between two given possibilities. Similar to the value indistinguishability scenario, the challenger creates two transactions with different metadata values (already known to the adversary) and mints only one transaction leaving the adversary make a guess as to what it is. The following vector represents the strongest scenario as "$(0_s 0_r 0_v 1_m)_\omega$-$((4,4)_s, (4,4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" as per the notion M-IND-PUBM-ACTIVE and it is formally defined below.

Definition 5.7. (**M-IND-PUB-ACTIVE**) A currency scheme Π is said to satisfy the anonymity notion M-IND-PUB-ACTIVE with respect to Metadata Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 0_r 0_v 1_m)_\omega$-$((4,4)_s, (4,4)_r, 3_v, 3_m,\ 5_t)_\psi$-$(2_\delta,\ 3_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{M-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Metadata Unlinkability (M-ULK): We define the property of unlinkability of metadata with a close analogy to value unlinkability. i.e. Given a transaction, it is not possible to correctly identify the metadata from two given hidden metadata values. Here we use the `GenerateMetadata` helper function to generate the data required for the game (Figure 6). Accordingly, we have the corresponding notion M-ULK-NILM-ACTIVE parameterised by, "$(0_s 0_r 0_v\ 1_m)_\omega$-$((4,4)_s, (4,4)_r,\ 3_v, 0_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$" representing the strongest case in this sense. The formal definition follows.



(a) ALL-IND game (b) NILL-IND game

Figure 18: Strongest and weakest anonymity games

Definition 5.8. (**M-ULK-NILM-ACTIVE**) A currency scheme Π is said to satisfy the anonymity notion M-ULK-NILM-ACTIVE with respect to Metadata Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 0_r 0_v 1_m)_\omega$-$((4,4)_s,\ (4,4)_r,\ 3_v, 0_m,\ 5_t)_\psi$-$(2_\delta, 3_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{M-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

5.2.3 Other useful anonymity notions

Further to above notions, we also formally define the strongest and weakest anonymity notions modelled in this framework as they are useful in benchmarking the anonymity landscape.

Strongest anonymity (ALL-IND) In this setting, the game models two senders and two recipients. The challenger creates two transactions t_{p_0} and t_{p_1} as before, but each transaction is created using distinct set of data; i.e. different sender, recipient, value and metadata (Figure 18 (a)). The strongest adversary in this scenario has the FULL knowledge and FULL power given by ALL-IND-FULL-FULL notion and parameterised by the vector $(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s,\ (4,4)_r,\ 3_v, 3_m,\ 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$. This setting models the highest level of anonymity achievable by a currency scheme and can be considered as "absolute fungibility". We provide the following formal definition in this connection.

Definition 5.9. (**ALL-IND-FULL-FULL**) A currency scheme Π is said to satisfy the anonymity notion ALL-IND-FULL-FULL with respect to indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(1_s 1_r 1_v 1_m)_\omega$-$((4,4)_s,\ (4,4)_r,\ 3_v, 3_m,\ 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{ALL-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

We also define another set of notions here which are useful in analysing the anonymity of many existing cryptocurrencies as those schemes are not secure when the randomness of the coins in a transaction (i.e. when $\psi_t > 1$). These represent slightly weaker anonymity notions with respect to S-IND, S-ULK, R-IND, R-ULK, V-IND and V-ULK as formalised below.

Definition 5.10. (**S-IND-PUBST-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Sender Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(1_s 0_r 0_v 0_m)_\omega$-$((3_{pk}, 0_{sk})_s, (4_{pk}, 4_{sk})_r, 3_v, 3_m,\ 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{S-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Definition 5.11. (**S-ULK-NILS-PUBT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Sender Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(1_s0_r0_v0_m)_\omega$-$((0,0)_s, (4,4)_r, 3_v,\ 3_m, 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{S-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Definition 5.12. (**R-IND-PUBRT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Recipient Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s1_r0_v0_m)_\omega$-$((4,4)_s, (3,0)_r, 3_v, 3_m, 1_t)_\psi$-$(2_\delta$,$0_\alpha$,$0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{R-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Definition 5.13. (**R-ULK-NILR-PUBT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Recipient Unlinkability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s1_r0_v0_m)_\omega$-$((4,4)_s, (0,0)_r,\ 3_v, 3_m, 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ is negligible.

$$\mathbf{Adv}^{R-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Definition 5.14. (**V-IND-PUBSRT-ACTIVE**) A currency scheme Π is said to be anonymous with respect to Value Indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s0_r1_v0_m)_\omega$-$((3,0)_s,\ (3,0)_r, 2_v, 3_m\ , 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{V-IND}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Definition 5.15. (**V-ULK-PUBSRT-NILV-FULL**) A currency scheme Π is said to be anonymous with respect to Value Unlinkability against an adversary $\mathcal{A}$ under a hidden adversarial initialisation, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s0_r1_v0_m)_\omega$-$((3,0)_s, (3,0)_r, 0_v, 3_m,\ 1_t)_\psi$-$(2_\delta$,$3_\alpha$,$1_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{V-ULK}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

Further, we consider the weakest adversary that can be modelled in our game. In this case, the game produces two identical transactions as opposed to the strongest scenario above (Figure 18 (b)). These transactions differ only in their randomness and the adversary has to identify the correct transaction. Hence, the weakest adversary in this case is a NIL-NIL adversary with no knowledge nor power, which is a passive adversary. This means that even δ=0 , meaning that the scheme has a hidden private state, which however may not be the case for most cryptocurrency schemes. Yet, we provide the following formalisation for comparison.

Definition 5.16. (**NIL-IND-NIL-NIL**) A currency scheme Π is said to satisfy the anonymity notion NIL-IND-NIL-NIL with respect to indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s0_r0_v0_m)_\omega$-$((0,0)_s,\ (0,0)_r,$ $0_v, 0_m,\ 0_t)_\psi$-$(0_\delta, 0_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{NIL-IND1}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

As many cryptocurrency schemes have public states, we can see that at the very least, the adversary can view the state, meaning that we can have δ=1 for most schemes. This will model an adversary with VIEW power with other parameters being zero. Hence, we define a slightly less weaker notion in this sense, which can be useful to model anonymity in some real world constructions.

Definition 5.17. (**NIL-IND-NIL-VIEW**) A currency scheme Π is said to satisfy the anonymity notion NIL-IND-NIL-VIEW with respect to indistinguishability against an adversary $\mathcal{A}$, if $\mathcal{A}$'s advantage of winning the Anonymity game defined by the parameters $(0_s 0_r 0_v 0_m)_\omega$-$((0,0)_s$, $(0,0)_r$, $0_v, 0_m$, $0_t)_\psi$-$(1_\delta$, $0_\alpha, 0_\beta)$ is negligible. i.e.

$$\mathbf{Adv}^{NIL-IND2}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta} = \left| \mathbf{Pr}\left[\mathbf{Exp}^{Anonymity}_{\Pi,\mathcal{A},\mathcal{O},\omega,\psi,\delta,\alpha,\beta}(\lambda) = 1\right] - 1/2 \right| \textit{ is negligible in } \lambda.$$

5.2.4 Representation of anonymity notions

In order to clearly represent above anonymity notions, we construct graphical illustrations as shown in figures 19 and 20. These diagrams are useful in comparing anonymity landscape across different implementations while illustrating the complex multidimensional diversity of adversarial parameters.

Figure 19 represents a comparison between the strongest anonymity notion ALL-IND-FULL-FULL against the weakest notion NIL-IND-NIL-NIL in our anonymity game. All other notions lay within the area bounded by these two notions. The larger the area covered by the graph of a given notion, the stronger is the level of anonymity. This is evident from the Figure 20, which represents two more anonymity notions related to S-IND and S-ULK corresponding to definitions 5.1 and 5.2, and shows that S-IND is stronger than S-ULK.

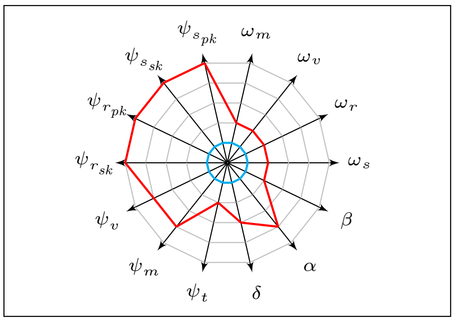


Figure 19: Strongest anonymity notion: ALL-IND-FULL-FULL (red); Weakest notion: NIL-IND-NIL-NIL (cyan)

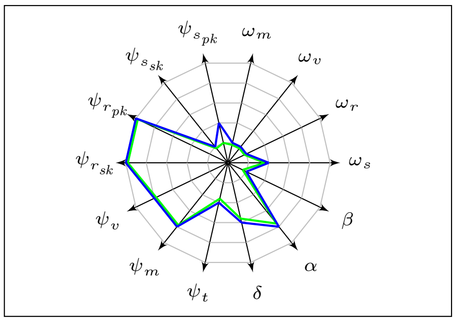


Figure 20: Anonymity notions S-IND-PUBS-ACTIVE (green) and S-ULK-NILS-ACTIVE (blue)

5.3 Theorems

As it is apparent from the definitions presented in the previous section, we can utilise the Anonymity game to realise a multitude of potential different attacker scenarios. Identifying the relationships among these is a worthwhile exercise in order to discern the meaningful aspects of anonymity captured by them.

