



Article

The Cryptographic Complexity of Anonymous Coins: A Systematic Exploration

Niluka Amarasinghe *, Xavier Boyen and Matthew McKague

School of Computer Science, Queensland University of Technology, Brisbane City, QLD 4000, Australia; xb@boyen.org (X.B.); matthew.mckague@qut.edu.au (M.M.)

* Correspondence: niluka.amarasinghe@qut.edu.au

Abstract: The modern financial world has seen a significant rise in the use of cryptocurrencies in recent years, partly due to the convincing lure 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 numerous 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, thereby enabling one to make meaningful comparisons irrespective of their implementation. Accordingly, our work lays the foundation for formalizing 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, as opposed to existing claims that are drawn upon at a higher level, thus missing out on the elemental factors underpinning anonymity.



Citation: Amarasinghe, N.; Boyen, X.; McKague, M. The Cryptographic Complexity of Anonymous Coins: A Systematic Exploration. *Cryptography* **2021**, *5*, 10. <https://doi.org/10.3390/cryptography5010010>

Received: 31 December 2020

Accepted: 18 February 2021

Published: 4 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: anonymity; security models; cryptocurrencies; anonymous coins; 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 [1,2].

Thanks to the blockchain technology, *trust*, the grease of financial transactions, can now be *inferential* rather than *axiomatic*. The decentralized 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* (By which we mean permissionless, fully decentralized, with democratic governance, and transparently operated—in other words, conducive to trust from first principles.) 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 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 [3] and approaches based on modal logic [4]. Some such studies present formalized terminologies to capture different aspects of anonymity [5], while some propose metrics that could be used to measure a quantitative

notion thereof, e.g., as a degree of anonymity [6]. 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 centralized 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 was 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, it 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, as transactions are linked to payment addresses in a big graph that is visible to all [7,8]. Detailed analyses of public bitcoin transaction data have shown that it is possible to uncover behavior patterns of Bitcoin users and trace their identities in real life [9–11].

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 deanonymization [12,13].

As rationalized in [14], despite a large number of studies around the topic of cryptocurrency anonymity, no standardized 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 in [7,8,14,15], which lifted the veil on the multiplicity of anonymity notions for cryptocurrencies, but stopped short of actually providing a crisp formalism for defining and using those notions.

Over the course of this study, we identified a very *fine-grained* structure for the intuitive notion of payment anonymity, parameterized through qualitative distinct definitions, which can be shown to be 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. Note 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 categorization 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 analyze 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 parameterized 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.

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.

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. [16] 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 analyzed in many different ways compromising the anonymity of users. Some have attempted to quantify such data as in [11], where behavioral patterns and transaction flows are studied at the user level. Meiklejohn et al. [17] present a different characterization 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. [18] 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 formalize 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. Androuraki et al. [10] 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, whereas profile indistinguishability refers to clustering of users based on addresses or transactions. This interpretation of unlinkability has been applied in several subsequent studies related to the anonymity of Bitcoin [19–21]. From these studies, it is apparent that Bitcoin anonymity cannot be defined at the transaction layer as addresses and transactions are linkable by the construction itself.

As Bitcoin receives much criticism to that effect, new currency schemes have emerged more promising anonymity expectations, which has led to the need for more concrete formalization of anonymity concepts. Zcash is one such scheme which supports two types of transactions: “shielded” and “unshielded”. Shielded transactions are encrypted, and therefore conceal 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* [22,23]. 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 [23].

Cryptonote is one of the protocols based on which several currency systems have been constructed with improved anonymity claims. Saberhagen [24], 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, then claimed to provide untraceable transactions and untraceable payments. However, these two properties have fallen to deanonymization attacks in many subsequent studies through the analysis of Monero transaction data [13,25,26].

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 [8,27]. Although it has been claimed in [23] that Zcash achieves fungibility through its use of zk-SNARK (zero-knowledge Succinct Non-Interactive Argument of Knowledge) proofs, the survey study in [8] makes the countermanding claim that Mimblewimble is the only cryptocurrency scheme to do so. Even so, the original Mimblewimble is insecure, and the fix proposed in [28], by making it preserve a lot more data, reintroduces the coin history removed in the original fungibility claim [29].

A recent work by Biryukov et al. [30] presents an experimental analysis on deanonymization 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. [31] proposed a formal model for blockchain systems by modeling 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 modeling anonymity based on the functionality of a currency scheme, as opposed to the underlying construction.

With this background, many have attempted to evaluate and compare the level of anonymity achieved by different cryptocurrencies through diverse means. Among the academic literature, the survey conducted by Khalilov et al. [7] presents a comprehensive categorization on a wide range of cryptocurrencies. They attempted to group the underlying constructions of these schemes around three aspects of anonymity: *untraceability*, *hidden*

values, and *hidden IP addresses*. A similar study was carried out by Conti et al. [8], which discusses the privacy aspects of Bitcoin and other cryptocurrencies as a comparison of advantages and disadvantages in terms of privacy and anonymity with respect to *unlinkability*, *untraceability*, *deniability*, and *fungibility*, yet without providing formal definitions. Further, the work in [14] presented a survey of several cryptocurrencies with respect to a set of qualitative anonymity properties such as *fungibility*, *unlinkability*, *untraceability*, *hidden values*, and *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. [15] 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: the ability to break links between transactions and hiding user identities. However, this categorization also, similar to the work in [7], 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. Table 1 compares the terminology used to model anonymity in above studies with our work.

Table 1. Comparison with similar work.

Paper/Study	Basis/Method	Type	Anonymity Notion/s	Applicable Cryptocurrencies
[16]	Experimental analysis	Quantitative	No formal notions	Bitcoin
[11]	Experimental analysis	Quantitative	No formal notions	Bitcoin
[17]	Experimental analysis	Quantitative	No formal notions	Bitcoin
[18]	Experimental analysis	Quantitative	No formal notions	Bitfodine
[10,19–21]	Quantitative analysis	Quantitative	Activity unlinkability and user profile indistinguishability	Bitcoin
[22,23]	Experimental analysis	Quantitative	Linkability	Zcash
[13,24–26]	Implementation techniques	Qualitative	Untraceability, unlinkability	Cryptonote
[8]	Based on distinct properties	Qualitative	Internal/external unlinkability, untraceability, fungibility, deniability (no formal definitions)	Bitcoin variants, CoinJoin and Cryptonote based variants
[30]	Network analysis	Quantitative	Anonymity degree	Bitcoin, Zcash
[31]	A transaction model based on blockchain semantics	Qualitative	No formal notions	Bitcoin, Ethereum and Hyperledger Fabric blockchains
[15]	Techniques used for anonymisation	Qualitative	Pseudonymity, Set anonymity, full anonymity, confidential transactions (No formal definitions)	Bitcoin and variants, Ethereum, Cryptonote based variants
[14]	Based on existing literature	Qualitative	Unlinkability, fungibility, untraceability, hidden values, unlinkability of IP addresses (No formal definitions)	Bitcoin and variants, Cryptonote based variants
[7]	Anonymity/privacy improvement methods	Qualitative	untraceability of input/output addresses and transactions, hidden values, hidden IP addresses (No formal definitions)	Bitcoin and variants, CryptoNote based, Mumblewimble
Our work	Anonymity framework based on a comprehensive adversarial model	Qualitative	Indistinguishability and Unlinkability of senders, recipients, value and metadata (formal definitions)	Any decentralised cryptocurrency scheme

1.3. Research Question

As evident from the above, there is no existing means for modeling anonymity uniformly across diverse constructions. Further, the presence of a wide range of notions suggests that anonymity depends on a number of factors. As a result, current claims of anonymity cannot be apprehended in a qualitative manner. Accordingly, we formulate our research question as follows.

How can we achieve a fine-grained systematization of anonymity modeling suitable for massively decentralized systems such as modern cryptocurrencies?

As such, we follow the research methodology below to devise a suitable means to achieve the above.

1.3.1. Research Methodology

Our research methodology is focused around constructing a common unifying mathematical framework, which is able to capture all the multiple security nuances in all existing and future schemes, following the accepted approach in cryptography which is to define security properties in terms of “games” or conceptual experiments. We choose game-based definitions over the Universal Composability (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.

The first step in this process is to define a generic cryptocurrency scheme which depicts the functionality of a typical decentralized currency scheme, while ensuring its correct and secure functionality. We do not consider the particulars of the underlying communication network or the consensus mechanism here as they may be unique to each scheme. Instead, we focus on the functionality which enables us to define a universal framework irrespective of the implementation-specific elements.

Based on this currency scheme, we then construct a comprehensive adversarial model, which encapsulates a wide range of attacker scenarios by investigating different security aspects. This lays the foundation for our conceptual framework, which is the essence of this work. Thereafter, we devise a game to model various settings affecting anonymity of a currency scheme, leading to a set of granular anonymity notions that are unique and precise, and are able to provide a unified framework for modeling anonymity.

1.3.2. Conceptual Framework

As already stated, this framework is formulated using game-based security modeling. It is a commonly accepted method in cryptography, which makes it easy to capture adversarial goals under well-defined adversarial capabilities. Each capability defines a unique attacker scenario, which is usually represented by a unique game (or a conceptual experiment). As in our case there are so many to consider, we came up with a methodology which involves defining a parametric experiment, which allows us to parameterize in terms of adversarial capabilities along multiple dimensions, each of which on a rising scale from no access to informational to manipulative to full control. Further details of this conceptual framework are discussed in Section 3 under Security.

1.4. Implication of the Research

As summarized in Table 1, the diversity of existing anonymity modeling methods highlights the inability to compare anonymity achieved by decentralized currency schemes. Conversely, our work uncovers the granularity of anonymity through a plethora of distinct definitions, each representing a different aspect of anonymity. While a multitude of separate definitions may seem absurdly excessive, we emphasize that these definitions arise naturally from considering the possible interactions between the adversary and the cryptocurrency. Indeed, our notions generalize 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. As a consequence, this work provides a means for modeling anonymity in a precise and qualitative manner.

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.

A 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. We use the notation given in Table 2 throughout the document in order to represent the scheme and its operations mathematically.

Table 2. 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	$p_0 \not\preceq p_1$
$p_0 \preceq p_1 \wedge p_0 \neq p_1$	$p_0 \prec p_1$
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	$ S $
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	$A B, A \setminus B$
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 $\langle statement \rangle$, then return 1	$\langle statement \rangle [condition]$
If $\langle condition \rangle$ is false after $\langle statement \rangle$, then return 0	$\langle statement \rangle \langle condition \rangle$
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	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}

2.1. A Generic Cryptocurrency Scheme

We define a generic cryptocurrency scheme as follows.

Definition 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}.

Functionality

Table 3 summarizes 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. The payment address creation process, *CreateAddr*, takes an 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.

Table 3. Functions.

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

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. The *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.

Note that we model only the generic functionality of a cryptocurrency scheme in this scheme. Therefore, 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 to realize the security implications of functionality of a scheme that may be outside our base model.

3. Correctness

The correctness of our model is established in terms of a set of experiments, which collectively ensure the expected functionality of the currency scheme.

3.1. Correctness Properties

We define several experiments in order to ensure the correctness of the currency scheme for all valid parameter values of λ and ρ (Table 4). Experiments corresponding to these properties are listed in Appendix A. Accordingly, the correctness of the proposed scheme is defined as follows.

Table 4. List of experiments for correctness.

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

Definition 2 (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 $X \in \{\text{init}, \text{create-addr}, \text{create-txn}, \text{extract-txn-data}, \text{mint}, \text{adjudicate}, \text{adj-monotonicity}, \text{create-checkpoint}, \text{retrieve-checkpoint}, \text{genesis-state}, \text{checkpoint-monotonicity}\}$, $\text{Exp}_{\pi}^X(\lambda, \rho)$ returns 1.

4. Security

We establish the security requirements for the proposed framework through a game-based approach. For this purpose, 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

In game-based security modeling, the first task is to establish the adversarial objective corresponding to each security property. Then, we identify the factors affecting the adversary's actions, which we term as adversarial capabilities. Both objectives and capabilities together form the variables in this model and are independent in their own right. Further, adversarial capabilities can be grouped in terms of what information the adversary knows (adversarial knowledge) and what actions the adversary can execute (adversarial power). Combining the objective and capabilities of the adversary, we formulate a conceptual

game, which mimics an instance of a given system between the adversary and a challenger, specifying the conditions on which the adversary can win the game. The adversary’s advantage of winning the game (or its success probability) is the factor that decides whether the system under study facing an adversary with the specific capabilities and objectives is secure. In this case, we can consider this advantage as the dependent variable in this framework. If the advantage is non-negligible, then we say that the system is not secure against that adversary.

In the context of cryptocurrencies, adversarial capabilities depend on a number of factors, which collectively results in a series of unique attacker scenarios. The adversary’s knowledge of public/secret keys of senders and recipients, value and metadata of a given transaction, and other transaction data correspond to adversarial knowledge. The ability to setup the initial state, to manipulate the state, and to cause minting to fail are the variables corresponding to adversarial power. Further, the adversarial objective can also be manifold, and we consider indistinguishability and unlinkability with respect to anonymity of senders, recipients, transactions etc. (we discuss these notions in detail in subsequent sections). Each adversarial goal, when combined with different capabilities, results in a large number of attacker scenarios. We construct a single parametric experiment to capture all possible adversarial capabilities, with each parameter combination resulting in a distinct security definition. This conceptualization is illustrated in Figure 1.