It is interesting to note that as we vary different security parameters in our model, their correlations result in a non-trivial lattice form as depicted in figures 21 and 22. These relations are interpreted as implications, equivalences and separations. The arrow "$\mapsto$" represents an

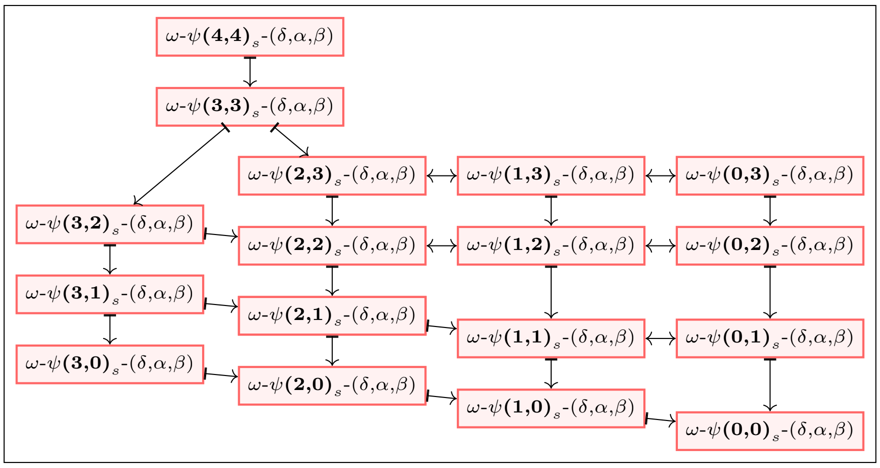


Figure 21: Relationship of anonymity notions for different sender addresses $(\psi_{pk}, \psi_{sk})_s$

implication in the direction of the arrow and a separation in the opposite direction whereas the double arrow "↔" shows an equivalence relation. In order to formalise these relationships, we define a set of theorems that will simplify the process of assessing the anonymity of a currency scheme and we describe them below.

Theorem 1. *For a currency scheme Π and for a given combination of ω, δ, α, ψ_{sk_s}, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t and β, the notion resulting from increasing the value of ψ_{pk_s} while holding others is strictly stronger than the former for the following scenarios:*

i. given that Π is secure in ω-$\psi((\mathbf{3,0})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{2,0})_s)$-(δ,α,β), ω-$\psi((\mathbf{1,0})_s)$-(δ,α,β) and ω-$\psi((\mathbf{0,0})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{3,0})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta) \rightarrow$
ω-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{0,0})_s)$-$(\delta,\alpha,\beta)$

ii. given that Π is secure in ω-$\psi((\mathbf{3,1})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{2,1})_s)$-(δ,α,β) and ω-$\psi((\mathbf{1,1})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,1})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{1,1})_s)$-$(\delta,\alpha,\beta)$

iii. given that Π is secure in ω-$\psi((\mathbf{3,2})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{2,2})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{3,2})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,2})_s)$-$(\delta,\alpha,\beta)$

iv. given that Π is secure in ω-$\psi((\mathbf{3,3})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{2,3})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{3,3})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,3})_s)$-$(\delta,\alpha,\beta)$

v. given that Π is secure in ω-$\psi((\mathbf{4,4})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{3,3})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{4,4})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{3,3})_s)$-$(\delta,\alpha,\beta)$

where $\omega = \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ and $\delta \in \{1,2\}$ (Figure 21).

Proof. (Part i) Assume a currency scheme Π which is secure against the anonymi- ty game defined by a given combination of ψ_{pk_r}, ψ_{sk_r}, ψ_v, ψ_m, ψ_t, α, β and with $(\psi_{pk_s}, \psi_{sk_s}) = (3,0)$. This means that senders' addresses are created with respect to identity information controlled by the adversary and senders' public keys are known throughout, yet secret keys are not known. Now consider a scenario with $(\psi_{pk_s}, \psi_{sk_s}) = (2,0)$, while having other parameters fixed. This means that the adversary has access to the public keys of senders through the oracles and the addresses are honestly generated. When compared to the former case, the adversary has less control in the latter scenario. Hence, we can conclude that if Π is secure against a more powerful adversary, then it is also secure against a less powerful adversary. i.e. ω-$\psi((3,0)_s)$-(δ,α,β) $\rightarrow$ ω-$\psi((2,0)_s)$-(δ,α,β). Similarly, if we consider the case where $(\psi_{pk_s}, \psi_{sk_s}) = (1,0)$ by only changing ψ_{pk_s}, then the adversary gets to know the public keys in the end with secret keys unknown throughout. In comparison with the case $(2,0)$, the adversary has less knowledge about the keys in the case $(1,0)$. Hence, it is clear that if Π is secure in $(2,0)$, it is also secure in $(1,0)$. i.e. ω-$\psi((2,0)_s)$-$(\delta,\alpha,\beta) \rightarrow$ ω-$\psi((1,0)_s)$-(δ,α,β). Similarly, $(0,0)$ case provides even less knowledge to the adversary compared to $(1,0)$. Hence, if Π is secure in $(1,0)$, it is also secure in $(0,0)$. i.e. ω-$\psi((1,0)_s)$-$(\delta,\alpha,\beta) \rightarrow$ ω-$\psi((0,0)_s)$-(δ,α,β).

(Part ii) Similar to part i, we can see that in this case $\psi_{sk_s} = 1$ is fixed. Hence, $\psi_{sk_s} = 3$ represents the strongest case, followed by $\psi_{pk_s} = 2$ and $\psi_{pk_s} = 1$. Following the same argument as above, we can see that $(3,1)$ is more powerful than $(2,1)$, followed by $(1,1)$. And hence the implication relations follow from that.

(Part iii) In this case $\psi_{sk_s} = 2$ is fixed and it follows from above that $(3,2)$ is more powerful than $(2,2)$ since the adversary has the control over identity. Hence, the equivalence relation follows.

(Part iv) This scenario represents the case where $\psi_{sk_s} = 3$. As before, we see that $(3,3)$ is stronger than $(2,3)$ since with the former case, the adversary has control over both the identity and the randomness over the latter case. Hence, it follows that $(3,3) \rightarrow (2,3)$.

(Part v) Consider the case where $(\psi_{pk_s}, \psi_{sk_s})$=$(4,4)$, in which the adversary has full control over the senders. In comparison, in the $(3,3)$ case, although the adversary gets to choose the identity and randomness, he does not have full control over the senders as the address creation is performed honestly. Hence, $(4,4)$ is more powerful than $(3,3)$ and along the same line of argument as before, we can say that $(4,4) \rightarrow (3,3)$. □

Theorem 2. *For a currency scheme Π and for a given combination of ω, δ, α, ψ_{pk_s}, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t and β, the notion resulting from increasing the value of ψ_{sk_s} while holding others is strictly stronger than the former for the following scenarios:*

i. given that Π is secure in ω-$\psi((\mathbf{0,3})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{0,2})_s)$-(δ,α,β), ω-$\psi((\mathbf{0,1})_s)$-(δ,α,β) and ω-$\psi((\mathbf{0,0})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{0,3})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{0,2})_s)$-$(\delta,\alpha,\beta) \rightarrow$
ω-$\psi((\mathbf{0,1})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{0,0})_s)$-$(\delta,\alpha,\beta)$

ii. given that Π is secure in ω-$\psi((\mathbf{1,3})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{1,2})_s)$-(δ,α,β) and ω-$\psi((\mathbf{1,0})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{1,3})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{1,2})_s)$-$(\delta,\alpha,\beta) \rightarrow$
ω-$\psi((\mathbf{1,0})_s)$-(δ,α,β)

iii. given that Π is secure in ω-$\psi((\mathbf{2,3})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{2,2})_s)$-(δ,α,β), ω-$\psi((\mathbf{2,1})_s)$-(δ,α,β) and ω-$\psi((\mathbf{2,0})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{2,3})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,2})_s)$-$(\delta,\alpha,\beta) \rightarrow$
ω-$\psi((\mathbf{2,1})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta)$

iv. given that Π is secure in ω-$\psi((\mathbf{3,3})_s)$-(δ,α,β), Π is also secure in ω- $\psi((\mathbf{3,2})_s)$-(δ,α,β), ω-$\psi((\mathbf{3,1})_s)$-(δ,α,β) and ω-$\psi((\mathbf{3,0})_s)$-(δ,α,β).
i.e. ω-$\psi((\mathbf{3,3})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{3,2})_s)$-$(\delta,\alpha,\beta) \rightarrow$
ω-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta) \rightarrow \omega$-$\psi((\mathbf{3,0})_s)$-$(\delta,\alpha,\beta)$

where $\omega = \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 21).*

Proof. This proof is similar to the proof of theorem 1 based on the fact that the knowledge of secret keys implies the knowledge of the public keys. $\square$