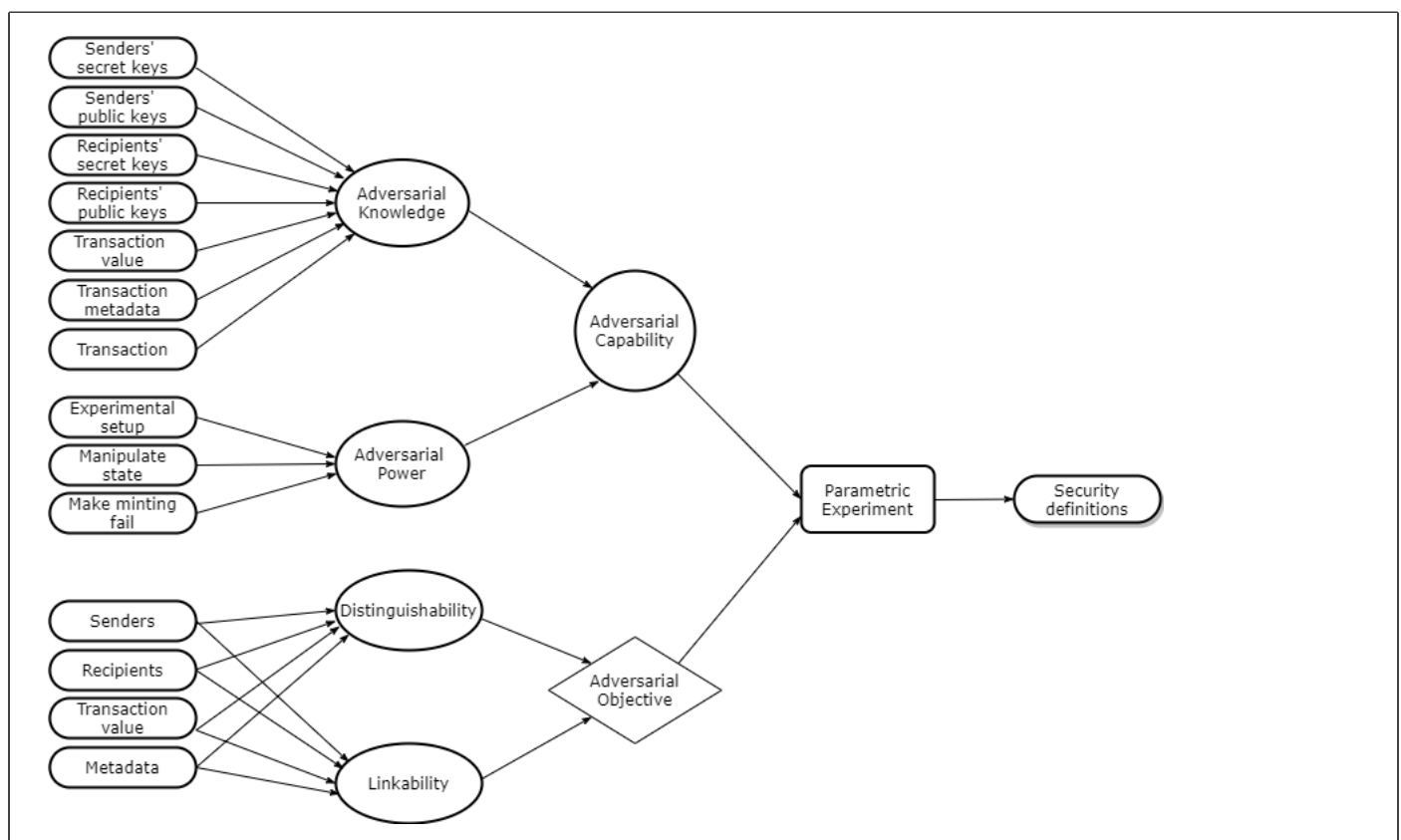


Figure 1. Conceptual diagram of the adversarial model.

We introduce a set of parameters to represent the above-stated variables. Adversary’s access to knowledge is parameterized by ψ , where access to specific information is represented by a subscript corresponding to entities, sender (ψ_s), recipient (ψ_r), value (ψ_v), and metadata (ψ_m). With regards to the adversarial power, parameter α represents how the state initialization in the experimental setup is handled. The parameter δ represents the ability to manipulate the state. In addition, the parameter β denotes whether the adversary

has the ability to cause minting to fail. Further, we introduce additional parameters later to represent adversarial goals to study anonymity considerations.

The adversary's access to knowledge ψ is modeled in terms of different levels of access to information, ranging from no access to informational, manipulative, and full access, respectively. These parameter values start from 0, and increasing values represent increasing level of access. 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 of the transaction 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 transactions 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 5.

In the case where $\alpha = 0$, the state initialization is performed honestly with hidden randomness. A value of 1 represents a scenario where an honest state initialization takes place with public randomness. For values $\alpha > 1$, the adversary has some control over the state initialization as listed in Table 5. With $\delta = 0$, the system state is hidden from the adversary. Similar to the state initialization, the adversary has control over the state when $\delta > 1$. The parameter β on the other hand, only takes two values 0 and 1 to say whether the adversary can cause minting to fail.

With this parameterization, we can capture a wide range of adversaries ranging from passive (with all parameters equal to zero) to static (with $\delta, \beta \leq 1$) and adaptive adversaries (with parameter values greater than 1).

4.2. 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 2). 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 ($\mathcal{O}_{hidMetadata}$). 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, whereas $T'_{\mathcal{O}}$ stores the randomness of the coins spent in transactions. 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 $\beta = 1$. The adversary has access to all available oracles, unless specifically mentioned with a specific subscript in the games. Table 6 summarizes 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. 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	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	-	-	-

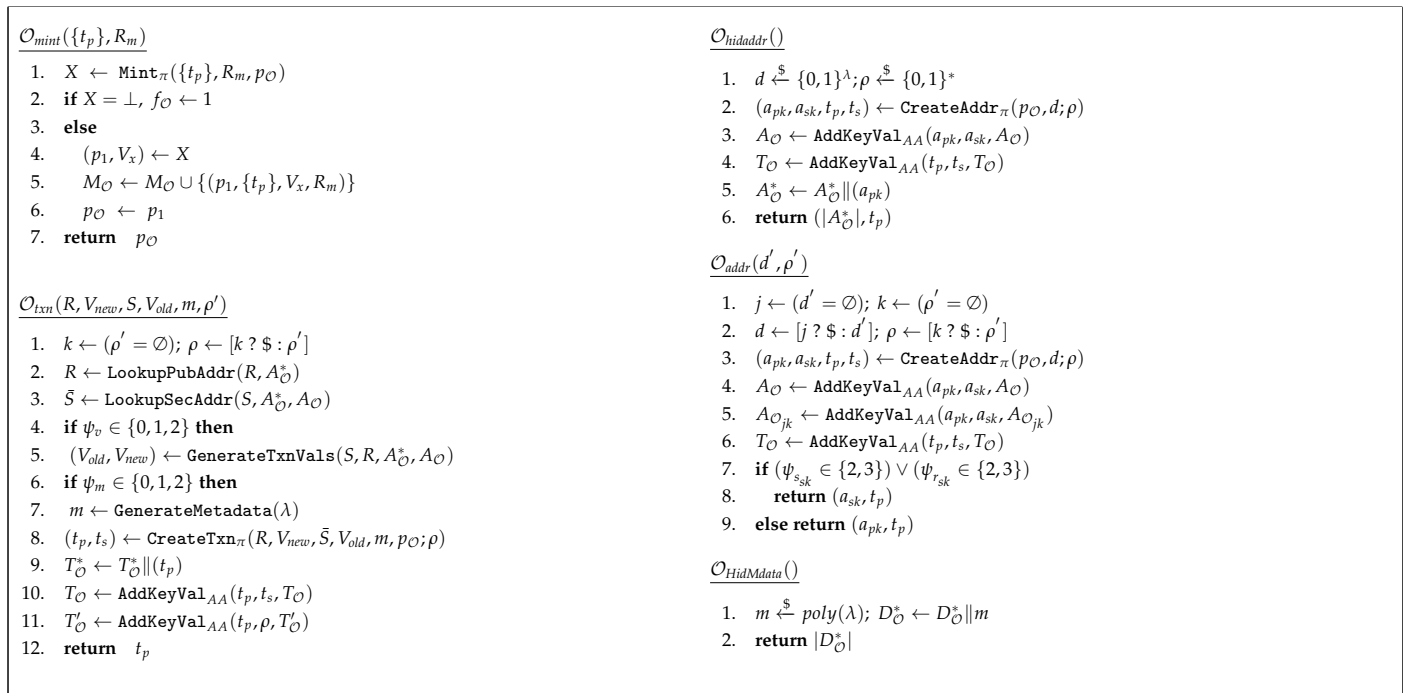


Figure 2. Oracle functions.

Table 6. 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

Additionally, we also define a set of helper functions to be used in the security games as given in Figure 3 to improve the clarity of the games. The SetupState function performs the state initialization based on INIT, whereas the RunAdversary function executes an instance of the adversary \mathcal{A} denoted by different subscripts based on ACT. LookupPubAddr and LookupSecAddr functions are used to obtain public keys and private keys from hidden addresses, respectively. 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, GenerateMetadata function generates metadata values required for a transaction when $\psi_m = 0$ while LookupMdata is used to obtain hidden metadata when $\psi_m \in \{0, 1\}$.

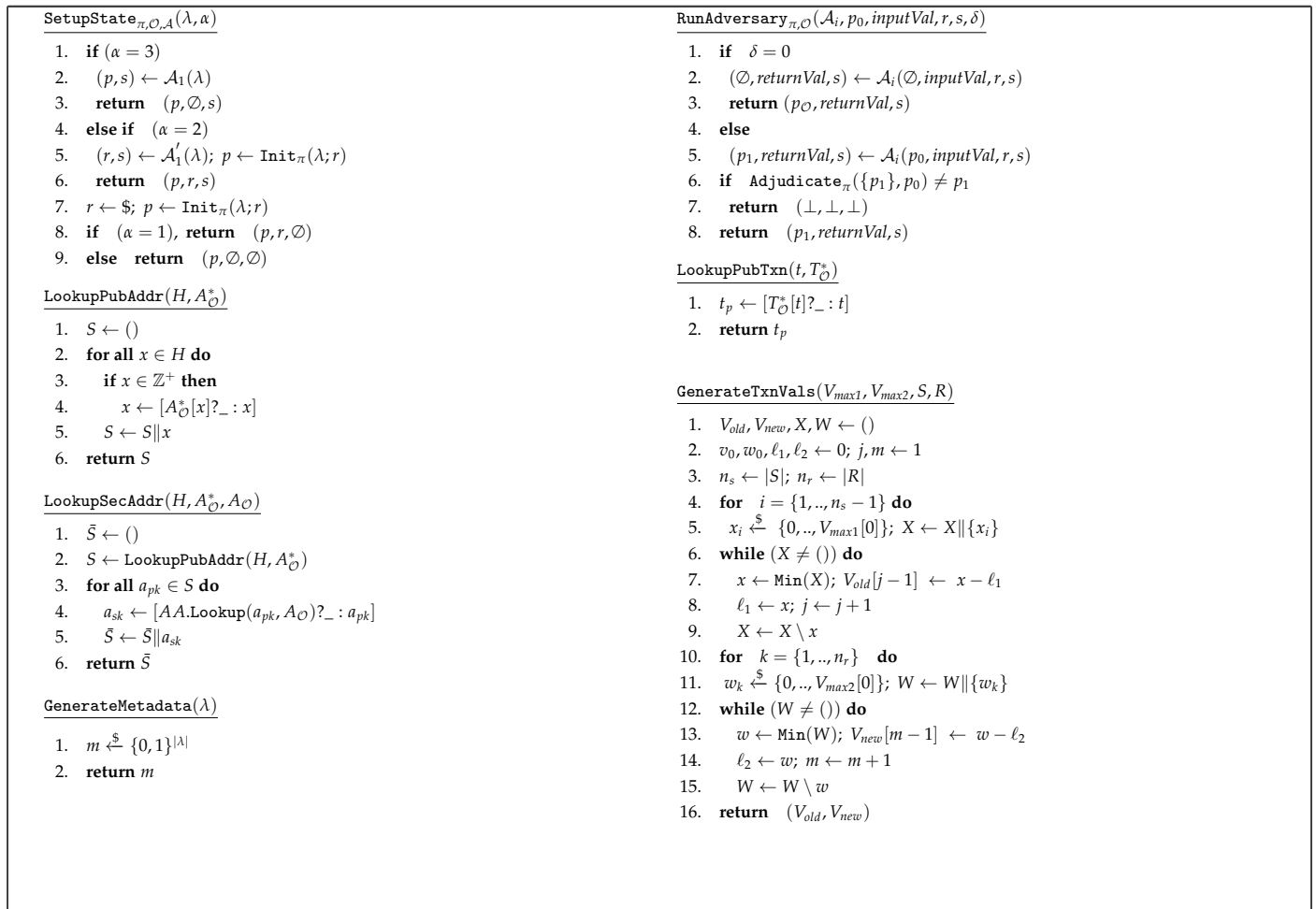


Figure 3. Helper functions.

4.3. Security Properties

First, we define a set of 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 *Anonymity*. Each property is demonstrated with respect to attacker’s goals and we construct appropriate games to model adversarial behavior explained earlier.