Theorem 3. *For a currency scheme* Π *and for a given combination of* ω, δ, α, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t *and* β, *the resulting notion from increasing the value of* $\psi_{s_{pk}}$ *while holding others fixed, is equivalent to the former when* $\psi_{pk_s} \leq \psi_{sk_s}$ *under the following scenarios;*

i. given that Π *is secure in* ω-$\psi((0,1)_s)$-(δ,α,β), *then* Π *is also secure in* ω- $\psi((1,1)_s)$-(δ,α,β) *and vice versa.*
i.e. ω-$\psi((0,1)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((1,1)_s)$-$(\delta,\alpha,\beta)$

ii. given that Π *is secure in* ω-$\psi((0,2)_s)$-(δ,α,β), *then* Π *is also secure in* ω- $\psi((1,2)_s)$-(δ,α,β) *and* ω-$\psi((2,2)_s)$-(δ,α,β), *and vice versa.*
i.e. ω-$\psi((0,2)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((1,2)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((2,2)_s)$-$(\delta,\alpha,\beta)$

iii. given that Π *is secure in* ω-$\psi((0,3)_s)$-(δ,α,β), *then* Π *is also secure in* ω- $\psi((1,3)_s)$-(δ,α,β) *and* ω-$\psi((2,3)_s)$-(δ,α,β), *and vice versa.*
i.e. ω-$\psi((0,3)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((1,3)_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega$-$\psi((2,3)_s)$-$(\delta,\alpha,\beta)$

where $\omega = \in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 21).*

Proof. (Part i) Assume a currency scheme Π which is secure against the anonymi- ty game defined by a given combination of ω, δ, α, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t, β and with $(\psi_{pk_s}, \psi_{sk_s}) = (0,1)$. This means that senders' public keys are hidden and the secret keys are also hidden but will be revealed in the end (because $\psi_{pk_s} = 1$) in the game. According to our construction, the knowledge of the secret keys implies the knowledge of the public keys. Hence, this scenario can be simplified to a case in which both secret keys and public keys are revealed in the end. Now, consider the case where $(\psi_{pk_s}, \psi_{sk_s}) = (1,1)$ while having all other parameters fixed. In this case, both secret keys and public keys are revealed in the end. As such, we can conclude that both cases represent the same amount of knowledge for the adversary (since all other parameters are constant) and hence both notions are equivalent. Hence, Π is also secure in the case where $(\psi_{pk_s}, \psi_{sk_s}) = (1,1)$. i.e. $\omega_{\bar{s}}$-$\psi((\mathbf{0,1})_s)$-$(\delta,\alpha,\beta) \leftrightarrow \omega_{\bar{s}}$-$\psi((\mathbf{1,1})_s)$-$(\delta,\alpha,\beta)$.

(Part ii) Similar to part i, $\psi_{sk_s} = 2$ in this case corresponds to the case where the addresses are honestly generated and secret keys are accessible by the adversary through the oracles during the game. This means that this case is the same irrespective of ψ_{sk_s} in $(0,2)$, $(1,2)$ and $(2,2)$ through the same line of argument as before. Hence the equivalence relation follows.

(Part iii) As before, $\psi_{sk_s} = 3$ models the case where the addresses are generated based on the randomness chosen by the adversary and the secret keys are already known to the adversary. Following the same argument, we can say that $(1,3)$, $(1,3)$ and $(2,3)$ scenarios are equivalent and hence, the above equivalence relation. $\square$

Theorem 4. *For a currency scheme* Π *for a given combination of* ω, δ, α, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t *and* β *(with* $\psi_t \neq 0$*), the notion resulting from decreasing the value of* $\psi_{s_{pk}}$ *while holding others is not necessarily stronger than the former for the following scenarios:*

i. ω-$\psi((\mathbf{0,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow$
ω-$\psi((\mathbf{3,0})_s)$-(δ,α,β)

ii. ω-$\psi((\mathbf{1,1})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{2,1})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta)$

iii. ω-$\psi((\mathbf{2,2})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,2})_s)$-$(\delta,\alpha,\beta)$

iv. ω-$\psi((\mathbf{2,3})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,3})_s)$-$(\delta,\alpha,\beta)$

v. ω-$\psi((\mathbf{3,3})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{4,4})_s)$-$(\delta,\alpha,\beta)$

where $\omega =\in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 21).*

Proof. (Part i)
(a). Consider a currency scheme where there is some special value associated with the public key of the addresses which could provide a hint about the secret key. For example, the hash of a value V that is related to the corresponding senders is associated with the public keys of the recipient addresses as a special value. Hence, being able to view the public keys of recipients will give an adversary additional information about the transaction, as opposed to having hidden addresses.

Assume that there exists a currency scheme Π_1 which is secure in $(\psi_{pk}, \psi_{sk})_s = (0,0)$. This means that the scheme is secure against an adversary who is unable to view the public keys. i.e. These are hidden addresses created by the oracle. Consider a modified currency scheme $\Pi_1^{'}$ derived from Π_1 such that the transaction creation process is modified as follows :

```
CreateTxn$_{\Pi_1'}$($args, R, V$) {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_1}$($args$)
    if SpecialValue($R$) = Hash($V$) then
      return $((t_p, args), t_s)$
    return $((t_p, \emptyset), t_s)$
}
```

All other operations in $\Pi^{'}$ are of the form: $f_{\Pi'} = f_{Pi}$. In this case, with the modified `CreateTxn` operation, if the adversary knows the transaction, then the public keys can be known and details about senders could be obtained. When a bit b is chosen, the adversary simulates $\Pi^{'}$ and if $\psi_t \geq 1$, the adversary is able to obtain additional information about the senders and hence it is not secure. Accordingly, this scheme is secure in $(0,0)$, but not necessarily secure in $(1,0)$. i.e. $\omega_{\bar{s}}$-$\psi((\mathbf{0,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega_{\bar{s}}$-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta)$.

(b). Consider a scheme Π_2 which is secure when $(\psi_{pk}, \psi_{sk})_s = (1,0)$. Accordingly, the adversary knows the public keys of senders at the end of the game. However, the adversary is not able to create or mint any more transactions involving those senders. With a similar construction as above, consider a modified scheme $\Pi_2^{'}$ with the a modified `Mint` operation, which leaks additional information about the senders from a transaction t_p via a special value V. i.e.

```
Mint$_{\Pi_2'}$($args, t_p, V$) {
    $(p', V_x) \leftarrow$ Mint$_{\Pi_2}$($args$)
    if SpecialValue($t_p$) = Hash($V$) then
      return $((p', args), V_x)$
    return $((p', \emptyset), V_x)$
}
```

Now, $\Pi_2^{'}$ is secure when $(\psi_{pk}, \psi_{sk})_s = (1,0)$ as the transactions cannot be minted after the reveal at the end of the game. However, when $(\psi_{pk}, \psi_{sk})_s = (2,0)$, the adversary has the knowledge of the senders at the start of the game, and hence is able to craft a transaction to reveal additional information about the senders through the above modification, making $\Pi_2^{'}$ not secure when $(\psi_{pk}, \psi_{sk})_s = (2,0)$. i.e. ω-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta)$.

(c). Consider another scheme Π_3 which is secure when $(\psi_{pk}, \psi_{sk})_s = (2,0)$. In this case, the adversary has access to the senders' public address throughout the game. Now consider a modified scheme $\Pi_3^{'}$ as before with a modified `CreateTxn` operation, where a special value is associated with the identity of the senders d, which leaks additional information about the senders' private keys when it is equal to the hash of the special value V. i.e.

```
CreateTxn$_{\Pi_3'}(args, d, V)$ {
    $(t_p, t_s) \leftarrow$ CreateTxn$_{\Pi_3}(args)$
    if SpecialValue$(d)$ = Hash$(V)$ then
        return $((t_p, args), t_s)$
    return $((t_p, \emptyset), t_s)$
}
```

The modified `CreateTxn` together with the knowledge of the transaction, allows the adversary gain additional information about the secret keys of the senders. Hence, Π_3' is secure when the senders' identity is unknown (i.e. $(\psi_{pk}, \psi_{sk})_s = (2, 0)$), but not secure when the identity is known. Hence, we conclude that, ω-$\psi((\mathbf{2,0})_s)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi((\mathbf{3,0})_s)$-$(\delta, \alpha, \beta)$.

(Parts ii to v) We can use the same line of argument as above to provide counter examples for other separations.

□

Theorem 5. *For a currency scheme Π for a given combination of ω, δ, α, $(\psi_{pk}, \psi_{sk})_r$, ψ_v, ψ_m, ψ_t and β (with $\psi_t \neq 0$), the notion resulting from decreasing the value of $\psi_{s_{sk}}$ while holding others is not necessarily stronger than the former for the following scenarios:*

i. ω-$\psi((\mathbf{2,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{2,1})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{2,2})_s)$-$(\delta,\alpha,\beta) \nrightarrow$ ω-$\psi((\mathbf{2,3})_s)$-(δ,α,β)

ii. ω-$\psi((\mathbf{3,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,1})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{3,2})_s)$-$(\delta,\alpha,\beta) \nrightarrow$ ω-$\psi((\mathbf{3,3})_s)$-(δ,α,β)

iii. ω-$\psi((\mathbf{1,0})_s)$-$(\delta,\alpha,\beta) \nrightarrow \omega$-$\psi((\mathbf{1,1})_s)$-$(\delta,\alpha,\beta)$

where $\omega =\in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 21).*

Proof. (Sketch) This proof is similar to the proof of Theorem 4. □

Note that the Theorems 1, 2, 3, 4 and 5 also hold for recipient addresses in the same manner and hence we do not provide separate theorems for the recipients here.