The *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. The *transaction binding* property establishes the requirement 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. *Spendability* guarantees that the funds associated with a payment address (a) cannot decrease unless the corresponding secret keys are known, i.e., $\text{Balance}_{aft}(a) < \text{Balance}_{bef}(a)$ only if secret key of a (a_{sk}) is known $\forall a$. The property of *Indemnification* requires that fund balances associated with the payment addresses that are not involved in a transaction should remain unchanged.

The *balance* 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 summarized for a set of transactions involved in one minting operation as follows:

$$\sum_{a \in \mathcal{S}} V_{old}(a) - \sum_{a \in \mathcal{R}} V_{new}(a) + \text{Minted units} = \sum_{a \in \mathcal{R}_m} V_x(a) \quad (1)$$

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

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

A single experiment is defined to capture all three goals, so that the balances cannot be manipulated by an adversary.

Positivity ensures that the fund balance corresponding to each payment address in the system is non-negative at all times. The *descendancy* property requires that an adversary should not be able to produce a valid system state, which does not descend from the genesis state.

We construct individual games (i.e., conceptual experiments) to represent all the above properties, and relevant details are included in Appendix B. Accordingly, we formally define the security of a currency scheme constructed as per Definition 1 in terms of those experiments as follows.

Definition 3 (Security of the currency scheme). For $Y \in \{\text{unforgeability, txn-binding, spendability, balance, indemnification, positivity, descendancy}\}$, a currency scheme Π is said to be $(\psi, \delta, \alpha, \beta)$ -secure with respect to Y if for every PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ and for all possible values of ψ, δ, α and β , the advantage of winning the security experiment $\text{Exp}_{\Pi, \mathcal{A}, \mathcal{O}, \psi, \delta, \alpha, \beta}^Y(\lambda)$ is negligible in $\lambda \in \mathbb{Z}^+$, i.e.,

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

where ε is a negligible function (\forall positive polynomials $p(\lambda)$, $\exists N$ such that $\forall \lambda > N$, $\varepsilon(\lambda) \leq 1/p(\lambda)$) in λ , $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$.

5. Anonymity

In this section, we demonstrate how we can model different aspects of anonymity of a currency scheme, in terms of the proposed framework. First, we formulate a parametric game to capture different 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 categorized 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 categorized as non-topological entities, without having a direct relationship to the transaction graph. We parameterize different scenarios where an attacker can manipulate these entities at various levels.

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 4). 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, based

on which the transaction data are varied. The adversarial inputs to the game are crafted based on ψ . We now define a common game to capture all possible attacker scenarios in this setting. Figure 5 illustrates the game.

```

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

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

Figure 4. Additional helper functions for the Anonymity game.

```

ExpAnonymity $\pi, A, \mathcal{O}, \omega, \psi, \delta, \alpha, \beta$ ( $\lambda$ )
1.  $A_{\mathcal{O}}, A_{\mathcal{O}_{11}}, A_{\mathcal{O}_{10}}, A_{\mathcal{O}_{00}}, T_{\mathcal{O}}, T'_{\mathcal{O}} \leftarrow AA.\text{Init}(); A_{\mathcal{O}}^*, T_{\mathcal{O}}^*, D_{\mathcal{O}}^* \leftarrow ()$ 
2.  $U \leftarrow \mathcal{O}; M_{\mathcal{O}} \leftarrow \{\}; f_{\mathcal{O}} \leftarrow 0$ 
3.  $(p_{\mathcal{O}}, r, s) \leftarrow \text{SetupState}_{\pi, \mathcal{O}, A}(\lambda, \alpha) \quad \langle p_{\mathcal{O}} \neq \perp \rangle$  ▷ State initialisation
4.  $(p_{\mathcal{O}}, (S_0, S_1, R_0, R_1, V_{old_0}, V_{new_0}, V_{old_1}, V_{new_1}, T, R_m, m_0, m_1, t_0, t_1, \rho_0, \rho_1, s) \leftarrow \text{RunAdversary}_{\pi, \mathcal{O}}(A_2, p_{\mathcal{O}}, (\mathcal{O}), r, s, \delta) \quad \langle p_{\mathcal{O}} \neq \perp \rangle$ 
5.  $(\omega_s, \omega_r, \omega_v, \omega_m) \leftarrow \omega;$ 
6.  $(\psi_{p_{ks}}, \psi_{s_{ks}}, \psi_{p_{kr}}, \psi_{s_{kr}}, \psi_v, \psi_m, \psi_t) \leftarrow \psi$ 
7. if  $\neg\{\text{CheckAdvConditions}(\omega, \psi, S_0, S_1, R_0, R_1, V_{old_0}, V_{new_0}, V_{old_1}, V_{new_1}, m_0, m_1, A_{\mathcal{O}}^*, A_{\mathcal{O}_{jk}}, D_{\mathcal{O}}^*)\}$  then ▷ Check adversarial conditions on inputs
8. return 0
9. if  $(\psi_t = 5)$  then
10.  $(t_{p_0}, t_{s_0}) \leftarrow t_0 \quad \langle \text{IsMintable}_{\pi}(\{t_{p_0}\} \cup T, p_{\mathcal{O}})^{\beta} \rangle$ 
11.  $(t_{p_1}, t_{s_1}) \leftarrow t_1 \quad \langle \text{IsMintable}_{\pi}(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\beta} \rangle$ 
12. else
13.  $t_{p_0} \leftarrow \mathcal{O}_{\text{Txn}}(R_0, V_{new_0}, S_0, V_{old_0}, m_0, \psi, p_{\mathcal{O}}, \rho_0) \quad \langle \text{IsMintable}_{\pi}(\{t_{p_0}\} \cup T, p_{\mathcal{O}})^{\beta} \rangle$ 
14.  $t_{p_1} \leftarrow \mathcal{O}_{\text{Txn}}(R_{\omega_r}, V_{new_{\omega_r}}, S_{\omega_s}, V_{old_{\omega_v}}, m_{\omega_m}, \psi, p_{\mathcal{O}}, \rho_1) \langle \text{IsMintable}_{\pi}(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\beta} \rangle$ 
15.  $b \stackrel{\$}{\leftarrow} \{0, 1\}$  ▷ Challenger picks a bit
16.  $(p_1, V_x) \leftarrow \text{Mint}_{\pi}(\{t_{p_b}\} \cup T, R_m, p_{\mathcal{O}})$ 
17.  $U \leftarrow \text{RevealData}(t_{p_b}, \psi, \omega, A_{\mathcal{O}}^*, T_{\mathcal{O}}^*, T_{\mathcal{O}}, p_1)$ 
18.  $(\cdot, b', \cdot) \leftarrow \text{RunAdversary}_{\pi, \mathcal{O}}(A_3, p_1, (U), r, s, \delta) \quad \langle \beta \vee (f_{\mathcal{O}} \neq 1) \rangle$ 
19. return  $b' \stackrel{?}{=} b$ 
    