Theorem 6. *For a currency scheme Π and for a given combination of ω, δ, ψ and β, if α is decreased while holding the others, the former notion is strictly stronger than the resulting notion for the following scenarios;*

i. given Π is secure in ω-ψ-$(\delta, \mathbf{3}_\alpha, \beta)$, then Π is also secure in ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$.
i.e. ω-ψ-$(\delta, \mathbf{3}_\alpha, \beta) \ \rightarrow\ \omega$-$\psi$-$(\delta, \mathbf{2}_\alpha, \beta)$

ii. given Π is secure in ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$, then Π is also secure in ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$.
i.e. ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta) \ \rightarrow\ \omega$-$\psi$-$(\delta, \mathbf{1}_\alpha, \beta)$

iii. given Π is secure in ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$, then Π is also secure in ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta)$.
i.e. ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta) \ \rightarrow\ \omega$-$\psi$-$(\delta, \mathbf{0}_\alpha, \beta)$

where $\omega =\in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 22(d)).*

$$\begin{tikzcd}
\omega\text{-}\psi\text{-}(\delta,\mathbf{3_\alpha},\beta) \arrow[d] \\
\omega\text{-}\psi\text{-}(\delta,\mathbf{2_\alpha},\beta) \arrow[d] \\
\omega\text{-}\psi\text{-}(\delta,\mathbf{1_\alpha},\beta) \arrow[d] \\
\omega\text{-}\psi\text{-}(\delta,\mathbf{0_\alpha},\beta)
\end{tikzcd}$$

(d) Varying α (Theorems 6 and 7.)

$$\begin{tikzcd}
\omega\text{-}\psi\text{-}(\mathbf{2_\delta},\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi\text{-}(\mathbf{1_\delta},\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi\text{-}(\mathbf{0_\delta},\alpha,\beta)
\end{tikzcd}$$

(a) Varying δ (Theorems 8 and 9.)

$$\begin{tikzcd}
\omega\text{-}\psi\text{-}(\delta,\alpha,\mathbf{1_\beta}) \arrow[d] \\
\omega\text{-}\psi\text{-}(\delta,\alpha,\mathbf{0_\beta})
\end{tikzcd}$$

(c) Varying β (Theorem 10 and 11.)

$$\begin{tikzcd}
\omega\text{-}\psi(\mathbf{5_t})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{4_t})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{3_t})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{2_t})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{1_t})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{0_t})\text{-}(\delta,\alpha,\beta)
\end{tikzcd}$$

(b) Varying ψ_t (Theorems 16 and 17.)

$$\begin{tikzcd}
\omega\text{-}\psi(\mathbf{3_v})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{2_v})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{1_v})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{0_v})\text{-}(\delta,\alpha,\beta)
\end{tikzcd}$$

(e) Varying ψ_v (Theorems 12 and 13.)

$$\begin{tikzcd}
\omega\text{-}\psi(\mathbf{3_m})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{2_m})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{1_m})\text{-}(\delta,\alpha,\beta) \arrow[d] \\
\omega\text{-}\psi(\mathbf{0_m})\text{-}(\delta,\alpha,\beta)
\end{tikzcd}$$

(f) Varying ψ_m (Theorems 14 and 15).

Figure 22: Relationships among notions based on α, δ, ψ_v, ψ_m, ψ_t and β.

Proof. (Part i) We start with a currency scheme Π that is secure against a ω-ψ-$(\delta, 3_\alpha, \beta)$ adversary. In this case, the adversary has the full control over the state initialisation for the anonymity game. Now, consider an adversary for $\alpha = 2$ with all other parameters the same. In this scenario, the adversary only has control to choose the randomness, but with an honest state initialisation. Hence, the adversary in the latter case is less powerful than the former. Thus it follows that Π is also secure against ω-ψ-$(\delta, 2_\alpha, \beta)$ given that Π is secure against a more powerful adversary in ω-ψ-$(\delta, 3_\alpha, \beta)$.

(Part ii) Similar to part i, $\alpha = 2$ adversary is more powerful than a $\alpha = 1$ adversary since the adversary has no control over state initialisation in the latter case. Hence, given a scheme Π is secure in ω-ψ-$(\delta, 2_\alpha, \beta)$, it is clear that Π is also secure in ω-ψ-$(\delta, 1_\alpha, \beta)$.

(Part iii) Applying the same argument as before, the adversary has less information when $\alpha = 0$ compared to $\alpha = 1$. Hence, given that a currency scheme Π is secure against ω-ψ-$(\delta, 1_\alpha, \beta)$, then Π is also secure against a less powerful adversary, ω-ψ-$(\delta, 0_\alpha, \beta)$. □

Theorem 7. *For a currency scheme Π and for a given combination of ω, δ, ψ and β (with $\delta \in \{0, 1\}$), if α is increased while holding the others, the system is necessarily secure in the* *resulting notion for the following scenarios;*

i. Given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$ *and* Π_0 *is not secure in* ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$*, then there exists a currency scheme* Π *which is secure in* ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$ *but not secure in* ω-ψ-$(\delta, \mathbf{3}_\alpha, \beta)$*.*
i.e. ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, \mathbf{3}_\alpha, \beta)$

ii. Given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$ *and* Π_0 *is not secure in* ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$*, then there exists a currency scheme* Π *which is secure in* ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$ *but not secure in* ω-ψ-$(\delta, \mathbf{2}_\alpha, \beta)$*.*
i.e. ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, \mathbf{2}_\alpha, \beta)$

iii. Given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta)$ *and* Π_0 *is not secure in* ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta)$*, then there exists a currency scheme* Π *which is secure in* ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta)$ *but not secure in* ω-ψ-$(\delta, \mathbf{1}_\alpha, \beta)$*.*
i.e. ω-ψ-$(\delta, \mathbf{0}_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, \mathbf{1}_\alpha, \beta)$

where $\omega =\in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}, \psi_t, \beta \in \{0, 1\}$ *and* $\delta \in \{0, 1, 2\}$ *(Figure 22(d)).*

Proof. (Part i) Assume that there exist two currency schemes Π_1 and Π_0 such that Π_1 is secure in ω-ψ-$(\delta, 2_\alpha, \beta)$ and Π_0 is not secure in ω-ψ-$(\delta, 2_\alpha, \beta)$. In the case of $\gamma = 2$, the state is initialised through an honest initialisation based on the randomness chosen by the adversary whereas when $\gamma = 3$, the adversary generates the initial state on his own with randomness of his choice. Consider a new currency scheme Π, where the state initialisation process takes place based on the selection of a bit b. If $b = 1$, then the initial state is decided by the honest behaviour as follows:

$$\texttt{Init}_\Pi(\lambda; r) = (\texttt{Init}_{\Pi_1}(\lambda, r_1), 1)$$

Otherwise, the adversary chooses to construct the initial state p_0 through: $p_0 = (\texttt{Init}_{\Pi_0}(\lambda; r_1), 0)$ simulating the insecure Π_0 protocol. Other functions in Π will be of the following form depending on the value of b:

$$f_\Pi((p, b), ...) = f_{\Pi_b}(p, ...)$$

In the case of $b = 1$, this construction returns $\texttt{Init}_{\Pi_1}$ which makes Π secure in ω-ψ-$(\delta, 2_\alpha, \beta)$. When $b = 0$, the adversary chooses to output $\texttt{Init}_{\Pi_0}$ simulating the insecure protocol Π_0. Π is not secure in ω-ψ-$(\delta, 3_\alpha, \beta)$ in this case. This shows that ω-ψ-$(\delta, 2_\alpha, \beta) \rightarrow \omega$-$\psi$-$(\delta, 3_\alpha, \beta)$ is false. Hence, we conclude that ω-ψ-$(\delta, 2_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, 3_\alpha, \beta)$.

(Part ii) Assume there exist two currency schemes Π_1 and Π_0 such that Π_1 is secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ and Π_0 is not secure in ω-ψ-$(\delta, 1_\alpha, \beta)$. $\gamma = 1$ represents an honest state initialisation with randomness over which the adversary does not have control, whereas when $\gamma = 2$, the adversary chooses the randomness with an honest state initialisation. Hence the difference between the two cases is whether the randomness is chosen by the adversary as per the algorithm `SetupState`. We now define a new currency scheme Π where the initialisation takes place upon a selection of a random string r_2 as follows depending on a bit b:

$$\texttt{Init}_\Pi(\lambda, (r_1, r_2)) = \begin{cases} (\texttt{Init}_{\Pi_1}(\lambda, r_1), 1), & \text{if } r_2 \neq 0...0 \\ (\texttt{Init}_{\Pi_0}(\lambda, r_1), 0), & \text{if } r_2 = 0...0 \end{cases}$$

Similarly, all functions in Π need to be dependent on r, similar to part i above. i.e.

$$f_\Pi(p, ...) = \begin{cases} f_{\Pi_1}(p, ...), & \text{if } p = (., 1) \\ f_{\Pi_0}(p, ...), & \text{if } p = (., 0) \end{cases}$$

According to this construction, in the honest scenario ($b = 1$), Π is secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ with since it returns state initialisation $(\texttt{Init}_{\Pi_1}(r), 1)$ which is ω-ψ-$(\delta, 1_\alpha, \beta)$ secure. However, it should be noted that r_2 can be 00...0 in the honest scenario also with a small probability of $1/2^\lambda$ which is negligible and hence we can assume that Π is secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ in the honest case. When $b = 0$, the adversary can choose r_2 to be 0..0 to simulate the insecure protocol Π_0 which makes Π insecure in ω-ψ-$(\delta, 2_\alpha, \beta)$. This means that ω-ψ-$(\delta, 1_\alpha, \beta) \rightarrow \omega$-$\psi$-$(\delta, 2_\alpha, \beta)$ is false since Π can be secure in ω-ψ-$(\delta, 1_\alpha, \beta)$ but not secure in ω-ψ-$(\delta, 2_\alpha, \beta)$. Thus we conclude that ω-ψ-$(\delta, 1_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, 2_\alpha, \beta)$.

(Part iii) Assume there exist two currency schemes Π_1 and Π_0 such that Π_1 is secure in ω-ψ-$(\delta, 0_\alpha, \beta)$ and Π_0 is not secure in ω-ψ-$(\delta, 0_\alpha, \beta)$. In this scenario, an honest state initialisation takes place and the adversary does not see the randomness involved whereas with $\gamma = 1$ state initialisation, the adversary is able to see the randomness.

Construct a new currency scheme Π as explained previously.

$$\texttt{Init}_\Pi(\lambda, (r_1, r_2)) = (\texttt{Init}_{\Pi_0}(\lambda, r_1), \texttt{Init}_{\Pi_1}(\lambda, r_1), r_2, 1)$$

In the honest case, this will output $(p_0, p_1, 1)$. Consider the $\gamma = 0$ game where, immediately after the `RunAdversary` process is run for the first time, a transaction is run by the adversary which would reveal the randomness and the bit b. i.e. (t_p, r, b). This would simulate a $\gamma = 1$ scenario and if the bit $b = 0$, then f_{Π_0} will be chosen. However, Π_0 is not secure when $\gamma = 0$, and thus it makes this situation insecure. Hence, we claim that Π is not secure with $\gamma = 1$ in this case since Π_0 is not secure when $\gamma = 0$. This shows that Π can be secure in ω-ψ-$(\delta, 0_\alpha, \beta)$ but not secure in ω-ψ-$(\delta, 1_\alpha, \beta)$. i.e. ω-ψ-$(\delta, 0_\alpha, \beta) \nrightarrow \omega$-$\psi$-$(\delta, 1_\alpha, \beta)$. □

Theorem 8. *For a currency scheme Π and for a given combination of ω, α, ψ and β, if δ is decreased while holding the others, the former notion is strictly stronger than the resulting notion for the following scenarios;*

i. given Π is secure in ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ then Π is also secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$.
i.e. ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta) \rightarrow \omega$-$\psi$-$(\mathbf{1}_\delta, \alpha, \beta)$

ii. given Π is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ then Π is also secure in ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$.
i.e. ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta) \rightarrow \omega$-$\psi$-$(\mathbf{0}_\delta, \alpha, \beta)$

where $\omega =\in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ and $\delta \in \{1, 2\}$ (Figure 22(a)).

Proof. $\delta = 2$ represents the strongest adversary with the capability to manipulate the state whereas the adversary is only able to view the state when $\delta = 1$, hence representing a weaker adversary. Similarly with $\delta = 0$, the state is private and hence the adversary is the weakest in this respect, being unable to view the state. Using the same proof technique as in Theorem 6, we can see that ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ represents a more powerful attacker than ω-ψ- $(\mathbf{1}_\delta, \alpha, \beta)$ and also ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ is more powerful than ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$, hence it is clear that both implications are true. □

Theorem 9. *For a currency scheme Π and for a given combination of ω, α, ψ and β (with $\alpha \neq 0$), if δ is increased while holding the others, the resulting notion is not necessarily stronger than the former notion for the following scenarios;*

i. given that two currency schemes Π_0 and Π_1 exists such that Π_1 is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ and Π_0 is not secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$, then there exists a currency scheme Π which is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ but not secure in ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$.
i.e. ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$-$\beta \nrightarrow \omega$-$\psi$-$(\mathbf{2}_\delta, \alpha, \beta)$

ii. *given that two currency schemes* Π_0 *and* Π_1 *exists such that* Π_1 *is secure in* ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ *and* Π_0 *is not secure in* ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$*, there exists a currency scheme* Π *which is secure in* ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ *but not secure in* ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$*.*
   i.e. ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta) \;\nrightarrow\; \omega$-$\psi$-$(\mathbf{1}_\delta, \alpha, \beta)$

where $\omega =\in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2,\ 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}$*,* $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 22(a)).*

Proof. (Part i) The difference between the two scenarios where $\delta = 1$ and $\delta = 2$ is that with $\delta = 2$, the adversary is able to modify the state whereas with $\delta = 1$, he can only view the state. Suppose that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ and Π_0 is not secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$. Assume that a currency scheme Π is constructed as before in the proof of part (ii) of the Theorem 7. The initialisation of the scheme Π results in: (p_0, p_1, r, b) with functions f_{Π_b}. In the honest construction, we have $b = 1$, with f_{Π_1} and the adversary is only able to view the state which simulates a ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ scenario. Π_1 is already secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ and hence Π is also secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$. Now assume that the adversary gets to choose the bit $b = 0$. In this case, functions f_{Π_0} will be executed and hence the adversary is now able to change the state through the functions f_{Π_0} that return the state as $(p_0^{'}, p_1, r, 0)$ which represent a ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ adversary. However, from the definition, Π_0 is not secure against this adversary since Π_0 is not secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$ and hence is not secure in ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ as well. Therefore, Π is also not secure in ω-ψ-$(\mathbf{2}_\delta, \alpha, \beta)$ in this case.

(Part ii) Suppose that a currency scheme Π_1 is secure in ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ and another currency scheme Π_0 is not secure in ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$. Define a new honest construction for a currency scheme Π as described in the proof of part (ii) of the Theorem 7 with the same initialisation and a modification to the `Mint` function: i.e.

```
Init_Π(λ, (r_1, r_2)) = (Init_Π0(λ, r_1), (Init_Π1(λ, r_1), r_2, 1))
Mint_Π((p_0, p_1, r, b), T, args) = {
    if $p_0$ and $p_1$ are inital states and $b = 1$ and $T = \{r\}$,
      then return $((p_0, p_1, r, 0), outputs)$
  }
```

In the honest case, $\texttt{Init}_\Pi$ will output $(p_0, p_1, 1)$ and all functions f_{Π_1} will be executed. Since the adversary is not able to view the state now, this simulates ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$. We claim that Π is ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ secure in this case since the adversary has the same amount of information as for Π_1 and Π_1 is secure in ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$. Then immediately after the state initialisation, the adversary mints $\{r\}$ which will return $(p_0, p_1, r, 0)$, setting bit $b = 0$. In this instance, the protocol simulates Π_0 which is not secure. Hence we can conclude that Π is secure in ω-ψ-$(\mathbf{0}_\delta, \alpha, \beta)$ but not secure in ω-ψ-$(\mathbf{1}_\delta, \alpha, \beta)$. □

Theorem 10. *For a currency scheme* Π *and for a given combination of* ω*,* α *and* ψ*, if* β *is decreased while holding the others, the former notion is strictly stronger than the resulting notion for the following scenarios;*

i. *given* Π *is secure in* ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta)$ *then* Π *is also secure in* ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$*.*
   i.e. ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta) \;\rightarrow\; \omega$-$\psi$-$(\delta, \alpha, \mathbf{0}_\beta)$

where $\omega =\in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2,\ 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}$*,* $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 22(a)).*

Proof. Consider a currency scheme Π which is secure in ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta)$. In this case, the adversary is able to cause minting to fail so that failed mint operations may leak information about the corresponding transaction. On the other hand, $\beta = 0$ represents a weaker adversary as no additional information is leaked in this case. As the scheme Π is secure against a more powerful adversary with $\beta = 1$, we can conclude that Π is also secure against any weaker adversary, and an adversary with $\beta = 0$. i.e. ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta) \;\rightarrow\; \omega$-$\psi$-$(\delta, \alpha, \mathbf{0}_\beta)$. □

Theorem 11. *For a currency scheme* Π *and for a given combination of* ω, α *and* ψ, *if* β *is decreased while holding the others, the former notion is strictly stronger than the resulting notion for the following scenarios;*

i. *given that* Π *is secure in* ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta)$ *then* Π *is not necessarily secure in* ω-ψ-$(\delta, \alpha, \mathbf{1}_\beta)$. *i.e.* ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta) \nrightarrow \omega$-$\psi$-$(\delta, \alpha, \mathbf{1}_\beta)$

where $\omega =\in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 22(a)).*

Proof. Assume that a currency scheme Π is secure in ω-ψ-$(\delta,\alpha,0_\beta)$. Consider a construction $\Pi^{'}$ similar to Π except that the `Mint` operation is modified with an input bit b and a special value V which reveals additional information about the transaction when $b = 1$ and when a mint operation fails. i.e.

```
Mint_{Π'}($args, b, V$) {
if  (Mint_Π($args$) =⊥) ∧ ($b$ = 1)   then
  return $V$
else return  Mint_Π($args$)
  }
```