```

Figure 5. Anonymity game.

Execution of the Game

The game is executed as follows. The state initialization 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 provides a set of data (to be used in creating transactions, minting, etc.) to the challenger, based on the values of ψ parameters with respect to senders, recipients, values, and metadata. These data 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 , together with the current state $p_{\mathcal{O}}$ (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., $\psi_{pk_s}, \psi_{sk_s}, \psi_{pk_r}, \psi_{sk_r} \in \{0, 1\}$. The adversary can make corresponding oracle calls to create hidden addresses, and opaque handles to those addresses are given in return. 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 ψ , the challenger checks whether the values returned by the adversary meet the expected criteria (through the `CheckAdvConditions` function) 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 " $\langle \text{IsMintable}_{\pi}(\{t_{p_1}\} \cup T, p_{\mathcal{O}})^{\bar{\beta}} \rangle$ " to represent this condition. In this case, when $\beta = 0, \bar{\beta} = 1$, and the game continues if `IsMintable()` = 1. When $\beta = 1, \bar{\beta} = 0$, and thus `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/she 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/her 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 linked 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 $\omega\text{-}\psi\text{-}(\delta, \alpha, \beta)$. The strongest adversary is modeled with the highest power (to manipulate the state initialization 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 emphasize the capabilities in power and knowledge specific to a given setting. Therefore, the highest level of anonymity modeled by the game is the notion ALL-IND-FULL-FULL and the weakest is the notion of NIL-IND-NIL-NIL. Accordingly, Table 7 lists some useful anonymity notions and their corresponding parameter vectors.

Table 7. Some useful anonymity notions.

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

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 [20]. 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 6).

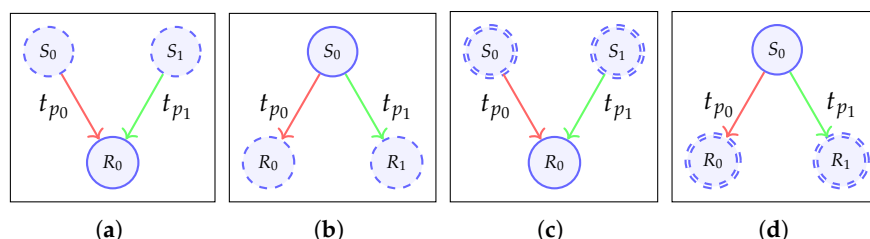


Figure 6. Topological anonymity notions. (Dashed outline: addresses with hidden secret keys, double-dashed outline: addresses with hidden public/private keys, Solid outline: both keys known). (a) Sender Indistinguishability (S-IND); (b) Recipient Indistinguishability (R-IND); (c) Sender Unlinkability (S-ULK); (d) Recipient Unlinkability (R-ULK).

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 6a 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 $\psi_{sk_s} = 0$ (PUBS knowledge) 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 if $KNW_t \neq 5$ (otherwise, the adversary provides the transactions). 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 initialization and the ability to manipulate the state, as well as with the highest knowledge of the transaction (i.e., $\psi_t = 5$). However, having $\beta = 1$ 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 and thus we consider an adversary with ACTIVE power (i.e., $\delta = 2, \alpha = 3, \beta = 0$). Accordingly, 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 4 (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.,

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

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 6c 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 (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.,

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

Recipient Indistinguishability (R-IND)

This notion is similar to sender indistinguishability, except with recipient addresses. Therefore, 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 6b, 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 have 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 6 (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.,

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

Recipient Unlinkability (R-ULK)

This property is referred to as the inability to link a transaction to the correct recipient. Figure 6d 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., $\psi_{pk_r}, \psi_{sk_r} = 0$. 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 7 (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 $(0_s 1_r 0_v 0_m)_{\omega} - ((4, 4)_s, (0, 0)_r, 3_v, 3_m, 5_t)_{\psi} - (2_{\delta}, 3_{\alpha}, 0_{\beta})$ is negligible.

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

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 could also hinder the privacy of users. Therefore, 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 7).

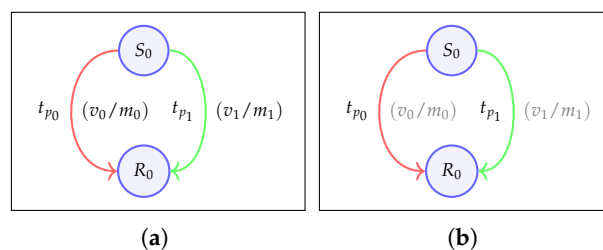


Figure 7. Non-topological Anonymity notions. (a) Value/Metadata indistinguishability. (b) Value/Metadata unlinkability (hidden values/metadata).

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 7a). 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. Therefore, the strongest notion in this scenario is given by V-IND-PUBSR-ACTIVE, which is represented by the vector $((0_s 0_r 1_v 0_m)_{\omega} - ((3, 0)_s, (3, 0)_r, 3_v, 3_m, 5_t)_{\psi} - (2_{\delta}, 3_{\alpha}, 0_{\beta}))$ with the following formal definition.

Definition 8 (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 $((0_s 0_r 1_v 0_m)_{\omega} - ((3, 0)_s, (3, 0)_r, 3_v, 3_m, 5_t)_{\psi} - (2_{\delta}, 3_{\alpha}, 0_{\beta}))$ is negligible, i.e.,

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

Value Unlinkability (V-ULK)

We define the property of unlinkability related to transaction value as the inability to correctly identify value of the minted transaction from two possible hidden values. In order to realize 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 3). 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 7b, 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 9 (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.,

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

Metadata Indistinguishability (M-IND)

Other transaction-related data such as scripts, IP addresses, etc. also pose a risk to anonymity as 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 modeling the effects imposed by the other layers of implementations such as the consensus scheme. Therefore, 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 $((0_s 0_r 0_v 1_m)_{\omega} - ((4, 4)_s, (4, 4)_r, 3_v, 2_m, 5_t)_{\psi} - (2_{\delta}, 3_{\alpha}, 0_{\beta}))$ as per the notion M-IND-PUBM-ACTIVE and it is formally defined below.

Definition 10 (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.,

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

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 3). Accordingly, we have the corresponding notion M-ULK-NILM-ACTIVE parameterized 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.

Definition 11 (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.,

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

5.2.3. Other Useful Anonymity Notions

Further to above notions, we also formally define the strongest and weakest anonymity notions modeled 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 8a). The strongest adversary in this scenario has the FULL knowledge and FULL power given by ALL-IND-FULL-FULL notion and parameterized 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.

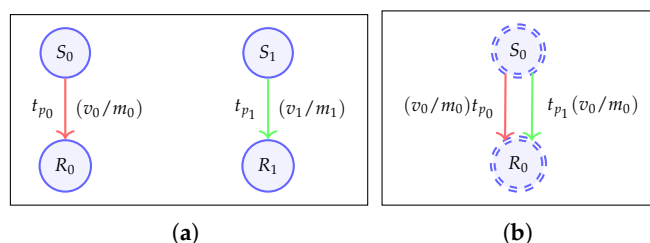


Figure 8. Strongest and weakest anonymity games. (a) ALL-IND game; (b) NILL-IND game.

Definition 12 (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.,

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

Weakest Anonymity (NIL-IND)

Here, we consider the weakest adversary that can be modeled in our game. In this case, the game produces two identical transactions as opposed to the strongest scenario above (Figure 8b). These transactions differ only in their randomness and the adversary has to identify the correct transaction. Therefore, 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 $\delta = 0$, 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 formalization for comparison.

Definition 13 (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 $(0_s 0_r 0_v 0_m)_{\omega} - ((0, 0)_s, (0, 0)_r, 0_v, 0_m, 0_t)_{\psi} - (0_{\delta}, 0_{\alpha}, 0_{\beta})$ is negligible, i.e.,

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

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 set $\delta = 1$ in our model for most schemes. This will model an adversary with VIEW power with other parameters being zero. Therefore, we define a slightly less weaker notion in this sense, which can be useful to model anonymity in some real world constructions.

Definition 14 (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.,

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

5.2.4. Representation of Anonymity Notions

In order to clearly represent above anonymity notions, we construct graphical illustrations as shown in Figures 9 and 10. These diagrams are useful in comparing anonymity landscape across different implementations while illustrating the complex multidimensional diversity of adversarial parameters.

Figure 9 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 10, which represents two more anonymity notions related to S-IND and S-ULK corresponding to Definitions 4 and 5, and shows that S-IND is stronger than S-ULK.

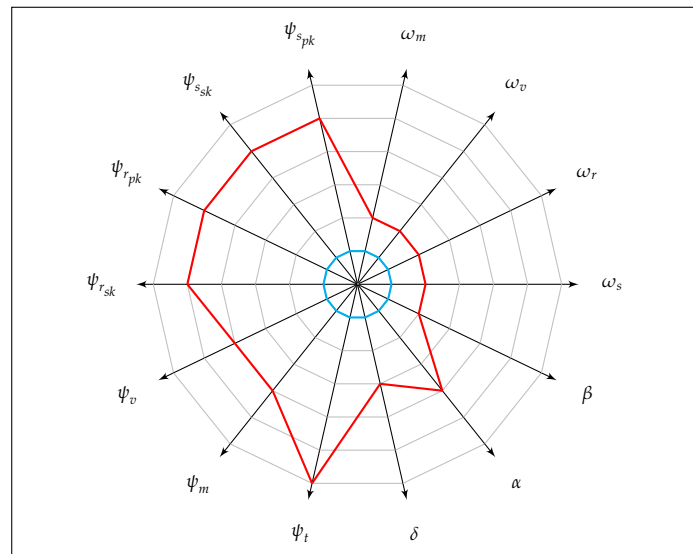


Figure 9. Strongest anonymity notion: ALL-IND-FULL-FULL (red). Weakest notion: NIL-IND-NIL-NIL (cyan).

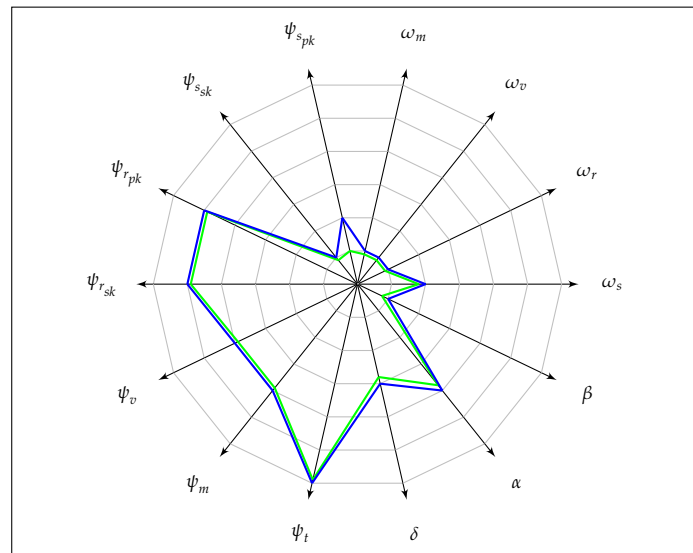


Figure 10. Anonymity notions S-IND-PUBS-ACTIVE (green) and S-ULK-NILS-ACTIVE (blue).

5.3. Theorems

As is apparent from the definitions presented in the previous section, we can utilize the Anonymity game to realize 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.

Note that as we vary different security parameters in our model, their correlations result in a non-trivial lattice form as depicted in Figures 11 and 12. These relations are interpreted as implications, equivalences, and separations. The arrow “ \rightarrow ” represents an implication in the direction of the arrow and a separation in the opposite direction, whereas the double arrow “ \leftrightarrow ” shows an equivalence relation. In order to formalize these relationships, we define a set of theorems that will simplify the process of assessing the anonymity of a currency scheme and we present the same below. Relevant proofs of these theorems are available in Appendix C.

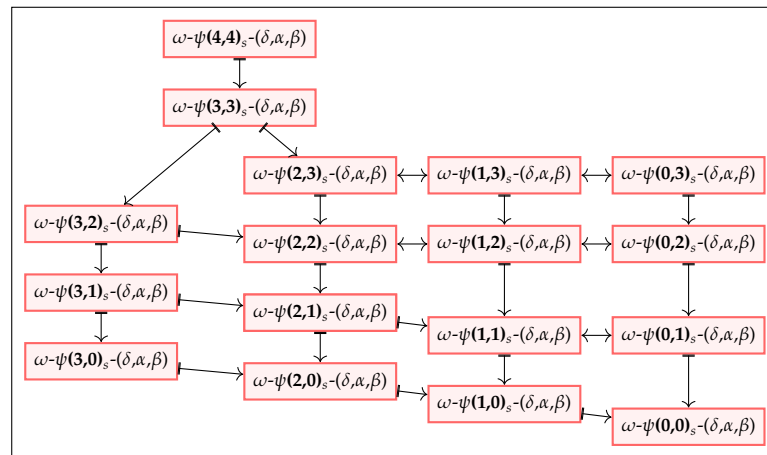


Figure 11. Relationship of anonymity notions for different sender addresses $(\psi_{pk}, \psi_{sk})_s$.



Figure 12. Relationships among notions based on $\alpha, \delta, \psi_v, \psi_m, \psi_t$, and β .

Theorem 1. For a currency scheme Π and for a given combination of $\omega, \delta, \alpha, \psi_{sk_s}, (\psi_{pk}, \psi_{sk})_r, \psi_v, \psi_m, \psi_t$, 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 $\omega-\psi((3,0)_s)-(\delta, \alpha, \beta)$, Π is also secure in $\omega-\psi((2,0)_s)-(\delta, \alpha, \beta)$, $\omega-\psi((1,0)_s)-(\delta, \alpha, \beta)$ and $\omega-\psi((0,0)_s)-(\delta, \alpha, \beta)$,
i.e., $\omega-\psi((3,0)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((2,0)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((1,0)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((0,0)_s)-(\delta, \alpha, \beta)$
- ii. given that Π is secure in $\omega-\psi((3,1)_s)-(\delta, \alpha, \beta)$, Π is also secure in $\omega-\psi((2,1)_s)-(\delta, \alpha, \beta)$ and $\omega-\psi((1,1)_s)-(\delta, \alpha, \beta)$,
i.e., $\omega-\psi((3,1)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((2,1)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((1,1)_s)-(\delta, \alpha, \beta)$
- iii. given that Π is secure in $\omega-\psi((3,2)_s)-(\delta, \alpha, \beta)$, Π is also secure in $\omega-\psi((2,2)_s)-(\delta, \alpha, \beta)$,
i.e., $\omega-\psi((3,2)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((2,2)_s)-(\delta, \alpha, \beta)$
- iv. given that Π is secure in $\omega-\psi((3,3)_s)-(\delta, \alpha, \beta)$, Π is also secure in $\omega-\psi((2,3)_s)-(\delta, \alpha, \beta)$,
i.e., $\omega-\psi((3,3)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((2,3)_s)-(\delta, \alpha, \beta)$
- v. given that Π is secure in $\omega-\psi((4,4)_s)-(\delta, \alpha, \beta)$, Π is also secure in $\omega-\psi((3,3)_s)-(\delta, \alpha, \beta)$,
i.e., $\omega-\psi((4,4)_s)-(\delta, \alpha, \beta) \rightarrow \omega-\psi((3,3)_s)-(\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk_r}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 11).

Theorem 2. For a currency scheme Π and for a given combination of $\omega, \delta, \alpha, (\psi_{pk_r}, \psi_{sk_r})_r, \psi_v, \psi_m, \psi_t$, 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 $\omega\text{-}\psi((0, 3)_s)\text{-}(\delta, \alpha, \beta)$, Π is also secure in $\omega\text{-}\psi((0, 2)_s)\text{-}(\delta, \alpha, \beta)$, $\omega\text{-}\psi((0, 1)_s)\text{-}(\delta, \alpha, \beta)$ and $\omega\text{-}\psi((0, 0)_s)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi((0, 3)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((0, 2)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((0, 1)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((0, 0)_s)\text{-}(\delta, \alpha, \beta)$
- ii. given that Π is secure in $\omega\text{-}\psi((1, 3)_s)\text{-}(\delta, \alpha, \beta)$, Π is also secure in $\omega\text{-}\psi((1, 2)_s)\text{-}(\delta, \alpha, \beta)$ and $\omega\text{-}\psi((1, 0)_s)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi((1, 3)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((1, 2)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((1, 0)_s)\text{-}(\delta, \alpha, \beta)$
- iii. given that Π is secure in $\omega\text{-}\psi((2, 3)_s)\text{-}(\delta, \alpha, \beta)$, Π is also secure in $\omega\text{-}\psi((2, 2)_s)\text{-}(\delta, \alpha, \beta)$, $\omega\text{-}\psi((2, 1)_s)\text{-}(\delta, \alpha, \beta)$ and $\omega\text{-}\psi((2, 0)_s)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi((2, 3)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((2, 2)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((2, 1)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((2, 0)_s)\text{-}(\delta, \alpha, \beta)$
- iv. given that Π is secure in $\omega\text{-}\psi((3, 3)_s)\text{-}(\delta, \alpha, \beta)$, Π is also secure in $\omega\text{-}\psi((3, 2)_s)\text{-}(\delta, \alpha, \beta)$, $\omega\text{-}\psi((3, 1)_s)\text{-}(\delta, \alpha, \beta)$ and $\omega\text{-}\psi((3, 0)_s)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi((3, 3)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((3, 2)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((3, 1)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((3, 0)_s)\text{-}(\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk_r}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 11).

Theorem 3. For a currency scheme Π and for a given combination of $\omega, \delta, \alpha, (\psi_{pk_r}, \psi_{sk_r})_r, \psi_v, \psi_m, \psi_t$, and β , the resulting notion from increasing the value of ψ_{sk_s} 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 $\omega\text{-}\psi((0, 1)_s)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi((1, 1)_s)\text{-}(\delta, \alpha, \beta)$, and vice versa.,
i.e., $\omega\text{-}\psi((0, 1)_s)\text{-}(\delta, \alpha, \beta) \leftrightarrow \omega\text{-}\psi((1, 1)_s)\text{-}(\delta, \alpha, \beta)$
- ii. given that Π is secure in $\omega\text{-}\psi((0, 2)_s)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi((1, 2)_s)\text{-}(\delta, \alpha, \beta)$ and $\omega\text{-}\psi((2, 2)_s)\text{-}(\delta, \alpha, \beta)$, and vice versa,
i.e., $\omega\text{-}\psi((0, 2)_s)\text{-}(\delta, \alpha, \beta) \leftrightarrow \omega\text{-}\psi((1, 2)_s)\text{-}(\delta, \alpha, \beta) \leftrightarrow \omega\text{-}\psi((2, 2)_s)\text{-}(\delta, \alpha, \beta)$
- iii. given that Π is secure in $\omega\text{-}\psi((0, 3)_s)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi((1, 3)_s)\text{-}(\delta, \alpha, \beta)$ and $\omega\text{-}\psi((2, 3)_s)\text{-}(\delta, \alpha, \beta)$, and vice versa,
i.e., $\omega\text{-}\psi((0, 3)_s)\text{-}(\delta, \alpha, \beta) \leftrightarrow \omega\text{-}\psi((1, 3)_s)\text{-}(\delta, \alpha, \beta) \leftrightarrow \omega\text{-}\psi((2, 3)_s)\text{-}(\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk_r}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 11).

Theorem 4. For a currency scheme Π for a given combination of $\omega, \delta, \alpha, (\psi_{pk_r}, \psi_{sk_r})_r, \psi_v, \psi_m, \psi_t$, and β (with $\psi_t = 1$), the notion resulting from decreasing the value of ψ_{sk_s} while holding others is not necessarily stronger than the former for the following scenarios:

- i. $\omega\text{-}\psi((0, 0)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((1, 0)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((2, 0)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((3, 0)_s)\text{-}(\delta, \alpha, \beta)$
- ii. $\omega\text{-}\psi((1, 1)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((2, 1)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((3, 1)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow$
- iii. $\omega\text{-}\psi((2, 2)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((3, 2)_s)\text{-}(\delta, \alpha, \beta)$
- iv. $\omega\text{-}\psi((2, 3)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((3, 3)_s)\text{-}(\delta, \alpha, \beta)$
- v. $\omega\text{-}\psi((3, 3)_s)\text{-}(\delta, \alpha, \beta) \nrightarrow \omega\text{-}\psi((4, 4)_s)\text{-}(\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk_r}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t = 1$, $\beta \in \{0, 1\}$, and $\delta \in \{0, 1, 2\}$ (Figure 11).

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 ψ_{sk} while holding others is not necessarily stronger than the former for the following scenarios:

- i. $\omega\text{-}\psi((\mathbf{2}, \mathbf{0})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{2}, \mathbf{1})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{2}, \mathbf{2})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{2}, \mathbf{3})_s)\text{-}(\delta, \alpha, \beta)$
- ii. $\omega\text{-}\psi((\mathbf{3}, \mathbf{0})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{3}, \mathbf{1})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{3}, \mathbf{2})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{3}, \mathbf{3})_s)\text{-}(\delta, \alpha, \beta)$
- iii. $\omega\text{-}\psi((\mathbf{1}, \mathbf{0})_s)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi((\mathbf{1}, \mathbf{1})_s)\text{-}(\delta, \alpha, \beta)$

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 11).

Note that the Theorems 1–4 also hold for recipient addresses in the same manner.

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 $\omega\text{-}\psi\text{-}(\delta, \mathbf{3}_\alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \mathbf{3}_\alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$
- ii. given Π is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$
- iii. given Π is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{0}_\alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\delta, \mathbf{0}_\alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 12d).

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 $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$, then there exists a currency scheme Π which is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{3}_\alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, \mathbf{3}_\alpha, \beta)$
- ii. Given that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$, then there exists a currency scheme Π which is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, \mathbf{2}_\alpha, \beta)$
- iii. Given that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{0}_\alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{0}_\alpha, \beta)$, then there exists a currency scheme Π which is secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{0}_\alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \mathbf{0}_\alpha, \beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, \mathbf{1}_\alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 12d).

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 $\omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta)$ then Π is also secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$
- ii. given Π is secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ then Π is also secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk_r} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 12a).

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 exist such that Π_1 is secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$, then there exists a currency scheme Π which is secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)\text{-}\beta \not\rightarrow \omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta)$
- ii. given that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$, there exists a currency scheme Π which is secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 12a).

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

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 12a).

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 $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{0}_\beta)$ then Π is not necessarily secure in $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{1}_\beta)$,
i.e., $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{0}_\beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{1}_\beta)$

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 12a).

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 $\omega\text{-}\psi\text{-}(\mathbf{3}_v)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi\text{-}(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{3}_v)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$
- ii. given Π is secure in $\omega\text{-}\psi\text{-}(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_v)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\mathbf{1}_v)\text{-}(\delta, \alpha, \beta)$
- iii. given Π is secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_v)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_v)\text{-}(\delta, \alpha, \beta)$,
i.e., $\omega\text{-}\psi\text{-}(\mathbf{1}_v)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\mathbf{0}_v)\text{-}(\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 12e).

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

secure in $\omega\text{-}\psi(\mathbf{1}_v)\text{-}(\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{1}_v)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$
 iii. given that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$
 and Π_0 is not secure in $\omega\text{-}\psi(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$, then there exists a currency scheme Π which is
 secure in $\omega\text{-}\psi(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi(\mathbf{3}_v)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{2}_v)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi(\mathbf{3}_v)\text{-}(\delta, \alpha, \beta)$
 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 12e).

Theorem 14. For a currency scheme Π and for a given combination of $\omega, \delta, \alpha, (\psi_{pk}, \psi_{sk})_s, \psi_v, \psi_t$, 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 $\omega\text{-}\psi(\mathbf{3}_m)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{3}_m)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$
- ii. given Π is secure in $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$
- iii. given Π is secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{0}_m)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{0}_m)\text{-}(\delta, \alpha, \beta)$

where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \gamma \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 12f).

Theorem 15. For a currency scheme Π and for a given combination of $\omega, \delta, \alpha, (\psi_{pk}, \psi_{sk})_s, \psi_m, \psi_t$, 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 $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$
 and Π_0 is not secure in $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$, then there exists a currency scheme Π which is
 secure in $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi(\mathbf{3}_m)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi(\mathbf{3}_m)\text{-}(\delta, \alpha, \beta)$
- ii. given that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$
 and Π_0 is not secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$, then there exists a currency scheme Π which is
 secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi(\mathbf{2}_m)\text{-}(\delta, \alpha, \beta)$
- iii. given that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi(\mathbf{0}_m)\text{-}(\delta, \alpha, \beta)$
 and Π_0 is not secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$, then there exists a currency scheme Π which is
 secure in $\omega\text{-}\psi(\mathbf{0}_m)\text{-}(\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{0}_m)\text{-}(\delta, \alpha, \beta) \not\rightarrow \omega\text{-}\psi(\mathbf{1}_m)\text{-}(\delta, \alpha, \beta)$

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 12f).

Theorem 16. For a currency scheme Π and for a given combination of $\omega, \delta, \alpha, (\psi_{pk}, \psi_{sk})_s, (\psi_{pk}, \psi_{sk})_r, \psi_v, \psi_m$, 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 $\omega\text{-}\psi(\mathbf{5}_t)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{4}_t)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{5}_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{4}_t)\text{-}(\delta, \alpha, \beta)$
- ii. given Π is secure in $\omega\text{-}\psi(\mathbf{4}_t)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{3}_t)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{4}_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{3}_t)\text{-}(\delta, \alpha, \beta)$
- iii. given Π is secure in $\omega\text{-}\psi(\mathbf{3}_t)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{2}_t)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{3}_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{2}_t)\text{-}(\delta, \alpha, \beta)$
- iv. given Π is secure in $\omega\text{-}\psi(\mathbf{2}_t)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{1}_t)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{2}_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{1}_t)\text{-}(\delta, \alpha, \beta)$
- v. given Π is secure in $\omega\text{-}\psi(\mathbf{1}_t)\text{-}(\delta, \alpha, \beta)$, then Π is also secure in $\omega\text{-}\psi(\mathbf{0}_t)\text{-}(\delta, \alpha, \beta)$,
 i.e., $\omega\text{-}\psi(\mathbf{1}_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(\mathbf{0}_t)\text{-}(\delta, \alpha, \beta)$

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 12b).

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 - $(\psi(\mathbf{0}_t))$ - $(\delta, \alpha, \beta) \rightarrow \Pi_1$ 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 - $(\psi(\mathbf{1}_t))$ - $(\delta, \alpha, \beta) \rightarrow \Pi_1$ 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 - $(\psi(\mathbf{2}_t))$ - $(\delta, \alpha, \beta) \rightarrow \Pi_1$ 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 - $(\psi(\mathbf{3}_t))$ - $(\delta, \alpha, \beta) \rightarrow \Pi_1$ 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 - $(\psi(\mathbf{4}_t))$ - $(\delta, \alpha, \beta) \rightarrow \Pi_1$ 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 12c).

Note 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., $(1111)_{\omega} - ((4, 4), (4, 4), 3, 3, 5)_{\psi} - (2_{\delta}, 3_{\alpha}, 1_{\beta}) \rightarrow \omega - ((\psi_{pk}, \psi_{sk})_s, (\psi_{pk}, \psi_{sk})_r, \psi_v, \psi_m, \psi_t) - \delta - \alpha - \beta$
where $\omega \in \{1, 0\}^4$, $\psi_{pk}, \psi_{sk} \in \{0, 1, 2, 3, 4\}$, $\psi_v, \psi_m, \alpha \in \{0, 1, 2, 3\}$, $\psi_t \in \{0, 1, 2, 3, 4, 5\}$, $\beta \in \{0, 1\}$ and $\delta \in \{0, 1, 2\}$ (Figure 11).

Proof. (sketch) This follows from the above theorems as illustrated in Figures 11 and 12 as this notion is the strongest among all. \square

Corollary 2 (IND \rightarrow ULK). For a currency scheme Π ,

- 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; and
- 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

Proof. (sketch) (Part i) From the definitions of S-IND (Definition 4) and S-ULK (Definition 5), 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 (Figure 13).

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

(Part iii) Follows from Theorem 12.

(Part iv) Follows from Theorem 14. \square

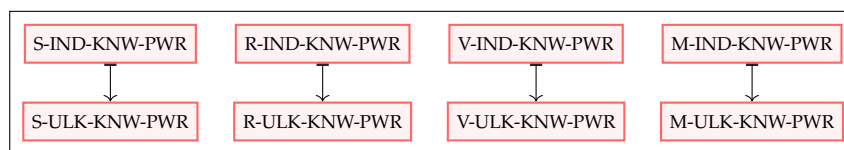


Figure 13. Relations between indistinguishability and unlinkability (Corollary 2).

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 13). Note that this notion is trivial in that no adversary can ever win the corresponding game as the transactions t_0 and t_1 are, aside from randomness, identical.

6. Discussion