When the bit $b = 0$, $\Pi^{'}$ functions similar to Π and hence is secure in ω-ψ-$(\delta,\alpha,0_\beta)$. However, when $b = 1$, with the modified `Mint` function models a scenario for β= 1 for $\Pi^{'}$, yet $\Pi^{'}$ is not secure here as the `Mint` operation leaks special information about the transaction in this case. Hence, we conclude that ω-ψ-$(\delta, \alpha, \mathbf{0}_\beta) \nrightarrow \omega$-$\psi$-$(\delta, \alpha, \mathbf{1}_\beta)$. □

Theorem 12. *For a currency scheme* Π *and for a given combination of* ω, δ, α, $(\psi_{pk}, \psi_{sk})_s$, ψ_m, ψ_t *and* β, *when the value of* ψ_v *is decreased while holding others fixed, the former notion is strictly stronger than the resulting notion under the following scenarios;*

i. *given* Π *is secure in* ω-$\psi(\mathbf{3}_v)$-(δ, α, β), *then* Π *is also secure in* ω-$\psi(\mathbf{2}_v)$- (δ, α, β). *i.e.* ω-$\psi(\mathbf{3}_v)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{2}_v)$-$(\delta, \alpha, \beta)$

ii. *given* Π *is secure in* ω-$\psi(\mathbf{2}_v)$-(δ, α, β), *then* Π *is also secure in* ω-$\psi(\mathbf{1}_v)$- (δ, α, β). *i.e.* ω-$\psi(\mathbf{2}_v)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{1}_v)$-$(\delta, \alpha, \beta)$

iii. *given* Π *is secure in* ω-$\psi(\mathbf{1}_v)$-(δ, α, β), *then* Π *is also secure in* ω-$\psi(\mathbf{0}_v)$- (δ, α, β). *i.e.* ω-$\psi(\mathbf{1}_v)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{0}_v)$-$(\delta, \alpha, \beta)$

where $\omega =\in \{1,0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0,1,2,\ 3,4\}$, $\psi_v, \psi_m \in \{0,1,2,3\}$, $\psi_t \in \{0,1,2,3,4,5\}, \beta \in \{0,1\}$ *and* $\delta \in \{1,2\}$ *(Figure 22(e)).*

Proof. As before, it is clear that $\psi_v = 3$ represents a stronger adversary compared to $\psi_v = 2$ and $\psi_v = 2$ adversary is stronger than $\psi_v = 1$ by our construction and $\psi_v = 1$ adversary is stronger than $\psi_v = 0$. Hence, given that a currency scheme Π is secure against a $\psi_v = 3$ adversary, then Π is also secure against $\psi_v = 2$. Similarly, Π is also secure in $\psi_v = 1$ then in $\psi_v = 0$. □

Theorem 13. *For a currency scheme* Π *and for a given combination of* ω, δ, α, $(\psi_{pk}, \psi_{sk})_s$, ψ_m, ψ_t *and* β *(with* $\psi_t = 1$*), the resulting notion from increasing the value of* ψ_v *while holding others fixed, the scheme is not necessarily secure in the resulting notion under the following scenarios;*

i. *given that two currency schemes* Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-$\psi(\mathbf{0}_v)$-(δ, α, β) *and* Π_0 *is not secure in* ω-$\psi(\mathbf{0}_v)$-(δ, α, β), *then there exists a currency scheme* Π *which is secure in* ω-$\psi(\mathbf{0}_v)$-(δ, α, β) *but not secure in* ω-$\psi(\mathbf{1}_v)$-(δ, α, β). *i.e.* ω-$\psi(\mathbf{0}_v)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi(\mathbf{1}_v)$-$(\delta, \alpha, \beta)$

ii. given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-$\psi(\mathbf{1}_v)$-(δ, α, β) *and* Π_0 *is not secure in* ω-$\psi(\mathbf{1}_v)$-(δ, α, β)*, then there exists a currency scheme* Π *which is secure in* ω-$\psi(\mathbf{1}_v)$-(δ, α, β) *but not secure in* ω-$\psi(\mathbf{2}_v)$-(δ, α, β)*.*
i.e. ω-$\psi(\mathbf{1}_v)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi(\mathbf{2}_v)$-$(\delta, \alpha, \beta)$

iii. given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-$\psi(\mathbf{2}_v)$-(δ, α, β) *and* Π_0 *is not secure in* ω-$\psi(\mathbf{2}_v)$-(δ, α, β)*, then there exists a currency scheme* Π *which is secure in* ω-$\psi(\mathbf{2}_v)$-(δ, α, β) *but not secure in* ω-$\psi(\mathbf{3}_v)$-(δ, α, β)*.*
i.e. ω-$\psi(\mathbf{2}_v)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi(\mathbf{3}_v)$-$(\delta, \alpha, \beta)$

where $\omega = \in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 22(e)).*

Proof. (Sketch)
The proof follows the same line of argument as the proof in Theorem 4, with the exception that the special value gives a hint about the transaction value instead of the senders. □

Theorem 14. *For a currency scheme* Π *and for a given combination of* ω*,* δ*,* α*,* $(\psi_{pk}, \psi_{sk})_s$*,* ψ_v*,* ψ_t *and* β*, when the value of* ψ_m *is decreased while holding others fixed, the former notion is strictly stronger than the resulting notion under the following scenarios;*

i. given Π *is secure in* ω-$\psi(\mathbf{3}_m)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{2}_m)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{3}_m)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{2}_m)$-$(\delta, \alpha, \beta)$

ii. given Π *is secure in* ω-$\psi(\mathbf{2}_m)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{1}_m)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{2}_m)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{1}_m)$-$(\delta, \alpha, \beta)$

iii. given Π *is secure in* ω-$\psi(\mathbf{1}_m)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{0}_m)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{1}_m)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{0}_m)$-$(\delta, \alpha, \beta)$

where $\omega = \in \{1, 0\}^4$*,* $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$*,* $\psi_v, \psi_m \in \{0, 1, 2, 3\}$*,* $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 22(f)).*

Proof. Adhering to the same line of argument, with $\psi_m = 3$, the adversary is more powerful than $\psi_m = 2$ adversary, since the adversary has full control over metadata in the former case. Hence it follows that given a currency scheme which is secure against a $\psi_m = 3$ adversary, the scheme is also secure against a less powerful $\psi_m = 2$ adversary. And through the same line of argument, it follows that $(\psi_m = 2) \rightarrow (\psi_m = 1)$ and $(\psi_m = 1) \rightarrow (\psi_m = 0)$. □

Theorem 15. *For a currency scheme* Π *and for a given combination of* ω*,* δ*,* α*,* $(\psi_{pk}, \psi_{sk})_s$*,* ψ_m*,* ψ_t *and* β *(with* $\psi_t = 1$*), the resulting notion from increasing the value of* ψ_m *while holding others fixed, the scheme is not necessarily secure in the resulting notion under the following scenarios;*

i. given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-$\psi(\mathbf{2}_m)$-(δ, α, β) *and* Π_0 *is not secure in* ω-$\psi(\mathbf{2}_m)$-(δ, α, β)*, then there exists a currency scheme* Π *which is secure in* ω-$\psi(\mathbf{2}_m)$-(δ, α, β) *but not secure in* ω-$\psi(\mathbf{3}_m)$-(δ, α, β)*.*
i.e. ω-$\psi(\mathbf{2}_m)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi(\mathbf{3}_m)$-$(\delta, \alpha, \beta)$

ii. given that two currency schemes Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-$\psi(\mathbf{1}_m)$-(δ, α, β) *and* Π_0 *is not secure in* ω-$\psi(\mathbf{1}_m)$-(δ, α, β)*, then there exists a currency scheme* Π *which is secure in* ω-$\psi(\mathbf{1}_m)$-(δ, α, β) *but not secure in* ω-$\psi(\mathbf{2}_m)$-(δ, α, β)*.*
i.e. ω-$\psi(\mathbf{1}_m)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi(\mathbf{2}_m)$-$(\delta, \alpha, \beta)$

iii. *given that two currency schemes* Π_0 *and* Π_1 *exist such that* Π_1 *is secure in* ω-$\psi(\mathbf{0}_m)$-(δ, α, β) *and* Π_0 *is not secure in* ω-$\psi(\mathbf{1}_m)$-(δ, α, β)*, then there exists a currency scheme* Π *which is secure in* ω-$\psi(\mathbf{0}_m)$-(δ, α, β) *but not secure in* ω-$\psi(\mathbf{1}_m)$-(δ, α, β)*.*
i.e. ω-$\psi(\mathbf{0}_m)$-$(\delta, \alpha, \beta) \nrightarrow \omega$-$\psi(\mathbf{1}_m)$-$(\delta, \alpha, \beta)$

where $\omega =\in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 22(f)).*

Proof. (Sketch)
The proof follows the same line of argument as the proof in Theorem 4, with the exception that the special value gives a hint about the transaction metadata instead of the senders. □