Our objective in this work was to find a solution to our research question, i.e., to formulate a method to achieve a fine-grained systematization of anonymity modeling suitable for decentralized systems such as modern cryptocurrencies. To this effect, 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. Through this model, we have constructed a unified means of analyzing the true level of anonymity achieved by any fully decentralized cryptocurrency system in a qualitative manner.

Our proposed anonymity model is centered around the idea of indistinguishability and it is elaborated around a group of entities related to cryptocurrency transactions, e.g., senders, recipients, values, and metadata. To this effect, we defined a comprehensive adversarial model encompassing a wide range of capabilities including knowledge and power. Adversarial knowledge is determined based on a range of access levels to the knowledge of secret/public keys of senders/recipients, transaction values, metadata, and transaction records. Power is defined based on different combinations of state initialization methods, ability to manipulate the state, and to cause minting to fail. These capabilities collectively represent an exhaustive adversarial model, which is capable of modeling 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 realization 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 modeling anonymity in the context of cryptocurrencies, yet our model 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. Thus, we have been able to achieve a fine-grained model as expected.

Building upon this model, we have provided formal definitions for a subset of anonymity notions that demonstrate baseline anonymity notions in indistinguishability, which is the fundamental property of anonymity. Inability to distinguish between two probable (known) entities related to a minted transaction under various adversarial settings was considered

as the basis for these notions. Moreover, we also defined *unlinkability*, a weaker notion of *indistinguishability*, in order to define an intermediary level of anonymity. In this case, we considered the indistinguishability between two entities that are unknown to the adversary in a given scenario. However, even without formal definitions, other notions also play a significant role in performing a rigorous analysis of anonymity aspects of cryptocurrencies. As such, the anonymity notions resulting from this framework provide a universal systematization of anonymity across different implementations.

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 multidimensional adversarial parameter model. While some correlations are trivial, there are others that may depend on the underlying constructions, all of which demonstrate the sophistication of anonymity in the context of cryptocurrencies. To this effect, we have formulated a set of theorems which is useful in investigating anonymity in decentralized cryptocurrency schemes and thus this formalization adds value to our framework. 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 [32].

In our attempt to grasp the anonymity landscape modeled by these notions, we saw that 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 [19–22]. Although this interpretation does not differentiate between senders and recipients, it closely relates to the sender and recipient unlinkability notions defined in our model. Conversely, ours present a more detailed conceptualization, where we model the level of knowledge of the payment addresses by varying different parameters in our notions, thereby representing many different variants of unlinkability in our notions. This clearly shows how our model is capable of covering more ground compared to very high-level notions discussed in the existing literature such as [22].

In the case of Monero, the unlinkability property has been discussed as 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 [25,33]. 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 [25,33]. In contrast, however, our notions are defined with respect to several dimensions addressing a wider scope of adversarial capabilities, and therefore capture even minor deviations in anonymity in this respect.

Further, the notion of k -anonymity has been utilized in some studies to visualize the anonymity landscape in terms of a quantitative measure where the parameter k denotes the size of the anonymity set (i.e., number of entities in a group) [7,25]. Our work here is orthogonal to this in the sense that ours provide a means for 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 analyze anonymity in minute detail. As noted earlier, this study 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, and this may be a future direction of study in this context.

Therein, our work shows the very complex nature of the level of anonymity demonstrated by any decentralized currency scheme through a fine-grained systematization. It is thus evident that existing claims for anonymity of different cryptocurrency schemes might only be anonymous in some aspects, e.g., a currency scheme which is claimed to possess

unlinkability, might not demonstrate unlinkability with respect to all entities. Therefore, claims for anonymity cannot be made lightly in the presence of such granularity.

7. Conclusions

In this paper, 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 particularized to varying security subjects and adversarial models. Together, these represent a precise and exhaustive recount of true anonymity achieved by a currency scheme.

Author Contributions: Conceptualization, N.A., X.B. and M.M.; methodology, N.A, X.B. and M.M.; implementation, N.A.; writing—original draft preparation, N.A.; writing—review and editing, N.A., X.B. and M.M.; supervision, X.B. and M.M. All authors have read and agreed to the published version of the manuscript.

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

Data Availability Statement: Data sharing not applicable. No new data were created or analysed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IND	Indistinguishability
ULK	Unlinkability
KNW/KNOWL	Knowledge
PWR	Power
S	Sender/s
R	Recipient/s
V	Value
M	Metadata

Appendix A. Experiments for Correctness of the Scheme

This section provides an extension to the correctness of the proposed currency scheme as briefly discussed in Section 3. Presented below is the background information related to correctness experiments listed in Table 4.

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 subjective functions with following mappings:

$$\text{Deserialise}_W : \{0,1\}^* \times \mathbb{N} \rightarrow \{\perp\} \cup W \quad \text{where } W \in \{\mathbb{P}, \mathbb{A}, \mathbb{T}, \mathbb{V}, \mathbb{M}\}$$

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 A1). ρ is represented as $\rho = (\rho_1, \rho_2, \dots)$ 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. In addition, 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 A2).

Figure A1 lists the functions used to generate input data for the correctness experiments based on bit strings $\rho_k \in \rho$. In addition, other helper functions used in correctness experiments are given in Figure A2.

<p>GenerateState(λ, ρ_k)</p> <ol style="list-style-type: none"> 1. $p \leftarrow \text{Deserialise}_p(\lambda, \rho_k)$ 2. if $\neg(\text{IsValidState}_\pi(p, \lambda))$ 3. return \perp 4. return p <p>GenerateTxns(p, λ, ρ_k)</p> <ol style="list-style-type: none"> 1. $\{(t_p, t_s)\} \leftarrow \text{Deserialise}_T(\lambda, \rho_k)$ 2. for all $(t_p, t_s) \in \{(t_p, t_s)\}$ do 3. if $\neg(\text{IsValidSecTxn}_\pi(t_p, t_s, p))$ 4. return \perp 5. return $\{(t_p, t_s)\}$ <p>GenerateTxnValues($R, \tilde{S}, \lambda, \rho_k$)</p> <ol style="list-style-type: none"> 1. $(V_{old}, V_{new}, m) \leftarrow \text{Deserialise}_V(\lambda, \rho_k)$ 2. if $(V_{old} \neq \tilde{S}) \vee (V_{new} \neq R)$ 3. return \perp 4. return (V_{old}, V_{new}, m) 	<p>GenerateAddr(p, λ, ρ_k)</p> <ol style="list-style-type: none"> 1. $((a_{pk}, a_{sk}) \leftarrow \text{Deserialise}_k(\lambda, \rho_{k+1}))$ 2. for all $(a_{pk}, a_{sk}) \in ((a_{pk}, a_{sk}))$ do 3. if $\neg(\text{IsValidPubAddr}_\pi(a_{pk}, p))$ 4. return \perp 5. if $\neg(\text{IsValidSecAddr}_\pi(a_{pk}, a_{sk}, p))$ 6. return \perp 7. return $((a_{pk}, a_{sk}))$ <p>GenerateMintAddr(p, λ, ρ_k)</p> <ol style="list-style-type: none"> 1. $R_m \leftarrow \text{Deserialise}_M(\lambda, \rho_{k+1})$ 2. for all $a_{pk} \in R_m$ do 3. if $\neg(\text{IsValidPubAddr}_\pi(a_{pk}, p))$ 4. return \perp 5. return R_m
--	---

Figure A1. Functions used to generate input data from bitstring ρ .

<p>SelectSubsetofStates($P_{set}, k, \tilde{\rho}, \mathfrak{a}$)</p> <ol style="list-style-type: none"> 1. $n \leftarrow (\{1, \dots, P_{set} \}; \rho_{k+1})$ 2. for $i \in \{1, \dots, n\}$ do 3. $p_i \leftarrow (P_{set}; \rho_{k+1+i})$ 4. $P_{set} \leftarrow P_{set} \setminus \{p_i\}$ 5. return $(P_{set}, k + 1 + i)$ <p>EvolveState($p, k, \tilde{\rho}, \mathfrak{a}$)</p> <ol style="list-style-type: none"> 1. $W \leftarrow \text{GenerateTxns}(p, \lambda, \rho_{k+1})$ 2. if $W = \perp$, return \perp 3. $\{(t_p, t_s)\} \leftarrow W$ 4. if $\neg(\text{IsValidMintable}_\pi(\{(t_p, t_s)\}, p))$, return \perp 5. $Y \leftarrow \text{GenerateMintData}(p, \lambda, \rho_{k+2})$ 6. if $Y = \perp$, return \perp 7. $R_m \leftarrow Y$ 8. $(p_1, V_x) \leftarrow \text{Mint}_\pi(\{(t_p, t_s)\}, R_m, p; \rho_{k+3})$ 9. return $(p_1, k + 3)$ 	<p>GenerateSetofStates($k, \tilde{\rho}, \mathfrak{a}$)</p> <ol style="list-style-type: none"> 1. $P_{set} \leftarrow \{\}$ 2. $m \leftarrow \rho - k$ 3. if $m < 1$, return \perp 4. $n \leftarrow (\{1, \dots, m - 1\}; \rho_{k+1})$ 5. $i \leftarrow k + 2$ 6. for $j \in \{1, \dots, n\}$ do 7. $p_j \leftarrow \text{GenerateState}(\lambda, \rho_i)$ 8. if $\text{IsValidState}_\pi(p_j, \lambda)$, $P_{set} \leftarrow P_{set} \cup \{p_j\}$ 9. $i \leftarrow i + 1$ 10. return (P_{set}, i)
---	--

Figure A2. Helper functions for correctness.

Figures A3 and A4 list all experiments that establish the correctness of the proposed scheme.

Exp_{II}^{init}(λ, ρ)

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

Exp_{II}^{create-addr}(λ, d, ρ)

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

Exp_{II}^{mint}(λ, ρ):

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

Exp_{II}^{adjudicate}(λ, ρ):

1. $p_0 \leftarrow \text{GenerateState}(\lambda, \rho_1)$ [$p_0 \neq \perp$]
2. $X \leftarrow \text{GenerateSetofStates}(1, \lambda, \rho)$ [$X \neq \perp$]
3. $(P_{set}, k) \leftarrow X$
4. $p' \leftarrow \text{Adjudicate}_\pi(P_{set}, p_0)$
5. **if** $p' = p_0$
6. **for all** $p_i \in P_{set}$ **do**
7. **if** $\text{IsValidState}_\pi(p_i, \lambda) \wedge (p_0 \prec p_i)$
8. **return** 0
9. **else**
10. **if** $\neg(\text{IsValidState}_\pi(p', \lambda)) \vee (p' \notin P_{set})$
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

Exp_{II}^{create-checkpoint}(λ, ρ):

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

Exp_{II}^{create-txn}(λ, ρ)

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

Exp_{II}^{extract-txn-data}(λ, ρ)

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

Exp_{II}^{genesis-state}(λ, ρ):

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

Figure A3. Correctness experiments 1.

<p>Exp_{II}^{checkpoint-monotonicity}(λ, ρ):</p> <ol style="list-style-type: none"> 1. $p_0 \leftarrow \text{GenerateState}(\lambda, \rho_1) [p_0 \neq \perp]$ 2. $p_0^c \leftarrow \text{CreateCheckpoint}_\pi(p_0, \lambda) [p_0^c \neq \perp]$ 3. $p_1 \leftarrow p_0^c$ 4. $i, n_c \leftarrow 1$ 5. while ($i < \rho$) do 6. $X \leftarrow \text{EvolveState}(p_1, i, \lambda, \rho) [X \neq \perp]$ 7. $(p_2, k) \leftarrow X$ 8. $b \xleftarrow{\\$} \{0, 1\}$ 9. if $b = 1$ 10. $H \leftarrow \text{CreateCheckpoint}_\pi(p_2, \lambda; \rho_{k+1})$ 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 $j \in \{1, \dots, n_c - 1\}$ do 17. $p_1^c \leftarrow \text{RetrieveCheckpointState}(p^c)$ 18. if $p_1^c = \perp$, return 0 19. $p^c \leftarrow p_1^c$ 20. return ($p^c \stackrel{?}{=} p_0^c$) 	<p>Exp_{II}^{adj-monotonicity}(λ, ρ):</p> <ol style="list-style-type: none"> 1. $p_0 \leftarrow \text{GenerateState}(\lambda, \rho_1) [p_0 \neq \perp]$ 2. $X \leftarrow \text{GenerateSetofStates}(2, \lambda, \rho) [X \neq \perp]$ 3. $(P_{set}, k) \leftarrow X$ 4. $(P_{test}, j) \leftarrow \text{SelectSubsetofStates}(P_{set}, k, \lambda, \rho)$ 5. $p_1 \leftarrow \text{Adjudicate}_\pi(P_{set}, p_0)$ 6. $X \leftarrow \text{EvolveState}(p_1, j, \lambda, \rho) [X \neq \perp]$ 7. $(p_2, j) \leftarrow X$ 8. $p_3 \leftarrow \text{Adjudicate}_\pi(P_{test} \cup \{p_2\}, p_1)$ 9. return $P_3 \stackrel{?}{=} P_2$ <p>Exp_{II}^{retire-checkpoint}(λ, ρ):</p> <ol style="list-style-type: none"> 1. $p_0 \leftarrow \text{GenerateState}(\lambda, \rho_1) [p_0 \neq \perp]$ 2. $p_1 \leftarrow p_0$ 3. while $\text{RetrieveCheckpointState}_\pi(p_1) \neq p_1$ do 4. $X \leftarrow \text{RetrieveCheckpointState}_\pi(p_1)$ 5. if $X = \perp$, return 0 6. $p' \leftarrow X$ 7. if $\neg(\text{IsValidState}_\pi(p', \lambda))$, return 0 8. if $p' \neq p_1$, return 0 9. $p_1 \leftarrow p'$ 10. return $\text{IsGenesisState}_\pi(p_1, \lambda)$
---	---

Figure A4. Correctness experiments 2.

Appendix B. Security Properties

In this section, we provide comprehensive details of the security properties discussed in Section 4.3, describing how individual games are executed in terms of experiments representing each scenario. For each property, we provide a description of the game, winning condition, and the relevant experiment.

Appendix B.1. Unforgeability

Appendix B.1.1. Game

In this game, the initial state is set up 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_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 A5.

Exp_{II, A, O, \psi, \delta, \alpha, \beta}^{unforgeability}(λ)

1. $A_O, T_O \leftarrow AA.\text{Init}(); M_O \leftarrow \{\}; f_O \leftarrow 0$
2. $(p_O, r, s) \leftarrow \text{SetupState}_{\pi, O, A}(\lambda, \alpha) \langle p_O \neq \perp \rangle$
3. $(p_O, (t_p, t_s), s) \leftarrow \text{RunAdversary}_{\pi, O}(\mathcal{A}_2, p_O, \mathcal{O}, r, s, \delta) \langle p_O \neq \perp \rangle$
4. $S \leftarrow \text{ExtractSenderPubAddr}_\pi(t_p, t_s, p_O)$
5. **return** $\text{IsValidPubTxn}_\pi(t_p, p_O) \wedge (AA.\text{Lookup}(t_p, T_O) \stackrel{?}{=} \perp) \wedge (S \cap AA.\text{Keys}(A_O) \neq \emptyset)$

Figure A5. Experiment for unforgeability.

Appendix B.1.2. Winning Condition

The adversary wins this game if he/she 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/she does not know the corresponding secret key a_{sk} .

Appendix B.2. Transaction Binding Property

Appendix B.2.1. 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 A6.

```

Exptxn-binding $\pi, \mathcal{A}, \mathcal{O}, \psi, \delta, \alpha, \beta$ ( $\lambda$ )
1.  $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.Init(); M_{\mathcal{O}} \leftarrow \{\}$ 
2.  $(p_{\mathcal{O}}, r, s) \leftarrow SetupState_{\pi, \mathcal{O}, \mathcal{A}}(\lambda, \alpha) \ (p_{\mathcal{O}} \neq \perp)$ 
3.  $(p_{\mathcal{O}}, t_s, s) \leftarrow RunAdversary_{\pi, \mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \mathcal{O}, r, s, \delta) \ (p_{\mathcal{O}} \neq \perp)$ 
4.  $t_p \leftarrow AA.Lookup(t_s, T_{\mathcal{O}})$ 
5. return  $(t_p \neq \perp) \wedge (IsValidSecTxn_{\pi}(t_p, t_s, p_{\mathcal{O}}))$ 

```

Figure A6. Experiment for transaction binding property.

Appendix B.2.2. 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.

Appendix B.3. Spendability

Appendix B.3.1. 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 A7). Then, the adversary evolves the state from that point onward, 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 A8).

Appendix B.3.2. Winning Condition

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

Appendix B.4. Balance Property

Appendix B.4.1. 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} , and 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 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 fail, the 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.

ExtractUnmintedTxns(T_O, M_O)

1. $T_M \leftarrow \bigcup_{m \in M_O} m[1]$
2. $T \leftarrow AA.Keys(T_O)$
3. $T' \leftarrow T \setminus T_M$
4. **return** T'

Figure A7. Function ExtractUnmintedTxns.

Exp^{spendability} _{$\pi, A, O, \psi, \delta, \alpha, \beta$} (λ)

1. $A_O, T_O, B \leftarrow AA.Init()$; $M_O \leftarrow \{\}$; $f_O \leftarrow 0$
2. $(p_O, r, s) \leftarrow SetupState_{\pi, O, A}(\lambda, \alpha)$ ($p_O \neq \perp$)
3. $(p_O, \emptyset, s) \leftarrow RunAdversary_{\pi, O}(\mathcal{A}_2, p_O, \emptyset, r, s, \delta)$ ($p_O \neq \perp$)
4. **for all** $a_{pk} \in A_O$ **do**
5. $a_{sk} \leftarrow AA.Lookup(a_{pk}, A_O)$; $bal \leftarrow GetBalance_{\pi}(a_{pk}, a_{sk}, p_O)$
6. $B \leftarrow AA.Insert(a_{pk}, bal, B)$
7. $T' \leftarrow ExtractUnmintedTxns(T_O, M_O)$
8. **for all** $t_p \in T'$ **do**
9. $t_s \leftarrow AA.Lookup(t_p, T_O)$
10. $R \leftarrow ExtractRecipientPubAddr_{\pi}(t_p, t_s, p_O)$
11. $V_{new} \leftarrow ExtractOutputVal_{\pi}(t_p, t_s, p_O)$
12. **for** $i \in \{0, \dots, |R|\}$ **do**
13. $a_{pk} \leftarrow R[i]$; $bal \leftarrow AA.Lookup(a_{pk}, B)$
14. $B \leftarrow AA.Update(a_{pk}, bal - V_{new}[i], B)$
15. $(p_O, \emptyset, s) \leftarrow RunAdversary_{\pi, O, mint}(\mathcal{A}_3, p_O, \emptyset, r, s, \delta)$ ($p_O \neq \perp$)
16. **for all** $a_{pk} \in A_O$ **do**
17. $a_{sk} \leftarrow AA.Lookup(a_{pk}, A_O)$; $bal' \leftarrow GetBalance_{\pi}(a_{pk}, a_{sk}, p_O)$
18. **if** $bal' < AA.Lookup(a_{pk}, B)$, **return** 1
19. **return** 0

Figure A8. Experiment for spendability.

Appendix B.4.2. Winning Condition

The 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:

$$Balance_{bef}(a) + V_{new}(a) - V_{old}(a) + V_{excess}(a) = Balance_{aft}(a) \quad \forall a \in \bar{S}, \bar{R}, \bar{R}_m$$

Figure A9 lists the corresponding game (Exp^{balance} _{$\pi, A, O, \psi, \delta, \alpha, \beta$}) which demonstrates this property.

Appendix B.5. Indemnification

Appendix B.5.1. 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_O in B_{bef} and also records the mint history M_1 . Then, the adversary evolves the state and outputs a set of transactions $\{(t_p, t_s)\}$ together with the current state p_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/she checks the closing balances of all addresses in A_O and ensures none of these addresses are in S or R (Figure A10).

$\text{Exp}_{\pi, A, \mathcal{O}, \Psi, \delta, \alpha, \beta}^{\text{balance}}(\lambda)$

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

Figure A9. Experiment for the Balance property.

$\text{Exp}_{\pi, A, \mathcal{O}, \Psi, \delta, \alpha, \beta}^{\text{indemnification}}(\lambda)$

1. $A_{\mathcal{O}}, T_{\mathcal{O}}, B_{\text{bef}} \leftarrow AA.\text{Init}(); M_{\mathcal{O}} \leftarrow \{\}; S, R \leftarrow (); f_{\mathcal{O}} \leftarrow 0$
2. $(p_{\mathcal{O}}, r, s) \leftarrow \text{SetupState}_{\pi, \mathcal{O}, A}(\lambda, \alpha) \quad \langle p_{\mathcal{O}} \neq \perp \rangle$
3. $(p_{\mathcal{O}}, \mathcal{O}, s) \leftarrow \text{RunAdversary}_{\pi, \mathcal{O}}(\mathcal{A}_2, p_{\mathcal{O}}, \mathcal{O}, r, s, \delta) \quad \langle (p_{\mathcal{O}} \neq \perp) \rangle$
4. $M_1 \leftarrow M_{\mathcal{O}}$
5. **for all** $a_{pk} \in AA.\text{Keys}(A_{\mathcal{O}})$ **do**
6. $a_{sk} \leftarrow AA.\text{Lookup}(a_{pk}, A_{\mathcal{O}})$
7. $bal \leftarrow \text{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})$
8. $B_{\text{bef}} \leftarrow AA.\text{Insert}(a_{pk}, bal, B_{\text{bef}})$
9. $(p_{\mathcal{O}}, (\{t_p, t_s\}), s) \leftarrow \text{RunAdversary}_{\pi, \mathcal{O}}(\mathcal{A}_3, p_{\mathcal{O}}, \mathcal{O}, r, s, 0) \quad \langle p_{\mathcal{O}} \neq \perp \rangle$
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. $S \leftarrow S \parallel \text{ExtractSenderPubAddr}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
14. $R \leftarrow R \parallel \text{ExtractRecipientPubAddr}_{\pi}(t_p, t_s, p_{\mathcal{O}})$
15. **for all** $a_{pk} \in AA.\text{Keys}(A_{\mathcal{O}})$ **do** $\langle a_{pk} \notin S \parallel R \rangle$
16. $a_{sk} \leftarrow AA.\text{Lookup}(a_{pk}, A_{\mathcal{O}})$
17. **if** $\text{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}}) \neq AA.\text{Lookup}(a_{pk}, B_{\text{bef}})$, **return** 1
18. **return** 0

Figure A10. Experiment for indemnification.

Appendix B.5.2. Winning Condition

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

Appendix B.6. Positivity

Appendix B.6.1. 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 A11).

$\text{Exp}_{\pi, A, \mathcal{O}, \psi, \delta, \alpha, \beta}^{\text{positivity}}(\lambda)$

1. $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\text{Init}(); M_{\mathcal{O}} \leftarrow \{\}; f_{\mathcal{O}} \leftarrow 0$
2. $(p_{\mathcal{O}}, r, s) \leftarrow \text{SetupState}_{\pi, \mathcal{O}, A}(\lambda, \alpha) \langle p_{\mathcal{O}} \neq \perp \rangle$
3. $(p_{\mathcal{O}}, (a_{pk}, a_{sk}), s) \leftarrow \text{RunAdversary}_{\pi, \mathcal{O}}(A_2, p_{\mathcal{O}}, \mathcal{O}, r, s, \delta) \langle p_{\mathcal{O}} \neq \perp \rangle$
4. **return** $(\text{IsValidSecAddr}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}})) \wedge (\text{GetBalance}_{\pi}(a_{pk}, a_{sk}, p_{\mathcal{O}}) < 0)$

Figure A11. Experiment for positivity.

Appendix B.6.2. Winning Condition

The adversary wins if the given address is valid and has a negative balance.

Appendix B.7. Descandancy

Appendix B.7.1. Game

In this game, the 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 A12).

$\text{Exp}_{\pi, A, \mathcal{O}, \psi, \delta, \alpha, \beta}^{\text{descandancy}}(\lambda)$

1. $A_{\mathcal{O}}, T_{\mathcal{O}} \leftarrow AA.\text{Init}(); M_{\mathcal{O}} \leftarrow \{\}; f_{\mathcal{O}} \leftarrow 0$
2. $(p_{\mathcal{O}}, r, s) \leftarrow \text{SetupState}_{\pi, \mathcal{O}, A}(\lambda, \alpha) \langle p_{\mathcal{O}} \neq \perp \rangle$
3. $(p_{\mathcal{O}}, \mathcal{O}, s) \leftarrow \text{RunAdversary}_{\pi, \mathcal{O}}(A_2, p_{\mathcal{O}}, \mathcal{O}, r, s, \delta) \langle p_{\mathcal{O}} \neq \perp \rangle$
4. $p' \leftarrow p_{\mathcal{O}}$
5. **while** $\text{IsValidState}_{\pi}(p', \lambda) \neq 0$ **do**
6. $p^c \leftarrow \text{RetrieveCheckpoint}(p')$
7. **if** $\text{IsGenesisState}_{\pi}(p^c, \lambda)$, **return** 0
8. $p' \leftarrow p^c$
9. **return** 1

Figure A12. Experiment for descandancy.

Appendix B.7.2. Winning Condition

Adversary wins if the loop ends up in an invalid state. Experiment for descandancy.

Appendix C. Proofs of Theorems

This section provides the proofs for the theorems defined in Section 5.3. In the cases where proofs are similar, only a proof sketch is provided, pointing out differences if any.

Appendix C.1. Proof of Theorem 1

Proof. (Part i) Assume a currency scheme Π which is secure against the anonymity game defined by a given combination of $\psi_{pk_r}, \psi_{sk_r}, \psi_v, \psi_m, \psi_t, \alpha, \beta$ 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. Therefore, 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., $\omega\text{-}\psi((3, 0)_s) - (\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((2, 0)_s) - (\delta, \alpha, \beta)$. 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 $(1, 0)$. Therefore, it is clear that if Π is secure in $(2, 0)$, it is also secure in $(1, 0)$,

i.e., $\omega\text{-}\psi((2,0)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((1,0)_s)\text{-}(\delta, \alpha, \beta)$. Similarly, (0,0) case provides even less knowledge to the adversary compared to (1,0). Therefore, if Π is secure in (1,0), it is also secure in (0,0), i.e., $\omega\text{-}\psi((1,0)_s)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi((0,0)_s)\text{-}(\delta, \alpha, \beta)$.

(Part ii) Similar to part i, we can see that in this case $\psi_{sk_s} = 1$ is fixed. Therefore, $\psi_{sk_s} = 3$ 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). Therefore, 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) as the adversary has the control over identity. Therefore, 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) as, with the former case, the adversary has control over both the identity and the randomness over the latter case. Therefore, 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/she does not have full control over the senders as the address creation is performed honestly. Therefore, (4,4) is more powerful than (3,3) and along the same line of argument as before, we can say that $(4,4) \rightarrow (3,3)$. \square

Appendix C.2. Proof of Theorem 2

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

Appendix C.3. Proof of Theorem 3

Proof. (Part i) Assume a currency scheme Π which is secure against the anonymity game defined by a given combination of $\omega, \delta, \alpha, (\psi_{pk}, \psi_{sk})_r, \psi_v, \psi_m, \psi_t, \beta$, 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. Therefore, 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 (as all other parameters are constant) and hence both notions are equivalent. Thus, Π is also secure in the case where $(\psi_{pk_s}, \psi_{sk_s}) = (1,1)$, i.e., $\omega_s\text{-}\psi((\mathbf{0},\mathbf{1})_s)\text{-}(\delta, \alpha, \beta) \leftrightarrow \omega_s\text{-}\psi((\mathbf{1},\mathbf{1})_s)\text{-}(\delta, \alpha, \beta)$.

(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, thus, the above equivalence relation. \square

Appendix C.4. Proof of Theorem 4

Proof. 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. Therefore, 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 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 Π' derived from Π such that the transaction creation process is modified as follows:

```

CreateTxn $_{\Pi'}$ (args, R, V) {
  (tp, ts) ← CreateTxn $_{\Pi}$ (args)
  if SpecialValue(R) = Hash(V) then
    return ((tp, args), ts)
  return ((tp, ∅), ts)
}

```

All other operations in Π' 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 Π' and if $\psi_t = 1$, the adversary is able to obtain additional information about the senders and therefore it is not secure. Accordingly, this scheme is secure in $(0, 0)$, but not necessarily secure in $(1, 0)$, i.e., $\omega_{s-\psi}((0, 0)_s) - (\delta, \alpha, \beta) \not\rightarrow \omega_{s-\psi}((1, 0)_s) - (\delta, \alpha, \beta)$.

Following the same line of argument, we can construct counter examples to prove the same for other relations. \square

Appendix C.5. Proof of Theorem 5

Proof. (Sketch) This proof is similar to the above proof of Theorem 4. \square

Appendix C.6. Proof of Theorem 6

Proof. (Part i) We start with a currency scheme Π that is secure against a $\omega\text{-}\psi\text{-}(\delta, 3_\alpha, \beta)$ adversary. In this case, the adversary has the full control over the state initialization 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 initialization. Therefore, the adversary in the latter case is less powerful than the former. Thus, it follows that Π is also secure against $\omega\text{-}\psi\text{-}(\delta, 2_\alpha, \beta)$ given that Π is secure against a more powerful adversary in $\omega\text{-}\psi\text{-}(\delta, 3_\alpha, \beta)$.

(Part ii) Similar to part i, $\alpha = 2$ adversary is more powerful than a $\alpha = 1$ adversary as the adversary has no control over state initialization in the latter case. Therefore, given that a scheme Π is secure in $\omega\text{-}\psi\text{-}(\delta, 2_\alpha, \beta)$, it is clear that Π is also secure in $\omega\text{-}\psi\text{-}(\delta, 1_\alpha, \beta)$.

(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 $\omega\text{-}\psi\text{-}(\delta, 1_\alpha, \beta)$, then Π is also secure against a less powerful adversary, $\omega\text{-}\psi\text{-}(\delta, 0_\alpha, \beta)$. \square

Appendix C.7. Proof of Theorem 7

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

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

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

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

In the case of $b = 1$, this construction returns Init_{Π_1} which makes Π secure in $\omega\text{-}\psi\text{-}(\delta, 2_\alpha, \beta)$. When $b = 0$, the adversary chooses to output Init_{Π_0} simulating the insecure protocol Π_0 . Π is not secure in $\omega\text{-}\psi\text{-}(\delta, 3_\alpha, \beta)$ in this case. This shows that $\omega\text{-}\psi\text{-}(\delta, 2_\alpha, \beta) \rightarrow \omega\text{-}\psi\text{-}(\delta, 3_\alpha, \beta)$ is false. Therefore, we conclude that $\omega\text{-}\psi\text{-}(\delta, 2_\alpha, \beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, 3_\alpha, \beta)$.

(Part ii) Assume there exist two currency schemes Π_1 and Π_0 such that Π_1 is secure in $\omega\text{-}\psi\text{-}(\delta, 1_\alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\delta, 1_\alpha, \beta)$. $\gamma = 1$ represents an honest state initialization with randomness over which the adversary does not have control, whereas when $\gamma = 2$, the adversary chooses the randomness with an honest state initialization. Therefore 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 initialization takes place upon a selection of a random string r_2 as follows depending on a bit b :

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

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

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

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

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

Construct a new currency scheme Π as explained previously.

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

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. Therefore, 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 $\omega\text{-}\psi\text{-}(\delta, 0_\alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\delta, 1_\alpha, \beta)$, i.e., $\omega\text{-}\psi\text{-}(\delta, 0_\alpha, \beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, 1_\alpha, \beta)$. \square

Appendix C.8. Proof of Theorem 8

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 $\delta = 1$, thus 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 $\omega\text{-}\psi\text{-}(2_\delta, \alpha, \beta)$ represents a more powerful attacker than $\omega\text{-}\psi\text{-}(1_\delta, \alpha, \beta)$ and also $\omega\text{-}\psi\text{-}(1_\delta, \alpha, \beta)$ is more powerful than $\omega\text{-}\psi\text{-}(0_\delta, \alpha, \beta)$, thus it is clear that both implications are true. \square

Appendix C.9. Proof of Theorem 9

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/she can only

view the state. Suppose that two currency schemes Π_0 and Π_1 exist such that Π_1 is secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ and Π_0 is not secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$. Assume that a currency scheme Π is constructed as before in the proof of part (ii) of the Theorem 7. The initialization 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 $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ scenario. Π_1 is already secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ and thus Π is also secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$. Now assume that the adversary gets to choose the bit $b = 0$. In this case, functions f_{Π_0} will be executed and therefore 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 $\omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta)$ adversary. However, from the definition, Π_0 is not secure against this adversary as Π_0 is not secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$ and thus is not secure in $\omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta)$ as well. Therefore, Π is also not secure in $\omega\text{-}\psi\text{-}(\mathbf{2}_\delta, \alpha, \beta)$ in this case.

(Part ii) Suppose that a currency scheme Π_1 is secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$ and another currency scheme Π_0 is not secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$. 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 initialization and a modification to the Mint function, i.e.,