Theorem 16. *For a currency scheme* Π *and for a given combination of* ω*,* δ*,* α*,* $(\psi_{pk}, \psi_{sk})_s$*,* $(\psi_{pk}, \psi_{sk})_r$ ψ_v*,* ψ_m *and* β*, when the value of* ψ_t *is decreased while holding others fixed, the former notion is strictly stronger than the resulting notion under the following scenarios;*

i. *given* Π *is secure in* ω-$\psi(\mathbf{5}_t)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{4}_t)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{5}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{4}_t)$-$(\delta, \alpha, \beta)$

ii. *given* Π *is secure in* ω-$\psi(\mathbf{4}_t)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{3}_t)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{4}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{3}_t)$-$(\delta, \alpha, \beta)$

iii. *given* Π *is secure in* ω-$\psi(\mathbf{3}_t)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{2}_t)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{3}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{2}_t)$-$(\delta, \alpha, \beta)$

iv. *given* Π *is secure in* ω-$\psi(\mathbf{2}_t)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{1}_t)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{2}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{1}_t)$-$(\delta, \alpha, \beta)$

v. *given* Π *is secure in* ω-$\psi(\mathbf{1}_t)$-(δ, α, β)*, then* Π *is also secure in* ω-$\psi(\mathbf{0}_t)$- (δ, α, β)*.*
i.e. ω-$\psi(\mathbf{1}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{0}_t)$-$(\delta, \alpha, \beta)$

where $\omega =\in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 22(b)).*

Proof. (Part i) Consider a currency scheme Π which is secure in ω-$\psi(\mathbf{5}_t)$-(δ, α, β). With $\psi_t = 5$, the adversary has the highest possible knowledge of the transaction as the adversary creates the transaction and hence is more powerful than any other adversary having the knowledge of $\psi_t < 5$ (while having other parameters fixed). This means that if a currency scheme Π is secure against a stronger adversary with $\psi_t = 5$, then Π is secure against less powerful adversaries; e.g. an adversary with $\psi_t = 4$. i.e. ω-$\psi(\mathbf{5}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{4}_t)$-$(\delta, \alpha, \beta)$.

(Part ii) Similarly, ω-$\psi(\mathbf{4}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{3}_t)$-$(\delta, \alpha, \beta)$ also holds as being able to choose the randomness for the transaction ($\psi_t = 4$) leaks additional information about the transaction to the adversary earlier in the game compared to knowing that at the end of the game ($\psi_t = 3$), which models a weaker adversary.

(Part iii) With $\psi_t = 3$, the knowledge of the randomness of the transaction (i.e. actual coins involved) provides more information to the adversary than just the secret part of the transaction t_s (i.e. $\psi_t = 2$). Hence, ω-$\psi(\mathbf{3}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{2}_t)$-$(\delta, \alpha, \beta)$ holds.

(Part iv) With the same argument, $\psi_t = 2$ represents a more powerful adversary than ψ_1 with the knowledge of just the public part of the transaction. i.e. ω-$\psi(\mathbf{2}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{1}_t)$-$(\delta, \alpha, \beta)$.

(Part v) In this case of $\psi_t = 1$, the adversary is able to view the transaction whereas when $\psi_t = 0$, the transaction is hidden. Hence, the former case shows a more powerful adversary than the latter case. Accordingly, ω-$\psi(\mathbf{1}_t)$-$(\delta, \alpha, \beta) \rightarrow \omega$-$\psi(\mathbf{0}_t)$-$(\delta, \alpha, \beta)$. □

Theorem 17. *For a currency scheme* Π *and for a given combination of* ω, δ, α, $(\psi_{pk}, \psi_{sk})_s$, ψ_v, ψ_m *and* β *the resulting notion from increasing the value of* ψ_t *while holding others fixed, the scheme is not necessarily secure in the resulting notion under the following scenarios;*

i. *Given that there exists a currency scheme* Π_1 *which is secure in* ω-$\psi(\mathbf{0}_t)$- (δ, α, β), *it does not necessarily imply that* Π_1 *is secure in* ω-$(\psi(\mathbf{1}_t)$-(δ, α, β). *i.e.* ω-$(\psi(\mathbf{0}_t)$-$(\delta, \alpha, \beta) \nrightarrow \Pi_1$ *is secure in* ω-$(\psi(\mathbf{1}_t)$-(δ, α, β).

ii. *Given that there exists a currency scheme* Π_1 *which is secure in* ω-$\psi(\mathbf{1}_t)$- (δ, α, β), *it does not necessarily imply that* Π_1 *is secure in* ω-$(\psi(\mathbf{2}_t)$-(δ, α, β). *i.e.* ω-$(\psi(\mathbf{1}_t)$-$(\delta, \alpha, \beta) \nrightarrow \Pi_1$ *is secure in* ω-$(\psi(\mathbf{2}_t)$-(δ, α, β).

iii. *Given that there exists a currency scheme* Π_1 *which is secure in* ω-$\psi(\mathbf{2}_t)$- (δ, α, β), *it does not necessarily imply that* Π_1 *is secure in* ω-$(\psi(\mathbf{3}_t)$-(δ, α, β). *i.e.* ω-$(\psi(\mathbf{2}_t)$-$(\delta, \alpha, \beta) \nrightarrow \Pi_1$ *is secure in* ω-$(\psi(\mathbf{3}_t)$-(δ, α, β).

iv. *Given that there exists a currency scheme* Π_1 *which is secure in* ω-$\psi(\mathbf{3}_t)$- (δ, α, β), *it does not necessarily imply that* Π_1 *is secure in* ω-$(\psi(\mathbf{4}_t)$-(δ, α, β). *i.e.* ω-$(\psi(\mathbf{3}_t)$-$(\delta, \alpha, \beta) \nrightarrow \Pi_1$ *is secure in* ω-$(\psi(\mathbf{4}_t)$-(δ, α, β)

v. *Given that there exists a currency scheme* Π_1 *which is secure in* ω-$\psi(\mathbf{4}_t)$- (δ, α, β), *it does not necessarily imply that* Π_1 *is secure in* ω-$(\psi(\mathbf{5}_t)$-(δ, α, β). *i.e.* ω-$(\psi(\mathbf{4}_t)$-$(\delta, \alpha, \beta) \nrightarrow \Pi_1$ *is secure in* ω-$(\psi(\mathbf{5}_t)$-(δ, α, β)

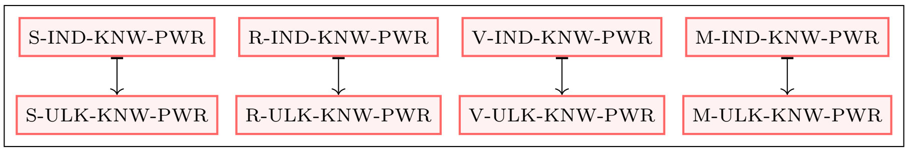
where $\omega = \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, \ 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\beta \in \{0, 1\}]$ *and* $\delta \in \{1, 2\}$*(Figure 22(c)).*

Proof. (Sketch)
This can be proven in a manner similar to Theorem 4, except that the special value leaks information about the transaction t_p or t_s in each scenario instead of sender addresses. □

It should be noted that in some cases the separations are not known to hold for all values of the unspecified parameters. Based on the above theorems, we also define the following corollaries.

Corollary 1. *(**Absolute Fungibility (ALL-IND-FULL-FULL)**) Given that a currency scheme* Π *is secure in the strongest anonymity notion (i.e. secure against the strongest possible adversary), then* Π *is also secure in any other notion (any other adversary).*
i.e. $(1_s 1_r 1_v 1_m)_\omega$-$((4, 4)_s, (4, 4)_r, 3_v, 3_m, 5_t)_\psi$-$(2_\delta, 3_\alpha, 1_\beta) \rightarrow \omega$-$\psi$-$(\delta, \alpha, \beta)$
where $\omega = \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, \ 3, 4\}$, $\psi_v, \psi_m \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}, \beta \in \{0, 1\}$ *and* $\delta \in \{1, 2\}$ *(Figure 21).*

Proof. (sketch) This follows from the above theorems as illustrated in figures 21 and 22 since this notion is the strongest among all. □



$$\begin{tikzcd}
\boxed{\text{S-IND-KNW-PWR}} \arrow[d, mapsto] & \boxed{\text{R-IND-KNW-PWR}} \arrow[d, mapsto] & \boxed{\text{V-IND-KNW-PWR}} \arrow[d, mapsto] & \boxed{\text{M-IND-KNW-PWR}} \arrow[d, mapsto] \\
\boxed{\text{S-ULK-KNW-PWR}} & \boxed{\text{R-ULK-KNW-PWR}} & \boxed{\text{V-ULK-KNW-PWR}} & \boxed{\text{M-ULK-KNW-PWR}}
\end{tikzcd}$$

Figure 23: Relations between indistinguishability and unlinkability (Corollary 2).