```

Init $\Pi$ ( $\lambda, (r_1, r_2)$ ) = (Init $\Pi_0$ ( $\lambda, r_1$ ), (Init $\Pi_1$ ( $\lambda, r_1$ ),  $r_2, 1$ ))
Mint $\Pi$ (( $p_0, p_1, r, b$ ),  $T, args$ ) = {
    if  $p_0$  and  $p_1$  are initial states and  $b = 1$  and  $T = \{r\}$ ,
        then return (( $p_0, p_1, r, 0$ ), outputs)
}

```

In the honest case, Init_Π will output $(p_0, p_1, 1)$ and all functions f_{Π_1} will be executed. As the adversary cannot view the state now, this simulates $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$. We claim that Π is $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$ secure in this case as the adversary has the same amount of information as for Π_1 and Π_1 is secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$. Then, immediately after the state initialization, 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. Therefore, we can conclude that Π is secure in $\omega\text{-}\psi\text{-}(\mathbf{0}_\delta, \alpha, \beta)$ but not secure in $\omega\text{-}\psi\text{-}(\mathbf{1}_\delta, \alpha, \beta)$. \square

Appendix C.10. Proof of Theorem 10

Proof. Consider a currency scheme Π which is secure in $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{1}_\beta)$. 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., $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{1}_\beta) \rightarrow \omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{0}_\beta)$. \square

Appendix C.11. Proof of Theorem 11

Proof. Assume that a currency scheme Π is secure in $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{0}_\beta)$. Consider a construction Π' 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 $\Pi'$ ( $args, b, V$ ) {
    if (Mint $\Pi$ ( $args$ ) =  $\perp$ )  $\wedge$  ( $b = 1$ ) then
        return  $V$ 
    else return Mint $\Pi$ ( $args$ )
}

```

When the bit $b = 0$, Π' functions similar to Π , and thus is secure in $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{0}_\beta)$. However, when $b = 1$, with the modified Mint function models a scenario for $\beta = 1$ for Π' , yet Π' is not secure here as the Mint operation leaks special information about the transaction in this case. Therefore, we conclude that $\omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{0}_\beta) \not\rightarrow \omega\text{-}\psi\text{-}(\delta, \alpha, \mathbf{1}_\beta)$. \square

Appendix C.12. Proof of Theorem 12

Proof. As before, we can say 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 $\psi_v = 0$. Therefore, 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 $\psi_v = 0$. \square

Appendix C.13. Proof of Theorem 13

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. \square

Appendix C.14. Proof of Theorem 14

Proof. Adhering to the same proofing methodology, we argue that $\psi_m = 3$ adversary is more powerful than $\psi_m = 2$ adversary, as the adversary has full control over metadata in the former case. Therefore, 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. Moreover, through the same line of argument, it follows that $(\psi_m = 2) \rightarrow (\psi_m = 1)$ and $(\psi_m = 1) \rightarrow (\psi_m = 0)$. \square

Appendix C.15. Proof of Theorem 15

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. \square

Appendix C.16. Proof of Theorem 16

Proof. (Part i) Consider a currency scheme Π which is secure in $\omega\text{-}\psi(5_t)\text{-}(\delta, \alpha, \beta)$. With $\psi_t = 5$, the adversary has the highest possible knowledge of the transaction as the adversary creates the transaction and thus 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., $\omega\text{-}\psi(5_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(4_t)\text{-}(\delta, \alpha, \beta)$.

(Part ii) Similarly, $\omega\text{-}\psi(4_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(3_t)\text{-}(\delta, \alpha, \beta)$ 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., $\psi_t = 2$). Therefore, $\omega\text{-}\psi(3_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(2_t)\text{-}(\delta, \alpha, \beta)$ 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., $\omega\text{-}\psi(2_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(1_t)\text{-}(\delta, \alpha, \beta)$.

(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. Therefore, the former case shows a more powerful adversary than the latter case. Accordingly, $\omega\text{-}\psi(1_t)\text{-}(\delta, \alpha, \beta) \rightarrow \omega\text{-}\psi(0_t)\text{-}(\delta, \alpha, \beta)$. \square

Appendix C.17. Proof of Theorem 17

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. \square

References

1. Herskind, L.; Katsikouli, P.; Dragoni, N. Privacy and Cryptocurrencies—A Systematic Literature Review. *IEEE Access* **2020**, *8*, 54044–54059. [CrossRef]
2. Sabry, F.; Labda, W.; Erbad, A.; Malluhi, Q. Cryptocurrencies and Artificial Intelligence: Challenges and Opportunities. *IEEE Access* **2020**, *8*, 175840–175858. [CrossRef]
3. Sweeney, L. K-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **2002**, *10*, 557–570. [CrossRef]
4. Tsukada, Y.; Mano, K.; Sakurada, H.; Kawabe, Y. Anonymity, Privacy, Onymity, and Identity: A Modal Logic Approach. In Proceedings of the 2009 International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009; Volume 3, pp. 42–51.
5. Pfitzmann, A.; Hansen, M. A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. Available online: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf,2010 (accessed on 3 August 2020).
6. Díaz, C.; Seys, S.; Claessens, J.; Preneel, B. Towards Measuring Anonymity. In *Privacy Enhancing Technologies*; Dingledine, R., Syverson, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 54–68.
7. Khalilov, M.C.K.; Levi, A. A Survey on Anonymity and Privacy in Bitcoin-like Digital Cash Systems. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2543–2585. [CrossRef]
8. Conti, M.; Kumar, S.; Lal, C.; Ruj, S. A survey on security and privacy issues of bitcoin. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3416–3452. [CrossRef]
9. Morris, L. Anonymity Analysis of Cryptocurrencies. Ph.D. Thesis, Rochester Institute of Technology, Rochester, NY, USA, 2015.
10. Androulaki, E.; Karame, G.O.; Roeschlin, M.; Scherer, T.; Capkun, S. Evaluating user privacy in bitcoin. In Proceedings of the International Conference on Financial Cryptography and Data Security, Okinawa, Japan, 1–5 April 2013; pp. 34–51.
11. Ron, D.; Shamir, A. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*; Sadeghi, A.R., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 6–24.
12. Miller, A.; Moeser, M.; Lee, K.; Narayanan, A. An Empirical Analysis of Linkability in the Monero Blockchain. *arXiv* **2017**, arXiv:1704.04299.
13. Möser, M.; Soska, K.; Heilman, E.; Lee, K.; Heffan, H.; Srivastava, S.; Hogan, K.; Hennessey, J.; Miller, A.; Narayanan, A. An empirical analysis of traceability in the monero blockchain. *Proc. Priv. Enhanc. Technol.* **2018**, *2018*, 143–163. [CrossRef]
14. Amarasinghe, N.; Boyen, X.; McKague, M. A Survey of Anonymity of Cryptocurrencies. In Proceedings of the Australasian Computer Science Week Multiconference, ACSW 2019, Sydney, Australia, 29–31 January 2019; pp. 2:1–2:10.
15. Alsalami, N.; Zhang, B. SoK: A Systematic Study of Anonymity in Cryptocurrencies. In Proceedings of the 2019 IEEE Conference on Dependable and Secure Computing (DSC), Hangzhou, China, 18–20 November 2019; pp. 1–9.
16. Reid, F.; Harrigan, M. An analysis of anonymity in the bitcoin system. In *Security and Privacy in Social Networks*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 197–223.
17. Meiklejohn, S.; Pomarole, M.; Jordan, G.; Levchenko, K.; McCoy, D.; Voelker, G.M.; Savage, S. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In Proceedings of the 2013 Conference on Internet Measurement Conference, IMC '13, Barcelona, Spain, 23–25 October 2013; pp. 127–140.
18. Spagnuolo, M.; Maggi, F.; Zanero, S. Bitiodine: Extracting intelligence from the bitcoin network. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 3–7 March 2014; pp. 457–468.
19. Möser, M.; Böhme, R. Anonymous Alone? Measuring Bitcoin's Second-Generation Anonymization Techniques. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW), Paris, France, 26–28 April 2017; pp. 32–41.
20. Ober, M.; Katzenbeisser, S.; Hamacher, K. Structure and Anonymity of the Bitcoin Transaction Graph. *Future Internet* **2013**, *5*, 237–250. [CrossRef]
21. Wijaya, D.A.; Liu, J.K.; Steinfeld, R.; Sun, S.F.; Huang, X. Anonymizing Bitcoin Transaction. In *Information Security Practice and Experience*; Bao, F., Chen, L., Deng, R.H., Wang, G., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 271–283.
22. Kappos, G.; Yousaf, H.; Maller, M.; Meiklejohn, S. An Empirical Analysis of Anonymity in Zcash. *CoRR* **2018**, abs/1805.03180. Available online: <http://xxx.lanl.gov/abs/1805.03180> (accessed on 16 November 2020).
23. Quesnelle, J. An Analysis of Anonymity in the Zcash Cryptocurrency. Master's Thesis, University of Michigan-Dearborn, Dearborn, MI, USA, 2018.
24. Van Saberhagen, N. 2013. Available online: <https://cryptonote.org/whitepaper.pdf> (accessed on 10 December 2020)
25. Wijaya, D.A.; Liu, J.; Steinfeld, R.; Liu, D.; Yuen, T.H. Anonymity Reduction Attacks to Monero. In *Information Security and Cryptology*; Guo, F., Huang, X., Yung, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 86–100.
26. Wijaya, D.A.; Liu, J.; Steinfeld, R.; Liu, D. Monero Ring Attack: Recreating Zero Mixin Transaction Effect. In Proceedings of the 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), New York, NY, USA, 1–3 August 2018; pp. 1196–1201.
27. Ruffing, T.; Moreno-Sanchez, P. ValueShuffle: Mixing Confidential Transactions for Comprehensive Transaction Privacy in Bitcoin. In *Financial Cryptography and Data Security*; Springer International Publishing: Cham, Switzerland, 2017; pp. 133–154.
28. Fuchsbauer, G.; Orrù, M.; Seurin, Y. Aggregate cash systems: A cryptographic investigation of mumblewimble. In *Advances in Cryptology—EUROCRYPT 2019*; Springer International Publishing: Cham, Switzerland, 2019; pp. 657–689.

29. Poelstra, A. 2016. Available online: <https://scalingbitcoin.org/he/papers/mimblewimble.pdf> (accessed on 14 August 2020)
30. Biryukov, A.; Tikhomirov, S. Deanonymization and Linkability of Cryptocurrency Transactions Based on Network Analysis. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS P), Stockholm, Sweden, 17–19 June 2019; pp. 172–184.
31. Cachin, C.; De Caro, A.; Moreno-Sanchez, P.; Tackmann, B.; Vukolic, M. The Transaction Graph for Modeling Blockchain Semantics. *IACR Cryptol. ePrint Arch.* **2017**, *2017*, 1070.
32. Amarasinghe, N.; Boyen, X.; McKague, M. The Complex Shape of Anonymity in Cryptocurrencies: Case Studies from a Systematic Approach. In Proceedings of the International Conference on Financial Cryptography and Data Security, Tokyo, Japan, 22–23 April 2021.
33. Kumar, A.; Fischer, C.; Tople, S.; Saxena, P. A Traceability Analysis of Monero’s Blockchain. In *Computer Security—ESORICS 2017*; Foley, S.N., Gollmann, D., Sneekenes, E., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 153–173.