Corollary 2. ***(IND $\rightarrow$ ULK)*** *For a currency scheme Π and for a given entity, indistinguishability with respect to that entity implies unlinkability with all other parameters which are not linked to the entity are kept fixed. i.e.*

i. given Π is secure in S-IND-KNW-PWR for a given adversarial knowledge KNW of recipients, value and metadata and given adversarial power PWR, then Π is also secure in S-ULK-KNW-PWR.
i.e. S-IND-KNW-PWR $\rightarrow$ S-ULK-KNW-PWR

ii. given Π is secure in R-IND-KNW-PWR for a given adversarial knowledge KNW of senders, value and metadata and given adversarial power PWR, then Π is also secure in R-ULK-KNW-PWR.
i.e. R-IND-KNW-PWR $\rightarrow$ R-ULK-KNW-PWR

iii. given Π is secure in V-IND-KNW-PWR for a given adversarial knowledge KNW of senders, recipients and metadata and given adversarial power PWR, then Π is also secure in V-ULK-KNW-PWR.
i.e. V-IND-KNW-PWR $\rightarrow$ V-ULK-KNW-PWR

iv. given Π is secure in M-IND-KNW-PWR for a given adversarial knowledge KNW of senders, recipients and value and given adversarial power PWR, then Π is also secure in M-ULK-KNW-PWR.
i.e. M-IND-KNW-PWR $\rightarrow$ M-ULK-KNW-PWR
(Figure 23)

Proof. (sketch) (Part i) From the definitions of S-IND (definition 5.1) and S-ULK (definition 5.2), the difference between the two notions is that $(\psi_{s_{pk}}, \psi_{s_{sk}}) = (2, 0)$ in S-IND and it is $(0, 0)$ in S-ULK. Then from Theorem 1, it follows that $(2, 0)_s \rightarrow (0, 0)_s$ and hence the implication follows.

(Part ii) Similarly, follows from Theorem 1 with respect to recipients.

(Part iii) Follows from Theorem 12.

(Part iv) Follows from Theorem 14. □

Conversely, the weakest adversary is represented by the notion NIL-IND-NIL-NIL represented by the vector $(0000)_\omega$-$((0, 0)_s,\ (0, 0)_r, 0_v, 0_m, 0_t)_\psi$-$(0_\delta, 0_\alpha,\ 0_\beta)$ with all entities hidden (definition 5.16). Note that this notion is trivial in that no adversary can ever win the corresponding game since the transactions t_0 and t_1 are, aside from randomness, identical.

6 Discussion

In this work we have developed a comprehensive framework which depicts the generic functionality of a cryptocurrency scheme, irrespective of the underlying implementation. We have established the soundness of our model while ensuring the functional correctness and security against a wide range of adversaries. Our main contribution is to propose a unified means of analysing the true level of anonymity achieved by a cryptocurrency system in a qualitative manner.

The proposed anonymity model is centered around the idea of indistinguishability and it is elaborated around a group of entities; e.g. senders, recipients, values and metadata. A comprehensive adversarial model is defined encompassing different combinations of state initialisation methods and adversary power, and is capable of modelling anonymity at a much granular level, resulting in a vast number of different notions per each test case (defined by ω). While some notions may not carry a meaningful realisation in a real currency scheme, a majority leads to a multitude of attacker scenarios, which may not have been thought possible otherwise. One may wonder why we need such granularity in modelling anonymity in the context of cryptocurrencies, yet our analysis outcome shows how a minute change such as varying one value in a single variable in the Anonymity game, could drastically affect the level of anonymity of a cryptocurrency.

Building upon this model, we have provided formal definitions for a set of anonymity notions that demonstrate baseline anonymity notions in indistinguishability. Moreover, we also define *unlinkability*, a weaker notion of *indistinguishability*, in order to define an intermediary level of anonymity notions. However, even without formal definitions, other notions also play a significant role in performing a rigorous analysis of anonymity aspects of cryptocurrencies.

Taking a step further, we have attempted to understand the relationships and interdependencies among these myriad notions. Consequently, we have identified several implications, equivalences and separations which provide useful comparisons in the multi-dimensional adversarial parameter model. While some correlations are trivial, there are others that may depend on the underlying constructions, all of which demonstrates the sophistication of anonymity in the context of cryptocurrencies. We have demonstrated how such complexity is evident in real world cryptocurrency schemes in a separate work, which focuses on specific case studies based on a subset of the general framework described here.

In our attempt to grasp the anonymity landscape modelled by these notions, we see some familiar anonymity notions referenced in existing literature present similar interpretations to some notions in our model. As mentioned at the outset, the most widely referenced notion with respect to the anonymity of cryptocurrencies is the concept of unlinkability. In the context of Bitcoin, unlinkability is interpreted as linking addresses to transactions and to real world user identities [9, 16, 17, 30]. This interpretation closely relates to the sender and recipient unlinkability notions defined in our model, yet ours present a more detailed approach, where we can model very minor aspects of unlinkability by varying different parameters in our notions.

In the case of Monero, unlinkability refers to the inability of deducing whether two transactions were intended to the same recipient addresses, which is closely analogous to the recipient unlinkability notion in our model [11, 29]. On the other hand, the notion of traceability, which is widely discussed in the context of Monero, closely correlates to the sender unlinkability notion in our work, [11, 29]. However, our notions are defined with respect to several dimensions addressing a wider scope of adversarial capabilities.

Further, the notion of k-anonymity has been utilised in some studies to visualise the anonymity landscape in terms of a quantitative measure [29, 10]. Our work here is orthogonal to this in the sense that ours provide a means of qualitative analysis of anonymity.

In essence, the notions we propose herein are in relation to the entities within a currency scheme, with a much wider span of attacker scenarios which helps to analyse anonymity in minute detail. As noted earlier, this stu- dy does not investigate the privacy aspects of the underlying consensus mechanism or the network of a cryptocurrency scheme. Yet, these layers may leak information independently from the currency scheme in which case it may affect the achievable level of anonymity.

Therein, our work shows the very complex nature of the level of anonymity demonstrated by any currency scheme. It is hence evident that existing claims for anonymity of different cryptocurrency schemes might only be anonymous in some aspects. i.e. A currency scheme which is claimed to possess unlinkability, might not demonstrate unlinkability with respect to all entities. Hence, claims for anonymity cannot be made lightly in the presence of such granularity.

7 Conclusion

In this report, we have presented a common framework that can be used to evaluate the level of anonymity associated with different cryptocurrency schemes, regardless of the implementation method. We provide a single formal experiment to capture a plethora of distinct security and privacy properties that we identify, and attempt to draw connections to existing terminology.

Our model defines a rigorous set 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 a currency scheme.

Acknowledgements

Xavier Boyen is the recipient of an Australian Research Council Future Fellowship and acknowledges generous support from the grant, number FT140101145.

References

[1] Alsalami, N., Zhang, B.: SoK: A systematic study of anonymity in cryptocurrencies. In: 2019 IEEE Conference on Dependable and Secure Computing (DSC). pp. 1–9 (Nov 2019)

[2] 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)

[3] 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)

[4] 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)

[5] 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)

[6] Conti, M., Kumar, S., Lal, C., Ruj, S.: A survey on security and privacy issues of bitcoin. IEEE Communications Surveys & Tutorials (2018)

[7] 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)

[8] Fuchsbauer, G., Orrù, M., Seurin, Y.: Aggregate cash systems: A cryptographic investigation of mimblewimble. In: EUROCRYPT (2019)

[9] Kappos, G., Yousaf, H., Maller, M., Meiklejohn, S.: An empirical analysis of anonymity in zcash. CoRR **abs/1805.03180** (2018)

[10] 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)

[11] Kumar, A., Fischer, C., Tople, S., Saxena, P.: A traceability analysis of monero's blockchain. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security – ESORICS 2017. pp. 153–173. Springer International Publishing, Cham (2017)

[12] 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)

[13] Miller, A., Moeser, M., Lee, K., Narayanan, A.: An empirical analysis of linkability in the monero blockchain. arXiv preprint arXiv:1704.04299 (2017)

[14] Morris, L.: Anonymity Analysis of Cryptocurrencies. Ph.D. thesis, Rochester Institute of Techology (2015)

[15] 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. Proceedings on Privacy Enhancing Technologies **2018**(3), 143–163 (2018)

[16] Möser, M., Böhme, R.: Anonymous alone? measuring bitcoin's second-generation anonymization techniques. In: 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW). pp. 32–41 (April 2017)

[17] 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

[18] 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

[19] Poelstra, A.: Mimblewimble (2016)

[20] Quesnelle, J.: An Analysis of Anonymity in the Zcash Cryptocurrency. Master's thesis, University of Michigan-Dearborn (2018)

[21] Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Security and privacy in social networks, pp. 197–223. Springer (2013)

[22] Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In: Sadeghi, A.R. (ed.) Financial Cryptography and Data Security. pp. 6–24. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

[23] 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 International Publishing, Cham (2017)

[24] 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)

[25] Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002)

[26] 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)

[27] Van Saberhagen, N.: Cryptonote v 2. 0 (2013), `https://cryptonote.org/whitepaper.pdf`

[28] Wijaya, D.A., Liu, J., Steinfeld, R., Liu, D.: Monero ring attack: Recreating zero mixin transaction effect. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). pp. 1196–1201 (Aug 2018)

[29] Wijaya, D.A., Liu, J., Steinfeld, R., Liu, D., Yuen, T.H.: Anonymity reduction attacks to monero. In: Guo, F., Huang, X., Yung, M. (eds.) Information Security and Cryptology. pp. 86–100. Springer International Publishing, Cham (2019)

[30] Wijaya, D.A., Liu, J.K., Steinfeld, R., Sun, S.F., Huang, X.: Anonymizing bitcoin transaction. In: Bao, F., Chen, L., Deng, R.H., Wang, G. (eds.) Information Security Practice and Experience. pp. 271–283. Springer International Publishing, Cham (2016)